Graduado en Ingeniería de la Salud

Aplicación de técnicas de aprendizaje automático para la predicción de respuesta patológica en cáncer colorrectal

Applying machine learning to predict pathology response in colorectal cancer

Realizado por

Alejandro Domínguez Recio

Tutorizado por

Mercedes Amor Pinilla

Eduardo Guzmán de los Riscos

Departamento

Lenguajes y Ciencias de la Computación

MÁLAGA, (mes y año)

Abstract

Machine learning (ML) techniques are transforming many aspects of our society nowadays. In the medical field and more specifically in oncology ML is being applied from laboratories to clinical practice. In this project, 4 ML models will be applied in the prognosis of 5-year survival using colorectal cancer omics data. At the same time, the bias in the predictions of the models will be evaluated considering as study variable 'RACE' and as sensitive attribute 'Black and African American'. The dataset used corresponds to a real cohort of 594 patients and was downloaded from the public repository The Cancer Genome Atlas TCGA. The integration of 3 types of omics data (RNAseq, degree of methylation and abundance of microorganisms in microbiome) associated with each sample was applied in the development of the ML models. The most significant variables from the omics datasets were selected by 2 sequentially applied feature selection methods. The 30 most significant feature from each omics dataset were applied in the development of ML models. The ML models were implemented using the *lgbm* libraries of Python and *Caret* of R. The performance of the models was evaluated by 5x2 Cross Validation. A selection of metrics were applied both in evaluating the performance of the ML models and in detecting bias in their predictions. The results showed significant differences in performance between the families of ML models applied. The analysis of the bias in the predictions was influenced by the unbalanced proportion of variables in the 'Race' variable, causing a decrease in the robustness of the results in the applied metrics.

Keywords: Machine Learning, Bias, Cancer, R, Python

Resumen

Las técnicas de machine learning (ML) estan transformando muchos de los aspectos nuestra sociedad hoy en día. En el campo de la medicina y más especificamente en la oncología el ML esta siendo aplicado desde los laboratorios hasta la práctica clínica. En este proyecto se aplicarán 4 modelos de ML en el prognóstico de supervivencia a 5 años a partir datos ómicos de cancer colorrectal. A la vez, se evaluará el sesgo en las predicciones de los modelos considerando como variable de estudio 'RACE' y como atributo sensible 'Black and African American'. El conjunto de datos usado pertenece a una cohorte real de 594 pacientes y fué descargado del repositorio público The Cancer Genome Atlas (TCGA). El tipo de datos ómico aociado a cada muestra y aplicado en el desarrollo de los modelos fue la integración de datos RNAseq, grado de metilación y abundancia de microorganismos en microbioma. Las variables mas significativas de los conjuntos de datos ómicos se realizó mediante 2 métodos de selección de variables secuencialmente aplicados. Las 30 variables mas signíficativas de cada conjunto de dato ómico fueron aplicadas en el desarrollo de los modelos de ML. Los modelos de ML fueron implementados usando las librerias *lgbm* de Python y *Caret* de R. El rendimiento de los modelos se evaluó mediante 5x2 Cross Validation. Una selección de métricas fueron aplicadas tanto en la evaluación del rendimiento de los modelos ML, como en la detección de bias en sus predicciones. Los resultados mostraron diferencias significativas en el rendimiento entre las familias de los modelos de ML aplicados. El análsis del sesgo en las predicciones se vió influida por la proporcion desbalanceada de las variables en la variable 'Race', provocando una disminución en la robustez de los resultados en las metricas aplicadas.

Palabras clave: **Machine Learning, Bias, Cancer, R, Python**

Glosario

ACC	Accuracy
ACC NS	Accuracy Categoria No Sensible
ACC S	Accuracy Categoria Sensible
ACC P	Accuracy Parity
AUC	Area Under ROC Curve
CRC	Cancer Colorectal
DT	Decision Tree
EO	Equalized Odds
EOP	Equal Opportunity
FNR	False Negative Rate
FNR NS	False Negative Rate Categoria No Sensible
FPR	False Positive Rate
GBDT	Gradient Boosting Decision Tree
IA	Inteligencia Artificial
IDE	Integrated Development Environment
KNN	K Nearest Neighbors
LGBM	Light Gradient Boosting Machine
ML	Machine Learning
NB	Naive Bayes
NGS	Next Generation Sequencing
PCA	Principal Component Analysis

PRE NS	Precisión Categoría No Sensible
PRE S	Precisión Categoría Sensible
PCA	Principal Component Analysis
PRO NS	Proportional Parity Categoria No Sensible
PRO S	Proportional Parity Categoria Sensible
PRO P	Proportional Parity
RF	Random Forest
SLE	Supervivencia Libre de Enfermedad
SVM	Support Vector Machine
SVM L	Support Vector Machine Lineal
SVM P	Support Vector Machine Polinómico
SVM R	Support Vector Machine Radial
TCGA	The Cancer Genome Atlas
TPR	True Positive Rate

Índice

Índice de figuras

Índice de cuadros

1 Introducción

1.1. Motivación

La cantidad de datos que se tiene hoy en día de los pacientes es imposible de asimilar por los especialistas sanitarios. Añadiendo la dificultad de que cada paciente tiene unas características genéticas y clínicas propias que influyen particularmente en el curso de las patologías. Los modelos de machine learning nos permiten obtener tanto información explicativa como predictiva de cómo influyen las distintas variables en un conjunto de datos en relación a algún fenómeno de interés existente en estos. Con este proyecto se pretende aplicar distintas técnicas de machine learning para la predicción y análisis de respuesta patológica en cáncer colorrectal. Además se abordarán el desafío de las identificación de equidad en los conjuntos de datos utilizados, aspecto el cuál tiene particular presencia y relevancia en conjuntos de datos clínicos. El proyecto será realizado sobre el conjunto de datos Colorectal Adenocarcinoma (TCGA, PanCancer Atlas), el cual será descargado desde el repositorio público 'cBioPortal for Cancer Genomics'.

La principal motivación de este proyecto es la predicción de prognósis en cancer colorectal a partir de la aplicación de modelos machine learning usando datos multi-ómicos. Adicionalmente se analizará el posible sesgo o bias en las predicciones de los modelos aplicados.

1.2. Objetivos

- Estudiar el estado del arte en modelos de machine learning aplicados a la predicción de prognósis de cancer colorectal, obteniendo una visión de aquellos que mejor resultado están dando hasta la fecha.
- Estudiar el estado del arte en técnicas de detección y mitigación de bias en el ámbito clínico. Adquiriendo conocimiento de aquellas técnicas de detección y mitigación de bias, con una aplicación más optima en el contéxto del proyecto.
- Evaluar la equidad/bias en el conjunto de datos aplicado, permitiendo determinar la existencia y el grado de bias en el conjunto datos.
- Integrar datos multi-ómicos en la predicción de prognósis en cancer colorectal. Abordando el problema del prognósis de cancer colorectal desde un punto de vista multi dimensional.
- Obtener conjuntos de características representativos a partir de técnicas de selección de variables en prognósis de cancer colorectal, permitiendo identificar aquellas variables con mayor influencia en la prognósis de cancer colorectal.
- Comparar/Evaluar el rendimiento de los modelos de machine learning aplicados con métricas de rendimiento/selección de métricas.
- Evaluar la equidad en la predicción de los modelos de machine learning aplicados mediante una selección de métricas de equidad.

1.3. Tecnologías usadas

Durante el desarrollo del proyecto se han utilizado los siguientes lenguajes de programación, librerias y entornos de desarrollo:

R

R es un lenguaje de programación open source de análisis estadístico [24]. Existen más de 4000 librerías disponibles en el repositorio público The Comprehensive R Archive Network (CRAN), con aplicación en machine learning, análisis de datos o bioinformática.

Actualmente se encuentra entre los 10 lenguajes de programación más usados mundialmente

Las **librerias R** aplicadas en el TFG se muestran en el Cuadro 1

Libreria	**Descripción general**
$caret$	Funciones aplicadas al entrenamiento de modelos de clasificación y regresión
$limma$	Análisis de expresión diferencial en estudios de RNAseq y microarrays
$fairness$	Cálculo de métricas aplicadas en la detección de sesgo en modelos
$ROCR$	Evaluación y visualización del rendimiento de clasificadores
$Biobase$	Funciones base de Bioconductor
$CORElearn$	Evaluación de variables
$dplyr$	Manipulación de estructuras de datos
$mltools$	Preprocesado y exploración de datos en machine learning.
$kernlab$	Modelos de machine learning basados en funciones kernel

Cuadro 1: Librerias R usadas en el TFG

Python

Python es un lenguaje de programación de alto nivel caracterizado por su suave curva de aprendizaje y su alta legibilidad. La sintáxis sencilla y flexible de Python permite desarrollar aplicaciones con menos código en comparación con otros lenguajes de programación de bajo nivel. Otra de sus características principales es el soporte de los paradigmas de programación, orientado a objetos, estructurado y funcional.

Actualmente se encuentra entre los 5 lenguajes de programación mas usados mundialmente, siendo comúnmente utilizado en machine learning.

Las **librerias Python** aplicadas en el TFG se muestran en el Cuadro 2

Libreria	**Descripción general**
numpy	Manipulación de arrays
pandas	Manipulación y exploración de datos
lightgbm	Implementación python de Light Gradient Boosting Machine
sklearn	Funciones aplicadas al flujo de trabajo completo del machine learning.

Cuadro 2: Librerias Python usadas en el TFG

RStudio

RStudio es un IDE que permite utilizar R de una forma sencilla y organizada mediante una interfaz gráfica de usuario [70].

Entre las características principales de RStudio se encuentran, ejecución de comandos R por consola en sesiones interactivas, uso de notebooks o administrador de librerias R.

Google Colaboratory

Google Colaboratory conocido como 'Colab' es una herramienta colaborativa web accesible a partir de una cuenta Google [11]. Colab permite ejecutar y compartir código Python en notebooks. Colab es principalmente aplicado a machine learning o análisis de datos. Adicionalmente de la función colaboratoria, una de las funciones mas destacadas de Colab es el acceso a recursos de computación GPU sin coste.

2

Contexto Médico del Cancer Colorectal

2.1. Introducción

Con el objetivo de poner en contexto la relevancía del cancer colorectal, en los siguientes puntos se detallará la patología desde un punto biológico y clínico:

2.2 Intestino Grueso. Anatomía y fisiología del intestino grueso.

2.3 Incidencia. Resumen de la incidencia del cancer colorrectal mundialmente.

2.4 Causas y Factores de Riesgo. Resumen de las principales causas y factores de riesgo conocidos en cancer colorectal.

2.5 Sintomatología. Mención de los sintomas clínicos presentados en cancer colorrectal.

2.6 Clasificación. Mención de los principales sistemas de clasificación de estadios del cancer colorrectal.

2.7 Diagnóstico. Mención de los sistema de diagnóstico de cancer colorrectal aplicados en la clínica, así como de los tipos de muestras usados.

2.8 Supervivencia. Resumen de los datos de supervencia en cancer colorrectal mundialmente.

2.2. Intestino Grueso

El intestino grueso se encuentra localizado en la cavidad abdominal y constituye la mayor porción del tracto gastrointestinal en términos de masa y longitud [25]. A la vez, el intestino grueso forma parte del sistema digestivo, siendo responsable de algunas de sus funciones principales.

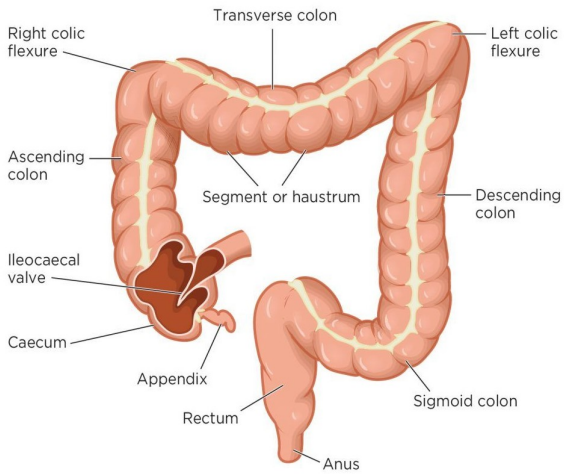


Figura 1: Anatomía Intestino Grueso [51]

Anatomía del Intestino Grueso

El intestino grueso tiene una longitud de aproximadamente 150 cm, extendiéndose desde la porción final del íleon hasta el ano [51, 25]. El intestino grueso esta formado por 5 partes principales:

Ciego

La parte inicial del intestino grueso comienza con el ciego. El ciego tiene una longitud de 6cm aproximadamente y es la prolongación de la parte final del íleon. En el ciego se continúa con la absorción de nutrientes y electrolitos no absorbidos por íleon.

Colon

La parte final del ciego conecta con el colon. El colon forma la parte más larga del intestino grueso. Este está formado por colon ascendente, flexión cólica derecha, colon transverso,

flexión cólica izquierda, colon descendente y colon sigmoide. El retroperitoneo (zona exterior de la cavidad peritoneal) alberga el colon ascendente, colon descendente y recto. El colon transverso y colon sigmoide se encuentra adherido por el mesocolon a la pared abdominal.

Recto, Canal Anal y Ano

La parte final del colon conecta con el recto. El recto ocupa los 20cm finales del tracto gastrointestinal y conecta distalmente con el canal anal y ano. La parte final del recto, llamada ampolla rectal tiene la funcion de almacenar las heces antes de ser expulsadas. [51]

Funciones del Intestino Grueso

Después de 8-9 horas tras la ingestión los alimentos pasan a través del intestino grueso [25]. Una vez que los alimentos llegan al intestino grueso en forma de quimo líquido, una serie de funciones básicas del sistema digestivo son producidas. Las funciones del intestino grueso son:

- **Absorción de agua y electrolitos:** Debido a la presencia de residuos en el colon se estimulan contracciones haustrales. Como resultado de las contracciones haustrales, los residuos son empujados a la siguiente porción haustral. A partir del tránsito generado por las contracciones se facilita la absorción del agua y electrolitos en los residuos. Ralentizando el tránsito y dando más tiempo al intestino grueso para absorber agua y electrolitos, contracciones antiperistálticas desplazan los residuos de alimentos hacia la válvula ileocecal. Aproximadamente el 10 % de agua restante en el quimo se absorbe en el intestino grueso [51].

- **Formación y transporte de heces:** Aproximadamente 150ml de cada 500ml que entran en el intestino grueso, llegan a ser heces. La heces están formadas mayormente por bacterias, células epiteliales de la mucosa intestinal, elementos inorgánicos, fibra y agua. A su vez entre su composición podemos encontrar grasas y proteínas no digeridas por el intestino delgado. De los 1.5l de fluidos que pasan al intestino grueso diariamente, 100ml son expulsados a través de las heces. El proceso de absorción de fluidos por el intestino grueso tiene una duración de entre 12-24 horas. [51]

- **Digestión química por el microbioma intestinal:** A partir de la fermentación por millones de bacterias localizadas en el intestino grueso, se produce la digestíón química de los residuos que pasan. Millones de bacterias localizadas en el intestino grueso tienen la función de la digestión de carbohidratos restantes tranferidos en el quimo desde el intestino delgado. La digestión de los carbohidratos restantes es realizada a partir de su fermentación, liberando hidrógeno, dióxido de carbono y metano. La protección contra bacterias potencialmente nocivas procedente del entorno externo, el equilibrio del sistema inmune o la síntesis de ciertas vitaminas son otras de las funciones asociadas al microbioma intestinal [51].

2.3. Incidencia

El cáncer colorectal (CRC) es el tercer cáncer más comunmente diagnosticado y la segunda causa principal de muerte relacionada al cáncer mundialmente [32].

Aproximadamente 2 millones de nuevos casos de CRC y sobre un 1 millón de muertes por CRC en 2020, correspondiendo al 10,7 % y 9,5 % de todos los nuevos casos de cáncer y muertes asociados mundialmente.

Alrededor del 60.4 % de los casos de CRC diagnosticados en 2020 fueron en pacientes con una edad comprendida entre 50 y 74 años, y sobre el 10 % de los casos con edad menor a 50 años. Añadiendo que, la mitad de las muertes asociadas al CRC ocurrieron en pacientes con una edad comprendidad entre 50 y 74 años.

La franja de edad 50-74 años es aquella con mayor número de nuevos diagnósticos de CRC, con el 60.4 % del total de los nuevos casos en 2020, como también aquella con mayor número de muertes, el 50 % del total de muertes asociadas al CRC. Por otro lado, la franja de edad de pacientes menores a 50 años se le asoció el 10 % del total de los nuevos casos de CRC en 2020.

Manteniendo la incidencia actual, se estima que en 2040 el número de nuevos diagnósticos de CRC alcanze los 3.2 millones de casos, con un incremento del 63 % respecto del 2020. Por consiguiente, el CRC representa un desafío global para los servicios sanitarios, tanto por su mortalidad como por el incremento de uso de los servicios sanitarios y gastos asociados [47, 32].

2.4. Causas y Factores de Riesgo

En la mayoría de los casos el CRC comienza como una ampolla en la mucosa intestinal, aunque también podría presentarse como un leve tumor inicial denominado adenoma, cual podría pogresar hasta un tumor maligno, dependiendo de su presentación histológica y tamano. En presencia de adenoma, el 60 % de estos son denominados adenomas simples y el 40 % adenomas multiples. Aproximadamente el 24 % de los pacientes con pólipos no tratados desarrollarán cancer. [26]

Entre los principales factores de riesgo podemos citar bajo consumo de frutas y verduras, alto consumo de carnes rojas y grasas, tabaquismo, mayores de 50 años o historial familiar de CRC. No obstante el 35 % de los factores de riesgo pueden ser explicados por factores hereditarios.

Existen mutaciones que representan el 90 % del total de las encontradas en familias con cáncer hereditario. Se estima que alrededor del 10 % de todos los casos de CRC son debidos a sindromes genéticos. En ausencia de síndromes genéticos, el historial familiar de CRC contribuye en un incremento del 25 % el riesgo de desarrollar CRC.

Mutaciones en genes involucrados en la silenciación de transcripción, control del ciclo celular, reparación de DNA o diferenciación celular son algunas de las causas moleculares del CRC. A su vez anomalías cromosomales o cambios epigenéticos involucrados en la proliferación celular son también causas relacionadas con la aparición de CRC. [15, 26]

Inversamente, factores como la ingesta de fármacos anti-inflamatorios no-esteroides reducen el riesgo de CRC, regulando la sobreexpresión del receptor del factor de crecimiento epidérmico. Además factores como una alta ingesta de fibra, frutas y verduras son factores protectores del desarrollo de CRC.[26],

2.5. Sintomatología

La sintomatología depende de la localización y tamaño tumor, como de la presencia o ausencia de metástasis. El dolor abdominal, sangrado rectal, alteración de los hábitos intestinales o pérdida involuntaria de peso, son los síntomas más comunes. El sangrado rectal es otro de los síntomas del CRC. No obstante cánceres proximales rara vez producen este síntoma, tanto porque la sangre llega mezclada con el residuo fecal como por la degradación química durante

el tránsito colónico. En los casos de sangrado oculto el paciente puede presentar anemia por deficiencia de hierro [26, 10].

2.6. Clasificación

Existen cuatro tipos principales de estadios en CRC, clasificados a partir del sistema TNM, sistema establecido por *The American Joint Committee on Cancer*. Las letras T,N y M representan los factores que se tienen en cuenta para la clasificació, siendo T el tamaño del tumor, N el número de nodos linfáticos afectados y M la presencia o ausencia de metástasis.[26]

Estadios TNM	Características
Estadio 0	Tis, N0, M0
Estadio I	T1, N0, M0 T2, N0, M0
Estadio IIA Estadio IIB	T3, N0, M0 T4, N0, M0
Estadio IIIA Estadio IIIB Estadio IIIC	T1-2, N1, M0 T3-4, N1, M0 T1-4, N2, M0
Estadio IV	T1-4, N0-2, M1

Figura 2: Sistema clasificación TNM [27]

A su vez, el CRC *Subtyping Consurtium* ha determinado cuatro consensuados subtipos moleculares CMS1, CMS2, CMS3 y CMS4. Estos fueron determinados a partir del análisis transcriptómico de expresión génica en celulas tumorales, estroma tumor-infiltrado y en microambiente tumoral. [15, 28]

En estadios pre-malignos, se ha identificado tres pathways principales que determinan los perfiles moleculares en CRC: inestabilidad cromosomal, alta inestabilidad de microsatélites y fenotipo metilador de islas CpG.[15]

2.7. Diagnóstico

El diagnóstico de CRC mediante analisis manual microscopico de muestras histopatológicas, es el método tradicional. Mediante la observación microscópica del patólogo, este determina el grado del tumor en base a los cambios estructurales observados en el tejido. Detectar las diferencias en las estructuras tumorales es un gran desafío para el diagnóstico manual de los patólogos, el cual es subjetivo y potencialmente propenso a errores. [54, 67]

El diagnóstico mediante análisis manual microscópico no es la única técnica de diagnóstico en CRC. Existen cinco categorías principales de técnicas de diagnóstico en CRC, las cuales dependen del tipo de datos analizado y de la metodología aplicada. Las cinco categorías principales de técnicas de diagnóstico en CRC son: análisis espectral, análisis de textura, análisis genético, análisis de sérum y análisis objeto-orientado. El análisis de textura, espectral y objeto-orientado son técnicas aplicadas al análisis de imágenes. Las técnicas de análisis genético y sérum están basadas en el análisis de muestras físicas.

- **Análisis de textura:** La textura es la combinación de patrones con regular o irregular frecuencia. Resultados del análisis de características asociadas a la textura muestran que estas son indenpendientes en cada estadio del CRC, permitiendo así, su aplicación como modelo de clasificación en todos sus estadios. Parámetros como la entropía o la correlación son usados para la cuantificación de la textura. [55, 54]

- **Análisis espectral:** Es una tecnología que obtiene información espacial y espectral usando imágenes y espectroscopia. Selecciona bandas espectrales de imágenes de CRC, y permite diferenciar entre tejidos normales y malignos. [17, 73, 54]

- **Análisis objeto-orientado:** Imágenes de biopsia CRC son segmentadas y clasificadas en base al conocimiento de fondo sobre el tamaño, localización y distribución espacial de los elementos del tejido del colon.

- **Análisis genético:** Estudio de las alteraciones y comportamiento génetico del CRC. Existen tres estados diferenciados que un gen podría experimentar sobre-expresión, sub-expresión y mutación. La información biológica asociada es obtenida a partir de técnicas tales como microarrays, RNAseq o next-generation sequencing (NGS). Posteriormente es analizada con técnicas y modelos estadísticos, como el machine learning. [54]

- **Análisis de sérum:** El sérum esta compuesto por una serie de elementos químicos. En presencia de células cancerígenas, los elementos que forman el sérum son alterados. A partir de técnicas basadas en la fluorescencia inducida por láser y la espectroscopia, se pueden detectar las alteraciones en la composición del sérum. Dichas alteraciones permiten obtener información sobre la presencia o estado de los tumores.[54, 38]

2.8. Supervivencia

La supervivencia de un paciente es una medida que determina la prognosis del paciente. La supervivencia a tiempo x establece el porcentaje en el que un paciente vive al menos un periodo de tiempo x después de haber sido diagnósticado . Para la estimación del porcentaje se tiene en cuenta el tipo histológico, estadio del tumor o factores particulares como el perfil genético del paciente o tumor.

Sin tener el cuenta la edad del paciente, tipo histológico o estadio del tumor, se estima que alrededor del 65 % y 58 % de los pacientes diagnósticados de CRC, tienen una supervivencia de 5 y 10 años respectivamente. En los países desarrollados la tasa de supervivencia a 5 anos es del 55 %, siendo Estados Unidos el país con la tasa de supervivencia más alta, un 65 %. Por otro lado, los paises en vias de desarrollo presentan una tasa de supervivencia a 5 años del 39 %, marcando África la supervivencia más baja, 14 %.

Mejoras en el tratamiento del CRC colorectal ha permitido mejorar las tasas de supervivencia en la mayoria de los países mundialmente. El diagnóstico temprano gracias a la incorporación de programas de cribado como la mejora de sus técnicas, ha sido otro de los que han ayudada a mejorar la supervivencia. No obstante la tasa de supervivencia de los pacientes con cáncer varía en función del estadio y el tipo de tumor, así como de otros factores como la edad, el sexo y la presencia de metástasis. La expresión o mutación de determinados genes también influye en la supervivencia. Por ejemplo, la expresión de los genes KISS1 y KSSR se ha relacionado con un menor riesgo de metástasis.

Investigaciones recientes implican a la microbiota intestinal en el progreso y respuesta a los tratamientos en CRC. Se ha demostrado que la microbiota intestinal participa en la actividad anticancerígena y la toxicidad de la inmunoterapia en investigaciones preclínicas y clínicas [56, 32, 15].

3

Estado del Arte

3.1. Introducción

En los últimos años, con la cantidad de datos biológicos disponibles, los problemas biológicos y específicamente los relacionados con el cáncer, están adquiriendo una complejidad inabarcable con los métodos tradicionales.

Tecnologías como los microarrays de genes, la secuenciación de nueva generación (NGS) y la espectrometría de masas son algunas de las principales tecnologías ómicas que están permitiendo obtener la cantidad de información biomédica actualmente disponible.

La cantidad de datos actualmente disponibles puede proceder tanto del estudio de artefactos biológicos, como la expresión génica, la secuenciación del exoma completo o el análisis del microbioma, como de la información clínica asociada a los pacientes.

El éxito de las técnicas de aprendizaje automático o machine learning, especialmente cuando se utilizan para procesar grandes cantidades de datos, está llevando a una tendencia de su aplicación en el análisis de datos biológicos. Estos métodos utilizan algoritmos de aprendizaje para encontrar patrones y mejorar el análisis en datos complejos y de alta dimensionalidad. Los problemas biológicos complejos, desde el análisis de supervivencia y el pronóstico de enfermedades hasta el análisis de rutas, se han estudiado con técnicas de machine learning. Además, debido al buen rendimiento del machine learning en datos de alta dimensionalidad, se ha aplicado a conjuntos de datos en los que el número de variables es mucho mayor que el de observaciones, situación común en los conjuntos de datos biológicos. [43]

3.2. Machine learning y problemas biomédicos

Las aplicaciones del aprendizaje automático en el estudio del cáncer y los procesos biológicos implicados son diversas.

La gran variedad de usos del machine learning en la investigación del cáncer se atribuye a la compatibilidad de las características y la estructura funcional de los algoritmos de machine learning con los problemas y datos biológicos.

El elevado número de dimensiones en datos biológicos, en combinación con el gran número de variables clínicas disponibles (estado patológico, estado de clasificación TNM, efecto de los fármacos, respuesta al tratamiento, etc.) crea unas condiciones ideales para el uso de técnicas de machine learning tanto supervisadas como no supervisadas.

En el caso del aprendizaje supervisado, los problemas pueden clasificarse en problemas de regresión (tiempo de supervivencia del paciente, expresión génica o edad individual) o problemas de clasificación (clasificación de los pacientes en función de su estado de genes reguladores, estadios de la enfermedad o metástasis, etc.) en función de la variable dependiente a predecir.

Además, es posible desarrollar modelos de clasificación y regresión para identificar diversos resultados ómicos incluso sin basarse en datos clínicos. Esto incluye la identificación de genes impulsores, la evaluación del estado de metilación y la determinación de tipos de mutación. En el caso del aprendizaje no supervisado, la aplicación más común es encontrar nuevos subtipos de la enfermedad[43] .

Siendo la predicción de la prognosis el problema biológico que se aborda en este TFG, en el siguiente subapartado se realiza una revisión de algunas de las aplicaciones del machine learning en este área:

Predicción de Prognosis

Según la terminología médica, la prognosis se define como: “ resultado probable de un episodio de enfermedad en relación a la perspectiva de recuperación de la misma, tal y como lo indican la naturaleza y los síntomas del caso ”.

Conocer el pronóstico es crucial para determinar la probabilidad de avance del cáncer y la esperanza de vida del paciente, lo que afecta directamente al tratamiento clínico del paciente.

El pronóstico de los distintos tipos de cáncer difiere significativamente debido a su heterogeneidad, su entorno y el comportamiento único de cada paciente.

El pronóstico suele predecirse en función de una serie de variables clínicas [45]. En función de la combinación de variables y del tipo de tumor existen índices pronósticos definidos, como

el sistema de estadificación Tumor-Nódulo-Metástasis (TNM), el Nottingham Prognostic Index (NPI) para el cáncer de mama, y el estadio de la Fédération Internationale de Gynécologie et d'Obstétrique (FIGO) para el ginecológico. Sin embargo, el enfoque tradicional sólo tiene en cuenta una serie de factores que son inadecuados para abordar la complejidad del cáncer. Por ejemplo, el sistema TNM sólo tiene en cuenta tres factores para predecir el pronóstico, sin considerar otros factores relacionados con el tumor y el paciente. Esta simplicidad de los sistemas tradicionales suele dar lugar a grupos de pacientes sobregeneralizados con escasa capacidad pronóstica.

Por otro lado, el aumento en la generación de datos ómicos debido al desarrollo de tecnologías de secuenciación de alto rendimiento está permitiendo estudiar el pronóstico del cáncer desde un punto de vista molecular. La investigación ha demostrado que los biomarcadores moleculares, los parámetros celulares y la expresión de determinados genes son potenciales predictores del pronóstico del cáncer [75].

Diferentes técnicas de machine learning se han aplicado a la predicción del pronóstico y los factores relacionados [45]. Uno de los factores más estudiados con técnicas de machine learning es el tiempo de supervivencia de los pacientes. El tiempo de supervivencia se ha predicho principalmente mediante datos de expresión junto con datos de metilación. Los dos datos más utilizados por las técnicas de machine learning para la predicción del pronóstico son los datos de expresión génica y los datos de metilación. La integración de datos de expresión junto con otros tipos de datos ómicos o clínicos ha sido otro enfoque en algunos trabajos de machine learning aplicados a la predicción del pronóstico.

La recurrencia tumoral es un factor clave en el pronóstico del cáncer. Se han aplicado técnicas de machine learning para identificar el factor de riesgo responsable de la recurrencia tumoral. Los datos de expresión transcripcional han sido el principal tipo de datos utilizados por las técnicas de machine learning en este campo [45].

Otro factor clave en la predicción del pronóstico es la clasificación de los tumores [45]. En este sentido, las técnicas de machine learning han utilizado principalmente datos de RNAseq como fuente de datos. De acuerdo con el perfil de expresión de cada paciente, modelos de machine learning identifican genes que tienen el potencial de ser clasificados en diferentes niveles de riesgo.

3.3. Algoritmos de Machine Learning en el Ámbito Biomédico

Los distintos tipos de machine learning, aprendizaje supervisado, no supervisado y aprendizaje profundo han sido aplicados ampliamente en el campo del cáncer [43].

Como se ha comentado en el apartado anterior, en el campo biomédico es habitual que en los datos el número de características sea muy superior al número de muestras u observaciones, como es el caso de los datos genómicos.

La distribución de los tipos de algoritmos previamente aplicados suele estar relacionada con los datos utilizados y sus características. Por ejemplo, las redes neuronales son mucho más sensibles o su rendimiento se ve muy afectado en conjuntos de datos con alta dimensionalidad y pocas observaciones, mientras que modelos como Random (RF), Support Vector Machine (SVM) o modelos lineales son más resilientes en estas situaciones [45].

En este sentido, debido a que los datos genéticos y son caracterizados por las características previamente mencionadas modelos como SVM o RF han sido ampliamente aplicados [39].

En el caso de las redes neuronales es el modelo predominante utilizado para el análisis o predicción de imágenes biomédicas. Al mismo tiempo se han aplicado otros tipos de modelos como K Nearest Neighbors (KNN) o Naive Bayes (NB).

Adicionalmente, las aplicaciones del machine learning se pueden dividir dependiendo del tipo de características o condiciones tumorales estudiadas. Por un lado, los modelos que se entrenan para predecir condiciones transversales o básicas. En esta situación se utilizan datos de diferentes cohortes para entrenar los modelos. Por otro lado, estudios que tratan diferentes cohortes de forma independiente. Estos estudios se centran principalmente en la mejora de modelos o análisis anteriores aplicados a una cohorte específica

En los siguientes puntos se hace un breve repaso a las principales técnicas de machine learning aplicadas al ámbito biomédico.

Aplicaciones del Aprendizaje Supervisado

En el campo biomédico, el aprendizaje supervisado, lo que incluye la clasificación, la regresión y ensemblers, se ha aplicado ampliamente en los últimos años.

Los problemas biológicos y las situaciones en las que se ha aplicado el aprendizaje supervisado son diversos. Por ejemplo, en [62] se utiliza una combinación de SVM, RF y Decision Tree

(DT) para el diagnóstico y la estratificación del cáncer colorrectal utilizando datos genómicos. Para el diagnóstico, RF alcanzó una precisión del 99,8 %, y para la estratificación, una precisión media del 91,52 %.

Del mismo modo, [41], utiliza una combinación de Regresión Lineal (LR), RF, SVM, KNN y NB para el diagnóstico de cancer colorrectal, pero en este caso se utilizaron registros médicos electrónicos. SVM obtuvo la mayor precisión entre los cinco modelos, alcanzando una precisión del 86,5 %.

Otros estudios han abordado la integración de datos, como [22], que combina datos clínicos, de imagen y genómicos para predecir posibles recaídas de carcinoma oral de células escamosas. La integración de datos se realizó creando modelos independientes para cada tipo de datos y combinando los resultados mediante un sistema de votación por mayoría. Se aplicaron redes bayesianas, RNA, SVM, DT y RF.

El análisis de supervivencia es otro factor crítico estudiado con machine learning. En el estudio [42], se aplicó la regresión de Cox univariante y multivariante para detectar características significativas en el pronóstico del cáncer colorrectal de aparición temprana.

Aplicaciones del Aprendizaje No Supervisado

El aprendizaje no supervisado se utiliza principalmente para la subtipificación o reclasificación de pacientes en una cohorte particular.

La subtipificación de la expresión génica aplicada al pronóstico del cáncer gástrico se realizó en [14] utilizando una factorización matricial no negativa (NMF). Se detectaron cuatro subgrupos a partir de 32 genes miembros de las tres principales vías asociadas al cáncer gástrico. A continuación, se desarrolló un modelo SVM para generar puntuaciones de riesgo destinadas a predecir la supervivencia global utilizando los cuatro subgrupos identificados anteriormente.

[1] utilizó una combinación de cuatro tipos de algoritmos no supervisados para encontrar relaciones no lineales entre los hábitos de patrones dietéticos y el cáncer colorrectal. Los modelos aplicados fueron t-distributed stochastic neighbor embedding (t-SNE), uniform manifold approximation and projection (UMAP), reglas de asociación Apriori, análisis de componentes principales (PCA) y análisis factorial (FA). T-SNE alcanzó el mayor rendimiento, siendo capaz de proporcionar una clara diferenciación visual entre los dos grupos estudiados, diagnóstico positivo y negativo de CRC.

Asimismo, [61] utilizó la agrupación jerárquica para la estratificación de pacientes con características clinicopatológicas y moleculares similares en una cohorte de CRC. Se identificaron tres clusters diferenciados. El análisis de supervivencia de los tres clusters mostró diferencias significativas entre ellos. El análisis multivariante específico de los clusters determinó que todos los clusters eran significativos para la supervivencia libre de enfermedad (SLE).

Aplicaciones del Aprendizaje Profundo

Las aplicaciones del aprendizaje profundo en el campo biomédico siguen una tendencia creciente. La mayoría de las aplicaciones de aprendizaje profundo se centran en problemas relacionados con datos de imagen, como la clasificación de imágenes en el diagnóstico clínico. Sin embargo es posible que su aplicación se encuentre en otros tipos de datos, como los datos genómicos o las historias clínicas electrónicas.

Por ejemplo, [46] utilizó una red neuronal convolucional de 3 dimensiones para la detección de pólipos basada en vídeos endoscópicos. Alcanzaron resultados de Area Under ROC cuerve (AUC) de 0,87.

Como se mencionó anteriormente, los datos genómicos son una de las principales fuentes de datos biomédicos. En este sentido, [20] aplicó un algoritmo de aprendizaje profundo de capa de entrada 512x512 para la detección de genes a partir de imágenes histopatológicas del tejido colorrectal.

La integración de datos de múltiples fuentes también ha sido abordada mediante técnicas de aprendizaje profundo.

En [30], se integraron imágenes histopatológicas y datos de mRNA-seq de cancer hepático. Se aplicó una red completamente convolucional para extraer características de las imágenes histopatológicas. La combinación de características extraídas y los genes más representativos se utilizaron para el análisis de supervivencia mediante una regresión de Cox.

Otro ejemplo de integración de datos es [12], donde la integración de datos multiómicos de diferentes tipos de tumores se realiza mediante la extracción de características con un modelo autoencoder de aprendizaje profundo. Las características extraídas se utilizaron para estimar puntuaciones de riesgo mediante una regresión de Cox.

3.4. Retos

Existe un conjunto de retos identificados que son necesarios de abordar para facilitar y permitir la integración del aprendizaje automático en la práctica clínica. Algunos de los principales retos para la integración del aprendizaje automático en la práctica clínica son los siguientes:

- **Integración de datos**

  El cáncer es un problema complejo, afectado por varios diversos factores y sus relaciones. Debido a los avances en las tecnologías de alta secuenciación, el cáncer y los procesos biológicos implicados en él pueden estudiarse desde un punto de vista genómico, transcriptómico o proteómico. Al mismo tiempo, abordar este problema según una única fuente de información se está quedando obsoleto [43].

  machine learning ha demostrado su capacidad para manejar problemas no lineales con alta dimensionalidad. Sin embargo, la alta dimensionalidad de los datos genómicos clínicos y el pequeño número de muestras hacen que estos modelos sean difíciles de entrenar y susceptibles de overfitting. Además, existen datos no genómicos, como las imágenes, que se utilizan en la práctica clínica como principal técnica de diagnóstico. Sin embargo, las imágenes clínicas suelen tener un tamaño de gigabytes, y su formato no estructurado hace que su integración con datos ómicos estructurados sea con las tecnicas actuales una tarea difícil [35].

- **Reproducibilidad de modelos de machine learning**

  En ciencia, un estudio es reproducible si, dado el mismo conjunto de datos y metodología de análisis, un grupo independiente puede obtener los mismos resultados observados en el estudio original [5]. En este sentido, los datos y el código de desarrollo de machine learning deben estar disponibles para ser reproducibles. En la actualidad, los problemas de privacidad, las cuestiones éticas y las exigencias legislativas en el ámbito sanitario hacen que compartir datos sea muy complejo. Incluso cuando los modelos o los datos pueden compartirse, problemas como la documentación incompleta de los parámetros o el coste de reproducir modelos de gran complejidad provocan que la reproducibilidad sea todo un reto.

Para abordar el problema de la reproducibilidad, existen una serie de directrices y checklists de mejores prácticas para guiar cada etapa de los proyectos de aprendizaje automático. Entre las directrices y listas de comprobación de mejores prácticas más reconocidas, se encuentra la lista de comprobación TRIPOD, compuesta por 22 elementos para guiar la presentación de informes de proyectos de machine learning [29].

- **Bias**

  Con la integración del machine learning en la práctica clínica existe una gran preocupación por la automatización y la propagación de los sesgos actuales en los datos [72].

  La infrarrepresentación o sobrerrepresentación de valores en una serie de características, también denominada desequilibrio de clases o class imbalance, es habitual en los datos sobre el cáncer. Por ejemplo, en los datos genómicos existe una distribución desigual conocida en los perfiles genómicos y exómicos secuenciados en el proyecto The Cancer Genome Atlas, con prevalencia de descendientes europeos sobre asiáticos, africanos e hispanos [16].

  Esta distribución desequilibrada puede ser el resultado de un acceso desigual o injusto a los recursos clínicos, por ejemplo, ciertos grupos étnicos tienen menos acceso a las pruebas de laboratorio por razones económicas o relacionadas con el entorno, por lo que las bases de datos tienen menos información sobre ellos.

  Teniendo en cuenta que los modelos machine learning dependen de la calidad y representación de los datos, si un modelo machine learning se entrena con datos de población general, el modelo funcionará peor en las etnias infrarrepresentadas en contraste con las etnias con mejor acceso a los recursos clínicos. Esta situación podría verse perpetuada por la automatización de los modelos machine learning en la práctica clínica y los datos generados por su uso.

4

Conjunto de Datos

4.1. TGCA Consortium

The Cancer Genomic Atlas (TCGA) es una base de datos genómicos de caracterización de distintos subtipos de cancer.

El proyecto TCGA fue creado en 2006 por el National Cancer Institute (NCI) y el National Human Genome Research Institue (NHGRI) de Estados Unidos [68, 43].

El objetivo del proyecto principal es avanzar en el conocimiento del cancer humano mediante el análisis de las características geneticas de sus distintos subtipos .

En su primera fase en 2006 un total de 3 subtipos de cáncer fueron caracterizados. En su segunda fase en 2009, se llegó a alcanzar una completa caracterización genómica de entre 20-25 subtipos de cancer.

Actualmente existe información genética de más de 30 subtipos de cáncer.

El acceso a la información en sus bases de datos es de libre acceso proporcionando a la comunidad científica una de las mayores bases de datos en la caracterización del cáncer. Esto ha permitido alcanzar mejoras en su diagnóstico y tratamiento.

La información genética disponible es obtenida a partir de tecnologías de secuenciación de alto rendimiento.

Distintas tipos de datos genéticos en base al tipo de técnica molecular aplicada para su análisis son accesibles. Entre ellas podemos destacar RNAseq, grado de metilación de DNA o perfil transcriptómico del microbioma.

4.1.1. RNAseq

Esencialmente el RNA es la copia de un fragmento de DNA que posteriormente sirve como código en la traducción de la proteína [21]. Los fragmentos de RNA proporcionan información

acerca de la actividad de transcripción o nivel de expresión de las celulas.

El RNA es cuantificado comúnmente mediante la tecnología RNAseq. Basicamente, la tecnología de secuenciación RNAseq identifica fragmentos de cDNA, los cuales son fragmentos de RNA en los que se ha producido un cambio de la base nitrogenada uracilo por timina, proceso denominado transcripción inversa. Posteriormente fragmentos de cDNA con una longitud definida, también denominados lecturas, son sequenciados. Una vez la secuenciación ha terminado, las lecturas son mapeadas contra un genoma o transcriptoma de referencia, o en algunos casos ensamblados sin referencia. Finalmente, y comúnmente tras un proceso de normalización de los resultados se obtiene la cantidad de lecturas asociadas a regiones de interés en el DNA o genes en el transcriptoma.

Los datos de RNAseq disponibles en TCGA han sido normalizados mediante el método RSEM. RSEM es un método de normalización basada en la cuantificación de la abundancia asociada a genes. Una de las características principales de RSEM es la ausencia de genoma de referencia en su proceso de normalización [40].

Los datos de RNAseq en TCGA están organizados por genes, nivel de expresión asociado y la muestra de estudio a la que pertenece.

4.1.2. Metilación

Definiendo epigenética como el conjunto de mecanismos del que disponen las células para regular su actividad genética sin la existencia de modificación del DNA en respuesta a factores extrínsicos [2, 68].

El proceso de metilación es uno de los mecanismo epigenéticos de las células para modular y regular su actividad.

El proceso de metilación se caracteriza por la adicción de un grupo metil a la cadena de ADN. En base a la posición del grupo metilo en la cadena de ADN se distingen distintas categorias de metilación. Metilaciones producidas en regiones del ADN con una citosina seguida por una guanina son denominadas regiones CpG.

El grado de regulación genética en una célula puede ser estudiado bajo la intensidad o cantidad de regiones CpG en el DNA.

Las regiones CpG son una de las alteraciones más prematuras y frecuentes en cáncer. Las zonas hipermetiladas involucradas en rutas homoestáticas son una de las principales causas

impulsoras del cáncer.

Los datos de metilación disponibles en TCGA están formados por identificadores CpG y sus coefientes Betas asociados a cada muestra en el estudio. Los coeficientes Beta definen el ratio entre los alelos metilado y no metilado, con un rango de valores 0-1.

4.1.3. Microbioma

El microbioma del cuerpo humano alberga entre 10-100 trillones de microorganismos, principalmente en el intestino grueso, donde se estima que residen el 97 % de estos [59, 69]. Se considera que el microbioma está formado por cada uno de estos organismos y sus genes.

Existe una larga evidencia que el microbioma desarrolla un importante rol en mantener procesos homoestáticos en el organismo humano. Por consiguiente, estados de disbiosis en el microbioma contribuyen a la patogénesis de multitud de enfermedades.

En cáncer, investigaciones previas sugieren que la ocurrencia, progresión o respuesta de terapias contra el cáncer está relacionada con la presencia o ausencia de ciertas comunidades de microorganismos en el microbioma.

Los datos asociado al microbioma disponibles en TCGA proveen información sobre el recuento log-CPM asociadados a microorganismos en las muestras de estudio.

4.2. Conceptos Preprocesamiento de Datos en Machine Learning

4.2.1. Selección de Variables

La selección de variables es el proceso de reducción de variables redundantes y menos significativas en un conjunto de datos.

En machine learning workflows es un paso habitual durante el preprocesado de datos. Reduciendo aquellas variables redundantes y menos significativas se consigue una reducción en el tiempo de entrenamiento, mejora en las predicciones de los modelos y facilita la comprensión de los datos.

En conjuntos de datos clínicos es usual una gran cantidad de variables asociadas a diferentes procesos biológicos estudiados. Concretamente en datos ómicos la selección de variables nos permite separar aquellas variables especialmente relevantes en el proceso de estudio.

Las técnicas de selección de variables se pueden dividir en tres tipos: filtrado, wrapper y integradas o embedded [4].

4.2.2. Integración de Datos

Existen tres enfoques principales aplicados en la integración de datos: integración temprana, intermedia y tardía [9]. Se caracterizan por la fase en la que se realiza la integración, definiéndose la fase temprana antes de entrenar el modelo, la fase intermedia cuando el modelo se está desarrollando y la fase tardía cuando el modelo se ha desarrollado. Explicamos con más detalle cómo funcionan estos enfoques:

Integración temprana. Es una simple concatenación de características de diferentes conjuntos de datos ómicos en una única matriz.

Integración intermedia. No concatena características. Se utilizan modelos machine learning para realizar la integración. La integración intermedia puede dividirse en general o específica.

En **integración tardía**. Los resultados de los modelos aplicados en los distintos tipos de datos se combinan manualmente.

4.2.3. Normalización

Un paso a estudiar en el preprocesado de datos en machine learning es la normalización o estandarización de sus características.

La normalización es clave para que los modelos de machine learning que determinen sus predicciones en base a las distincias entre observaciones, no se vean afectados en su ajuste por outliers o diferentes escalas en las características con los que son entrenados. Esto dada la situación en la que el nivel de predicción de las características que componen el conjunto de datos no está relacionada con el rango de valores en estas.

Es decir, dado el caso de tener un conjunto de datos con las variables A y B, y una clase o variable de estudio que queremos predecir a partir de estas. El nivel de predicción de A y B no está relacionado por el rango de valores existe en estas. Sin embargo, dado el caso de que A tuviera un rango entre 0 y 1, y B entre 0-200. En tal caso, aún sin ser B una variable con una capacidad predictoria mayor que A, un modelo basado en distancias se verá afectado

por el rango de valores en esta, y dará mayor importancia a B, alterando así la capacidad de generalización de este.

Aplicando técnicas de normalización o estandanrización nos aseguramos que el total del conjunto de variables comparten el mismo rango, evitando así el posible sesgo creado por los rango de valores en estas.

Dentro de las técnicas de normalización y estandarización de datos existen multitud de variaciones dependiendo de la escala a la que se ajusten los datos o los parámetros instrínsecos de estos que se utilizen en su aplicación. Entre las técnicas de normalización y estandarización más comunes se encuentran min max-escaler, standard scaler o robust scaler [48].

4.3. Técnicas de Preprocesado de Datos Aplicadas en el TFG

4.3.1. Selección de Variables: Expresión Diferencial

La expresión diferencial o de abundancia permite obtener información sobre la diferencia en el comportamiento biológico distintos contextos biológicos [21].

La expresión diferencial o de abundancia esta basada en que las células u organismos biológicos tienen una expresión o abundancia distinta en base al contexto biológico en el que se encuentren. Su estudio es comumente aplicado en la práctica clínica. Por ejemplo, en la investigación oncológica,es de especial interés determinar las diferencias de expresión entre tejidos sanos y patológicos con el fin de conocer mejor los genes y sus respectivas funciones biológicas asociadas [21, 52].

4.3.2. Selección de Variables: Ganancia de Información

La técnica de selección de variables, Information Gain (IG) o Ganancia de Información nos proporciona el nivel de información asociada a una variable respecto de la clase de estudio [6]. Cuanto mayor valor IG, más información proporciona la variable sobre la clase. Las siguientes ecuaciones nos permiten obtener el valor IG:

$$IG(C, X) = Entropy(C) - \sum_{n=1} \frac{X_x}{X} * Entropy(X_x)$$

$$Entropy = -\sum_{n=1}^{c} P(x_i) log_2 P(X_i)$$

Siendo C la clase, X el vector que representa la variable dentro del conjunto de datos, e x los valores dentro del vector X. En la ecuacion de entropia, c simboliza el número de clases y $P(X_i)$ la probabilidad de ocurrencia de una instancia de la clase i [7].

4.3.3. Normalización: Min-Max

El método de normalización Min-Max transforma cada valor en una variable en el rango [0-1].

El valor normalizado de cada variable se calcula a partir de la siguiente fórmula[48]:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Siendo X_{norm} el valor de la variable normalizada, X el valor previo y X_{min} , X_{max}, los valores minimos y máximos de la columna respectivamente.

4.3.4. Integración de Datos: Integración Temprana

Dada la definición en la sección 4.2.2, la integración temprana es una concatenación de características de diferentes conjuntos de datos ómicos en una única matriz

5
Métodos

5.1. Conceptos Machine Learning

El aprendizaje automático o machine learning es un subconjunto de la inteligencia artificial (IA) en el que los modelos de aprendizaje están diseñados para aprender y predecir patrones en datos a partir de datos anteriores [39, 43].

El proceso de aprendizaje en los modelos de machine learning puede dividirse en dos partes: 1) modelado de los datos y las interdependencias o patrones dentro de estos 2) clasificación de nuevas instancias basadas en las dependencias o patrones aprendidos.

Las predicciones o resultados de los modelos de machine learning se basan en los datos de entrenamiento. Por lo tanto, es crucial proporcionar datos de calidad y generalizados para el rendimiento del modelo.

A la vez técnicas de nachine learning pueden dividirse según el tipo de problema y las características de los datos implicados [58, 39]:

- **Aprendizaje supervisado** En el aprendizaje supervisado, los datos de entrenamiento tienen un output o etiqueta asociada. Los problemas de aprendizaje supervisado pueden dividirse en dos tipos:

  - **Clasificación.** La tarea principal es predecir o clasificar un output o clase asociada a nuevas instancias desconocidas, basándose en el conocimiento aprendido de datos etiquetados anteriores. La etiqueta o clase a predecir es categórica.

    Los problemas de clasificación en los que la salida tiene dos categorías se definen como problemas de clasificación binaria, y para más de dos categorías a predecir se definen como problemas de clasificación multiclase.

  - **Regresión.** En este caso, el modelo pretende predecir un output o clase numérica.

- **Ensemblers.** Combina varios clasificadores para obtener un mejor rendimiento general en comparación con un único modelo aplicado de forma independiente.

- **Aprendizaje no supervisado** En el aprendizaje no supervisado, el output o clase no se proporciona en los datos de entrenamiento. Estos modelos se utilizan para el análisis y preprocesamiento de datos complejos. Las principales técnicas no supervisadas son [58, 39]:

  - **Clustering.** El clustering se centra en encontrar estructuras o clusters en los datos.
  - **Reducción de la dimensionalidad** Estas técnicas intentan encontrar un subconjunto más pequeño de las características originales con una pérdida de información mínima.

Evaluación de Modelos Machine Learning: Cross Validation

Los modelos de machine learning aprenden de las características y relaciones existentes en los con los que son entrenados. Así la capacidad de generalización de las predicciones de un modelo de machine learning dependerá de como de generales son los datos con los que es entrenado.

Usualmente en machine learning los modelos son entrenados mediante un cojunto de training y evaluados mediante uno de test. De esta forma, la distribución de las características esta sujeta una eleccion aleatoria de las instancias que formaran dichos conjuntos, propiciando que el modelo sea entrenado o evaluado con datos sesgados o con una capacidad de generalización menor. Y por consiguiente, creando un modelo menos robusto hacia nuevos datos y una evaluación de estos menos fiable.

Cross validation es una técnica de evaluación de modelos que permite obtener unos resultados mas consistentes que la típica partición aleatoria de los datos en entrenamiento y test [48, 76].

A la vez, cross validation tiene distintas variaciones en base al número de particiones del conjunto de datos, la distribución de las clases en estos o del tipo de datos con el que se trabaje. No obstante, el conjunto de técnicas de cross validation comparten el siguiente principio:

Algoritmo Cross Validation

- Realizar k particiones del conjunto total de datos.
- Repetición de un bucle k veces
- Entrenamiento de un modelo usando k-1 particiones del conjunto inicial de datos y evaluando en la partición restante mediante una métrica de rendimiento seleccionada.

  El rendimiento final del modelo es dado por la media de las k resultados obtenidos en cada iteración del bucle.

Métricas de Evaluación de Modelos de Machine Learning

La métricas de evaluación de modelos nos permiten cuantificar el rendimiento de estos en base al valor de sus predicciones en contraste con el valor real.

Dependiendo del modelo o del problema al que se aplique los valores de sus predicciones pueden ser continuos, de probabilidad o enteros.

Siendo un problema de clasificación en el que se aplica este proyecto, nos centraremos en métricas para el rendimiento de modelos aplicadas en este tipo de situaciones.

En problemas de clasificación el valor de las predicciones de los modelos son numéricos enteros o de probabilidad a los que se les aplica un cierto umbral para categorizar su respuesta.

5.2. Conceptos Bias en Machine Learning

En los siguientes puntos se explicarán los principales enfoques de detección y mitigación de bias en machine learning [65].

Técnicas de mitigación

La mitigación de bias en los sistemas de machine learning tiene como objetivo reducir el efecto del desequilibrio en la distribución de características en la predicción de los modelos.

Dependiendo de la fase del flujo de trabajo de machine learning en la que se aplique la técnica, se dividen en técnicas aplicadas a nivel de datos, algoritmo o híbrida.

Técnicas a nivel de datos

Las técnicas a nivel de datos se aplican a los datos de entrenamiento durante la etapa de preprocesamiento para que la distribución de clases sea más equilibrada [37]. Básicamente, reducen o aumentan las muestras en las clases mayoritarias o minoritarias, respectivamente. Estas técnicas también se denominan técnicas de remuestreo.

Las principales ventajas de las técnicas a nivel de datos son: 1) la aplicación de técnicas de remuestreo es independiente de los modelos machine learning [50], 2) relativamente fácil de aplicar.

Sin embargo, no se excluyen desventajas como la eliminación de patrones significativos o información útil asociada al submuestreo [34], o el coste computacional y el riesgo de sobreajuste debido a la duplicación de muestras de clases minoritarias durante el sobremuestreo [77].

El submuestreo y el sobremuestreo pueden aplicarse simultáneamente reduciendo las limitaciones antes mencionadas [37].

Por ejemplo, SMOTE-ENN utiliza la técnica de sobremuestreo de minorías sintéticas SMOTE en combinación con la técnica de submuestreo Edited Nearest Neighbors ENN. SMOTE crea nuevas muestras sintéticas mediante la interpolación de los K Nearest Neighbors (kNN) de cada una de las muestras minoritarias [13]. ENN evalúa cada muestra mediante k-NN utilizando el resto de muestras. Todas las muestras clasificadas incorrectamente se descartarán, y las muestras restantes formarán el conjunto de datos editado [8].

Técnicas a nivel de algoritmo

Las técnicas a nivel de algoritmo modifican o alteran los modelos de machine learning a nivel de algoritmo, añadiendo un coste o peso a la clase mayoritaria [65]. Los métodos a nivel de algoritmo pueden dividirse en métodos basados en el reconocimiento (se entrenan con muestras de la clase mayoritaria y reconocen la clase minoritaria como valores atípicos), métodos sensibles al coste de clasificación (ajustan el coste de la clasificación errónea para

equilibrar las clases mayoritaria y minoritaria) y ensemblers (utilizan y combinan la predicción de múltiples modelos mediante mecanismos como bagging, stacking o boosting).

Basadas en el reconocimiento

Los métodos basados en el reconocimiento modelan patrones o características de los datos de entrenamiento compuestos por una clase, con el fin de predecir posibles valores atípicos [44, 65].

Los métodos basados en el reconocimiento son una modificación de los clasificadores de una clase. Estos se adaptan a los problemas de desequilibrio de clases utilizando la clase mayoritaria como no atípica, y detectando las clases minoritarias como valores atípicos.

Los clasificadores de una clase tienen un buen rendimiento en conjuntos de datos en los que la clase minoritaria carece de estructura o el número de instancias es demasiado bajo o inexistente durante la fase de entrenamiento. Sin embargo, como los clasificadores se entrenan utilizando sólo una muestra de clase, se descarta toda la información sobre la clase minoritaria.

Coste-sensitivos

En los métodos sensibles al coste, se asigna un coste de clasificación errónea diferente a cada clase [66, 44]. Este algoritmo intenta minimizar el coste de la clasificación errónea en lugar de cualquier otra métrica de evaluación.

En el campo médico es los métodos sensibles al coste de clasificación son especialmente interesante debido a que en escenarios como el cáncer, es más peligrosa y costosa la clasificación errónea de un falso negativo (paciente con cáncer pero clasificado como sano), que un falso positivo.

Los enfoques basados en costes incluyen la entropía cruzada ponderada, la función de pérdida de dados multiclase y la pérdida focal.

Ensemblers

Los métodos basados en el aprendizaje conjunto combinan el conocimiento de múltiples aprendices débiles e integran sus resultados en un modelo robusto utilizando diferentes mecanismos de votación con el objetivo de obtener un mejor rendimiento que utilizando los algoritmos constituyentes individualmente [19, 44].

En función del mecanismo de votación utilizado, pueden dividirse en tres categorías principales: bagging, stacking y boosting.

Técnicas Hibridas

Las técnicas híbridas combinan métodos a nivel de datos en la fase de preprocesamiento con métodos a nivel de algoritmo durante el desarrollo del modelo [3, 37].

En primer lugar, se procesan los datos y se equilibran las clases. A continuación, los modelos machine learning se alteran o modifican internamente. De este modo, los modelos no están demasiado sesgados debido al ajuste previo de la distribución de clases.

Técnicas de Detección de Bias

Un aspecto crucial para abordar el problema del sesgo en machine learning es poder cuantificar el grado de equidad en los resultados de predicción de los modelos de machine learning [50].

Las métricas comunes de machine learning que evalúan el rendimiento general de los modelos de machine learning sin tener en cuenta los resultados del modelo para las clases minoritarias pueden conducir a resultados poco fiables en conjuntos de datos desequilibrados.

En este sentido, para medir el sesgo en el rendimiento de los modelos de machine learning, se han desarrollado notaciones matemáticas específicas denominadas técnicas de detección de sesgos o métricas de equidad [65].

5.3. Algoritmos Machine Learning Aplicados en el TFG

5.3.1. Light Gradient Boosting Machine (LightGBM)

El algoritmo Light Gradient Boosting machine (LGBM) es un tipo de ensemble de machine learning dentro de la categoría gradient boosting la cual utiliza un ensemble de árboles de decisión para realizar sus predicciones. [71, 23].

Los algoritmos de gradient boosting utilizan un modelo inicial débil que se mejora secuencialmente añadiendo modelos débiles que intentan reducir los errores de predicción de todos los modelos anteriores. El proceso de gradient boosting consta de tres componentes principales:

- **Función de pérdida**. La función de pérdida mide el rendimiento del modelo a la hora de modelar los datos. En otras palabras, la función de pérdida cuantifica la suma de errores o residuos para cada instancia de los datos. El objetivo del modelo es minimizar la función de pérdida. Según el tipo de problema, se aplica una función de pérdida diferente. Para problemas de regresión se utiliza el error medio, mientras que para problemas de clasificación se utiliza una función de pérdida log-verosimilitud binomial negativa [33, 49].

- **El modelo débil.** El modelo débil realiza las predicciones. Normalmente, los modelos débiles son árboles de decisión en los algoritmos de boosting. Los modelos débiles funcionan ligeramente mejor que una elección aleatoria, principalmente porque suelen estar restringidos para evitar un ajuste excesivo a los datos.

- **Modelos aditivos**. Los modelos aditivos se añaden secuencialmente utilizando el descenso de gradiente para minimizar la pérdida del modelo anterior. Cada iteración actualiza los residuos asociados a la diferencia entre el valor predicho del último modelo y el observado en los datos. El nuevo modelo añadido intenta minimizar los resultados de error del anterior.

LGBM supera los problemas de eficacia y escalabilidad de su predecesor Gradient Boosting Decision Tree (GBDT) en grandes conjuntos de datos de alta dimensionalidad [60].

Además, LGBM tiene un alto rendimiento a un menor coste computacional en contraste con otros algoritmos de boosting reconocidos como XGBoost.

Las principales características estructurales que permiten a LGBM alcanzar un alto rendimiento de manera eficiente son [23, 36]:

- **Gradient-based One-side Sampling (GOSS)**. La idea principal de GOSS es asignar a las muestras mal clasificadas una mayor probabilidad de ser incluidas en el conjunto de datos en las siguientes iteraciones. El valor absoluto del gradiente determina el grado de clasificación errónea. Las muestras con mayores gradientes provocan mayores variaciones en la función de pérdida, dándoles prioridad durante el proceso de entrenamiento. Por el contrario, las muestras con un gradiente bajo se descartan aleatoriamente.

- **Exclusive Feature Bunding (EFB).** El objetivo de EFB es reducir eficazmente el número de características. Las características mutuamente excluyentes se agrupan en una única característica. De este modo, al igual que en GOSS, se reduce el tiempo de entrenamiento del modelo.

- **Leaf-wise tree growth algorithm**El algoritmo de crecimiento de árbol por hojas tiende a alcanzar menores pérdidas que los métodos por profundidad. El algoritmo por hojas selecciona las hojas con mayor variación en la función de pérdida para su crecimiento. El crecimiento por hojas tiende a sobreajustarse cuando el número de muestras es bajo. La modificación de la profundidad máxima del árbol permite limitar el sobreajuste.

- **Processing of categorical features** LGBM realiza una división óptima de las características categóricas mediante un método de agrupación. Al evitar el procesamiento de características categóricas, los datos no son fragmentados debido a transformaciones numéricas con el consiguiente ahorro de tiempo de procesamiento.

- **Histogram-based splitting algorithm** El algoritmo de división basado en histogramas agrega los valores de una determinada característica numérica en grupos. De este modo, se mejora el tiempo de entrenamiento y el uso de memoria.

Los pasos realizados en LGBM son los siguientes [33, 64]:
Input: Data de entrenamiento $(X_i, y_i)_{i=1}^{N}$
Output: Modelo LGBM $\widehat{y}_i^{(t)}$, being t the t-th iteration siendo t la t-th iteración

- **Paso 1.** Agrupar características que se excluyen mutuamente aplicando EFB.

- **Paso 2.** Construir un árbol de decisión como aprendiz débil. El aprendiz débil se define como una constante.

$$\widehat{y}_i^{(0)} = f_0 = 0$$

- **Paso 3.** Construir un nuevo árbol de decisión minimizando la función de pérdida. La selección de la función de pérdida depende del tipo de problema aplicado (categórico, continuo, supervivencia, etc...).

$$f_t(X_i) = argmin_{f_t} L(y_i, \widehat{y}_i^{(t-1)} + f_t(X_i)$$

- **Paso 4.** Reajustar los datos aplicando GOSS.

- **Paso 5.** Construir un nuevo árbol de decisión de forma aditiva.

$$\widehat{y}_i^{(t)} = \widehat{y}_i^{(t-1)} + f_t(X_i)$$

- **Paso 6.** Repita los pasos 3,4 y 5 hasta que el modelo alcance la condición de parada.

- **Paso 7.** Obtener el modelo final.

$$\widehat{y}_i^{(t)} = \sum_{t=0}^{M-1} f_t(X_i)$$

5.3.2. Support Vector Machines (SVM)

Support Vector Machines (SVM) es un algoritmo de machine learning supervisado con aplicación el problemas de clasificación y regresión [53]. Basicamente, intenta encontrar un hiperplano que maximize la distancia entre los puntos de las distintas clases del conjunto de datos haciendo uso de soportes vectoriales. Los soportes vectoriales definen la distancia entre los puntos de las distintas clases y el hiperplano de clasificación.

Los algoritmos SVM pueden ser lineales y no lineales. Dentro de los SVM lineales existen dos subcategorias: hard margen and soft margen. SVMs de hard margen separan completamente los clases en un conjunto de datos mediante un hiperplano. Por otro lado, SVMs de soft margen introducen una penalización a aquellos puntos incorrectamente clasificados por el hiperplano.

Cuando el espacio de características no es linealmente separable SVM aplica distintas funciones kernels para estos espacios, en una dimensión superior que lo permita.

En base al tipo de funcion kernel utilizada existen distintos tipos de SVMs. Los kernels más comunes son polinómicos, radial o sigmoidal.

Las principales ventajas de SVM son su robustez ante outliers y su óptimo rendimiento en conjunto de datos con un elevado número de dimensiones.

5.4. Tecnicas y Métricas de Evaluación de Modelos Machine Learning Aplicadas en el TFG

5.4.1. Evaluación de Modelos Machine Learning: 5x2 Cross Validation

5x2 es un tipo de k-fold cross calidation que permite una evaluación de modelos y una estimación de hiperparámetros más robusta que k-fold cross validation [18].

Los pasos del algoritmo de 5x2 cross validation son los siguientes:

1. Division en 5 folds o particiones estratificadas en base a la clase de estudio.
2. Bucle externo. Itera tantas veces como el número de particiones del conjunto de datos inicial, en este caso realiza 5 iteraciones. Por cada una de las iteraciones destina una partición distinta a la evaluación del modelo y el resto como conjunto de datos de entrenamiento.
3. Bucle interno. Por cada iteración del bucle externo recibe el conjunto de entrenamiento y lo divide en 2 particiones de igual tamano, con la clase de estudio estratificada. Internamente realiza 2 iteraciones destinando en cada una de ellas una partición distinta a la evaluación del modelo y selección de hiperparámetros y el resto como conjunto de entrenamiento.Por cada combinación de hiperparámetros a evaluar se repite bucle.

   El valor del rendimiento del modelo con una combinación específica de hiperparámetros es la media de los resultados de rendimiento obtenidos en las 2 iteraciones del bucle. El método de búsqueda de la combinación más óptima aplicado es la búsqueda exhaustiva o grid search.
4. Bucle externo. La mejor combinación de hiperparámetros obtenida en el bucle interno es usada para entrenar un modelo con el total del conjunto de entrenamiento de la iteración en curso. Posteriormente el modelo es evaluado en el conjunto de validación asociado a la misma iteración que con la que ha sido entrenado.

   El rendimiento global del modelo es obtenido a partir de la media de los resultados obtenidos en el total de las iteraciones del bucle externo.

Nos basaremos para la evaluación de los modelos en las métricas Accuracy (ACC), Area Under ROC curve (AUC) y F1-score.

5.4.2. Métricas de Evaluación de Modelos de Machine Learning: Accuracy

La métrica de evaluación Accuracy (ACC) mide el ratio de predicciones correctas de un modelo entre el total de predicciones realizadas.

El rango de valores que puede alcanzar ACC es [0,1], siendo cero asociado a ninguna predicción correcta entre el total y de predicciones, y 1, todas la predicciones correctas del total realizadas.

5.4.3. Métricas de Evaluación de Modelos de Machine Learning: AUC

AUC mide la capacidad de un modelo para distinguir entre clases. Para ello hace uso de la curva ROC. La curva ROC define ratio entre true positives TP y false negatives FN en un determinado umbral de probabilidad de clasificación.

El rango de valores de AUC es [0,1], determinando 0 un modelo que clasifica las clases de forma opuesta, y 1, aquel con una capacidad de distinción entre clases total. A su vez, un valor AUC de 0.5, define un modelo en el que el total de sus predicciones están asociadas una sola clase.

5.4.4. Métricas de Evaluación de Modelos de Machine Learning: F1-Score

F1-score o también llamada la media armónica de la precisión y recall del modelo, se calcula a partir de la siguiente fórmula [63]:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Siendo la precisión y recall:

- Precisión: Proporción de verdaderos positivos sobre el total de predicciones positivas realizas.
- Recall: Proporción de verdaderos positivos sobre el total de valores positivos.

5.5. Métricas de Detección de Bias en Modelos Machine Learning Aplicadas en el TFG

En las siguientes subsecciones se describe un conjunto de técnicas de detección de sesgos aplicadas previamente en machine learning aplicado al contexto biomedico:

5.5.1. Métricas de Detección de Bias en Modelos Machine Learning: Equal Opportunity

Cuando existe una clase de preferencia en la predicción del modelo, la precisión para la clase de preferencia debe ser igual entre el grupo protegido y el no protegido para que se cumpla la igualdad de oportunidades [74, 31]. La disparidad de True Positive Rate TPR o False Negative Rate FNR entre grupos debería ser cero en circunstancias ideales. La igualdad de oportunidades (en este caso, la modelización de TPR) se escribe formalmente como:

$$P(\widehat{Y} = 1|A = 0, Y = 1) = P(\widehat{Y} = 1|A = 1, Y = 1)$$

siendo $\widehat{Y}$ el valor predicho, A la clase protegida e Y el valor real.

5.5.2. Métricas de Detección de Bias en Modelos Machine Learning: Error Rate

Calcula la tasa de error de las predicciones en los grupos protegidos y desprotegidos dividiendo el número de predicciones incorrectas por el número total de predicciones [31]. La disparidad de la tasa de error entre grupos debería ser cero en una situación ideal.

5.5.3. Métricas de Detección de Bias en Modelos Machine Learning: Equal Odds

Tanto en los grupos protegidos como en los no protegidos, el TPR y el FPR deben ser iguales [74, 31]. Se impone que los individuos que obtienen resultados similares reciban el mismo trato. Así, la probabilidad de recibir un valor positivo es independiente del grupo y relacionada con el individuo. La igualdad de probabilidades es la métrica de equidad más restrictiva de las antes mencionadas, al tiempo que garantiza el máximo nivel de equidad algorítmica.La igualdad de oportunidades se escribe formalmente como:

$$P(\widehat{Y} = 1|A = 0, Y = y) = P(\widehat{Y} = 1|A = 1, Y = y), y \in 0, 1$$

siendo $\widehat{Y}$ el valor predicho, A la clase protegida e Y el valor real.

6
Implementación

En las siguientes subapartados se detalla el proceso de implementación realizado.

6.1. Carga de Datos

El conjunto de datos utilizado *Colorectal Adenocarcinoma (TCGA, PanCancer Atlas)* perteneciente al repositorio TCGA, fue descargado desde el portal *www.cbioportal.org*. El conjunto total de datos descargado contenía un total de 22 subconjuntos tabulares de datos con información clínica y ómica asociada a 594 muestras de cancer colorectal.

Del total de los 22 subconjuntos se seleccionó 5 de ellos: data clinical patient, data clinical sample, data microbiome, data methylation hm27 hm450 merged y data mrna seq v2 rsem.

Descripción de los conjuntos de datos utilizados:

- **data clinical patient**: Información clínica asociada a cada paciente (edad, estadio, raza, dias de seguimiento, etc...).
- **data clinical sample**: Información asociada a cada muestra (paciente id, tipo de cancer, tipo de tejido, etc...)
- **data microbiome**: Firmas microbianas (log-cpm) en contraste con los estudios de secuenciación del transcriptoma completo del TCGA.
- **data methylation hm27 hm450 merged**: Valores de metilación normalización entre plataformas (hm27 y hm450).
- **data mrna seq v2 rsem**: Valores de expresión mRNA normalizados mediante RSEM.

Una vez cargados en el workspace los subconjuntos de datos seleccionados, se renombraron de la siguiente forma: data clinical patient (clinical data), data clinical sample (clinical sample data), data microbiome (microbiome data), data methylation hm27 hm450 merged (methylation data) y data mrna seq v2 rsem (genomic raw data)

```
# Load clinical sample data
clinical_sample_data <- read.delim(RAW_CLINICAL_SAMPLE, header = FALSE, sep = "\t", dec =
    ".", fill = TRUE)

# Load clinical data
clinical_data <- read.delim(RAW_CLINICAL_PATIENT, header = FALSE, sep = "\t", dec = ".",
    fill = TRUE)

# Load methylation data
methylation_data <- read.table(RAW_METHYLATION, header = FALSE, sep = "\t", dec = ".",
    fill = TRUE)

# Load genomic raw data
genomic_data <- read.table(RAW_GENOMIC, header = FALSE, sep = "\t", dec = ".", fill = TRUE
    )

# Load microbiome data
microbiome_data <- read.table(RAW_MICROBIOME, header = FALSE, sep = "\t", dec = ".", fill
    = TRUE)
```

Listing 1: Carga de datos

6.2. Formato de los Datos

Debido a las diferencias en la organización de filas y columnas en los 5 subconjuntos de datos, se realizaron las siguientes manipulaciones:

1. Posicionamiento inicial de nombres de columnas y filas.
2. Asignación de los atributos clinicos como nombre de columnas en los conjuntos de datos clinical data y clinical sample.

3. Asignación sample id como nombre de columnas en los conjuntos de datos ómicos.
4. Adicción de la columna sample id al conjunto de datos clinical data procedente del conjunto de datos clinical sample data. Para su correcta ordenación se utilizó como clave el atributo patient id existente en ambos subconjuntos.
5. Asignación de la columna sample id como nombre de filas en los conjuntos clinical data y clinical sample data.
6. Asignación como nombres de fila las firmas microbianas, hugo symbol y CpG identificador, en los conjuntos de datos microbiome data, genomic raw data y methylation data respectivamente.
7. Eliminación de aquellas observaciones con sample id duplicado.
8. Ordenación por orden alfabético los nombres de filas o columnas asociados a sample id.

6.3. Análisis de los Datos

Clinical data

Filas	594
Columnas	3
Columnas Discretas	3
Columnas Continuas	0
Total Columnas Nulas	0
NAs	239
Observaciones Totales	1782
Filas Completas	361

Cuadro 3: Summary clinical data

- **Distribución Valores Nulos**

La proporción de valores nulos en la variable Race representan el 39 % de las observaciones del conjunto original de datos. En las variables OS MONTHS y OS STATUS los valores nulos representan el 1 % de las observaciones.

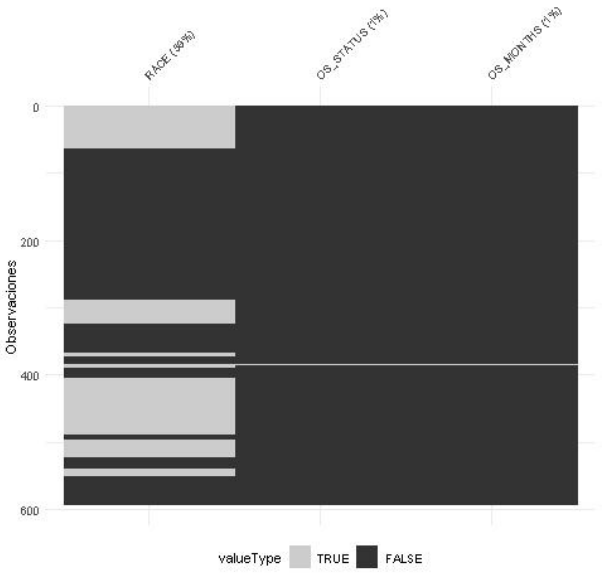
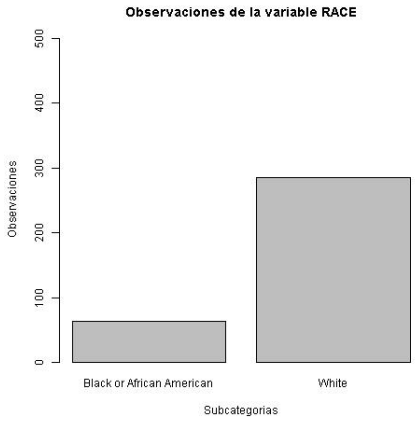


Figura 3: Proporción NAs en clinical data

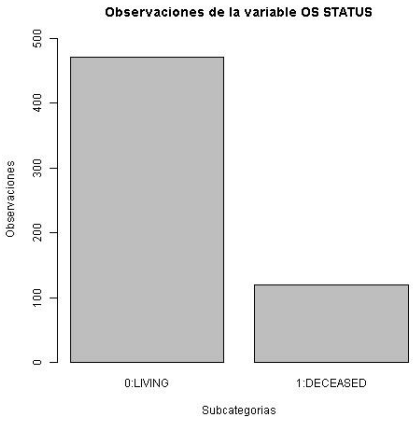
- **Distribución Variable Race**

Siendo Race la variable sensible a la cual le aplicaremos el análisis de bias en la predicción de los modelos de machine learning, estudiaremos mas profundamente sus distribución. Como se ha comentado en el apartado anterior, los valores nulos están representado en el 39 % de sus observaciones.

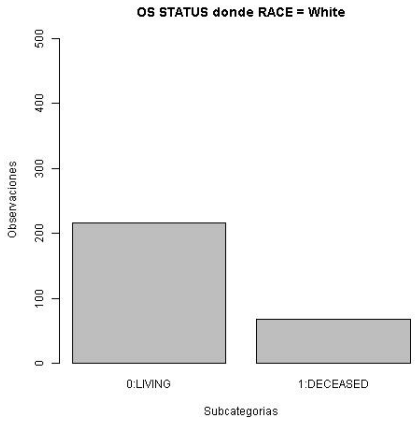
Aspecto fundamental en el análisis de bias es la proporción de las distintas categorias en la variable sensible. Siendo 81,6 % de observaciones Race = White y el 18,4 % Race = Black or African American.



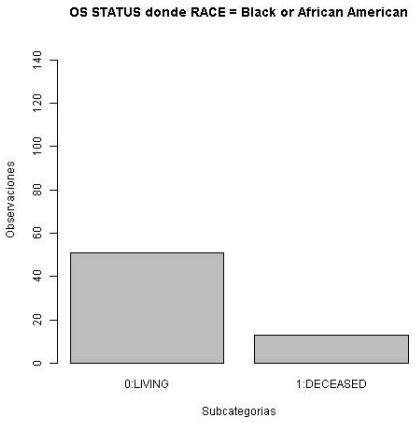
(a) Barplot variable RACE



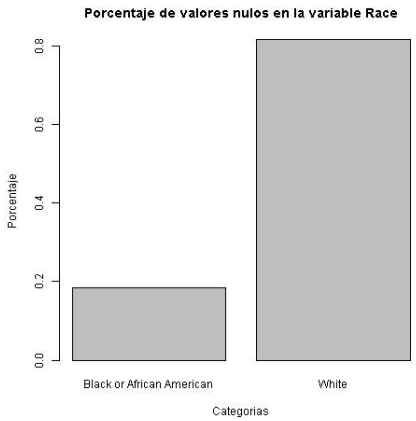
(b) Barplot variable OS STATUS



(c) Barplot variable OS STATUS donde RACE = White



(d) Barplot variable OS STATUS donde RACE = Black or African American



(e) Porcentaje de valores nulos en la variable Race

Figura 4: Barplots variable RACE

Clinical sample data

Filas	594
Columnas	18
Columnas Discretas	17
Columnas Continuas	0
Total Columnas Nulas	1
NAs	753
Observaciones Totales	10692
Filas Completas	0

Cuadro 4: Summary clinical sample

- **Distribución Valores Nulos**

  Siendo el uso del conjunto de datos clinical sample data, el mapeo entre las variables PATIENT ID y SAMPLE ID, nos centramos en observar la proporción de valores nulos en dichas variables.

  Tras el análisis, el conjunto de datos clinical sample id no presenta observaciones con valores nulos en las variables PATIENT ID y SAMPLE ID.

Microbiome data

Filas	1406
Columnas	583
Columnas Discretas	583
Columnas Continuas	0
Total Columnas Nulas	0
NAs	0
Observaciones Totales	819698
Filas Completas	1406

Cuadro 5: Summary microbiome data

- **Distribución Valores Nulos**

  El conjunto de datos microbiome data no presenta valores nulos en ninguna de sus observaciones.

Methylation data

Filas	11256
Columnas	590
Columnas Discretas	590
Columnas Continuas	0
Total Columnas Nulas	0
NAs	3581
Observaciones Totales	6641040
Filas Completas	10567

Cuadro 6: Summary methylation data

- **Distribución Valores Nulos**

Nulos por Observación	Observaciones	Porcentaje de Observaciones
0	116	19.6610169491525
1	111	18.8135593220339
2	77	13.0508474576271
3	59	10
5	40	6.77966101694915
4	36	6.10169491525424
6	21	3.55932203389831
7	14	2.3728813559322
11	14	2.3728813559322
8	11	1.86440677966102
9	9	1.52542372881356
14	8	1.35593220338983
12	7	1.1864406779661
13	7	1.1864406779661
10	6	1.01694915254237

Cuadro 7: Porcentajes de valores nulos en el conjunto de datos methylation data

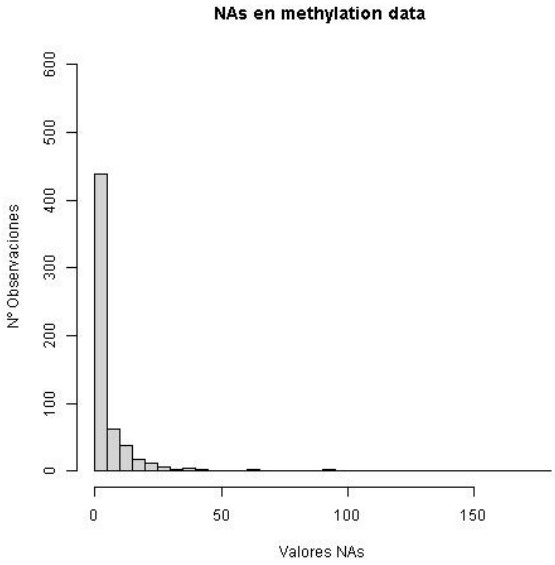


Figura 5: Histograma de valores nulos en el conjunto de datos methylation data

Entre las observaciones mas significativas en el análisis de los valores nulos en el conjunto de datos de metilación destacamos: 1) las observaciones con presencia de valores nulos no suelen tener más de 50 valores nulos por cada observación, 2) el 19.9 % de las observaciones no presentan valores nulos, 3) entre aquellas observacion con presencia de valores nulos, el 60 % de estas presentan menos de 7 valores nulos por observación.

Genomic data

Filas	20512
Columnas	592
Columnas Discretas	592
Columnas Continuas	0
Total Columnas Nulas	0
NAs	684632
Observaciones Totales	12143104
Filas Completas	17496

Cuadro 8: Summary genomic data

- **Distribución Valores Nulos**

La distribucion de valores nulos en el conjunto de datos genomic raw data es representada por dos grupos, 365 observaciones sin presencia de valores nulos y 227 observaciones con 3016 valores nulos.

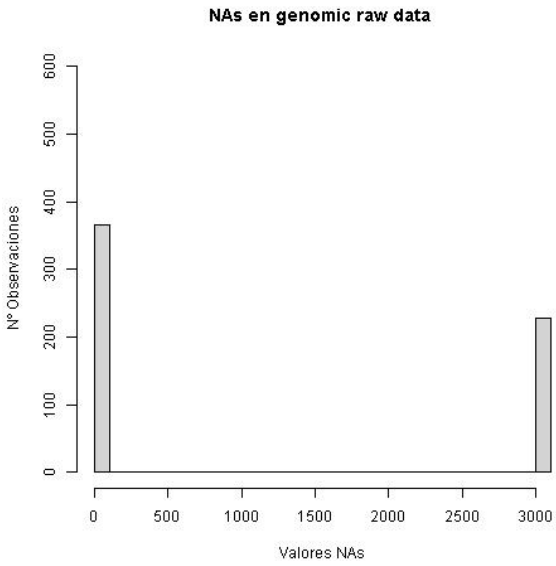


Figura 6: Histograma valores nulos en conjunto de datos genomic raw data

6.4. Preprocesado de Datos

En base al conjunto de base datos (clínico u ómico), diferentes criterios de preprocesado de datos fueron aplicados.

- **Imputación de valores nulos en clinical data** Dado el objetivo de clasificación de la prognosis, aquellas observaciones con valores nulos en las variables OS STATUS y OS MONTHS fueron eliminadas. Dado el objetivo de detección de sesgo en la predicción de los modelos, aquellas observaciones con valores nulos en la variable RACE fueron eliminadas.

- **Imputación de valores nulos en los conjuntos de datos ómicos** Variables (firmas microbianas, hugo symbol o CpG identificador) con un porcentaje de valores nulos mayor al 20 % fueron eliminadas.

	Clinical	Genomic	Methylation	Microbiome
Filas	348	17496	10567	1406
Columnas	3	592	590	583
Columnas Discretas	2	0	0	0
Columnas Continuas	1	592	589	582
Total Columnas Nulas	0	0	1	1
Filas completas	348	1796	0	0
NAs	0	0	10567	1406
Observaciones Totales	1044	10357632	6234530	819698

Cuadro 9: Summaries Conjuntos de Datos Post-Preprocesado

```
#### Genomic raw data imputation

# Delete variables with na in more than 20% of the samples
perc <- PERC
col_perc <- perc * ncol(genomic_data) / 100
genomic_data <- genomic_data[rowSums(is.na(genomic_data)) < col_perc, ]

#### Microbiome data imputation

# Delete variables
perc <- PERC
col_perc <- perc * ncol(microbiome_data) / 100
microbiome_data <- microbiome_data[rowSums(is.na(microbiome_data)) < col_perc, ]

#### Methylation data imputation

# Delete variables with na values
methylation_data <- methylation_data[rowSums(is.na(methylation_data)) < 1, ]
```

Listing 2: Preprocesado

6.5. Selección de Observaciones a Instante de Tiempo Fijo

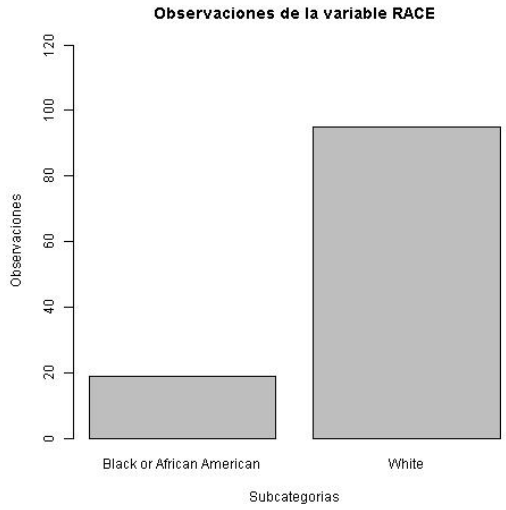
La selección de observaciones a instante de tiempo fijo es el proceso por el cual a partir del total de observaciones se seleccionan aquellas de las que se tenga información del estado de supervivencia en el instante de tiempo fijado. Para ello se tiene en cuenta la relación entre el estado de supervivencia, periodo de tiempo de seguimiento y el instante de tiempo fijado. Las variables OS STATUS, OS MONTHS definen el estado de supervivencia y periodo de seguimiento respectivamente.

En la selección de observaciones a instante de tiempo fijo, se aplicaron las siguientes condiciones:

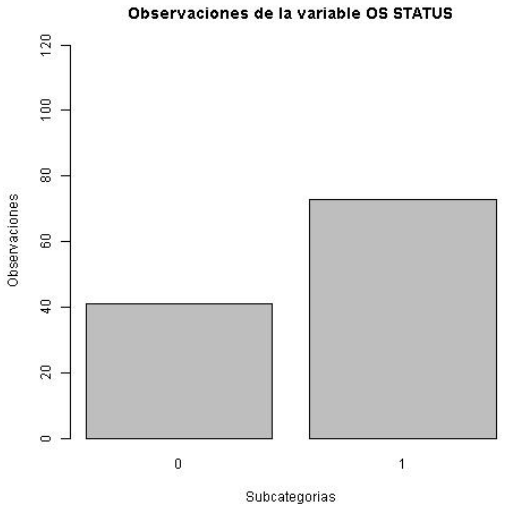
En la selección de observaciones a instante de tiempo fijo, se aplicaron las siguientes condiciones:

1. Seleccion **instante de tiempo fijo en 5 años (1825 días)**
2. Observaciones con tiempo de estudio menor o igual a 1825 días y no hallan sufrido el evento, no se tiene información del estado en el tiempo fijado y se elimina la observación.

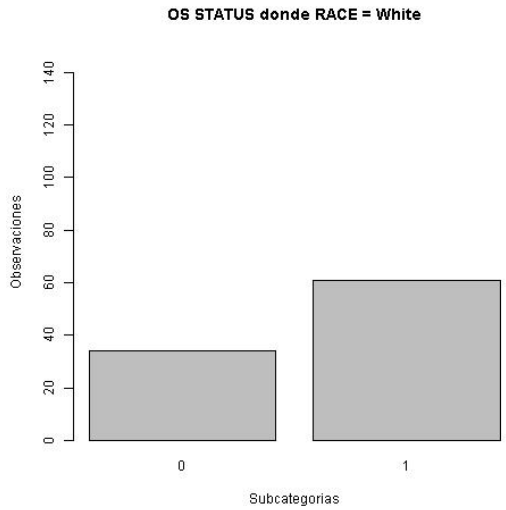
   Se define como evento, el valor positivo en la variable OS STATUS. OS STATUS define el estado de supervivencia en el tiempo de estudio.
3. Observaciones con tiempo de estudio menor o igual a 1825 días y hallan sufrido el evento, conservan dicho estado en el instante fijado
4. Observaciones con tiempo de estudio mayor a 1825 días y hallan sufrido el evento, en el instante fijado se le asigna evento no sufrido
5. Observaciones con tiempo de estudio mayor a 1825 días y no hallan sufrido el evento, conservan dicho estado en el instante fijado
6. Actualizamos tiempo de las observaciones que superen el tiempo fijado



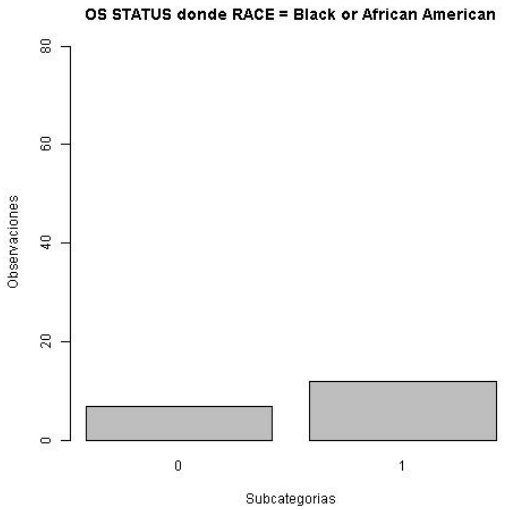
(a) Barplot variable RACE



(b) Barplot variable OS STATUS



(c) Barplot variable OS STATUS donde RACE = White



(d) Barplot variable OS STATUS donde RACE = Black or African American

Figura 7: Barplots clinical data fixed

Filas	Columnas	Columnas Discretas	Columnas Continuas	NAs
113	3	2	1	0

Cuadro 10: Summary clinical data fixed

```r
#' Keeps observations that match the time condition
#'
#' @description
#' Selects observations for which there is information on the survival status at a fixed
    time instant, considering the relationship between survival status, follow-up time
    period, and the fixed time point.
#'
#' @param df_clinical A Clinical data frame object with the columns OS_STATUS and OS_
    MONTHS, and sample_id as row names.
#' @param int_time An integer defining the fixed time instant in days.
#'
#' @return A data frame object. The output has the following characteristics:
#'
#' * Observations that match the time condition.
#' * Same columns as df_clinical with OS_STATUS and OS_MONTHS modified to the new time
    point.
#'
#' @export
#'
#' @examples
create_data_fixed <- function(df_clinical, int_time) {
  tryCatch(
    expr = {
      # Check arguments
      if (!is.numeric(int_time)) {
        stop("Non-numeric time")
      } else if (int_time < 0) {
        stop("Non-positive time value")
      } else if (!is.data.frame(df_clinical)) {
        stop("Non-data frame object data")
      } else if (any(!(c("OS_MONTHS", "OS_STATUS") %in% colnames(df_clinical)))) {
        stop("Cannot find 'OS_MONTHS' or 'OS_STATUS' in column names")
      } else {
        ## Create a dataframe at the fixed time point
        clinical_data_fixed <- df_clinical

        ## Observations with a study time less than or equal to 1825 days and have not
    suffered the event; there is no information on the status at the fixed time, and the
    observation is eliminated.
        clinical_data_fixed <-
          clinical_data_fixed[-c(
            which(
              clinical_data_fixed$OS_MONTHS <= int_time &
                clinical_data_fixed$OS_STATUS == 0
```

```
          )
        ),]

      ## Observations with a study time of less than or equal to 1825 days and have
   suffered the event; retain this state at the fixed time.
      clinical_data_fixed$OS_STATUS[which(clinical_data_fixed$OS_MONTHS <= int_time &
                                          clinical_data_fixed$OS_STATUS == 1)] <- 1

      ## Observations with a study time greater than 1825 days and have suffered the
   event; they have not suffered it yet at the fixed date.
      clinical_data_fixed$OS_STATUS[which(clinical_data_fixed$OS_MONTHS > int_time &
                                          clinical_data_fixed$OS_STATUS == 1)] <- 0

      ## Observations with a study time greater than 1825 days and have not suffered the
    event; retain this state at the fixed int_time.
      clinical_data_fixed$OS_STATUS[which(clinical_data_fixed$OS_MONTHS > int_time &
                                          clinical_data_fixed$OS_STATUS == 0)] <- 0

      # Update the time of observations that exceed the set time
      clinical_data_fixed$OS_MONTHS[which(clinical_data_fixed$OS_MONTHS > int_time)] <-
        int_time

      ## Subset of the population followed up to a fixed time point
      clinical_data_fixed <-
        clinical_data_fixed[which(clinical_data_fixed$OS_MONTHS <= int_time),]

      return(clinical_data_fixed)
    }
  },
  error = function(e) {
    print(sprintf(
      "An error occurred in create_data_fixed at %s : %s",
      Sys.time(),
      e
    ))
  }
 )

}
```

Listing 3: Funcion create data fixed

6.6. Selección de Variables

Expresión diferencial

El análisis de expresión diferencial se aplicó a los conjuntos de datos ómicos, considerando variables diferencialmente expresadas aquellas con **p.value <0.05**

La selección de variables mediante analisis de expresión diferencial se realizó utilizando el paquete de Bioconductor *Limma* [57] de R.

Limma a partir del ajuste de un modelo lineal generalizado para cada variable pemite aplicar diferentes diseños experimentales y probar distintas hipótesis. Además, haciendo uso de métodos bayesianos provee resultados estables en conjuntos de datos con reducido número de observaciones.

```
#' Apply differential analysis to omics data frames
#'
#' @param df_clinical Data frame clinical_data derived of create_data_fixed
#' @param dfs_omics List of omics data frames
#' @param p_val Double type defining the P-value to be applied has filter and selecting
    the most representative variables
#'
#' @return A list of 3 lists. The characteristics of each list are the following:
#'
#' * List 1 as ´diff_dfs´, store the omics data frames with the variables with p value < p
    _val
#' * List 2 as ´diff_names´, store a vector per each omic data frame of omic variables
    names which have passed the p value filter
#' * List 3 as ´top_table´, store a topTable element per each omic data frame with more
    information about it differential analysis
#' associated
#' @export
#'
#' @examples
diff_express <- function(df_clinical, dfs_omics, p_val) {
  tryCatch(
    expr = {
      # Check arguments
      if (!is.data.frame(df_clinical)) {
        error("Non data.frame type in df_clinical")
      } else if (!is.list(dfs_omics)) {
```

```
    error("Non list type in df_clinical")
  } else if (!is.double(p_val)) {
    error("Non double type in p_val")
  } else {
    results <-
      list(
        diff_dfs = list(),
        diff_names = list(),
        top_table = list()
      )

    for (i in 1:length(dfs_omics)) {
      # Transform omics dataframes as data matrix
      omics_matrix <- as.matrix(dfs_omics[i][[1]])

      # Expression set
      express_set <- Biobase::ExpressionSet(log2(omics_matrix + 1))

      # Update patient data in expression set
      Biobase::pData(express_set) <-
        df_clinical[c(Biobase::sampleNames(express_set)), ]

      # Model matrix
      mm <-
        model.matrix(~ 0 + OS_STATUS, data = Biobase::pData(express_set))

      # lmFit
      fit <- limma::lmFit(express_set, mm)

      # makeContrast
      contrast <-
        limma::makeContrasts(OS_STATUS0 - OS_STATUS1, levels = colnames(coef(fit)))

      # contrast.fit
      contrast_fit <- limma::contrasts.fit(fit, contrast)

      # eBayes
      bayes <- limma::eBayes(contrast_fit)

      # topTable
      top_table <- limma::topTable(bayes, sort.by = "P", n = Inf, )

      # Filter topTable based on p value
      diff_names <-
        rownames(top_table[which(top_table$P.Value < p_val), ])
```

```
          # Update results
          results$diff_dfs[i][[1]] <-
            dfs_omics[i][[1]][c(diff_names), ]
          results$diff_names[i][[1]] <- diff_names
          results$top_table[i][[1]] <- top_table
        }

        return(results)
      }
    },
    error = function(e) {
      print(sprintf("An error occurred in diff_express at %s : %s",
                    Sys.time(),
                    e))
    }
  )

}
```

Listing 4: Funcion diff express

Variable	**P.Value**
HSD17B10	2.32524924130574e-05
ZNF845	6.18228298166226e-05
SEMA3C	0.00011867987073504
G6PD	0.000142529808154183
SFT2D3	0.000240141478693963
HAUS7	0.000369065656049104
CLDN14	0.00036959004641193
PDE8A	0.000475280466389251
DOPEY2	0.000495846026738871
SH3D19	0.000523605046895921

Cuadro 11: Resultados expresión diferencial en genomic data (10 genes con p-value menor).

Variable	P.Value
cg01785339	2.24674719371477e-05
cg08312191	6.89723320176604e-05
cg24088751	0.000125820221788175
cg09890200	0.000135685318566912
cg05940463	0.000138025340854671
cg06490988	0.000164609572948849
cg06248182	0.000196344245448456
cg20346096	0.000210507395242455
cg26649834	0.000227713882168525
cg22796458	0.000229480233691522

Cuadro 12: Resultados expresión diferencial en methylation data (10 CpG con p-value menor).

Variable	P.Value
Alloiococcus	0.000735064728317517
Kushneria	0.000805474685031206
Ornithinimicrobium	0.00102141874783942
Maritalea	0.00143707185292297
Lentisphaera	0.00211249394216761
Lambdapapillomavirus	0.00306437390913374
Elusimicrobium	0.00359584229382626
Halobacillus	0.00423540804359464
Deferribacter	0.00431968397831372
Selenomonas	0.00504026568166995

Cuadro 13: Resultados expresión diferencial en microbiome data (10 microbiome variables con p-value menor).

Ganancia de información

Tras la selección de variables realizada mediante expresión diferencial, se realizó una **segunda selección de variables** aplicando el método ganancia de información. Se seleccionaron las **30 variables de cada conjunto de datos ómicos con mayor ganancia de información** respecto de la variable OS STATUS .

La seleccion de variables mediante ganancia de información se realizó aplicando el paquete *CORElearn* de R.

Variable	**IG**
G6PD	0.160916348032529
CASD1	0.148861680175459
SDSL	0.140629397218552
ABCD1	0.138190071248909
ERMAP	0.133388286607654
SERTAD4	0.130467921795087
DDX41	0.130324943003182
PSMB7	0.128572720894852
PSMD9	0.12591292305814
CTAGE6	0.125441237848683

Cuadro 14: Resultados ganancia de informacion en genomic data (10 genes con mayor IG).

Variable	IG
G6PD	0.160916348032529
CASD1	0.148861680175459
SDSL	0.140629397218552
ABCD1	0.138190071248909
ERMAP	0.133388286607654
SERTAD4	0.130467921795087
DDX41	0.130324943003182
PSMB7	0.128572720894852
PSMD9	0.12591292305814
CTAGE6	0.125441237848683

Cuadro 15: Resultados ganancia de informacion en genomic data + Race variable (10 genes con mayor IG).

Variable	IG
cg00309204	0.144101729299358
cg18139900	0.140629397218552
cg20426860	0.139847070955544
cg11481490	0.138190071248909
cg08907850	0.13384912183506
cg04284814	0.13384912183506
cg14709524	0.133496290602001
cg09809242	0.13306237608947
cg12903171	0.131930967720138
cg09890200	0.130837517821722

Cuadro 16: Resultados ganancia de informacion en methylation data (10 CpG con mayor IG).

Variable	IG
cg00309204	0.144101729299358
cg18139900	0.140629397218552
cg20426860	0.139847070955544
cg11481490	0.138190071248909
cg08907850	0.13384912183506
cg04284814	0.13384912183506
cg14709524	0.133496290602001
cg09809242	0.13306237608947
cg12903171	0.131930967720138
cg09890200	0.130837517821722

Cuadro 17: Resultados ganancia de informacion en methylation data + Race variable (10 CpG con mayor IG).

Variable	IG
Alloiococcus	0.133315937716431
Kushneria	0.0935081507882575
Halobacillus	0.0876302210494473
Pontibacter	0.0856556092598862
Sorangium	0.0808490582650077
Ornithinimicrobium	0.0785149080413622
Omegapapillomavirus	0.0766947394516089
Gloeobacter	0.0753604847585367
Rubellimicrobium	0.0747160782296764
Jeotgalicoccus	0.0718154650847453

Cuadro 18: Resultados ganancia de informacion en microbiome data (10 microbiome variables con mayor IG).

Variable	IG
Alloiococcus	0.133315937716431
Kushneria	0.0935081507882575
Halobacillus	0.0876302210494473
Pontibacter	0.0856556092598862
Sorangium	0.0808490582650077
Ornithinimicrobium	0.0785149080413622
Omegapapillomavirus	0.0766947394516089
Gloeobacter	0.0753604847585367
Rubellimicrobium	0.0747160782296764
Jeotgalicoccus	0.0718154650847453

Cuadro 19: Resultados ganancia de informacion en microbiome data + Race variable (10 microbiome variables con mayor IG).

```
#' Apply the information gain filter
#'
#' @param dataframe Data frame to which the filter is applied
#' @param int_n_features Integer defining the number of most representative features to be
    selected in the returned data frame
#'
#' @return A list of objects. Each object has the following characteristics:
#'
#' * Object 1: df_ig, storing a data frame with the set of most representative features
   defined by 'int_n_features' as row names and sample_ids as column names.
#' * Object 2: result_ig, storing the information gain value associated with each variable
    in the original data frame 'dataframe'.
# @export
#'
#' @examples
informationGain_fs <- function(dataframe, int_n_features) {
  tryCatch(
    expr = {
      # Check arguments
      if (!is.data.frame(dataframe)) {
        stop("Argument 'dataframe' must be a data frame")
      } else if (!is.numeric(int_n_features)) {
```

```
        stop("Argument 'int_n_features' must be an integer")
      } else {
        ig_coreLearn <-
          attrEval(OS_STATUS == 1 ~ ., data = dataframe, estimator = "InfGain")

        df_result <-
          dataframe[, names(ig_coreLearn[order(ig_coreLearn, decreasing = TRUE)[1:int_n_
    features]])]
        df_result$OS_STATUS <- dataframe$OS_STATUS

        ig_return <-
          data.frame(Columns = names(ig_coreLearn[order(ig_coreLearn, decreasing = TRUE)])
    , IG = ig_coreLearn[order(ig_coreLearn, decreasing = TRUE)])

        elements <- list(df_ig = df_result, result_ig = ig_return)

        return(elements)
      }
    },
    error = function(e) {
      print(sprintf("An error occurred in informationGain_fs at %s : %s",
                    Sys.time(),
                    e))
    }
  )
}
```

Listing 5: Función informationGain fs

6.7. Integración de Datos Ómicos

La integración de las 30 variables mas representativas asociadas a cada conjunto de datos ómico se realizó mediante la función early integration.

```
#' Apply early integration to a set of data frames
#'
#' @param list_dfs A list of data frames
#'
#' @return A data frame formed by concatenating variables from the datasets provided in '
    list_dfs'.
```

```r
#'
#' @export
#'
#' @examples
early_integration <- function(list_dfs) {
  tryCatch(
    expr = {
      # Check if list_dfs is a list of data frames
      if (!is.list(list_dfs) || !all(sapply(list_dfs, is.data.frame))) {
        stop("Argument 'list_dfs' must be a list of data frames")
      } else {
        dataframe <- list_dfs[[1]]

        for (i in 2:length(list_dfs)) {
          dataframe <- cbind(dataframe, list_dfs[[i]])
        }

        dataframe <- dataframe[!duplicated(as.list(dataframe))]

        return(dataframe)
      }
    },
    error = function(e) {
      print(sprintf("An error occurred in 'early_integration' at %s: %s",
                    Sys.time(),
                    e))
    }
  )
}
```

Listing 6: Función early integration

6.8. Entrenamiento y Evaluación de Modelos

El entrenamiento y evaluación de los modelos de machine learning **LGBM**, **SVM Radial**, **SVM Polinómico** y **SVM Lineal**, se realizó mediante la función train model.

```r
#' Training machine learning models applying hyperparameter grid search optimization
#'
#' @param char_model Character defining the model name
```

```
#' @param param_exp_grid expand.grid() element defining the hyperparameter space search
#' @param df_data Data frame with the data
#' @param list_cv_index List of train data indexes
#'
#' @return A list of objects. The characteristics of each object are as follows:
#'
#' * Object 1: 'df_results', stores a data frame with model evaluation and bias prediction
     detection metrics results for each fold.
#' * Object 2: 'list_best_params', stores a list with the hyperparameters with the best
    results in each fold.
#' @export
#'
#' @examples
train_model <- function(char_model, param_exp_grid, df_data, list_cv_index) {
  tryCatch(
    expr = {
      if (!is.character(char_model)) {
        stop("Argument 'char_model' must be a character")
      } else if (!is.data.frame(df_data)) {
        stop("Argument 'df_data' must be a data frame")
      } else if (!is.list(list_cv_index) || !all(sapply(list_cv_index, is.numeric))) {
        stop("Argument 'list_cv_index' must be a list of numeric vectors")
      }

      df_results <- data.frame(Model = character(), Fold = numeric(), ACC = numeric(), AUC
     = numeric(), F1_score = numeric(), PRO_S = numeric(), PRO_NS = numeric(), PROPP_S =
    numeric(), PROPP_NS = numeric(), SEN_S = numeric(), SEN_NS = numeric(), EO_S = numeric
    (), EO_NS = numeric(), PRE_S = numeric(), PRE_NS = numeric(), PRP_S = numeric(), PRP_
    NS = numeric(), ACC_S = numeric(), ACC_NS = numeric(), ACCP_S = numeric(), ACCP_NS =
    numeric(), FNR_S = numeric(), FNR_NS = numeric(), FNRP_S = numeric(), FNRP_NS =
    numeric(), FPR_S = numeric(), FPR_NS = numeric(), FPRP_S = numeric(), FPRP_NS =
    numeric(), GZ_S = numeric(), GZ_NS = numeric())

      list_best_params <- list()

      for (i in 1:length(list_cv_index)) {
        training_data <- df_data[list_cv_index[[i]], ]
        levels(training_data$OS_STATUS) <- make.names(levels(factor(training_data$OS_
    STATUS)))
        test_data <- df_data[-list_cv_index[[i]], ]
        levels(test_data$OS_STATUS) <- make.names(levels(factor(test_data$OS_STATUS)))

        set.seed(3000)
        training_train_control <- trainControl(method = "cv", number = 2, search = "grid",
     classProbs = TRUE, summaryFunction = twoClassSummary)
```

```
    if (char_model == "svmLinear2") {
      training_model <- train(OS_STATUS ~ ., data = training_data, method = "
svmLinear2", trControl = training_train_control, tuneGrid = param_exp_grid, metric = "
ROC")

      final_model <- train(OS_STATUS ~ ., data = training_data, method = "svmLinear2",
 trControl = trainControl(method = "none", classProbs = TRUE), tuneGrid = training_
model$bestTune, metric = "ROC")
    }

    if (char_model == "svmRadial") {
      training_model <- train(OS_STATUS ~ ., data = training_data, method = "
svmRadialSigma", trControl = training_train_control, tuneGrid = param_exp_grid, metric
 = "ROC")

      final_model <- train(OS_STATUS ~ ., data = training_data, method = "
svmRadialSigma", trControl = trainControl(method = "none", classProbs = TRUE),
tuneGrid = training_model$bestTune, metric = "ROC")
    }

    if (char_model == "svmPoly") {
      training_model <- train(OS_STATUS ~ ., data = training_data, method = "svmPoly",
 trControl = training_train_control, tuneGrid = param_exp_grid, metric = "ROC")

      final_model <- train(OS_STATUS ~ ., data = training_data, method = "svmPoly",
trControl = trainControl(method = "none", classProbs = TRUE), tuneGrid = training_
model$bestTune, metric = "ROC")
    }

    test_predictions <- predict(final_model, test_data %>% select(-c("OS_STATUS")))

    cf_matrix_test <- confusionMatrix(test_predictions, test_data$OS_STATUS)
    accuracy_test <- as.vector(cf_matrix_test$overall[1])
    F1_test <- as.vector(cf_matrix_test$byClass[7])

    ## Binarize the outcome
    test_predictions <- as.character(test_predictions)
    test_predictions[test_predictions == "X0"] <- 0
    test_predictions[test_predictions == "X1"] <- 1
    test_data$OS_STATUS <- as.character(test_data$OS_STATUS)
    test_data$OS_STATUS[test_data$OS_STATUS == "X0"] <- 0
    test_data$OS_STATUS[test_data$OS_STATUS == "X1"] <- 1
```

```
    df_prediction <- data.frame(predictions = as.numeric(test_predictions), OS_STATUS
= as.numeric(test_data$OS_STATUS), RACE = clinical_data_fixed$RACE[c(which(row.names(
clinical_data_fixed) %in% row.names(test_data)))])
    # data(df_prediction)
    pred_AUC_test <- prediction(as.numeric(df_prediction$predictions), as.numeric(df_
prediction$OS_STATUS))
    perf_AUC_test <- performance(pred_AUC_test, "auc")
    AUC_value_test <- perf_AUC_test@y.values[[1]]

    eq_odds <- equal_odds(
      data = df_prediction,
      outcome = "OS_STATUS",
      group = "RACE",
      preds = "predictions",
      base = "Black or African American"
    )

    eq_odds_sens_val <- eq_odds$Metric[1, 1]
    eq_odds_sens_par_val <- eq_odds$Metric[2, 1]
    eq_odds_nsens_val <- eq_odds$Metric[1, 2]
    eq_odds_nsens_par_val <- eq_odds$Metric[2, 2]

    pred_rt_parity <- pred_rate_parity(
      data = df_prediction,
      outcome = "OS_STATUS",
      group = "RACE",
      preds = "predictions",
      base = "Black or African American"
    )

    pred_rt_sens_val <- pred_rt_parity$Metric[1, 1]
    pred_rt_sens_par_val <- pred_rt_parity$Metric[2, 1]
    pred_rt_nsens_val <- pred_rt_parity$Metric[1, 2]
    pred_rt_nsens_par_val <- pred_rt_parity$Metric[2, 2]

    acc_par <- acc_parity(
      data = df_prediction,
      outcome = "OS_STATUS",
      group = "RACE",
      preds = "predictions",
      base = "Black or African American"
    )

    acc_sens_val <- acc_par$Metric[1, 1]
    acc_sens_par_val <- acc_par$Metric[2, 1]
```

```r
acc_nsens_val <- acc_par$Metric[1, 2]
acc_nsens_par_val <- acc_par$Metric[2, 2]

fnr_par <- fnr_parity(
  data = df_prediction,
  outcome = "OS_STATUS",
  group = "RACE",
  preds = "predictions",
  base = "Black or African American"
)

fnr_sens_val <- fnr_par$Metric[1, 1]
fnr_sens_par_val <- fnr_par$Metric[2, 1]
fnr_nsens_val <- fnr_par$Metric[1, 2]
fnr_nsens_par_val <- fnr_par$Metric[2, 2]

fpr_par <- fpr_parity(
  data = df_prediction,
  outcome = "OS_STATUS",
  group = "RACE",
  preds = "predictions",
  base = "Black or African American"
)

fpr_sens_val <- fpr_par$Metric[1, 1]
fpr_sens_par_val <- fpr_par$Metric[2, 1]
fpr_nsens_val <- fpr_par$Metric[1, 2]
fpr_nsens_par_val <- fpr_par$Metric[2, 2]

res_prop <- prop_parity(
  data = df_prediction,
  outcome = "OS_STATUS",
  group = "RACE",
  preds = "predictions",
  base = "Black or African American"
)

res_prop_sens_val <- res_prop$Metric[1, 1]
res_prop_sens_par_val <- res_prop$Metric[2, 1]
res_prop_nsens_val <- res_prop$Metric[1, 2]
res_prop_nsens_par_val <- res_prop$Metric[2, 2]

gz_s <- fpr_par$Metric[3, 1]
gz_ns <- fpr_par$Metric[3, 2]
```

```
    df_results_add <- data.frame(
      Model = as.character(char_model),
      Fold = as.numeric(i),
      ACC = as.numeric(accuracy_test),
      AUC = as.numeric(AUC_value_test),
      F1_score = as.numeric(F1_test),
      PRO_S = as.numeric(res_prop_sens_val),
      PRO_NS = as.numeric(res_prop_nsens_val),
      PROPP_S = as.numeric(res_prop_sens_par_val),
      PROPP_NS = as.numeric(res_prop_nsens_par_val),
      SEN_S = as.numeric(eq_odds_sens_par_val),
      SEN_NS = as.numeric(eq_odds_nsens_par_val),
      EO_S = as.numeric(eq_odds_sens_par_val),
      EO_NS = as.numeric(eq_odds_nsens_par_val),
      PRE_S = as.numeric(pred_rt_sens_val),
      PRE_NS = as.numeric(pred_rt_nsens_val),
      PRP_S = as.numeric(pred_rt_sens_par_val),
      PRP_NS = as.numeric(pred_rt_nsens_par_val),
      ACC_S = as.numeric(acc_sens_val),
      ACC_NS = as.numeric(acc_nsens_val),
      ACCP_S = as.numeric(acc_sens_par_val),
      ACCP_NS = as.numeric(acc_nsens_par_val),
      FNR_S = as.numeric(fnr_sens_val),
      FNR_NS = as.numeric(fnr_nsens_val),
      FNRP_S = as.numeric(fnr_sens_par_val),
      FNRP_NS = as.numeric(fnr_nsens_par_val),
      FPR_S = as.numeric(fpr_sens_val),
      FPR_NS = as.numeric(fpr_nsens_val),
      FPRP_S = as.numeric(fpr_sens_par_val),
      FPRP_NS = as.numeric(fpr_nsens_par_val),
      GZ_S = as.numeric(gz_s),
      GZ_NS = as.numeric(gz_ns)
    )

    df_results <- rbind(df_results, df_results_add)

    list_best_params[i][[1]] <- training_model$bestTune
  }

  df_results_add_final <- data.frame(
    Model = as.character("Global"),
    Fold = NA,
    ACC = as.numeric(mean(df_results$ACC)),
    AUC = as.numeric(mean(df_results$AUC)),
```

```
      F1_score = as.numeric(mean(df_results$F1_score)),
      PRO_S = as.numeric(mean(df_results$PRO_S)),
      PRO_NS = as.numeric(mean(df_results$PRO_NS)),
      PROPP_S = as.numeric(mean(df_results$PROPP_S)),
      PROPP_NS = as.numeric(mean(df_results$PROPP_NS)),
      SEN_S = as.numeric(mean(df_results$SEN_S)),
      SEN_NS = as.numeric(mean(df_results$SEN_NS)),
      EO_S = as.numeric(mean(df_results$EO_S)),
      EO_NS = as.numeric(mean(df_results$EO_NS)),
      PRE_S = as.numeric(mean(df_results$PRE_S)),
      PRE_NS = as.numeric(mean(df_results$PRE_NS)),
      PRP_S = as.numeric(mean(df_results$PRP_S)),
      PRP_NS = as.numeric(mean(df_results$PRP_NS)),
      ACC_S = as.numeric(mean(df_results$ACC_S)),
      ACC_NS = as.numeric(mean(df_results$ACC_NS)),
      ACCP_S = as.numeric(mean(df_results$ACCP_S)),
      ACCP_NS = as.numeric(mean(df_results$ACCP_NS)),
      FNR_S = as.numeric(mean(df_results$FNR_S)),
      FNR_NS = as.numeric(mean(df_results$FNR_NS)),
      FNRP_S = as.numeric(mean(df_results$FNRP_S)),
      FNRP_NS = as.numeric(mean(df_results$FNRP_NS)),
      FPR_S = as.numeric(mean(df_results$FPR_S)),
      FPR_NS = as.numeric(mean(df_results$FPR_NS)),
      FPRP_S = as.numeric(mean(df_results$FPRP_S)),
      FPRP_NS = as.numeric(mean(df_results$FPRP_NS)),
      GZ_S = NA,
      GZ_NS = NA
    )

    df_results <- rbind(df_results, df_results_add_final)

    results <- list(df_results = df_results, list_best_params = list_best_params)

    return(results)
  },
  error = function(e) {
    print(sprintf("An error occurred in train_model at %s : %s",
                  Sys.time(),
                  e))
  }
)
```

Listing 7: Función train model

sigma	0.001, 0.0015, 0.01, 0.015, 0.1, 0.5, 1, 5, 10, 50, 100
C	0.001, 0.0015, 0.01, 0.015, 0.1, 0.5, 1, 5, 10, 50, 100

Cuadro 20: Grid de hyperparámetros evaluados en SVM Radial

C	0.001, 0.0015, 0.01, 0.015, 0.1, 0.5, 1, 5, 10, 50, 100
degree	1,2,3
scale	1

Cuadro 21: Grid de hyperparámetros evaluados en SVM Polinómico

C	0.01, 0.015, 0.1, 0.5, 1, 5, 10, 50, 100

Cuadro 22: Grid de hyperparámetros evaluados en SVM Lineal

learning_rate	0.005, 0.01
n_estimators	8,16,24,32
num_leaves	6,8,12,16,20
boosting_type	'gbdt', 'dart'
objective	'binary'
max_bin	255, 510
random_state	300
colsample_bytree	0.64, 0.65, 0.66
subsample	0.7,0.75
reg_alpha	1,1.2
reg_lambda	1,1.2,1.4]

Cuadro 23: Grid de hyperparámetros evaluados en LGBM

7
Resultados

7.1. Resultados de la Selección de Variables

La selección de variables se realizo a los conjuntos de datos ómicos de RNAseq (DO1), metilación (DO2) y microbioma (DO3) de forma independiente y considerando como variable de estudio o clase, el estado de supervivencia 'OS STATUS' en el instante de tiempo fijado a 5 años.

Los métodos de selección de variables aplicados secuencialmente fueron: analisis de expresión diferencial o abundancia y ganancia de información. Se consideraron variables diferencialmente expresadas en los conjuntos de datos D01, DO2 y DO2 aquellas con un *p-value* <0.05. En la primera selección de variables mediante analisis de expresión diferencial respecto de la variable de estudio y cumpliendo con la condición de *p-value* fijada, se obtuvieron 17496 variables en DO1, 10632 en DO2 y 1406 en DO3. Posteriormente, se aplicó el método de selección de variables mediante ganancia de información a las variables obtenidas mediante análisis de expresión diferencial. Mediante la segunda selección de variables se seleccionaron de cada uno de los conjuntos de datos ómicos utilizados, las 30 variables más significativas respecto de la de estudio con supervivencia positiva. El conjunto de variables formado por las 30 variables de cada conjunto de datos ómicos fue integrado mediante integración temprana, formando un conjunto de datos de 90 variables.

El conjunto de datos (DF1) resultado de la integración de temprana y formado por 90 variables, fue destinado para la fase de clasificación mediante modelos de machine learning. Adicionalmente se creó un segundo conjunto de datos (DF2), el cual se creó a partir de DF1, añadiendo la variable Raza, con el objetivo de estudiar tanto su efecto como predictor, como en el posterior análisis de bias en la predicción de los modelos.

7.2. Resultados de Clasificación de los Modelos de Machine Learning

Los modelos LGBM, SVM Lineal, SVM Polinómico y SVM Radial fueron utilizados para la clasificación de supervivencia en el instante de tiempo fijado. La evaluación del rendimiento de los modelos de machine learning aplicada fue realizada mediante 5x2 Cross Validation. La métrica utilizada para la selección de hiperparámetros fue AUC. Las métricas utilizadas en la evaluación del rendimiento del modelo fueron AUC, Accuracy y F1-score. Los resultados de clasificación de los modelos aplicados en los conjuntos de datos DF1 y DF1, se muestran en los Cuadro 24 y Cuadro 25 respectivamente.

Se realizan las siguientes observaciones:

- DF2 proporciona una mejora en el rendimiento de los modelos en todas las métricas.
- SVM Lineal obtiene los mejores resultados en todas las métricas en los dos conjuntos de datos aplicados. No obstante las diferencias con SVM Radial y SVM Linear son mínimas.
- LGBM obteniendo a 'priori' los mejores resultados F1-score, se descartan los resultados debido a la incapacidad de diferenciación de sus predicciones entre clases. Todas las predicciones obtenidas pertenecen a una clase.
- En el conjunto de datos DF2, SVM Lineal obtiene los peores resultados en todas las métricas. Aunque de nuevo las diferencias son mínimas respecto de SVM Radial y SVM Polinómico.

Métricas	SVM L	SVM R	SVM P	LGBM
ACC	83.2 %	84.0 %	84.1 %	63.7 %
AUC	80.2 %	79.7 %	80.3 %	64.0 %
F1 Score	74.3 %	72.3 %	74.9 %	77.8 %

Cuadro 24: Resultados globales (5x2 cross validation) de clasificación sobre el conjunto datos DF1

Métricas	SVM L	SVM R	SVM P	LGBM
ACC	83.2 %	85.7 %	85.8 %	63.7 %
AUC	80.8 %	82.7 %	82.8 %	64.7 %
F1 Score	74.8 %	76.7 %	78.1 %	77.8 %

Cuadro 25: Resultados globales (5x2 cross validation) de clasificación sobre el conjunto datos DF2

7.3. Resultados en la Detección de Bias

El sesgo o bias en las predicciones de los modelos aplicados considerando como variable sensible RACE y como categoria de referencia 'Black or African American' se evaluo utilizando como métricas Proportional Parity (PRO P), Equalized Odds (EO), Equal Opportunity (considerando disparidad en TPR y FPR) y Accuracy disparity (ACC P). En el Cuadro 26 y Cuadro 27 se muestra el análisis de bias en las predicción de los modelos aplicados en los conjuntos de datos DF1 y DF2 respectivamente. Debido a la proporción de las predicciones asociadas a la variable sensible Race "Black or African American.[a]lgunas métricas no fueron posible ser calculadas.

Resultados asociados al conjunto de datos DF1:

- SVM Radial proporciona los mejores resultados en las métricas Proportional parity con 89,4 % y Accuracy sobre el grupo sensible, alcanzando un 66 %. Siendo estas de las métricas calculadas, las consideradas más relevantes para el contexto del problema.

- El modelo SVM Radial proporcina los mejores resultados en 4 de las métricas calculadas, SVM Polinómico en 3 y SVM Lineal en 1.
- LGBM obtuvo los peores resultadas en el total de las metricas calculadas.
- Las diferencias en la mayoria métricas calculadas es mínima entre los tres modelos SVM aplicados.

Resultados asociados al conjunto de datos DF2:

- SVM Radial proporciona los mejores resultados en las métricas Proportional parity con un 89,3 % y comparte mejor Accuracy sobre el grupo sensible, con SVM Lineal alcanzando un 64.3 %.
- El modelo SVM Radial comparte mejor resultado en tres de las meétricas calculadas junto con SVM Polinómico.
- LGBM obtuvo los peores resultadas en el total de las metricas calculadas.
- De igual forma que en los resultados asociados a DF1, las diferencias en las métricas calculas es mínima entre los tres modelos SVM aplicados.

Diferencias en los resultados entre los conjuntos de datos DF1 y DF2.

- Se obtiene mejores resultados en modelos aplicados al conjunto de datos DF1 en 3 de las métricas calculadas, no existe diferencia en 3 métricas y se obtienen mejores resultados en 3 métricas en el conjunto de datos DF2.
- Del total de resultados asociados a los dos conjuntos de datos, SVM Radial proporciona los mejores en 8 de las métricas calculadas y SVM Linear obtiene los mejores resultados en 3 de ellas.

Métricas	**SVM L**	**SVM Rl**	**SVM P**	**LGBM**
$PROS$	84.3 %	82.3 %	64.3 %	100 %
$PRONS$	67.9 %	72.7 %	72.1 %	100 %
$PROP$	83.1 %	89.4 %	NA	100 %
$PRES$	60.8 %	63.3 %	NA	61.6 %
$PRENS$	86.6 %	85.4 %	84.8 %	63.4 %
$ACCS$	64.3 %	66.3 %	44.3 %	61.6 %
$ACCNS$	85.7 %	86.3 %	87.1 %	63.4 %
$FNRNS$	7.2 %	2.9 %	2.9 %	0 %
$FPRNS$	26.3 %	32.9 %	29.2 %	100 %

Cuadro 26: Resultados de las métricas de detección de bias en la predicción de modelos sobre el conjunto datos DF1

Métricas	**SVM L**	**SVM Rl**	**SVM P**	**LGBM**
$PROS$	80.3 %	80.3 %	60.3 %	100 %
$PRONS$	67.9 %	70.1 %	72.1 %	100 %
$PROP$	88.2 %	89.3 %	NA	100 %
$PRES$	63.3 %	63.3 %	NA	61.6 %
$PRENS$	86.6 %	85.4 %	88.4 %	63.4 %
$ACCS$	64.3 %	64.3 %	44.3 %	61.6 %
$ACCNS$	85.7 %	89.1 %	88.8 %	63.4 %
$FNRNS$	7.2 %	2.9 %	1.5 %	0 %
$FPRNS$	26.3 %	25.8 %	26.7 %	100 %

Cuadro 27: Resultados de las métricas de detección de bias en la predicción de modelos sobre el conjunto datos DF2

8

Conclusiones y Líneas Futuras

8.1. Conclusiones

Siendo los objetivos principales del TFG el aplicar distintos modelos de machine learning en la predicción de la prognosis de cancer colorectal, a partir datos ómico reales y analizar el sesgo en las predicciones de los modelos en base a un atributo o característica sensible se realizan las siguientes conclusiones:

Tras una primer filtrado de los datos en base ha aquellas características que cumplieran la condición fijada en la prediccion de la prognosis a 5 años, se observó que la reducción de observaciones provocaba un incremento en el desbalanceo de categorias dentro de la variable sensible Race. Se intentó reducir el periodo de tiempo de estudio con el objetivo de conseguir un mayor número de observaciones, no obstante a menor era el tiempo de observación las variables de los conjuntos de datos ómicos perdían significancia mediante analisis diferencial. Mostrando que en base a la pregunta o condición de estudio la robustez del analisis del sesgo se veía comprometida. Finalmente se optó por fijar el instante de tiempo a 5 años, periodo de tiempo el cual se consideró que obtenía el mejor balance entre observaciones y significancia de las variables ómicas.

La seleccion de variables mediante analisis de expresión fue un factor que determinó tanto la propia selección de variables como el periodo de tiempo de estudio. La segunda técnica de selección de variables por ganancia de información es un método relativamente facil de implementar e independiente de los modelos de ML. No obstante, siendo la significancia de las variables calculada de forma independiente respecto de la variable de estudio, carece de la información proporcionada por relaciones entre estas.

La técnica de integración temprana permitió de una forma sencilla la integración de los tres tipos de datos ómicos. No obstante, no proporciona información sobre la relación entre las características, aspecto el cual es de interesante estudio.

En relación a los resultados de LGBM, estos mostrarón que dicho modelo no consiguió diferenciar entre clases, marcado esto por un ratio de Falsos Negativos de 1 en los conjuntos de datos DF1 y DF2. Entre las posibles causas se encuentran una reducida búsqueda de hiperparámetros o las características propias de los datos.

Los 3 tipos SVMs aplicado han mostrado ser consistentes en los 2 conjuntos de datos aplicados. Un mayor espacio de busqueda de hiperpárametros como un diferente procesamiento de los datos se postulan como posibles opciones de mejora.

Otro aspecto que involucró un efecto directo en la robustez de los resultados en la detección del sesgo fue el método de validación 5x2 cross validation. Debido al reducido número de observaciones propio de conjunto de datos, la reducción añadida por la condición de tiempo fijada y el desbalance entre las observaciones dentro de la variable sensible, muchas de las particiones evaluadas en 5x2 cross validation no tenían variabilidad suficiente dentro de la variable sensible para proporcinar significancia en algunas de las métricas calculadas.

Dados los objetivo iniciales y con la experiencia tras su desarrollo, se concluye que los distintos modelos machine aplicados ha permitido obtener unos resultados relativamente aceptable incluso en un conjunto de datos con un reducido número de características, no obstante la incorporación de analisis del sesgo en sus predicciones ha mostrado las limitaciones en los conjuntos de datos para un robusto analisis de este.

8.2. Líneas Futuras

Se proponen las siguientes lineas futuras en base de los resultados y obstáculos encontrados durante el desarrollo del TFG: 1) Análisis de los perfíles ómicos asociados a las observaciones dentro de la variable Race con el objetivo de encontrar aquellas variables mas significativas en la predicción de prognosis. 2) En relación al punto anterior, la aplicación de técnicas de integración de datos que permitan obtener información sobre la relación entre características en distintos tipos de datos, abordando el estudio desde un punto de vista multidimensional. 3) La alicación de distintos métodos de selección de variables, con el objetivo tanto de, analizar su efecto en los modelos de ML, como del análisis biológico de su resultado. 4) Abordar el problema del desbalanceo de clases en la variable sensible mediante técnicas de mitigación de bias a nivel de datos.

Referencias

[1] Hanif Abdul Rahman, Mohammad Ashraf Ottom e Ivo D Dinov. «Machine learning-based colorectal cancer prediction using global dietary data». En: BMC cancer 23.1 (2023), pág. 144.

[2] Santoshi Acharjee et al. «Mechanisms of DNA methylation and histone modifications». En: vol. 197. Elsevier B.V., ene. de 2023, págs. 51-92. ISBN: 9780443186691. DOI: 10.1016/bs.pmbts.2023.01.001.

[3] Sajid Ahmed et al. «Hybrid Methods for Class Imbalance Learning Employing Bagging with Sampling Techniques». En: (2017), págs. 1-5. DOI: 10.1109/CSITSS.2017.8447799.

[4] Micheal Olaolu Arowolo et al. «A survey of dimension reduction and classification methods for RNA-Seq data on malaria vector». En: Journal of Big Data 8 (1 dic. de 2021). ISSN: 21961115. DOI: 10.1186/s40537-021-00441-x.

[5] Andrew L. Beam, Arjun K. Manrai y Marzyeh Ghassemi. «Challenges to the Reproducibility of Machine Learning Models in Health Care». En: JAMA - Journal of the American Medical Asso 323 (4 ene. de 2020), págs. 305-306. ISSN: 15383598. DOI: 10.1001/jama.2019.20866.

[6] Houda Benhar, Ali Idri y Mohamed Hosni. «Impact of threshold values for filter-based univariate feature selection in heart disease classification». En: HEALTHINF 2020 - 13th International (2020), págs. 391-398. DOI: 10.5220/0008947403910398.

[7] Erkan Bostanci et al. «Machine Learning Analysis of RNA-seq Data for Diagnostic and Prognostic Prediction of Colon Cancer». En: Sensors 23 (6 mar. de 2023). ISSN: 14248220. DOI: 10.3390/s23063080.

[8] Andrei Z. Broder, Alfred M. Bruckstein y Jack Koplowitz. «On the performance of edited nearest neighbor rules in high dimensions». En: IEEE Transactions on Systems, Man, and Cybernetics SMC-15.1 (1985), págs. 136-139. DOI: 10.1109/TSMC.1985.6313401.

[9] Zhaoxiang Cai et al. «Machine learning for multi-omics data integration in cancer». En: iScience 25 (2 feb. de 2022). ISSN: 25890042. DOI: 10.1016/J.ISCI.2022.103798.

[10] Acevedo TMT Calva AM. «Revisión y actualización general en cáncer colorrectal». En: Anales de Radiología México (2009), 8(1):99-115.

[11] Michael Canesche et al. «Google Colab CAD4U: Hands-on cloud laboratories for digital design». En: 2021 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE. 2021, págs. 1-5.

[12] Hua Chai et al. «Integrating multi-omics data through deep learning for accurate cancer prognosis prediction». En: Computers in biology and medicine 134 (2021), pág. 104481.

[13] Nitesh V Chawla et al. «SMOTE: synthetic minority over-sampling technique». En: Journal of artificial intelligence research 16 (2002), págs. 321-357.

[14] Jae-Ho Cheong et al. «Development and validation of a prognostic and predictive 32-gene signature for gastric cancer». En: Nature communications 13.1 (2022), pág. 774.

[15] Fortunato Ciardiello et al. «MC-Clinical management of metastatic colorectal cancer in the era of precision medicine». En: CA: A Cancer Journal for Clinicians 72 (4 jul. de 2022), págs. 372-401. ISSN: 0007-9235. DOI: 10.3322/caac.21728.

[16] Irene Dankwa-Mullan y Dilhan Weeraratne. «1B- Artificial Intelligence and Machine Learning Technologies in Cancer Care: Addressing Disparities, Bias, and Data Diversity». En: Cancer Discovery 12 (6 jun. de 2022), págs. 1423-1427. ISSN: 21598290. DOI: 10.1158/2159-8290.CD-22-0373.

[17] GL Davis et al. «Spectral/spatial analysis of colon carcinoma». En: LABORATORY INVESTIGATION. Vol. 83. 1. LIPPINCOTT WILLIAMS & WILKINS 530 WALNUT ST, PHILADELPHIA, PA 19106-3621 USA. 2003, 320A-321A.

[18] Thomas G Dietterich. Approximate Statistical Tests for Comparing Supervised Classiication Learning 1997.

[19] Xibin Dong et al. «A survey on ensemble learning». En: Frontiers of Computer Science 14 (2020), págs. 241-258.

[20] Amelie Echle et al. «Clinical-grade detection of microsatellite instability in colorectal tumors by deep learning». En: Gastroenterology 159.4 (2020), págs. 1406-1416.

[21] Ciaran Evans, Johanna Hardin y Daniel Stoebel. «Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions». En: (sep. de 2016). URL: http://arxiv.org/abs/1609.00959.

[22] Konstantinos P Exarchos, Yorgos Goletsis y Dimitrios I Fotiadis. «Multiparametric decision support system for the prediction of oral cancer reoccurrence». En: IEEE Transactions on Informat 16.6 (2011), págs. 1127-1134.

[23] Piotr Florek y Adam Zagdański. «Benchmarking state-of-the-art gradient boosting algorithms for classification». En: (mayo de 2023). URL: http://arxiv.org/abs/2305.17094.

[24] Federico M Giorgi, Carmine Ceraolo y Daniele Mercatelli. «The R language: an engine for bioinformatics and data science». En: Life 12.5 (2022), pág. 648.

[25] Van de Graaff KM. «Anatomy and physiology of the gastrointestinal tract». En: Pediatr Infect Dis (1986).

[26] Juan José Granados-Romero et al. «Colorectal cancer: a review». En: International Journal of Research i 5 (11 oct. de 2017), pág. 4667. ISSN: 2320-6071. DOI: 10.18203/2320-6012.ijrms20174914.

[27] Charua Guindic. «Guías clínicas de diagnóstico y tratamiento del carcinoma de colon y recto. Tratamiento del cáncer de colon y recto». En: Revista de Gastroenterología de México (2008), págs. 21-125.

[28] Justin Guinney et al. «The consensus molecular subtypes of colorectal cancer». En: Nature Medicine 21 (11 nov. de 2015). Table Molecular Subtypes, págs. 1350-1356. ISSN: 1546170X. DOI: 10.1038/nm.3967.

[29] Shannon Haymond y Stephen R. Master. «How Can We Ensure Reproducibility and Clinical Translation of Machine Learning Applications in Laboratory Medicine?» En: Clinical Chemistry 68 (3 mar. de 2022), págs. 392-395. ISSN: 15308561. DOI: 10.1093/clinchem/hvab272.

[30] Jiaxin Hou et al. «Integrative Histology-Genomic Analysis Predicts Hepatocellular Carcinoma Prognosis Using Deep Learning». En: Genes 13.10 (2022), pág. 1770.

[31] Jonathan Huang, Galal Galal y Mahesh Vaidyanathan. «Review Evaluation and Mitigation of Racial Bias in Clinical Machine Learning Models: Scoping Review». En: (). DOI: 10.2196/36388. URL: https://medinform.jmir.org/2022/5/e36388.

[32] Yufei Jiang et al. «Global pattern and trends of colorectal cancer survival: a systematic review of population-based registration data». En: Cancer Biology and Medicine 19 (2 feb. de 2022), págs. 175-186. ISSN: 20953941. DOI: 10.20892/j.issn.2095-3941.2020.0634.

[33] Dongzi Jin et al. «SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism». En: Computers Security 97 (2020), pág. 101984. DOI: 10.1016/j.cose.2020.101984. URL: https://doi.org/10.1016/j.cose.2020.101984.

[34] Wattana Jindaluang, Varin Chouvatut y Sanpawat Kantabutra. «Under-sampling by algorithm with performance guaranteed for class-imbalance problem». En: 2014 International Computer (2014), págs. 215-221.

[35] Mingon Kang, Euiseong Ko y Tesfaye B. Mersha. «A roadmap for multi-omics data integration using deep learning». En: Briefings in Bioinformatics 23 (1 ene. de 2022). ISSN: 14774054. DOI: 10.1093/bib/bbab454.

[36] Guolin Ke et al. «LightGBM: A Highly Efficient Gradient Boosting Decision Tree». En: (). URL: https://github.com/Microsoft/LightGBM..

[37] Matloob Khushi et al. «A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data». En: IEEE Access 9 (2021), págs. 109960-109975. DOI: 10.1109/ACCESS.2021.3102399.

[38] Saraswati Koppad et al. «MC-C-Machine Learning-Based Identification of Colon Cancer Candidate Diagnostics Genes». En: Biology 11 (3 mar. de 2022). ISSN: 20797737. DOI: 10.3390/biology11030365.

[39] Konstantina Kourou et al. «Machine learning applications in cancer prognosis and prediction». En: Computational and Structural Biotechnology Journal 13 (2015), págs. 8-17. ISSN: 20010370. DOI: 10.1016/j.csbj.2014.11.005.

[40] Bo Li y Colin N. Dewey. «RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome». En: BMC Bioinformatics 12 (ago. de 2011). ISSN: 14712105. DOI: 10.1186/1471-2105-12-323.

[41] Hui Li et al. «Colorectal cancer detected by machine learning models using conventional laboratory test data». En: Technology in Cancer Research & Treatment 20 (2021), pág. 15330338211058352.

[42] Zongyu Liang et al. «Log odds of positive lymph nodes show better predictive performance on the prognosis of early-onset colorectal cancer». En: International Journal of Colorectal Disea 38.1 (2023), pág. 192.

[43] Jose Liñares-Blanco, Alejandro Pazos y Carlos Fernandez-Lozano. «Machine learning analysis of TCGA cancer data Distributed under Creative Commons CC-BY 4.0». En: (). DOI: 10.7717/peerj-cs.584.

[44] Fei Tony Liu, Kai Ming Ting y Zhi-Hua Zhou. «Isolation-Based Anomaly Detection». En: ACM Trans. Knowl. Discov. Data (2012).

[45] Barbara Lobato-Delgado, Blanca Priego-Torres y Daniel Sanchez-Morillo. «Combining Molecular, Imaging, and Clinical Data Analysis for Predicting Cancer Prognosis». En: Cancers 14 (13 jul. de 2022). ISSN: 20726694. DOI: 10.3390/cancers14133215.

[46] Masashi Misawa et al. «Artificial intelligence-assisted polyp detection for colonoscopy: initial experience». En: Gastroenterology 154.8 (2018), págs. 2027-2029.

[47] Eileen Morgan et al. «Global burden of colorectal cancer in 2020 and 2040: Incidence and mortality estimates from GLOBOCAN». En: Gut 72 (2 sep. de 2022), págs. 338-344. ISSN: 14683288. DOI: 10.1136/gutjnl-2022-327736.

[48] Ravil Muhamedyev. «Machine learning methods: An overview». En: Computer modelling & new techn 19.6 (2015), págs. 14-29.

[49] Alexey Natekin y Alois Knoll. «Gradient boosting machines, a tutorial». En: Frontiers in Neurorobotics 7 (DEC 2013). ISSN: 16625218. DOI: 10.3389/fnbot.2013.00021.

[50] Hien M. Nguyen, Eric W. Cooper y Katsuari Kamei. «A comparative study on sampling techniques for handling class imbalance in streaming data». En: (2012), págs. 1762-1767. DOI: 10.1109/SCIS-ISIS.2012.6505291.

[51] & Williams N Nigam Y Knight J. «Gastrointestinal tract 5: The anatomy and functions of the large intestine». En: Nursing Times (2019), págs. 50-53.

[52] Gregory D. Poore et al. «Microbiome analyses of blood and tissues suggest cancer diagnostic approach». En: Nature 579 (7800 mar. de 2020), págs. 567-574. ISSN: 14764687. DOI: 10.1038/s41586-020-2095-1.

[53] Alka Rani et al. «Machine learning for soil moisture assessment». En: Elsevier, ene. de 2022, págs. 143-168. ISBN: 9780323852142. DOI: 10.1016/B978-0-323-85214-2.00001-X.

[54] Saima Rathore et al. «MC-A recent survey on colon cancer detection techniques». En: IEEE/ACM Transactions on Computational Biology and Bioinformatics 10 (3 2013), págs. 545-563. ISSN: 15455963. DOI: 10.1109/TCBB.2013.84.

[55] Saima Rathore et al. «Texture analysis for liver segmentation and classification: a survey». En: 2011 Frontiers of Information Technology. IEEE. 2011, págs. 121-126.

[56] Prashanth Rawla, Tagore Sunkara y Adam Barsouk. «Epidemiology of colorectal cancer: Incidence, mortality, survival, and risk factors». En: Przeglad Gastroenterologiczny 14 (2 2019), págs. 89-103. ISSN: 18974317. DOI: 10.5114/pg.2018.81072.

[57] Matthew E Ritchie et al. «limma powers differential expression analyses for RNA-sequencing and microarray studies». En: Nucleic Acids Research 43 (7 2015). DOI: 10.1093/nar/gkv007.

[58] Stuart Russell y Peter Norvig. Artificial Intelligence: A Modern Approach. 3.ª ed. Prentice Hall, 2010.

[59] Gregory D. Sepich-Poore et al. The microbiome and human cancer. Mar. de 2021. DOI: 10.1126/science.abc4552.

[60] Abhibhav Sharma y Buddha Singh. «AE-LGBM: Sequence-based novel approach to detect interacting protein pairs via ensemble of autoencoder and LightGBM». En: Computers in Biology a 125 (2020), pág. 103964. DOI: 10.1016/j.compbiomed.2020.103964. URL: https://doi.org/10.1016/j.compbiomed.2020.103964.

[61] Manish Pratap Singh et al. «Unsupervised machine learning-based clustering identifies unique molecular signatures of colorectal cancer with distinct clinical outcomes». En: Genes & Diseases (2023).

[62] Ying Su et al. «Colon cancer diagnosis and staging classification based on machine learning and bioinformatics analysis». En: Computers in biology and medicine 145 (2022), pág. 105409.

[63] Abdel Aziz Taha y Allan Hanbury. «Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool». En: BMC medical imaging 15.1 (2015), págs. 1-28.

[64] Altyeb Taha y Sharaf Malebary. «Hybrid classification of Android malware based on fuzzy clustering and the gradient boosting machine». En: Neural Computing and Applications 33 (jun. de 2021), págs. 1-12. DOI: 10.1007/s00521-020-05450-0.

[65] Erdal Tasci et al. «IntroBias-Bias and Class Imbalance in Oncologic Data—Towards Inclusive and Transferrable AI in Large Scale Oncology Data Sets». En: Cancers 14 (12 jun. de 2022). ISSN: 20726694. DOI: 10.3390/cancers14122897.

[66] Nguyen Thai-Nghe, Zeno Gantner y Lars Schmidt-Thieme. «Cost-sensitive learning methods for imbalanced data». En: (2010), págs. 1-8. DOI: 10.1109/IJCNN.2010.5596486.

[67] Mai Tharwat et al. «MC-C-Colon Cancer Diagnosis Based on Machine Learning and Deep Learning: Modalities and Analysis Techniques». En: Sensors 22 (23 dic. de 2022). ISSN: 14248220. DOI: 10.3390/s22239250.

[68] Katarzyna Tomczak, Patrycja Czerwińska y Maciej Wiznerowicz. The Cancer Genome Atlas (TCGA): A 2015. DOI: 10.5114/wo.2014.47136.

[69] Luke K. Ursell et al. «Defining the human microbiome». En: Nutrition Reviews 70 (SUPPL. 1 ago. de 2012). ISSN: 00296643. DOI: 10.1111/j.1753-4887.2012.00493.x.

[70] John Verzani. Getting started with RStudio. .º'Reilly Media, Inc.", 2011.

[71] S. Vibinchandar y Dr V. Krishnapriya. «BREAST CANCER DATA FEATURE SELECTION USING ENSEMBLE LIGHT GRADIENT BOOSTING TECHNIQUE». En: Journal of Pharmaceutic 13 (2022), págs. 3228-3238. ISSN: 22297723. DOI: 10.47750/pnr.2022.13.S08.398.

[72] Kerstin N. Vokinger, Stefan Feuerriegel y Aaron S. Kesselheim. «IntroBias-Mitigating bias in machine learning for medicine». En: Communications Medicine 1 (1 ago. de 2021). DOI: 10.1038/s43856-021-00028-w.

[73] Chunying Wang et al. «A review of deep learning used in the hyperspectral image analysis for agriculture». En: Artificial Intelligence Review 54.7 (2021), págs. 5205-5253.

[74] Jie Xu et al. «Algorithmic fairness in computational medicine». En: (2022). DOI: 10.1016/j. URL: http://creativecommons.org/licenses/by/4.0/.

[75] Zugang Yin et al. Application of artificial intelligence in diagnosis and treatment of colorectal cancer: 2023. DOI: 10.3389/fmed.2023.1128084.

[76] Xinyu Zhang y Chu-An Liu. «Model averaging prediction by K-fold cross-validation». En: Journal of Econometrics 235.1 (2023), págs. 280-301.

[77] Zhuoyuan Zheng, Yunpeng Cai y Ye Li. «Oversampling method for imbalanced classification». En: Computing and Informatics 34.5 (2015), págs. 1017-1037.

Apéndice A
Desarrollo de Funciones R

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA