Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Pronóstico de resección y esperanza de vida en glioblastoma mediante Machine Learning

Resection and life expectancy prognosis in glioblastoma through Machine Learning.

La Laguna, 14 de julio de 2023

D. **Rafael Arnay del Arco**, con N.I.F. 78.569.591-G profesor Titular de Universidad adscrito al Departamento de Ingeniería informática y de sistemas de la Universidad de La Laguna, como tutor

D. **Javier Hernández Aceituno**, con N.I.F. 54.054.736-K profesor ayudante doctor adscrito al Departamento de Ingeniería informática y de sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A N

Que la presente memoria titulada:

"Pronóstico de resección y esperanza de vida en glioblastoma mediante Machine Learning"

ha sido realizada bajo su dirección por D. **Airam Rafael Luque León**,

con N.I.F. 79.072.491-D.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 14 de julio de 2023

Agradecimientos

Deseo expresar mi más sincero agradecimiento a todas las personas que han contribuido de manera significativa en la realización de este trabajo de fin de grado.

En primer lugar, quiero agradecer a mis tutores académicos, Rafael Arnay del Arco y Javier Hernández Aceituno, por su orientación, apoyo y dedicación a lo largo de este proyecto. Su experiencia y conocimientos han sido fundamentales para mi desarrollo académico y la culminación exitosa de este trabajo.

También quiero agradecer a mi familia y amigos por su incondicional apoyo durante toda mi etapa universitaria. Sus palabras de aliento, comprensión y motivación han sido un gran impulso en los momentos más desafiantes.

Por último, quiero reconocer a todos aquellos que, de una forma u otra, han sido parte de mi trayectoria académica. Agradezco también a todas las personas que, de manera indirecta, han contribuido a mi formación a través de sus investigaciones, libros, artículos y recursos disponibles.

Licencia

Resumen

El glioblastoma es un tipo de cáncer cerebral que resulta en una alta tasa de mortalidad y presenta variabilidad en las respuestas al tratamiento. La resección del tumor es una de las opciones terapéuticas principales, aunque su eficacia puede variar entre pacientes. El presente estudio busca explorar esta variabilidad a través de modelos de aprendizaje automático, utilizando un conjunto de datos clínicos que incluye información sobre el volumen del tumor y otras posibles afecciones derivadas de este mismo.

Estos modelos de aprendizaje automático, desarrollados utilizando la librería de sklearn, tienen como objetivo clasificar entre resección positiva o no y generar un pronóstico de la esperanza de vida. Se empleó una búsqueda en cuadrícula para la optimización de hiperparámetros y validación cruzada para la robustez del modelo.

Los resultados mostraron que los modelos de aprendizaje automático fueron capaces de generar clasificaciones de calidad para determinar la presencia de resección positiva. Sin embargo, la precisión en la generación de pronósticos de esperanza de vida no alcanzó un nivel satisfactorio.

Este trabajo demuestra la utilidad de las técnicas de aprendizaje automático en la interpretación de datos clínicos y en la toma de decisiones terapéuticas en glioblastoma, aunque también revela limitaciones que deben ser superadas para mejorar la precisión de los pronósticos.

Palabras clave: aprendizaje automático, glioblastoma, resección, pronóstico, esperanza de vida, sklearn, validación cruzada, búsqueda en cuadrícula.

Abstract

Glioblastoma is a type of brain cancer resulting in a high mortality rate and presents variability in treatment responses. Tumour resection is a main therapeutic option, although its effectiveness may vary among patients. This study aims to explore this variability through machine learning models, using a clinical dataset that includes information about the tumour volume and the involvement of the corpus callosum, among other factors.

These machine learning models, developed using the sklearn libraries, aim to classify between positive resection or not and to generate a life expectancy prognosis. A grid search was used for hyperparameter optimization and cross-validation for model robustness.

The results showed that machine learning models were able to generate quality classifications to determine the presence of positive resection. However, accuracy in generating life expectancy forecasts did not reach a satisfactory level.

This work demonstrates the utility of machine learning techniques in interpreting clinical data and in making therapeutic decisions in glioblastoma, but it also reveals limitations that need to be overcome to improve the accuracy of the forecasts.

Keywords: Machine learning, glioblastoma, resection, prognosis, life expectancy, sklearn, cross-validation, grid search

Índice general

Capítulo 1. Introducción 1

1.1 Fases del proyecto 2

Capítulo 2. Estado del arte 4

Capítulo 3. Análisis y tratamiento de los datos 6

3.1 Presentación y análisis de los datos 6

3.1.1 Distribución de los datos 7

3.1.2 Estudio de los valores 9

Primer dataset 9

Segundo dataset 10

3.1.3 Estudio de correlaciones 11

3.2 Tratamiento de los datos 13

Capítulo 4. Generación de modelos 14

4.1 Modelos seleccionados 14

4.1.1 Regresión logística: 14

4.1.2 Gaussian Naive Bayes: 14

4.1.3 Regresión lineal 15

4.1.4 Árboles de decisión 16

4.1.5 Bosques aleatorios 16

4.1.6 SVM (Support Vector Machine) 17

4.1.7 K vecinos más cercanos 18

4.1.8 MLP (Multi-Layer Perceptron) 18

4.2 Entrenamiento de los modelos 19

4.2.1 Experimentos 19

4.2.2 Técnicas usadas en el entrenamiento 20

Cross Validation 20

Grid Search 21

Hiperparámetros 22

Capítulo 5. Análisis de resultados 25

5.1 Clasificación 25

5.2 Regresión 26

Capítulo 6. Conclusiones y líneas futuras 29

Capítulo 7. Summary and Conclusions 30

Capítulo 8. Presupuesto del proyecto 31

Capítulo 9. Bibliografía 32

Índice de figuras

Ilustración 3.1 Distribución de las variables del primer dataset 8
Ilustración 3.2 Distribución de las variables del segundo dataset 9
Ilustración 3.3 Mapa de correlaciones entre las variables del primer dataset 12
Ilustración 3.4 Mapa de correlaciones entre las variables del segundo dataset 12

Índice de tablas

Tabla 3.1 Estadísticas del primer dataset 10
Tabla 3.2 Estadísticas del segundo dataset 11
Tabla 4.1 Asignación de hiperparámetros problema de clasificación 24
Tabla 4.2 Asignación de hiperparámetros problema de regresión 24
Tabla 5.1 Resultados por experimentos en clasificación 25
Tabla 5.2 Resultados por experimento en regresión 27
Tabla 8.1 Presupuesto de las actividades 31

Capítulo 1. Introducción

El glioblastoma es un tipo de cáncer cerebral agresivo y mortal que afecta a un gran número de personas en todo el mundo. Aproximadamente el 46.1% de todos los tumores cerebrales malignos primarios son glioblastomas, sin embargo, en comparación con otros tipos de cáncer, la incidencia anual es baja, con una tasa de aproximadamente 3.2 por cada 100,000 personas [1]. A pesar de los avances en el tratamiento, la tasa de supervivencia de los pacientes con glioblastoma sigue siendo baja. Por esta razón, se han llevado a cabo numerosas investigaciones para mejorar el diagnóstico y tratamiento de esta enfermedad.

La habilidad para estimar el volumen de resección y la expectativa de vida de los pacientes, utilizando variables médicas, constituye un desafío esencial en el ámbito de la medicina. Estas proyecciones son vitales para ofrecer un tratamiento personalizado en el manejo de enfermedades graves, habilitando a los profesionales de la salud a tomar decisiones fundamentadas con relación al enfoque clínico y así brindar a los pacientes la mejor atención posible.

La medicina fundamentada en evidencias ha probado ser un recurso inestimable en el proceso de toma de decisiones clínicas [2], y los progresos en tecnología y análisis de datos han facilitado el desarrollo de nuevas estrategias de pronóstico. En concreto, la implementación de modelos de Aprendizaje Automático se ha vuelto esencial en la anticipación de resultados médicos, dado que estos modelos poseen la habilidad de aprender a partir de los datos y detectar patrones complejos que pueden no ser perceptibles para los profesionales de la salud.

La generación de modelos de Machine Learning para el pronóstico de resección y esperanza de vida de los pacientes se ha convertido en un área de investigación activa debido a su potencial impacto en la atención clínica. Al conseguir información precisa acerca de la probabilidad de éxito de una resección quirúrgica y la evaluación de la expectativa de vida de los pacientes, los médicos pueden personalizar los planes de tratamiento, tomando en cuenta factores individuales y de esta forma optimizar los resultados para los pacientes.

En este marco, el propósito de este trabajo de fin de grado es diseñar y evaluar modelos de Aprendizaje Automático que empleen un conjunto específico de variables médicas para generar proyecciones precisas de resección y expectativa de vida de los pacientes. Al lograrlo, se espera aportar al progreso de la medicina de precisión y brindar un instrumento complementario para los profesionales de la salud en el proceso de toma de decisiones clínicas.

La justificación de este trabajo se basa en la necesidad de avanzar en el campo de la medicina de precisión y explotar el potencial de los modelos de Aprendizaje Automático en la generación de pronósticos precisos de resección y esperanza de vida de los pacientes.

La medicina de precisión aspira a superar el enfoque tradicional de "talla única" en el diagnóstico y tratamiento de enfermedades, reconociendo que cada individuo es único y puede

responder de manera diferente a los tratamientos [3]. Al emplear modelos de Aprendizaje Automático, se puede analizar una gran cantidad de datos clínicos y genéticos para identificar patrones y correlaciones que pueden ser desafiantes de detectar por métodos tradicionales. Estos modelos pueden aprender de los datos existentes y generar pronósticos precisos, permitiendo una toma de decisiones clínicas más fundamentada y una atención más individualizada.

La generación de pronósticos precisos de resección y esperanza de vida de los pacientes tiene implicaciones significativas en la práctica clínica. Por un lado, estos pronósticos pueden asistir a los médicos en determinar la factibilidad de una resección quirúrgica y evaluar el riesgo de complicaciones asociadas. Esto puede ser especialmente importante en el caso de enfermedades complejas y de alto riesgo, donde la toma de decisiones precisa es de vital importancia.

Por otro lado, la estimación precisa de la esperanza de vida puede incidir en la selección del tratamiento y en la planificación de cuidados paliativos, permitiendo una atención más apropiada y centrada en las necesidades personales de los pacientes. Además, esta información también puede ser útil para los pacientes y sus familias, brindándoles una comprensión más clara de la situación y ayudándoles a tomar decisiones fundamentadas.

El desarrollo de modelos de Aprendizaje Automático para la generación de pronósticos de resección y expectativa de vida posee el potencial de mejorar la eficacia y eficiencia de la atención médica. Al suministrar a los profesionales de la salud una herramienta complementaria basada en evidencia, se pueden minimizar los errores de diagnóstico, optimizar los recursos y mejorar los resultados clínicos. Además, estos modelos pueden contribuir al avance de la medicina de precisión y fomentar un enfoque más individualizado en la atención médica.

Por todo lo anteriormente mencionado, este trabajo se justifica por la necesidad de desarrollar modelos de Aprendizaje Automático para la generación de pronósticos precisos de resección y esperanza de vida de los pacientes. Estos modelos tienen el potencial de mejorar la toma de decisiones clínicas, personalizar el tratamiento y mejorar los resultados para los pacientes. Además, contribuyen al avance de la medicina de precisión al aprovechar el poder de los datos y la inteligencia artificial en la atención sanitaria.

1.1 Fases del proyecto

Este proyecto se dividió en un conjunto de fases, las cuales permitieron abordar de manera estructurada y sistemática el objetivo principal. A continuación, se describen brevemente las fases principales del proyecto:

1. **Análisis y preprocesamiento de los datos**: Debido a que el médico asociado facilitó los datos con los que se trabajará, la primera tarea a realizar fue llevar a cabo un proceso de análisis y comprensión inicial de los datos, para posteriormente realizar un preprocesamiento, para asegurar la calidad y coherencia de los mismo. En esta fase se realizaron tareas de limpieza de datos inconsistentes o faltantes, normalización de variables, eliminación de valores atípicos y selección de características relevantes para el pronóstico, basados en un análisis de las estadísticas y cómo se relacionan entre sí.

2. **Selección de modelos de Machine Learning:** A lo largo de esta fase se evaluaron diferentes algoritmos y técnicas de Machine Learning para seleccionar aquellos algoritmos que mejor

se adapten a los problemas. Se considerarán tanto modelos de clasificación como de regresión, dependiendo de la naturaleza de las variables objetivo.

3. **Entrenamiento y evaluación de los modelos:** Una vez seleccionados los modelos de Machine Learning, se procedió a entrenarlos utilizando los datos previamente preparados. Se realizó una evaluación exhaustiva de los modelos utilizando múltiples métricas de rendimiento, como la precisión, la sensibilidad, la especificidad o el error cuadrático medio, entre otras.

4. **Ajuste y optimización de los modelos:** En esta fase, se realizaron ajustes y optimizaciones de hiperparámetros en los modelos, con el objetivo de mejorar su rendimiento.

5. **Análisis e interpretación de resultados:** Finalmente, una vez obtenidos múltiples modelos en base a diferentes experimentos se analizaron e interpretaron los resultados obtenidos de los mismos. De igual forma se examinaron los factores más influyentes en el pronóstico de resección y esperanza de vida, y se discutieron la calidad de los mejores modelos y las posibles implicaciones clínicas de los hallazgos encontrados.

Capítulo 2. Estado del arte

La aplicación de técnicas de Aprendizaje Automático (ML) y Aprendizaje Profundo (DL) en el diagnóstico, pronóstico de tumores cerebrales ha sido un área de investigación activa en la última década, demostrando un potencial significativo para mejorar la atención al paciente.

Los modelos de Machine Learning, como los Árboles de Decisión, K-Nearest Neighbors (K-NN), Máquinas de Soporte Vectorial (SVM) y el Algoritmo de Random Forest se han utilizado en numerosas investigaciones para clasificar. Por ejemplo, un estudio de 2010 realizado por A Osareh et al. utilizó el algoritmo Support Vector Machine (SVM), junto con el algoritmo K-Nearest Neighbors para el diagnóstico de cáncer de mama [4]. El modelo logró una alta precisión, superando los métodos de clasificación tradicionales.

Además, los algoritmos de Machine Learning no se limitan a tareas de clasificación, sino que también han demostrado ser extremadamente útiles para tareas de regresión en diversas áreas. Como ejemplo, un estudio reciente utilizó técnicas de regresión de aprendizaje automático para estimar la distribución de la relación señal-ruido generalizada (GSNR) de caminos de luz no establecidos en redes ópticas. Este enfoque permitió a los operadores de red tomar decisiones más informadas sobre el despliegue de caminos de luz, destacando la versatilidad y la capacidad de estas técnicas para manejar una amplia gama de problemas [5].

Otra tecnología cuyo uso ha aumentado con el paso de los años, es el Deep Learning, siendo este una subcategoría de ML que utiliza redes neuronales con múltiples capas. El Deep Learning ha demostrado ser particularmente útil en la interpretación de imágenes médicas. Las Redes Neuronales Convolucionales (CNN), en particular, han sido ampliamente utilizadas para la clasificación y segmentación de tumores cerebrales. Por ejemplo, un estudio de 2020 de McKinley et al. utilizó CNN para analizar las imágenes de resonancia magnética (MRI) y predecir la necesidad de resección quirúrgica de tumores cerebrales [6]. Otro estudio destacado que emplea Redes Neuronales Convolucionales es el realizado por Fung Fung Ting y su equipo. En esta investigación, utilizaron técnicas de Deep Learning, con Redes Neuronales Convolucionales, para la clasificación precisa del cáncer de mama. Los resultados fueron excepcionales, logrando una tasa de precisión superior al 89%, lo que demuestra el potencial significativo de estas técnicas en el diagnóstico de enfermedades graves [7].

En ciertos contextos, la combinación de técnicas de Machine Learning (ML) y Deep Learning (DL) ha permitido a los investigadores optimizar la precisión de sus modelos predictivos. En 2022 un estudio propuso un marco de aprendizaje conjunto para diagnosticar la enfermedad de Alzheimer utilizando datos de neuroimagen. En este estudio, un equipo de investigadores encabezado por el autor principal Fradet, G, se utilizó una Red Neuronal Convolucional en 3D (3D-ResNet) para extraer características estructurales 3D de los datos de neuroimagen, y al mismo tiempo aplicó la técnica de Machine Learning Extreme Gradient Boosting (XGBoost) a nivel de vóxel para identificar los grupos de vóxeles más significativos en la imagen. Los datos demográficos de los pacientes y las puntuaciones de pruebas cognitivas se combinaron con las predicciones de 3D-ResNet y XGBoost para proporcionar una predicción final del diagnóstico [8]. Este enfoque logró un alto rendimiento en la detección de Alzheimer durante la primera visita de los pacientes, logrando un AUC (Area Under the Curve) promedio de 96% durante las pruebas. Este resultado ilustra cómo la integración

de técnicas de ML y DL combinadas puede mejorar la precisión de los modelos de diagnóstico en el ámbito de la neuroimagen.

En este punto se han mencionado y analizado algunas técnicas de inteligencia artificial. Podemos concluir que, a pesar de los avances significativos en la aplicación de ML y DL en la resección de tumores cerebrales y la predicción de la esperanza de vida, existen desafíos en su implementación a gran escala. Estos desafíos incluyen la necesidad de grandes conjuntos de datos de alta calidad para entrenamiento, la interpretabilidad de los modelos de ML y DL, y la validación y aprobación clínica y regulatoria [9]. Además, los modelos existentes a menudo se centran en un único tipo de tumor, como el glioblastoma, y pueden no ser aplicables a otros tipos de tumores. Por lo tanto, se necesitan más investigaciones para desarrollar modelos que puedan aplicarse a una amplia gama de tumores.

Capítulo 3. Análisis y tratamiento de los datos

Dentro de este capítulo se realizará una presentación de los datos iniciales, acompañada de un análisis de sus atributos, tales como procedencia, magnitud, distribución, entre otros. De la misma forma, se efectuará un análisis de las correlaciones entre las variables y las variables objetivo, con el fin de proporcionar una comprensión integral de los datos que se manejarán. Finalmente, se explicará el procedimiento adoptado para el tratamiento de los datos y se justificarán tales decisiones.

3.1 Presentación y análisis de los datos

Los datos utilizados en este estudio proceden de dos fuentes principales, el primer conjunto de datos proporcionado por el médico Julio Manuel Plata Bello, del Hospital Universitario de Canarias, y un segundo conjunto de datos público, de la Universidad de Pensilvania. El primer conjunto de datos consta de 136 registros con más de 50 variables médicas registradas para cada paciente. Sin embargo, siguiendo las recomendaciones del médico, se eligieron solo algunas variables específicas para su inclusión en el análisis, correspondientes a los datos de segmentación del tumor, junto con otras 3 variables que indican características adicionales de este: Subventricular, Afección de 1cm en cápsula y Afección del cuerpo calloso. Esta selección se realizó para simplificar el proceso de entrenamiento, además de la cohesión entre ambos conjuntos de datos, ya que, excepto las 3 variables mencionadas anteriormente, ambos conjuntos de datos cuentan con variables similares. A continuación, se enumeran las variables seleccionadas.

- **Edad**: La edad en el momento del diagnóstico del paciente, ya que se ha observado que la edad puede influir en la respuesta al tratamiento y en el pronóstico de la enfermedad.
- **Sexo**: El género del paciente.
- **Volumen tumoral (porcentaje)**: El volumen del tumor en relación con el volumen total del cerebro. Este dato proporciona información sobre la extensión del tumor y puede ser útil para predecir el éxito de la resección y la progresión de la enfermedad.
- **Volumen edema (porcentaje)**: El volumen del edema cerebral en relación con el volumen total del cerebro. El edema cerebral es una respuesta inflamatoria común en las lesiones cerebrales y su evaluación puede ser relevante para el pronóstico.
- **Volumen necrosis (porcentaje)**: El volumen de la necrosis en relación con el volumen total del cerebro. La necrosis es un proceso de muerte celular y su presencia puede ser indicativa de la agresividad tumoral.
- **Ratio Tumor/Edema**: La relación entre el volumen del tumor y el volumen del edema cerebral. Esta relación puede proporcionar información sobre la impasividad y la respuesta del tumor al tratamiento.

- **Ratio Tumor/Necrosis**: La relación entre el volumen del tumor y el volumen de la necrosis cerebral. Esta relación puede ser útil para evaluar la agresividad del tumor y su respuesta al tratamiento.

- **Subventricular**: Un indicador binario que representa la afectación de la zona subventricular por parte del tumor. Se ha observado que la presencia de tumor en esta área puede estar asociada con un peor pronóstico.

- **Afección de Cuerpo Calloso**: Un indicador binario que representa la afectación del cuerpo calloso por parte del tumor. El cuerpo calloso es una estructura clave en la conectividad cerebral y su afectación puede tener implicaciones clínicas significativas.

- **Afección de 1cm en cápsula**: Un indicador binario que representa la afectación de una región específica de la cápsula del tumor. Esta variable se incluyó debido a su relevancia clínica en el pronóstico.

- **Resección disco**: Un indicador binario que representa si el paciente ha sido sometido a una resección quirúrgica o no, superior al 90%. Esta variable es la variable objetivo para el problema de clasificación.

- **Periodo de exitus**: El período de tiempo transcurrido desde el diagnóstico hasta el fallecimiento del paciente. Esta la variable objetivo para generar los pronósticos de la esperanza de vida.

Reanudando el análisis de las características de los conjuntos de datos, el segundo conjunto consta de 672 registros y 9 variables médicas. Sin embargo, debido a la falta de datos de la segmentación del tumor, se empleó la herramienta OncoHabitats [10] para generar estos. No obstante, se necesitaron las resonancias magnéticas de los pacientes para generar dicha información y solo se dispuso de las resonancias de los primeros 110 pacientes, por lo que los datos se limitan a dichos registros. Cabe destacar que las tres variables adicionales presentes en el primer conjunto de datos no pudieron ser obtenidas para este segundo conjunto de datos.

3.1.1 Distribución de los datos

Con el fin de facilitar la comprensión de los datos, se realizará un análisis de la distribución de cada conjunto de datos de manera individual. Las ilustraciones 3.1 y 3.2 muestran las distribuciones de las variables.

Al enfocarnos en las variables objetivo, notamos un ligero desbalance favoreciendo a una resección positiva en el primer conjunto de datos, el cual se intensifica aún más en el segundo conjunto.

Por otro lado, al examinar la distribución de la variable "periodo exitus", en el primer conjunto de datos, observamos una forma similar a la de una distribución chi-cuadrado. La concentración de datos en el extremo izquierdo de la gráfica se asemeja a la "cola" izquierda de dicha distribución. Esta "cola" indica una alta frecuencia de observaciones en valores más bajos. Conforme nos desplazamos hacia la derecha, la frecuencia de los datos disminuye gradualmente, lo que es coherente con la característica decreciente de una distribución chi-cuadrado.

En cambio, para el segundo dataset, la gráfica muestra una distribución de datos con una campana extremadamente vertical. Esto indica que la mayoría de los datos se agrupan en una franja ajustada alrededor de un valor central. La alta densidad de datos en esta región denota una marcada tendencia hacia un valor particular, con escasa dispersión hacia los extremos. A medida que nos alejamos del valor central, la frecuencia de los datos disminuye de manera abrupta, generando una forma de distribución afilada y estrecha.

Revisando de nuevo el primer conjunto de datos con sus características adicionales, se puede observar un ligero desequilibrio para las variables 'afección de 1cm en cápsula' y 'afección del cuerpo calloso'. Sin embargo, esto no sucede para la variable 'subventricular', la cual presenta un gran desequilibrio a favor de una afección en la zona subventricular.

Para concluir con el análisis de la distribución de las variables, es importante resaltar que, en relación con el resto de las variables de los conjuntos de datos, no se pudo establecer una aproximación satisfactoria a una distribución normal. Esto se debe principalmente a la presencia notable de múltiples picos y valles en los gráficos, lo que sugiere una alta variabilidad y complejidad en los datos.

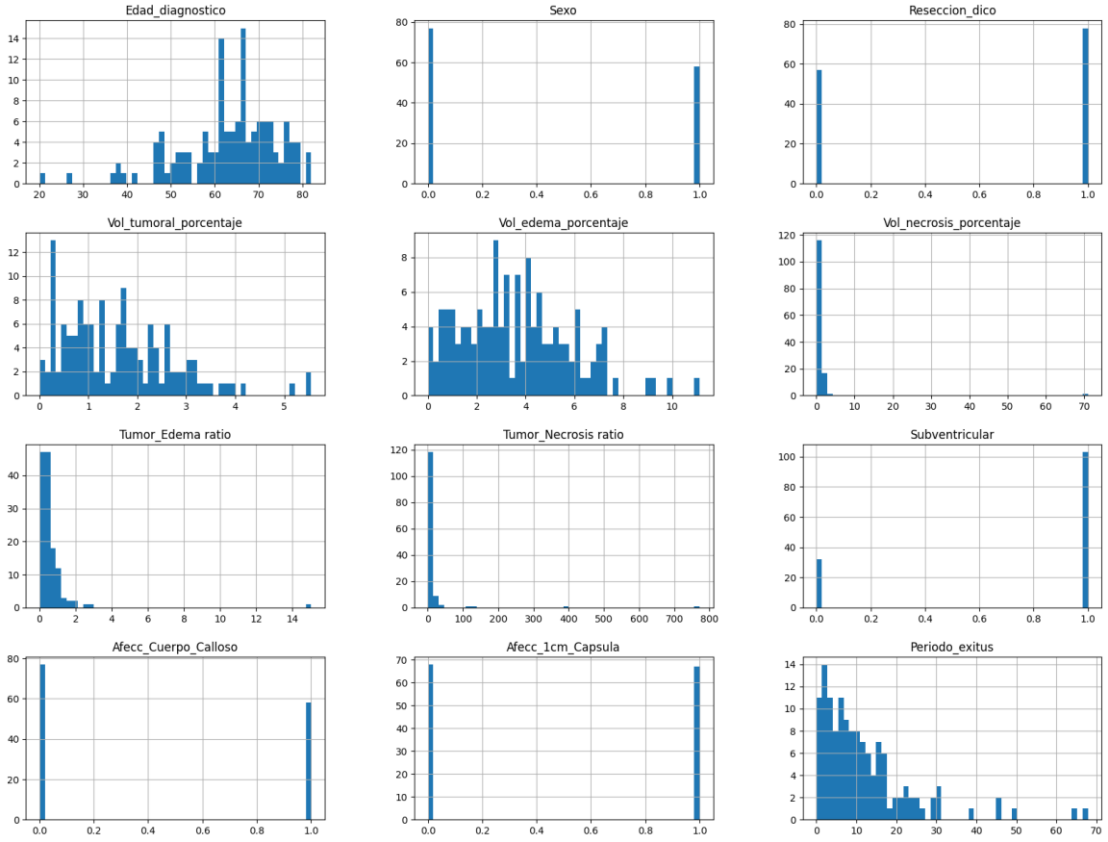

Ilustración 3.1 Distribución de las variables del primer dataset

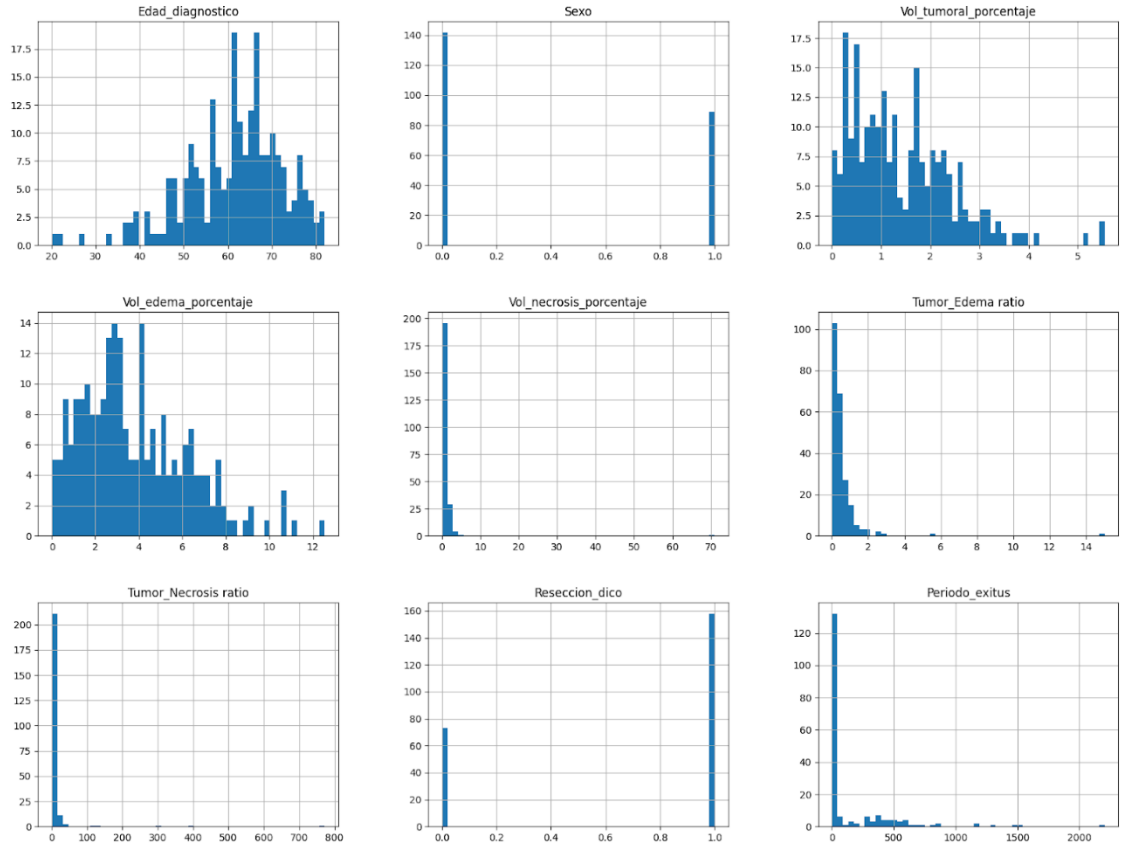

Ilustración 3.2 Distribución de las variables del segundo dataset

3.1.2 Estudio de los valores

Habiendo contextualizado y detallado la naturaleza de las variables en las secciones previas, nos adentraremos ahora en un análisis exploratorio de datos (conocido como EDA por sus siglas en inglés). El propósito de esta sección es proporcionar un resumen cuantitativo detallado de los datos disponibles en cada uno de los dos conjuntos de datos que se examinan en este estudio.

El EDA es una etapa esencial en cualquier análisis de datos y nos permitirá entender a fondo las características inherentes a nuestros datos. Esta comprensión se logrará principalmente a través de estadísticas descriptivas que brindarán información sobre la tendencia central, la dispersión y la distribución de nuestros datos.

Recordemos que cada variable representa un aspecto distinto del glioblastoma en los pacientes estudiados y cada una tiene su propio rango de valores y su propia distribución particular. Las diferencias en estas características son esenciales para informar el desarrollo posterior de modelos de aprendizaje automático, y para garantizar que los resultados obtenidos sean robustos y significativos.

Primer dataset

Al analizar los datos de la tabla 3.1, observamos que la variable Edad tiene una media de alrededor de 63.3 años, con una desviación estándar de 11.1 años. Esto indica que hay una variabilidad considerable en la edad al momento del diagnóstico. La edad mínima registrada es de 20 años, y la máxima es de 82 años.

En cuanto a la variable Sexo, la media de 0.43 sugiere que hay una mayoría de mujeres en el conjunto de datos. Además, la desviación estándar cercana a 0.5 indica una distribución bastante equitativa entre los sexos.

La variable Porcentaje de volumen tumoral tiene un promedio de aproximadamente 1.6%, con una desviación estándar de 1.14%, lo que indica que hay una variabilidad considerable en el volumen tumoral en la muestra de pacientes. Similarmente, las variables Porcentaje de volumen de edema y Porcentaje de volumen de necrosis también muestran una amplia variabilidad con desviaciones estándar de 2.23% y 6.09% respectivamente.

Por último, las variables Subventricular, Afección de cuerpo calloso, Afección de 1cm en capsula, y Resección son variables binarias, donde 1 indica la presencia de la condición y 0 su ausencia. Para Subventricular, la media de 0.76 sugiere una alta prevalencia de afecciones en la zona subventricular. Afección de cuerpo calloso y Afección de 1cm tienen medias de 0.43 y 0.50 respectivamente, sugiriendo una prevalencia más equilibrada. Resección, con una media de 0.58, sugiere que más de la mitad de los pacientes han tenido una resección positiva.

	Edad	Sexo	Vol. Tumoral (%)	Vol. Edema (%)	Vol. Necrosis (%)	Subventricular	Afecc. Cuerpo Calloso	Afecc. 1cm Capsula	Resección	Periodo exitus
Media	63.3	0.43	1.62	3.61	1.13	0.76	0.43	0.5	0.58	11.84
Desviación típica	11.12	0.5	1.14	2.23	6.09	0.43	0.5	0.50	0.5	11.86
Mínimo	20	0	0	0	0	0	0	0	0	0
25%	58	0	0.75	1.83	0.12	1	0	0	0	3.63
50%	65	0	1.48	3.33	0.41	1	0	0	1	8.3
75%	70.5	1	2.32	5.03	0.94	1	1	1	1	15.36
Máximo	82	1	5.55	11.12	71	1	1	1	1	67.9

Tabla 3.1 Estadísticas del primer dataset

Segundo dataset

Volviendo con el segundo dataset, se puede apreciar con respecto a la tabla 3.2, que la variable Edad tiene un promedio de aproximadamente 62.46 años con una desviación estándar de 12.36 años, lo que indica una variabilidad significativa en la edad al momento del escaneo. La edad mínima y máxima registradas son de 18.65 y 88.50 años respectivamente. Además, se aprecia un cierto desbalanceo a favor de los hombres con una media de 0.3964.

Las variables Porcentaje de volumen tumoral, Porcentaje de volumen de edema, y Porcentaje de volumen de necrosis tienen promedios de 1.22%, 4.29% y 0.82% respectivamente. Sus desviaciones estándar respectivas de 0.89%, 2.84% y 0.93% indican una variabilidad considerable en estas características entre los pacientes. Es importante tener en cuenta que estas variables solo tienen 110 observaciones cada una, lo que sugiere que hay muchos valores faltantes en estos campos.

	Edad	Sexo	Vol. Tumoral (%)	Vol. Edema (%)	Vol. Necrosis (%)	Resección	Periodo exitus
Media	62.46	0.4	1.22	4.291455	0.818000	0.83333	12.36
Desviación típica	12.36	0.49	0.89	2.836244	0.928535	0.37	10.40
Mínimo	18.65	0	0.04	0.280000	0.000000	0	0.09
25%	55.26	0	0.46	2.285000	0.162500	1	4.62
50%	63.02	0	1.05	3.260000	0.515000	1	10.71
75%	71.25	1	1.87	6.387500	1.082500	1	16.22
Máximo	88.50	1	3.86	13.990000	4.710000	1	64.61

Tabla 3.2 Estadísticas del segundo dataset

3.1.3 Estudio de correlaciones

En el ámbito del análisis de datos, uno de los aspectos fundamentales es comprender las relaciones existentes entre las variables. Estas relaciones pueden proporcionar una visión más profunda y significativa de los datos, revelando patrones, asociaciones y dependencias ocultas. Una técnica frecuentemente aplicada para indagar en estas conexiones es el análisis de correlaciones.

El análisis de correlaciones posibilita la cuantificación del nivel de relación entre parejas de variables en un conjunto de datos. Este índice estadístico, conocido como coeficiente de correlación, expresa la dirección y la intensidad del vínculo entre dos variables. Un coeficiente de correlación próximo a +1 señala una correlación positiva intensa, mientras que un valor cercano a -1 representa una correlación negativa de igual intensidad. En contraposición, un coeficiente próximo a 0 implica una correlación débil o inexistente.

Las ilustraciones 3.3 y 3.4 evidencian las correlaciones entre las variables de cada conjunto de datos. Si nos enfocamos en las variables objetivo (Resección y periodo de exitus), se observa una correlación mínima en comparación con las demás variables, resaltando una correlación más elevada entre las mismas variables objetivo del primer conjunto de datos. Sin embargo, debido a la naturaleza del estudio, esta correlación solo será útil en la generación de modelos de regresión. Con relación al primer conjunto de datos, se destaca una variable por su notable correlación negativa: Afección de 1cm en cápsula, con una correlación de -0,47. Continuando con el primer dataset no existen grandes correlaciones entre las variables, con la excepción de las ya mencionadas y las variables afección subventricular y afección del cuerpo calloso, con una correlación de -0,26 y -0,28 respectivamente. Por otro lado, con respecto al segundo dataset no se encuentran correlaciones destacables.

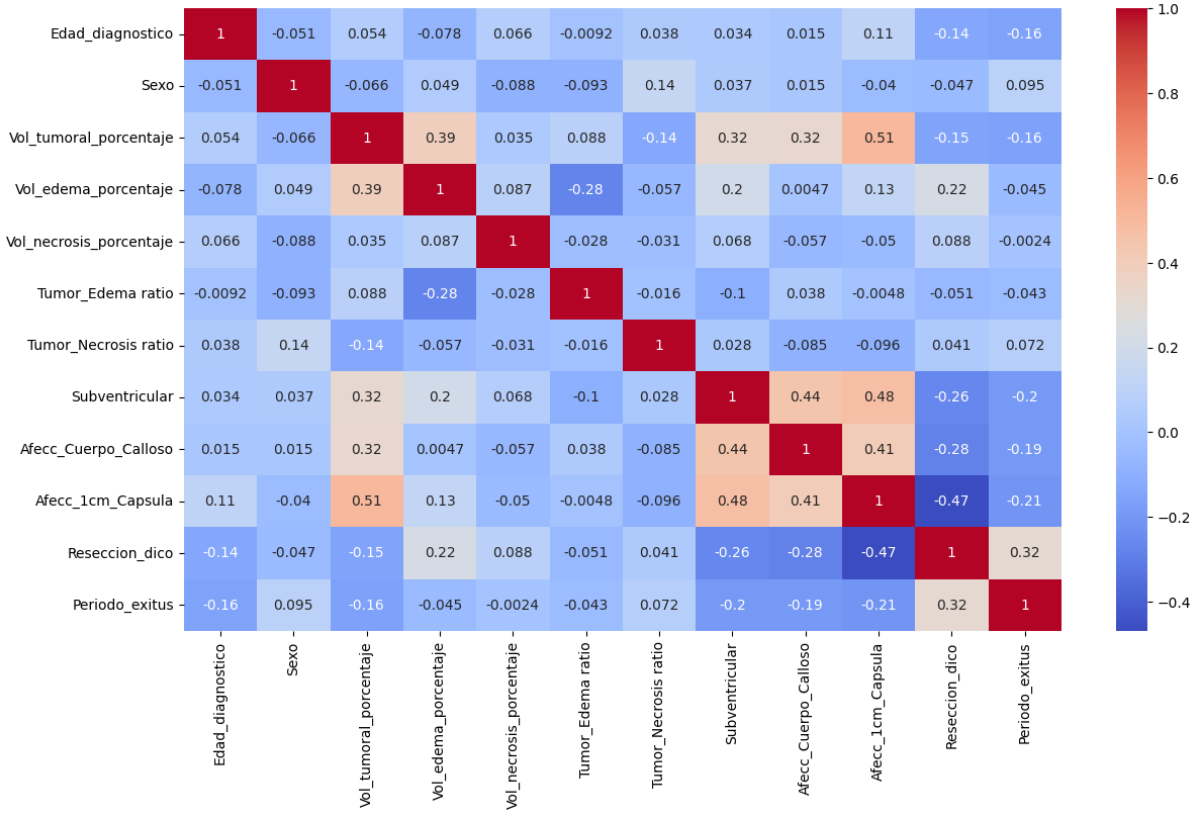

Ilustración 3.3 Mapa de correlaciones entre las variables del primer dataset

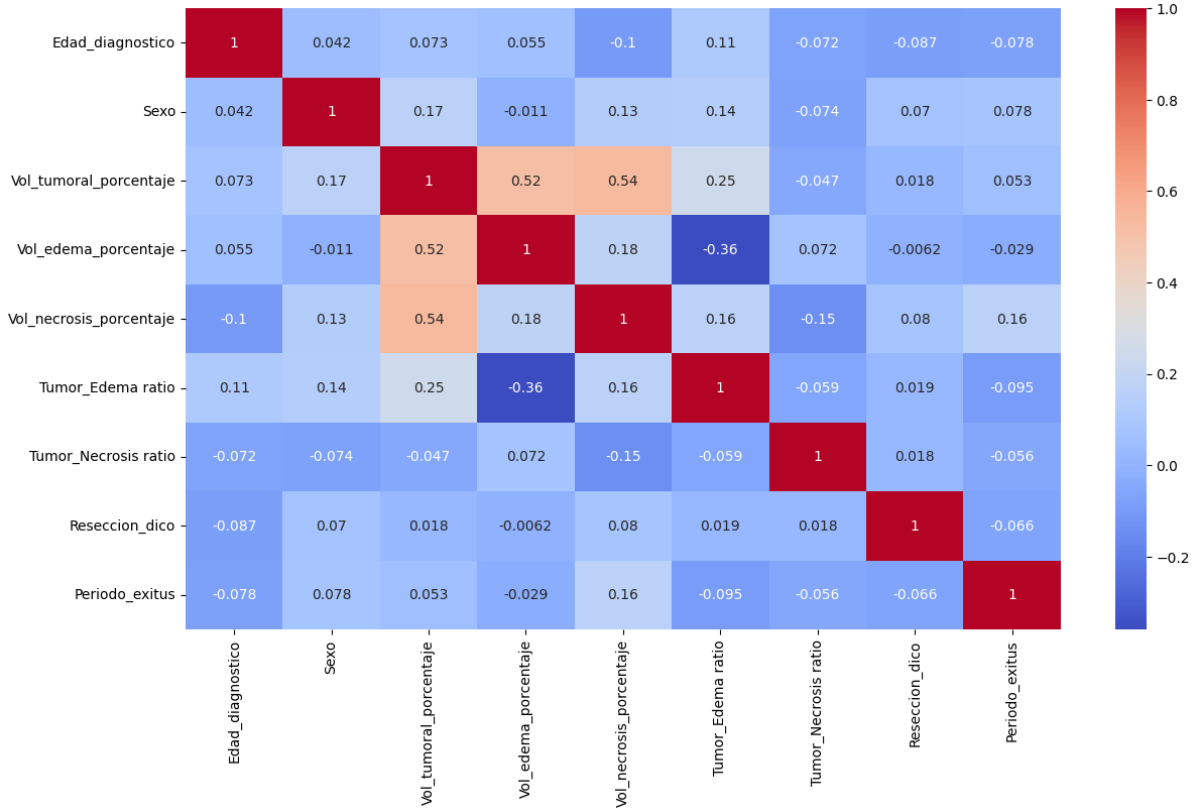

Ilustración 3.4 Mapa de correlaciones entre las variables del segundo dataset

3.2 Tratamiento de los datos

Previo a la implementación de los datos en los modelos de Machine Learning, se efectuó un preprocesamiento de los datos. Inicialmente, se normalizaron los valores para manipular enteros y flotantes y se estandarizaron los identificadores de cada columna, lo que optimizó la comparación y el análisis de estos. Además, se llevaron a cabo ajustes particulares en el segundo conjunto de datos para integrar la información de segmentación obtenida a través de los informes generados por la herramienta OncoHabitats.

Por otro lado, debido a una discrepancia en las unidades de medida de la variable objetivo que alude a la expectativa de vida de los pacientes, que en un conjunto de datos se expresa en días y en el otro en meses, fue necesario normalizar ambas. Se decidió trabajar con meses, resultando en una variable objetivo menos precisa, pero facilitando el entrenamiento de los modelos.

Con relación al manejo de datos faltantes, se optó por eliminar las filas que presentaban valores nulos en cualquier variable. Este proceder se llevó a cabo con el propósito de preservar la integridad y consistencia de los datos empleados en el análisis. Al remover las filas con valores ausentes, se asegura que el conjunto de datos usado para entrenar y evaluar los modelos sea completo y no contenga información incorrecta o incompleta. De este modo, se trabaja exclusivamente con información verídica proveniente del seguimiento de los pacientes.

Además de los procedimientos de preprocesamiento mencionados previamente, se realizó una normalización de los datos para aumentar la coherencia y facilitar su interpretación. Para este fin, se empleó una técnica de normalización estándar, multiplicando cada valor por la media y dividiéndolo por la desviación estándar de la variable correspondiente. Esta normalización permite que todas las variables tengan una escala similar y elimina cualquier sesgo o diferencia en la distribución de los datos. Así, se asegura que todas las variables aporten de manera equitativa en los modelos de Machine Learning usados para el análisis de los datos.

Capítulo 4. Generación de modelos

En este capítulo se presentarán los modelos seleccionados para los problemas inicialmente planteados. Además de explicar el funcionamiento de cada uno de estos. De igual forma se detallan las técnicas empleadas para el entrenamiento de estos, junto con los hiperparámetros específicos de cada modelo.

4.1 Modelos seleccionados

En este epígrafe se indican y explican los modelos seleccionados para los problemas de regresión y clasificación. Cabe resaltar que se seleccionaron una gran variedad de modelos para incrementar la posibilidad de generar modelos de calidad.

4.1.1 Regresión logística:

La regresión logística es un modelo de aprendizaje automático aplicado principalmente en la clasificación binaria. Pese a lo que sugiere su nombre, no es un modelo de regresión en el sentido convencional, sino un modelo de naturaleza probabilística. La regresión logística tiene como objetivo calcular la probabilidad de que un evento pertenezca a una categoría específica en base a las variables de entrada.

Este modelo se fundamenta en la función logística, también llamada función sigmoide, para efectuar la estimación. La función sigmoide aplica una transformación a una combinación lineal de las variables predictoras, generando un valor comprendido entre 0 y 1, que simboliza la probabilidad de pertenencia a la categoría positiva.

Durante el proceso de entrenamiento del modelo, los coeficientes de regresión se optimizan utilizando un algoritmo, como podría ser el descenso de gradiente. El objetivo es incrementar la verosimilitud de los datos observados basándose en las probabilidades generadas por el modelo.

Una vez entrenado, el modelo de regresión logística puede realizar la clasificación de nuevos ejemplos. Dado un conjunto de variables predictoras, el modelo calcula la probabilidad de pertenencia a la clase positiva utilizando los coeficientes aprendidos durante el entrenamiento. Luego, se aplica un umbral para convertir las probabilidades en etiquetas de clase, por ejemplo, asignando la etiqueta "1" si la probabilidad supera un umbral determinado, y la etiqueta "0" en caso contrario.

La regresión logística es un modelo lineal generalizado, es decir, puede manejar tanto variables numéricas como categóricas. Además, puede ser extendida para realizar clasificaciones multiclase utilizando técnicas como la regresión logística multinomial o la regresión logística ordinal, aunque esta no sea de interés en el ámbito de este estudio.

4.1.2 Gaussian Naive Bayes:

El clasificador Naive Bayes Gaussiano es un algoritmo de aprendizaje automático basado en el teorema de Bayes y la presunción de independencia condicional. Se utiliza para la clasificación de datos y se basa en la distribución normal o Gaussiana de los atributos.

El Naive Bayes Gaussiano asume que los atributos se distribuyen de manera independiente entre las clases, pero siguen una distribución normal dentro de cada clase. Este modelo es apropiado cuando los atributos continuos se ajustan a una distribución normal y cuando la independencia condicional es una suposición plausible para el problema en cuestión.

Durante el entrenamiento, el algoritmo estima los parámetros estadísticos de cada clase, como la media y la varianza, para cada característica. Estos parámetros se utilizan para construir el modelo de distribución normal de cada clase. El modelo aprendido se utiliza para calcular la probabilidad de que un nuevo ejemplo pertenezca a cada clase utilizando el teorema de Bayes.

El teorema de Bayes establece que la probabilidad de que un evento ocurra dado otro evento relacionado puede ser calculada utilizando la probabilidad condicional y las probabilidades marginales. En el caso del clasificador Naive Bayes Gaussiano, se calcula la probabilidad de que un ejemplo pertenezca a cada clase dado sus valores de características observadas.

Una vez calculadas las probabilidades de pertenencia a cada clase, el clasificador asigna el ejemplo a la clase con la probabilidad más alta. Este enfoque se basa en la suposición "naive" (ingenua) de independencia condicional entre las características, lo que permite simplificar el cálculo de las probabilidades mediante la multiplicación de las probabilidades de cada característica.

El clasificador Gaussiano Naive Bayes es computacionalmente eficiente y puede gestionar grandes cantidades de datos. No obstante, la suposición de independencia condicional puede no ser válida en todos los contextos, lo que puede afectar al rendimiento del modelo. Además, este clasificador no es apto para atributos categóricos o discretos, ya que asume una distribución normal en los atributos continuos, lo cual podría generar complicaciones para su implementación en este estudio debido a la presencia de numerosas variables categóricas en varios experimentos.

4.1.3 Regresión lineal

El modelo de regresión lineal es un algoritmo de aprendizaje automático utilizado para resolver problemas de regresión. Se basa en la relación lineal entre las características de entrada y la variable objetivo.

El objetivo del modelo de regresión lineal es hallar la línea recta que mejor se ajuste a los datos de entrenamiento, buscando minimizar la suma de los cuadrados de los errores entre las predicciones generadas por el modelo y los valores reales. En su versión más sencilla, el modelo de regresión lineal puede ser representado por la ecuación:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

Donde *y* es la variable objetivo a predecir, x_1, x_2,..., x_n son las características de entrada, y w_0, w_1, w_2, ..., w_n son los coeficientes del modelo que se ajustan durante el entrenamiento.

El entrenamiento del modelo de regresión lineal implica encontrar los valores óptimos para los coeficientes w_0, w_1, w_2, ..., w_n que minimicen la función de costo. Habitualmente, esta función se define como el error cuadrático medio (MSE, por sus siglas en inglés) o alguna de sus variantes.

Es importante destacar que el modelo de regresión lineal asume una relación lineal entre las características y la variable objetivo. Si los datos no siguen una relación lineal, el rendimiento del modelo puede ser limitado. En tales escenarios, se podrían aplicar técnicas de preprocesamiento o explorar modelos no lineales sofisticados.

4.1.4 Árboles de decisión

El algoritmo de árboles de decisión es, al igual que todos los algoritmos implementados, un método de aprendizaje automático supervisado. Este es utilizado tanto para problemas de clasificación como de regresión. Comienza con un conjunto de datos de entrenamiento y construye un modelo de árbol que se divide en ramas y nodos de decisión. Cada nodo representa una característica del conjunto de datos y cada rama representa una decisión basada en esa característica. El objetivo es crear un árbol que haga predicciones precisas y tenga la menor complejidad posible.

El proceso de construcción del árbol comienza con el nodo raíz, que contiene todo el conjunto de datos. Luego, se selecciona una característica para dividir el conjunto de datos en subconjuntos más pequeños en función de ciertos criterios. Los criterios comunes son la ganancia de información o la reducción de impureza, que evalúan la homogeneidad de los subconjuntos resultantes. La característica elegida debe ser la que maximice la ganancia de información o la reducción de la impureza.

Una vez que se ha realizado una división, se repite el proceso para cada subconjunto resultante en cada nodo hijo. Esto continúa recursivamente hasta que se cumple algún criterio de parada, como alcanzar una profundidad máxima, un número mínimo de muestras por hoja o una impureza mínima.

Cuando se trata de clasificación, cada hoja del árbol representa una clase y las muestras se asignan a la clase mayoritaria en esa hoja. Para la regresión, en cambio, cada hoja contiene un valor numérico que es la media o mediana de las muestras en esa hoja.

En resumen, el algoritmo de árboles de decisión construye un árbol jerárquico que realiza divisiones en función de características para clasificar o predecir valores. La construcción se basa en la selección de características óptimas y la división de datos en subconjuntos puros. Este enfoque permite interpretar fácilmente las decisiones tomadas por el modelo y puede manejar datos categóricos y numéricos, aportando gran versatilidad, la cual puede ser de gran ayuda al trabajar con datos de diferentes tipos, como es el caso particular de este proyecto.

Por otro lado, centrándonos en las diferencias claves en cuanto a la hora de su implementación para cada tarea, podemos apreciar ciertas diferencias. Para el problema de clasificación, el modelo utiliza medidas como la ganancia de información o la reducción de la impureza para decidir cómo dividir los datos y construir el árbol. Cada hoja del árbol representa una clase y las muestras se asignan a la clase mayoritaria en esa hoja. Por otro lado, el modelo aplicado a problemas de regresión, cada hoja del árbol contiene un valor numérico que es la media o mediana de las muestras en esa hoja. En lugar de utilizar medidas de impureza, el algoritmo utiliza la reducción del error cuadrado medio (MSE) o la reducción del error absoluto medio (MAE) para decidir cómo realizar las divisiones y construir el árbol.

4.1.5 Bosques aleatorios

El algoritmo de bosques aleatorios combina múltiples árboles de decisión para realizar predicciones. Su objetivo principal es reducir la varianza y el sobreajuste que pueden presentarse en un solo árbol de decisión.

El proceso de construcción de un bosque aleatorio comienza creando múltiples árboles de decisión independientes y no correlacionados entre sí. Para cada árbol, se utiliza una técnica llamada "muestreo con reemplazo" o bootstrap para crear un conjunto de datos de entrenamiento diferente. Esto significa que se seleccionan muestras del conjunto de datos original de forma aleatoria, permitiendo que una misma muestra aparezca múltiples veces o que algunas muestras no se incluyan.

Tras la creación del conjunto de datos de entrenamiento para cada árbol, se procede a construir los árboles de decisión de la misma manera que en el algoritmo de árboles de decisión. Sin embargo, hay una diferencia clave en el proceso de construcción: en cada nodo, en lugar de considerar todas las características para hacer una división, se selecciona un subconjunto aleatorio de características. Este enfoque se conoce como "aleatoriedad de características" y tiene el propósito de evitar que los árboles dependan demasiado de una única característica dominante.

Una vez que se han construido todos los árboles del bosque, para realizar una predicción, se toma la decisión de cada árbol y se elige la clase mayoritaria en el caso de un problema de clasificación o se calcula la media de las predicciones en el caso de un problema de regresión. La idea detrás de esto es que cada árbol tiene su propia "opinión" y, en conjunto, los árboles pueden producir una predicción más robusta y precisa.

Además de reducir la varianza y el sobreajuste, los bosques aleatorios también proporcionan medidas de importancia de características. Estas medidas evalúan cuánto contribuye cada característica al rendimiento del modelo y son útiles para seleccionar las características más relevantes en un problema de aprendizaje automático.

En resumen, el algoritmo de bosques aleatorios combina múltiples árboles de decisión independientes mediante el muestreo con reemplazo y la aleatorización de características. Esto reduce la varianza y el sobreajuste, y produce predicciones más robustas y precisas.

4.1.6 SVM (Support Vector Machine)

Las Máquinas de Vectores de Soporte (SVM) tienen como objetivo principal encontrar el hiperplano óptimo que maximice la separación entre las clases o se ajuste mejor a los datos.

En el caso de la clasificación, el algoritmo SVM busca un hiperplano en un espacio de alta dimensión que pueda separar las muestras de diferentes clases de manera óptima. Un hiperplano es una representación geométrica de una dimensión menos que los datos de entrada. Si los datos no son linealmente separables, SVM utiliza una técnica llamada "kernel trick" para transformar los datos en un espacio de mayor dimensión donde sí puedan ser separados por un hiperplano.

El objetivo es encontrar el hiperplano que maximice el margen, que es la distancia entre el hiperplano y las muestras más cercanas de cada clase. Estas muestras se conocen como vectores de soporte. SVM busca minimizar el error de clasificación y, al mismo tiempo, maximizar el margen, lo cual se conoce como la función de pérdida hinge.

En el caso de la regresión, el algoritmo SVM busca un hiperplano que minimice la cantidad de muestras que se encuentran fuera de un rango de tolerancia alrededor del hiperplano. El objetivo es encontrar el hiperplano que mejor se ajuste a los datos de entrenamiento, minimizando la función de pérdida epsilon-insensitive.

En ambos casos, SVM utiliza un enfoque de optimización convexa para encontrar los parámetros óptimos del modelo. La optimización se basa en resolver un problema de programación cuadrática, donde se busca encontrar los pesos y los sesgos que definen el hiperplano óptimo.

En resumen, el algoritmo SVM busca encontrar el hiperplano óptimo que maximice la separación entre clases o se ajuste mejor a los datos en un espacio de alta dimensión. Utiliza técnicas como el kernel trick para manejar datos no linealmente separables y resuelve problemas de optimización convexa para encontrar los parámetros del modelo.

4.1.7 K vecinos más cercanos

En esencia, el algoritmo de K vecinos se basa en la premisa de que los puntos de datos similares tienden a pertenecer a la misma clase o tener valores similares.

En el caso de la clasificación, el algoritmo K vecinos más cercanos encuentra los K puntos de datos más cercanos a una muestra de prueba en el espacio de características. La etiqueta de clase más común entre los K vecinos se asigna como la etiqueta de la muestra de prueba. La elección del valor de K es crucial, ya que un valor demasiado bajo puede llevar a sobreajuste y un valor demasiado alto puede llevar a subajuste.

En el caso de la regresión, el algoritmo K vecinos más cercanos encuentra los K puntos de datos más cercanos a una muestra de prueba y calcula la media o mediana de los valores de las muestras vecinas como el valor de predicción para la muestra de prueba.

Para calcular la proximidad entre las muestras, el algoritmo utiliza una métrica de distancia, como la distancia euclidiana, la distancia de Manhattan o la distancia de Minkowski. La elección de la métrica de distancia depende del problema y la naturaleza de los datos.

Además, el algoritmo puede verse afectado por la normalización de características, ya que las características con diferentes escalas pueden tener un impacto desproporcionado en la distancia. Por dicha razón se tomó la decisión de normalizar los datos antes de entrenar los modelos.

En resumen, el algoritmo de K vecinos más cercanos encuentra los K puntos de datos más cercanos a una muestra de prueba y utiliza su información para realizar clasificación o regresión. La elección del valor de K y la métrica de distancia son aspectos clave en el funcionamiento del algoritmo.

4.1.8 MLP (Multi-Layer Perceptron)

El Perceptrón Multicapa (MLP, por sus siglas en inglés) es un tipo de red neuronal artificial y está compuesto por múltiples capas de neuronas, incluyendo una capa de entrada, una o más capas ocultas y una capa de salida.

Cada neurona en la capa de entrada recibe los valores de las características de entrada. Cada neurona en las capas ocultas y la capa de salida tiene un conjunto de pesos y un sesgo asociados. Estos pesos y sesgos se ajustan durante el proceso de entrenamiento de la red para mejorar las predicciones.

El funcionamiento del MLP se basa en la propagación hacia adelante (forward propagation) y la retropropagación del error (backpropagation). En la propagación hacia adelante, los valores de entrada se multiplican por los pesos y se pasan a través de una función de activación en cada

neurona para generar las salidas. Las funciones de activación comunes incluyen la función sigmoide, la función ReLU o la función tangente hiperbólica.

Una vez que se generan las salidas de la capa de salida, se calcula el error entre las predicciones y los valores reales. Luego, en la retropropagación del error, se ajustan los pesos y sesgos en sentido contrario, propagando el error hacia atrás a través de la red. Esto se realiza utilizando un algoritmo de optimización, como el descenso de gradiente, para minimizar el error y actualizar los pesos de manera iterativa.

El proceso de entrenamiento continúa iterativamente, ajustando los pesos y sesgos hasta que se alcance un criterio de convergencia, como un número máximo de iteraciones o un error mínimo aceptable. Una vez que la red está entrenada, se puede utilizar para hacer predicciones en nuevos datos.

En cuanto a su aplicación para problemas de clasificación. Utiliza una función de activación en la capa de salida, como la función sigmoide o la función softmax, para generar las probabilidades de pertenencia a cada clase y realizar la clasificación.

Por otro lado, para problemas de regresión, la capa de salida utiliza una función de activación lineal o identidad para generar una salida numérica directa, sin transformación adicional.

4.2 Entrenamiento de los modelos

En este apartado se detallan los experimentos realizados, los pasos previos a la formación de los modelos y las estrategias implementadas durante dicho entrenamiento. Finalmente, se expondrán los conjuntos de hiperparámetros establecidos para cada modelo.

4.2.1 Experimentos

Para asegurar la máxima calidad de los modelos, se generaron diversas instancias para una variedad de experimentos, tanto para la problemática de clasificación como para la de regresión. Los experimentos se efectuaron en un entorno en el que se contaba con dos conjuntos de datos con ciertas diferencias. Ambos conjuntos compartían las mismas columnas, pero el primero contaba con algunas columnas adicionales que el segundo conjunto no incluía. Esta discrepancia representó un reto en términos de cómo aprovechar efectivamente ambas fuentes de datos, especialmente si se pretendía utilizar las columnas adicionales.

Para enfrentar este desafío, se diseñaron experimentos que implicaban el uso de distintas combinaciones de características. Las alternativas consideradas en los experimentos fueron las siguientes:

- **Uso de ratios o no (R)**: Se exploró la inclusión de ratios en el análisis de las variables. Las ratios son medidas que relacionan dos variables entre sí, lo cual puede proporcionar información adicional sobre la relación entre ellas.
- **Uso del primer dataset o de ambos (1 o 2)**: Se evaluó si era más adecuado utilizar exclusivamente el primer conjunto de datos o combinar ambos conjuntos para generar un conjunto de datos más completo. Al utilizar solo el primer conjunto, se aprovechaban las columnas adicionales disponibles, pero se perdía la información contenida en el segundo conjunto. Al combinar ambos conjuntos, se obtenía un conjunto de datos más

completo, aunque con la limitación de que las columnas adicionales solo estarían disponibles para las muestras del primer conjunto.

- **Uso de las variables adicionales del primer dataset o no (M)**: Dado que el primer conjunto de datos contenía variables adicionales, se investigó si incluir estas variables mejoraba el rendimiento de los modelos en comparación con el uso exclusivo del segundo conjunto de datos. Sin embargo, debido a la falta de datos correspondientes a estas variables adicionales en el segundo conjunto, su inclusión estaba limitada a las muestras del primer conjunto donde estuvieran disponibles. Esta consideración fue importante para evaluar si el beneficio de incluir las variables adicionales justificaba el uso de un conjunto de datos potencialmente más limitado.
- **Uso de filtrado de valores atípicos o no (F)**: Se consideró la opción de aplicar técnicas de filtrado de valores atípicos para eliminar observaciones que se alejan significativamente de la tendencia general de los datos. Esto en base a una disparidad de más del doble del rango intercuartílico de los volúmenes y los ratios. Esto se hizo para evaluar si el filtrado de valores atípicos mejoraba la calidad de los pronósticos de esperanza de vida o la capacidad de clasificación de resección. El filtrado de valores atípicos podría ayudar a eliminar datos ruidosos o atípicos que podrían afectar negativamente el rendimiento de los modelos.

Estas combinaciones de opciones posibilitaron una exploración exhaustiva de las diversas configuraciones posibles, permitiendo evaluar el impacto de cada una en el desempeño de los modelos de Machine Learning en la generación de pronósticos de esperanza de vida y en la clasificación de resección. La comparación de los resultados obtenidos en cada combinación de características brindó una comprensión más profunda de qué variables eran más relevantes, cómo se relacionaban entre sí y cómo influyen en el rendimiento general de los modelos.

4.2.2 Técnicas usadas en el entrenamiento

Se empleó una combinación de técnicas de Machine Learning con el objetivo de asegurar un entrenamiento óptimo de los modelos. Además, se realizó una división del conjunto de datos en una proporción de 80/20 para entrenamiento y prueba, respectivamente. El 80% de los datos se utilizó para entrenar el modelo, mientras que el 20% restante se reservó para evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento. Esta división en entrenamiento/prueba permitió obtener una estimación realista del desempeño del modelo en datos nuevos y no utilizados durante el proceso de entrenamiento. A continuación, se describe cómo se implementaron estas técnicas en cada modelo.

Cross Validation

La técnica de cross-validation, o validación cruzada, es un método de evaluación de modelos de Machine Learning que busca entender cuán bien el modelo se desempeñaría con datos nuevos, no vistos durante el entrenamiento. Esta técnica es ampliamente utilizada tanto para comparar diferentes modelos como para ajustar los hiperparámetros de un modelo específico.

El proceso de validación cruzada implica dividir el conjunto de datos en dos partes, donde una se usa para entrenar el modelo y la otra para evaluar su rendimiento. Este procedimiento se realiza varias veces, cambiando la partición de los datos en cada iteración. Este método permite un mejor aprovechamiento de los datos disponibles, ya que cada observación se usa tanto para el entrenamiento como para la evaluación en diferentes iteraciones.

Existen diferentes tipos de validación cruzada, sin embargo, la elegida para el entrenamiento de los modelos es la validación cruzada de k-iteraciones. En este enfoque, el conjunto de datos se divide en 'K' subconjuntos del mismo tamaño. El modelo se entrena 'K' veces, y en cada vez, se utiliza un subconjunto diferente como conjunto de prueba y los 'K-1' restantes como conjunto de entrenamiento. Luego, los resultados de cada iteración se promedian para obtener una única métrica de rendimiento.

Cabe resaltar que, en cada modelo se implementó la técnica de validación cruzada con 5 pliegues. Esta estrategia posibilitó obtener una evaluación más sólida del rendimiento del modelo, al utilizar el conjunto de datos completo tanto para el entrenamiento como para la validación. Esto previene la dependencia de una única división en entrenamiento/prueba y evita problemas como el sobreajuste por memorización de resultados, lo cual es especialmente relevante debido a la escasez de datos necesarios para lograr modelos de alta calidad.

Grid Search

Grid Search, también conocido como búsqueda en cuadrícula, es un método de hiper ajuste utilizado para optimizar los parámetros de un modelo de aprendizaje automático. Los hiperparámetros son configuraciones que se establecen antes de entrenar un modelo y que pueden afectar la velocidad y calidad del aprendizaje del modelo. Por ejemplo, en un modelo de regresión logística, un hiperparámetro podría ser el "learning rate", que determina qué tan rápido el modelo se ajusta a los datos.

El método de Grid Search se llama así porque, en esencia, se crea una cuadrícula de hiperparámetros y se itera sistemáticamente a través de todas las combinaciones posibles. Para cada combinación de hiperparámetros, se entrena y valida un modelo, y luego se registra el rendimiento del modelo.

Aquí están los pasos básicos para implementar la búsqueda en cuadrícula:

- **Definición de la cuadrícula de parámetros**: El primer paso es definir el conjunto de hiperparámetros que se desea ajustar. Esto creará la "cuadrícula" de combinaciones a probar.
- **Selección de la métrica de rendimiento**: Se debe decidir cuál será la métrica de rendimiento que utilizarás para evaluar el rendimiento de cada combinación de hiperparámetros. Esto puede ser algo como la precisión, el AUC, el error cuadrático medio, entre otros, dependiendo del tipo de problema que se está tratando de resolver.
- **Búsqueda y evaluación**: Luego, se realiza la búsqueda en cuadrícula, que esencialmente implica entrenar un modelo para cada combinación de hiperparámetros en la cuadrícula. Para cada modelo, se evalúa el rendimiento utilizando la métrica de rendimiento elegida.
- **Selección del mejor conjunto de parámetros**: Una vez que todos los modelos se han entrenado y evaluado, se puede comparar los resultados y seleccionar el conjunto de hiperparámetros que dio como resultado el mejor rendimiento.

Esta técnica fue empleada para cada modelo, lo que implicó explorar diferentes combinaciones de hiperparámetros específicos de cada modelo, se evaluando el rendimiento del modelo en cada combinación de hiperparámetros utilizando la validación cruzada. Esto permitió

identificar los mejores valores de los hiperparámetros optimizando el rendimiento de los modelos para los problemas en cuestión.

Hiperparámetros

Los modelos empleados son tan buenos como los hiperparámetros que se seleccionan para su funcionamiento, por lo que son de suma importancia para obtener modelos de calidad. En esta sección se detallan los hiperparámetros empleados por los modelos de clasificación y regresión utilizados en el estudio. A continuación, se explican los hiperparámetros seleccionados para cada modelo.

Regresión lineal

- **fit_intercept**: Indica si se debe calcular la intersección para este modelo. Si se establece en False, no se calculará la intersección (el modelo asume que los datos están centrados).

Arboles de decisión

- **max_depth**: Define la profundidad máxima que puede tener el árbol. En caso de ser None, la expansión de los nodos continuará hasta que todas las hojas sean puras o contengan menos muestras que las especificadas en “min_samples_split”.
- **min_samples_leaf**: El número mínimo de muestras necesarias para estar en un nodo hoja.

Bosques aleatorios

- **n_estimators**: Se refiere a la cantidad de árboles que se generan en el bosque durante el entrenamiento.
- **max_depth**: Similar al caso de los árboles de decisión, define la profundidad máxima de cada árbol.
- **min_samples_leaf**: Especifica el número mínimo de muestras que un nodo debe tener para ser considerado nodo hoja.

Máquinas de soporte de vectores

- **C**: Controla el equilibrio entre obtener un margen de separación más grande y minimizar las clasificaciones erróneas. Un valor menor de C implica un margen más suave, mientras que un valor mayor implica un margen más estricto.
- **kernel**: Define la función que se utiliza para transformar los datos en un espacio de mayor dimensión. Las opciones comunes incluyen 'linear', 'poly', 'rbf' y 'sigmoid'.
- **degree**: Grado de la función del kernel 'poly'. (Es ignorado por todos los otros kernels)
- **gamma**: Es el coeficiente que se usa para los kernels 'rbf', 'poly' y 'sigmoid'. Determina la influencia de una única muestra de entrenamiento.
- **coef0**: Es el término independiente en la función del kernel. Solo es relevante para los kernels 'poly' y 'sigmoid'.
- **probability**: Cuando se establece en verdadero, permite la estimación de las probabilidades de las clases.

K vecinos más cercanos

- **n_neighbors**: Especifica la cantidad de vecinos más cercanos que se deben considerar al hacer predicciones.

Perceptrón multicapa

- **hidden_layer_sizes**: El número de neuronas en las capas ocultas.
- **solver**: Define el algoritmo que se utiliza para la optimización de pesos. Las opciones incluyen 'lbfgs', 'sgd' y 'adam'.
- **alpha**: Es el parámetro de penalización L2, que añade un término de regularización al objetivo de aprendizaje para prevenir el sobreajuste.

Regresión logística

- **C**: Es el inverso del parámetro de regularización lambda. Un valor más pequeño de C aumentará la regularización, evitando el sobreajuste.
- **penalty**: Define la norma que se utiliza en la penalización. Las opciones pueden ser 'l1', 'l2', 'elasticnet' o 'none', que definen la penalización aplicada a los coeficientes durante la optimización.
- **solver**: Especifica el algoritmo a utilizar para la optimización del problema. Las opciones son 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'.

Para concluir, se presentarán los valores de los hiperparámetros utilizados en cada modelo. En las tablas 4.1 y 4.2 se detallan las asignaciones de los hiperparámetros correspondientes. Estos valores de hiperparámetros fueron determinados a través de un proceso de ajuste mediante prueba y error. Se llevaron a cabo varios experimentos en los cuales se variaron sistemáticamente los valores de los hiperparámetros con el objetivo de encontrar la configuración óptima. En cada iteración, se evaluó el rendimiento del modelo utilizando métricas relevantes. Los valores de hiperparámetros que ofrecieron los mejores resultados en términos de rendimiento fueron seleccionados y utilizados para el análisis final.

Modelo	Hiperparámetros	Lista de Valores
Logistic Regression	C,	np.logspace(-3, 3, 7)
	Penalty	['l1', 'l2']
	Solver	['liblinear', 'lbfgs']
Decision Tree Classifier	max_depth	[None, 3, 4, 5, 10]
	min_samples_leaf	[1, 2, 3, 4, 5, 8, 10, 15]
Random Forest Classifier	n_estimators	[5, 10, 20, 50, 100]
	max_depth	[1, 2, 3, 4]
	min_samples_leaf	[1, 2, 5]
Support Vector Classifier	coef0	[0.0, 1.0]
	C	np.logspace(-3, 4, 10)
	Kernel	['linear', 'rbf']
	Degree	[2, 3, 5]
	Probability	[True]
K-Nearest Neighbors	n_neighbors	[1, 2, 5, 10]
Multi-layer Perceptron	hidden_layer_sizes	[(100,), (10,), (5,), (200,)]
	Solver	['lbfgs', 'adam']
	Alpha	[0.0001, 0.001, 0.01, 0.1]

Tabla 4.1 Asignación de hiperparámetros problema de clasificación

Modelo	Hiperparámetros	Lista de Valores
Linear Regression	fit_intercept'	[True, False]
Decision Tree Regressor	max_depth'	[None, 3, 5, 7, 10]
	min_samples_leaf'	[1, 2, 3, 5, 10, 15]
Random Forest Regressor	n_estimators'	[100, 200, 500]
	max_depth'	[None, 5, 10]
	min_samples_leaf'	[1, 5, 10]
Support Vector Regressor	C	np.logspace(-3, 3, 7)
	Kernel	['linear', 'rbf']
	Degree	[2, 3]
	Gamma	['scale', 'auto']
K-Nearest Neighbors	n_neighbors	[1, 2, 5, 10]
Multi-layer Perceptron	hidden_layer_sizes	[(100,), (10,), (5,), (2,)]
	Solver	['lbfgs', 'adam']
	Alpha	[0.0001, 0.001, 0.01, 0.1]

Tabla 4.2 Asignación de hiperparámetros problema de regresión

Capítulo 5. Análisis de resultados

En este capítulo, se proporcionará un desglose de las métricas utilizadas para evaluar el rendimiento de los modelos asociados a cada problema, junto con un análisis de dichas métricas.

5.1 Clasificación

Para la medición de la calidad de cada modelo se extrajeron diferentes métricas siendo estas las siguientes:

- **T (Threshold)**: Umbral de decisión que se utiliza en modelos de clasificación para decidir entre las diferentes clases.
- **Bacc (Balanced Accuracy):** Versión de la precisión que tiene en cuenta el equilibrio entre las tasas de verdaderos positivos y verdaderos negativos. Es especialmente útil en conjuntos de datos desequilibrados.
- **P (Precision):** Proporción de verdaderos positivos entre todas las predicciones positivas. Un valor alto indica que cuando el modelo predice la clase positiva, es probable que sea correcto.
- **R (Recall):** Proporción de verdaderos positivos entre todas las instancias positivas reales. Un valor alto indica que el modelo es bueno para encontrar todas las instancias positivas.
- **ESP (Specificity):** Proporción de verdaderos negativos entre todas las instancias negativas reales. Un valor alto indica que el modelo es bueno para evitar falsos positivos.
- **F1 (F1 Score)**: Promedio armónico de la precisión y la exhaustividad. Un valor alto indica que el modelo tiene un buen equilibrio entre la precisión y la exhaustividad.

Para evaluar la eficacia de los modelos de clasificación generados se extrajeron los mejores clasificadores para cada experimento basándonos en la métrica de precisión equilibrada (Bacc). Dichos resultados se pueden apreciar en la tabla 5.1.

Experimento	Clasificador	T	Bacc	P	R	ESP	F1
C_model_1	Logistic Regression	0.5	0.8	0.7778	0.9333	0.6667	0.8485
C_model_2	Decision Tree	0.75	0.7215	0.8462	0.7097	0.7333	0.7719
C_model_F_1	Decision Tree	0.5	0.8182	0.7895	1	0.6364	0.8824
C_model_F_2	Logistic Regression	0.5638	0.65	0.7742	0.8	0.5	0.7869
C_model_F_R_1	Logistic Regression	0.6058	0.7179	0.7692	0.7692	0.6667	0.7692
C_model_F_R_2	MLP	0.5	0.7343	0.8276	0.9231	0.5455	0.8727
C_model_F_R_M_1	Logistic Regression	0.5433	0.906	0.9231	0.9231	0.8889	0.9231
C_model_F_M_1	Random Forest	0.5375	0.9333	1	0.8667	1	0.9286
C_model_R_1	Random Forest	0.4338	0.75	0.7143	1	0.5	0.8333
C_model_R_2	KNN	1	0.5581	0.7273	0.5161	0.6	0.6038
C_model_R_M_1	Logistic Regression	0.5	0.8083	0.8125	0.8667	0.75	0.8387
C_model_M_1	Random Forest	0.5	0.85	0.8667	0.8667	0.8333	0.8667

Tabla 5.1 Resultados por experimentos en clasificación

En el análisis de los resultados de los modelos de clasificación, se puede observar un amplio rango de desempeño, donde cada modelo tiene su propio conjunto de métricas que resalta sus fortalezas y debilidades.

En el modelo C_model_F_M_1 (Random Forest) se destaca como el más prometedor en términos de precisión, presentando un puntaje perfecto de 1.0. Esto indica que cuando este modelo predice una clase, es altamente probable que sea correcto. Además, su valor F1, que es un promedio ponderado de precisión y exhaustividad, es igualmente impresionante, lo que sugiere que este modelo es excelente tanto en la identificación de verdaderos positivos como en la evitación de falsos positivos.

Por otro lado, el modelo C_model_R_2 (KNN) muestra un rendimiento bastante inferior en todas las métricas. Con una precisión y exhaustividad bajas, y un valor F1 mediocre, este modelo parece tener dificultades para clasificar correctamente las instancias. Es especialmente notable su baja exhaustividad, lo que indica que el modelo tiene problemas para identificar todas las instancias positivas.

Otro modelo que merece atención es C_model_F_R_M_1 (Logistic Regression). Este modelo muestra un rendimiento bastante alto en todas las métricas, destacando por su equilibrio entre precisión y exhaustividad, lo cual es esencial en muchos problemas de clasificación.

El modelo C_model_F_1 (Decision Tree), por otro lado, presenta un caso intrigante, ya que, a pesar de su precisión y especificidad moderadas, logra una exhaustividad perfecta de 1.0. Este modelo podría ser particularmente útil en escenarios donde identificar todos los casos positivos sea de suma importancia, aunque a costa de algunos falsos positivos.

Los modelos C_model_1 (Logistic Regression) y C_model_F_R_1 (Logistic Regression) presentan rendimientos medios con una precisión y exhaustividad bastante altas. Sin embargo, la especificidad de ambos modelos es relativamente baja, lo que indica una mayor tasa de falsos positivos.

En conclusión, los modelos C_model_F_M_1 (Random Forest) y C_model_F_R_M_1 (Logistic Regression) se destacan como los más prometedores, dependiendo de las necesidades específicas del problema en cuestión. Sin embargo, todos los modelos, excepto C_model_R_2 (KNN), muestran un desempeño considerable y podrían ser útiles en distintos contextos, dependiendo de qué métricas se consideren más importantes.

5.2 Regresión

Para la medición de la calidad de cada modelo se extrajeron diferentes métricas siendo estas las siguientes:

- **MSE (Mean Squared Error):** Medida de error que se calcula como el promedio de los errores cuadrados. Cuanto más pequeño es el valor, mejor es el rendimiento del modelo.
- **RMSE (Root Mean Squared Error):** Raíz cuadrada del MSE. Esta métrica tiene la ventaja de estar en las mismas unidades que la variable dependiente.
- **MAE (Mean Absolute Error):** Media de los errores absolutos. Es una métrica fácil de interpretar, ya que mide cuánto se desvía, en promedio, las predicciones del modelo de los valores reales.

- **R2 (R-squared):** Coeficiente de determinación que indica cuánta variabilidad de los datos puede ser explicada por el modelo. Típicamente los valores de R2 varían entre 0 y 1, donde los valores más cercanos a 1 indican un mejor ajuste del modelo. Sin embargo, en ciertos casos, como en la regresión lineal con ajuste insuficiente, R2 puede resultar negativo, lo que indica que el modelo es peor que un modelo que simplemente predice la media de los datos.
- **MRE (Mean Relative Error):** Error relativo promedio. No tiene un rango limitado y puede ser negativo si el pronóstico sistemáticamente sobrestima el valor actual.
- **APE (Absolute Percentage Error):** Medida absoluta del error en términos porcentuales. Es muy útil para comparar el error en diferentes escalas.

Para evaluar la eficacia de los modelos de regresión se aplicó la misma técnica de extraer los mejores modelos por experimento, en este caso en base a el error cuadrático medio (MSE). Dichos resultados se pueden apreciar en la tabla 5.2.

Experimento	Clasificador	MSE	RMSE	MAE	R2	MRE	APE
R_model_1	MLP	72.7153	8.5273	5.5764	0.224	1.2786	0.9353
R_model_2	Decision Tree	191.5354	13.8396	9.4263	0.0879	1.7735	1.7241
R_model_F_1	Random Forest	197.0298	14.0367	8.6053	0.0077	1.5949	0.9173
R_model_F_2	Linear Regression	164.3112	12.8184	7.9828	0.1014	1.3274	0.0403
R_model_F_R_1	Random Forest	116.7613	10.8056	7.878	0.057	2.1096	0.4303
R_model_F_R_2	Random Forest	263.5039	16.2328	9.8557	-0.0452	1.8985	3.8407
R_model_F_R_M_1	Random Forest	121.5908	11.0268	7.8828	0.018	2.2456	0.5117
R_model_F_M_1	Random Forest	211.1314	14.5304	8.6309	-0.0634	1.5604	1.4118
R_model_R_1	Linear Regression	76.232	8.7311	6.4611	0.1865	1.8443	-0.9805
R_model_R_2	KNN	204.9871	14.3174	9.565	0.0238	1.6641	1.7108
R_model_R_M_1	Linear Regression	83.839	9.1564	6.5864	0.1053	1.9321	-1.6881
R_model_M_1	Linear Regression	82.6938	9.0936	6.4781	0.1175	1.9076	-1.6325

Tabla 5.2 Resultados por experimento en regresión

El análisis de la tabla de resultados de los modelos de regresión arroja resultados interesantes y variados. El modelo R_model_1 (MLP) muestra el menor MSE y RMSE entre todos los modelos, lo que indica que este modelo, en promedio, está más cerca del valor verdadero en comparación con los demás. Sin embargo, su coeficiente R2 es bajo, lo que significa que sólo explica un 22.4% de la variación de los datos.

Por otro lado, el modelo R_model_2 (Decision Tree) tiene un MSE y un RMSE bastante altos, lo que indica un mayor error en las predicciones. Sin embargo, su coeficiente R2 es apenas superior al del modelo MLP, lo que sugiere que también tiene una capacidad limitada para explicar la variación de los datos.

El modelo R_model_F_1 (Random Forest) muestra el tercer MSE y RMSE más altos de todos los modelos, lo que indica que este modelo puede tener un error considerable en sus predicciones. Este hecho se refleja en su coeficiente R2 extremadamente bajo, que es el segundo más bajo de todos los modelos.

El modelo R_model_F_2 (Linear Regression) tiene un rendimiento moderado en términos de MSE y RMSE, pero su coeficiente R2 es relativamente alto en comparación con la mayoría de los

demás modelos, lo que sugiere que tiene una capacidad razonable para explicar la variación de los datos.

Los modelos R_model_F_R_1, R_model_F_R_2, R_model_F_R_M_1 y R_model_F_M_1 (todos Random Forest) muestran una amplia gama de resultados en términos de MSE y RMSE, pero todos tienen coeficientes R2 muy bajos, lo que indica que tienen una capacidad limitada para explicar la variación de los datos.

Los modelos R_model_R_1, R_model_R_2, R_model_R_M_1 y R_model_M_1 (todos Linear Regression y KNN) también muestran una amplia gama de resultados en términos de MSE y RMSE, pero todos tienen coeficientes R2 bajos a moderados, lo que sugiere una capacidad limitada para explicar la variación de los datos.

Teniendo en cuenta la distribución de la variable objetivo, expuestas en el capítulo 3, la cual tiene una media de 16.98 y una desviación estándar de 13.12, el modelo R_model_1 (MLP) parece tener el mejor desempeño en términos de error en las predicciones.

Las conclusiones que se pueden extraer en cuanto a los modelos de regresión generados, es que los modelos de regresión no han logrado obtener buenos resultados. No obstante, esto es derivado de una casi nula correlación de los datos con la variable objetivo como se expuso en el capítulo 3, denotando una baja calidad de los datos sumado a una cantidad de datos insuficiente para generar un modelo de calidad a la hora de generar un modelo que sea capaz de predecir una esperanza de vida.

Capítulo 6. Conclusiones y líneas futuras

El proyecto de desarrollo de modelos de Machine Learning para la clasificación de resecciones y pronóstico de esperanza de vida en pacientes con glioblastoma representa una solución potencial para un problema real y significativo en la oncología. Considerando este objetivo, los resultados obtenidos pueden considerarse satisfactorios en parte: se lograron generar modelos eficaces para la clasificación de resección positiva o negativa utilizando variables clínicas, lo que podría proporcionar una herramienta valiosa para los profesionales de la salud en la toma de decisiones terapéuticas, mejorando la calidad y esperanza de vida de los pacientes.

Sin embargo, la calidad de los modelos de regresión para el pronóstico de la esperanza de vida no alcanzó un nivel satisfactorio. Si se dispusiera de más y diversos datos de pacientes con glioblastoma (especialmente datos etiquetados), podrían implementarse métodos más avanzados de Machine Learning, para mejorar la precisión del pronóstico o incluso reentrenar los modelos actuales con una mayor cantidad de estos.

Además, si los datos resultantes de este estudio se procesaran de manera adecuada, podrían ser útiles para entrenar futuros modelos que puedan realizar predicciones más precisas sobre las características clínicas de los pacientes y, por lo tanto, mejorar la efectividad del tratamiento. En definitiva, este trabajo ilustra tanto el potencial, como las limitaciones de los modelos de Machine Learning en el campo de la oncología, y pone de manifiesto la necesidad de recopilar y procesar más datos para mejorar la eficacia de estas herramientas.

En futuros trabajos, será esencial mejorar la precisión del modelo para la tarea de regresión y el pronóstico de la esperanza de vida. Para lograr esto, podría ser beneficioso recopilar una mayor cantidad de datos clínicos y explorar la inclusión de otras variables que puedan tener un impacto significativo en el pronóstico de la esperanza de vida. Además, se podrían probar otros métodos de Machine Learning más avanzados, como las redes neuronales profundas, que pueden ser más efectivas en la identificación de patrones complejos en los datos. También sería interesante investigar cómo se pueden incorporar estas herramientas de Machine Learning en la práctica clínica diaria para ayudar a los médicos a tomar decisiones más informadas sobre el tratamiento. Finalmente, se podría considerar el desarrollo de una interfaz de usuario amigable que permita a los médicos y otros profesionales de la salud utilizar fácilmente estos modelos de Machine Learning, y así facilitar su adopción en la atención sanitaria de rutina.

Capítulo 7. Summary and Conclusions

The project of developing machine learning models for the classification of resections and prognosis of life expectancy in patients with glioblastoma represents a potential solution to a real and significant problem in oncology. Considering this objective, the results obtained can be partially considered satisfactory: effective models were generated for the classification of positive or negative resection using clinical variables, which could provide a valuable tool for healthcare professionals in therapeutic decision-making, improving the quality and life expectancy of patients.

However, the quality of the regression models for the prognosis of life expectancy did not reach a satisfactory level. If more and diverse data from patients with glioblastoma were available (especially labelled data), more advanced methods of Machine Learning could be implemented, to improve the accuracy of the prognosis or even retrain the current models with a greater amount of these.

In addition, if the data resulting from this study were processed appropriately, they could be useful for training future models that can make more accurate predictions about the clinical characteristics of patients and, therefore, improve the effectiveness of the treatment. In short, this work illustrates both the potential and the limitations of Machine Learning models in the field of oncology and highlights the need to collect and process more data to improve the effectiveness of these tools.

In future works, it will be essential to improve the accuracy of the model for the regression task and the prognosis of life expectancy. To achieve this, it could be beneficial to collect a larger amount of clinical data and explore the inclusion of other variables that may have a significant impact on the prognosis of life expectancy. In addition, more advanced Machine Learning methods could be tested, such as deep neural networks, which can be more effective in identifying complex patterns in the data. It would also be interesting to investigate how these Machine Learning tools can be incorporated into daily clinical practice to help doctors make more informed decisions about treatment. Finally, the development of a user-friendly interface that allows doctors and other health professionals to easily use these Machine Learning models could be considered, thus facilitating their adoption in routine healthcare.

Capítulo 8. Presupuesto del proyecto

En la tabla 8.1 se desglosan tanto las tareas realizadas, como el tiempo invertido en cada una de ellas. Todo esto acompañado del precio correspondiente y el cómputo total de costes.

El salario medio de un Ingeniero Informático ronda los 36.500€ brutos al año, o unos 3.000€ brutos al mes[1]. Eso sumado a que un mes cuenta con 20 días laborables y suponiendo una jornada laboral de 8 horas, el coste por hora trabajada ronda los 19€ euros. El número de horas de trabajo que establece la guía docente de esta asignatura es de 300 horas en total.

Por todo lo anteriormente mencionado, aplicando las horas de trabajo dedicadas a cada fase del proyecto y sumando los recursos necesarios para ejecutar el proyecto, se presenta el presupuesto del proyecto, con un coste total de 5.700€.

Actividades	Cantidad (h)	Precio (€)
Análisis y preprocesado de los datos	100	1.900
Selección de modelos de Machine Learning	50	950
Entrenamiento y evaluación de los modelos	50	950
Ajuste y optimización de hiperparámetros	40	760
Análisis e interpretación de resultados	60	1.140
Total	300	5.700

Tabla 8.1 Presupuesto de las actividades

[1] Salario de un Ingeniero Informático en España: https://www.jobted.es/salario/ingeniero-informatico

Capítulo 9. Bibliografía

[1] Wirsching, H., & Weller, M. (2017). Glioblastoma. Malignant Brain Tumors: State-of-the-Art Treatment, , 265-288.

[2] Sackett, D. L. (1997). Evidence-based medicine. Paper presented at the Seminars in Perinatology, , 21(1) 3-5.

[3] Kosorok, M. R., & Laber, E. B. (2019). Precision medicine. Annual Review of Statistics and its Application, 6, 263-286.

[4] Osareh, A., & Shadgar, B. (2010, April). Machine learning techniques to diagnose breast cancer. In 2010 5th international symposium on health informatics and bioinformatics (pp. 114-120). IEEE.

[5] Ibrahimi, M., Abdollahi, H., Rottondi, C., Giusti, A., Ferrari, A., Curri, V., & Tornatore, M. (2021). Machine learning regression for QoT estimation of unestablished lightpaths. Journal of Optical Communications and Networking, 13(4), B92-B101.

[6] Laukamp, K. R., Thiele, F., Shakirin, G., Zopfs, D., Faymonville, A., Timmer, M., ... & Borggrefe, J. (2019). Fully automated detection and segmentation of meningiomas using deep learning on routine multiparametric MRI. European radiology, 29, 124-132.

[7] Ting, F. F., Tan, Y. J., & Sim, K. S. (2019). Convolutional neural network improvement for breast cancer classification. Expert Systems with Applications, 120, 103-115.

[8] Nguyen, D., Nguyen, H., Ong, H., Le, H., Ha, H., Duc, N. T., & Ngo, H. T. (2022). Ensemble learning using traditional machine learning and deep neural network for diagnosis of Alzheimer's disease. IBRO Neuroscience Reports, 13, 255-263.

[9] Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nature Medicine, 25, 44-56.

[10] Universidad Politécnica de Valencia. (n.d.). Oncohabitats. https://www.oncohabitats.upv.es/