

TRABAJO FIN DE GRADO



UCAM

UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

Desarrollo e interpretación de modelos Machine
Learning y Deep Learning para predicción de cáncer de
mama

Autora:

María Isabel Cornejo Rojas

Director:

Dr. Antonio Jesús Banegas Luna

Murcia, mayo de 2024

TRABAJO FIN DE GRADO



UCAM

UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

Desarrollo e interpretación de modelos Machine
Learning y Deep Learning para predicción de cáncer de
mama

Autora:

María Isabel Cornejo Rojas

Director:

Dr. Antonio Jesús Banegas Luna

Murcia, mayo de 2024

<https://colab.research.google.com/drive/1aqWwAyFFQyWthocZQnrV27VZdkWJchLB?usp=sharing>

Agradecimientos

A mis amigos de toda la vida, Joaquín y Andrea, que a pesar de la distancia estuvieron conmigo cada vez que los necesité.

A Antonio, por todos sus consejos y ayuda, no solo en este proyecto, sino en la vida, gracias a él descubrí lo que quiero hacer en mi vida.

A mi pareja, por creer en mí, apoyarme, ayudarme y comprenderme durante este proyecto.

A mi mami, por su paciencia, amor y tiempo dedicado, por alentarme desde la distancia, aunque siempre está en mi corazón.

Finalmente, a mi padre, la persona que más admiro en el mundo, quien ha sacrificado mucho por mí y sigue haciéndolo cada día de su vida.

Gracias bubu, te amo.

Listado de Abreviaturas

ANN, Artificial Neural Network

DL, Deep Learning

EDA, Exploratory Data Analysis (Análisis de datos exploratorio)

GAN, Generative Adversarial Network (Red Generativa Antagónica)

GB, Gradient Boosting

IA, Inteligencia Artificial

ML, Machine Learning

MLP, Multilayer Perceptron

PB, Product Backlog

PO, Product Owner

ÍNDICE

Contenido

1. INTRODUCCIÓN	19
1.1. Motivación	20
1.2. Definición	22
1.3. Objetivos propuestos	22
1.3.1. Objetivos generales	22
1.3.2. Objetivos específicos	22
2. ESTUDIO DEL MERCADO (O ESTADO DEL ARTE)	23
2.1. Conceptos relevantes del dominio de aplicación.	23
2.2. Relación con proyectos con la misma funcionalidad	24
2.2.1. Breast cancer prediction with transcriptome profiling using feature selection and machine learning methods.	24
2.2.2. Deep learning-based feature-level integration of multi-omics data for breast cancer patients survival analysis	25
2.2.3. Explainable AI for Breast Cancer Detection: A LIME-Driven Approach	26
2.2.4. Aspectos de Interpretabilidad	27
2.2.5. Comparación	29
2.3. Estudio de viabilidad	30
2.3.1. Alcance del proyecto	30
2.3.2. Estudio de la situación actual	31
2.3.3. Estudio y valoración de las alternativas de solución	31
2.3.4. Selección de la solución	34
3. METODOLOGÍAS USADAS	39
3.1. Estudio de metodologías ágiles	40
3.1.1. Scrum	40
3.1.2. Kanban	41

3.1.3.	Extreme Programming (XP)	41
3.1.4.	Adaptive Software Development (ASD)	41
3.2.	Elección de metodología	42
3.3.	Uso de Scrum en el proyecto	43
3.3.1.	Roles en el equipo Scrum	44
4.	TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO..	47
4.1.	Tecnologías	47
4.1.1.	Análisis de datos y modelado	47
4.1.2.	Frameworks de aprendizaje automático	47
4.2.	Herramientas	48
4.2.1.	Entorno de desarrollo	48
4.2.2.	Gestión de versiones y colaboración	48
4.2.3.	Visualización de datos	48
4.2.4.	Herramientas de documentación y presentación	48
5.	ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN	49
5.1.	Organización de Sprints	49
5.3.	Estimaciones utilizadas	51
5.3.1.	Estimación de prioridad	51
5.3.2.	Estimación de esfuerzo	51
5.4.	Tabla definitiva del Product Backlog con las estimaciones de prioridad y esfuerzo	51
5.5.	Realizar una planificación temporal del proyecto	61
5.5.1.	Estimación de tiempo	61
5.5.2.	Estimación de escenarios	63
5.5.3.	Planificación temporal	64
5.6.	Realizar una valoración de la dedicación y el coste económico	67
6.	DESARROLLO DEL CONTENIDO DEL PROYECTO	69

6.1.	Diseño y organización de código	69
6.2.	Especificación Sprints	71
6.2.1.	Sprint 0	71
6.2.2.	Sprint 1	74
6.2.3.	Sprint 2	79
6.2.4.	Sprint 3	102
6.2.5.	Sprint 4	112
6.2.4.	Sprint 5	122
7.	DESPLIEGUE Y PRUEBA DE LA SOLUCIÓN	131
7.1.	Plan de pruebas	131
7.2.	Indicadores del desempeño y/o utilidad.	144
7.2.1.	Metodología de Medición	146
7.2.2.	Indicadores Clave de Rendimiento (KPIs)	147
7.2.3.	Análisis de Tendencias	148
7.3.	Escalabilidad / extensión / mantenimiento / Soporte.....	149
7.3.1.	Escalabilidad y extensión.....	149
7.3.2.	Mantenimiento y soporte.....	149
7.4.	Plan de formación de usuarios.....	150
7.4.1.	Despliegue notebook	150
7.4.2.	Interpretación de imágenes	150
8.	CONCLUSIONES	155
8.1.	Objetivos alcanzados	144
8.2.	Conclusiones del trabajo y personales.....	155
8.3.	Vías futuras.....	156
9.	BIBLIOGRAFÍA	157

ÍNDICE DE ELEMENTOS GRÁFICOS

TABLAS

Tabla 1. Ventajas de Scrum y ASD. Elaboración propia.	42
Tabla 2. Desventajas de Scrum y ASD. Elaboración propia	43
Tabla 3. Roles definidos en el proyecto. Elaboración propia.....	44
Tabla 4. Product Backlog. Elaboración propia.....	50
Tabla 5. Requisitos técnicos específicos. Elaboración propia.	52
Tabla 6. Tabla de velocidad y tiempo estimado. Elaboración propia.....	64
Tabla 7. Estimación económica del proyecto. Elaboración propia.	67
Tabla 8. Informe Sprint 0. Elaboración propia.	71
Tabla 9. Conjunto de datos considerados. Elaboración propia	73
Tabla 10. Informe Sprint 1. Elaboración propia.	74
Tabla 11. Informe Sprint 2. Elaboración propia.	79
Tabla 12. Tabla comparación de la evaluación de los datos de Regresión Logística. Elaboración propia.	97
Tabla 13. Tabla comparación de la evaluación de los datos de Gradient Boosting. Elaboración propia.	98
Tabla 14. Tabla comparación de la evaluación de los datos de Random Forest. Elaboración propia.	99
Tabla 15. Comparación de evaluación modelos ML. Elaboración propia.	100
Tabla 16. Informe Sprint 3. Elaboración propia.	102
Tabla 17. Comparación de evaluación de diferentes configuraciones ANN. Elaboración propia.	104
Tabla 18. Comparación de evaluación de diferentes configuraciones MLP. Elaboración propia.	107
Tabla 19. Comparación de evaluación modelos DL. Elaboración propia.	109
Tabla 20. Comparación mejores modelos ML y DL. Elaboración propia.	110
Tabla 21. Informe Sprint 4. Elaboración propia.	112
Tabla 22. Tabla de comparación de los mejores modelos. Elaboración propia.	113
Tabla 23. Tabla de Consenso Global. Elaboración propia.	127
Tabla 24. Requisitos técnicos específicos. Elaboración propia.	132
Tabla 25. Caso de Prueba [US_1].....	135

Tabla 26. Caso de Prueba [US_2.1].....	135
Tabla 27. Caso de Prueba [US_2.2].....	135
Tabla 28. Caso de Prueba [US_2.3].....	136
Tabla 29. Caso de Prueba [US_2.4].....	136
Tabla 30. Caso de Prueba [US_3].....	136
Tabla 31. Caso de Prueba [US_4.1].....	137
Tabla 32. Caso de Prueba [US_4.2].....	137
Tabla 33. Caso de Prueba [US_5].....	137
Tabla 34. Caso de Prueba [US_6.1].....	138
Tabla 35. Caso de Prueba [US_6.2].....	138
Tabla 36. Caso de Prueba [US_6.3].....	138
Tabla 37. Caso de Prueba [US_7.1].....	139
Tabla 38. Caso de Prueba [US_7.2].....	139
Tabla 39. Caso de Prueba [US_7.3].....	139
Tabla 40. Caso de Prueba [US_7.4].....	140
Tabla 41. Caso de Prueba [US_8.1].....	140
Tabla 42. Caso de Prueba [US_8.2].....	140
Tabla 43. Caso de Prueba [US_8.3].....	141
Tabla 44. Caso de Prueba [US_8.4].....	141
Tabla 45. Caso de Prueba [US_9].....	141
Tabla 46. Caso de Prueba [US_10.1].....	142
Tabla 47. Caso de Prueba [US_10.2].....	142
Tabla 48. Caso de Prueba [US_11].....	142
Tabla 49. Tabla de Indicadores clave de Rendimiento.....	146

ILUSTRACIONES

Ilustración 1. Prevalencia de cáncer de mama a nivel mundial. Elaboración propia.	21
Ilustración 2. Metodología Tradicional "Cascada". Elaboración propia	39
Ilustración 3. Metodología ágil. Elaboración propia.	40
Ilustración 4. Línea de tiempo de cada Sprint. Elaboración propia.....	49
Ilustración 5. Leyenda planificación temporal. Elaboración propia.	64
Ilustración 6. Planificación temporal del Sprint 0. Elaboración propia.	65

Ilustración 7. Planificación temporal Sprint 1. Elaboración propia.	65
Ilustración 8. Planificación temporal Sprint 2. Elaboración propia.	65
Ilustración 9. Planificación temporal Sprint 3. Elaboración propia.	66
Ilustración 10. Planificación temporal Sprint 4. Elaboración propia.	66
Ilustración 11. Planificación temporal Sprint 5. Elaboración propia.	66
Ilustración 12. Esquema de organización código. Elaboración propia.	69
Ilustración 13. Captura de una parte del dataset sintético usando Label Encoder como codificador.	76
Ilustración 14. Captura de una parte del dataset sintético usando One-Hot Encoder como codificador.	76
Ilustración 15. Captura de dataset sintético con el uso de ambos codificadores	77
Ilustración 16. Histograma de Edad. Código Google Colab.	82
Ilustración 17. Pairplot de Atributos Numéricos. Código Google Colab.	82
Ilustración 18. Gráfico de Dispersión. Código Google Colab.	84
Ilustración 19. Frecuencia de Breast Quadrant. Código Google Colab.	84
Ilustración 20. Distribución de Breast Quadrant por diagnóstico. Código Google Colab.	85
Ilustración 21. Gráfico de dispersión Código Google Colab.	86
Ilustración 22. Detalle dataset sintético. Código Google Colab.	87
Ilustración 23. Matriz de Correlación. Código Google Colab.	88
Ilustración 24. Gráficos de barras de variables. Código Google Colab.	90
Ilustración 25. Variación de tamaños del tumor en relación con el diagnóstico. Código Google Colab.	91
Ilustración 26. Distribución de la edad entre dataset original y sintético. Código Google Colab.	92
Ilustración 27. Dispersión de la edad entre dataset original y sintético. Código Google Colab.	93
Ilustración 28. Dispersión de Tumor Size (cm). Código Google Colab.	94
Ilustración 29. División de validación, entrenamiento y prueba. Código Google Colab.	95
Ilustración 30. Matriz de confusión Gradient Boosting. Código Google Colab.	113
Ilustración 31. Matriz de confusión MLP. Código Google Colab.	114

Ilustración 32. Captura de caso 1 de interpretabilidad LIME, Gradient Boosting. Código Google Colab	116
Ilustración 33. Captura de caso 2 de interpretabilidad LIME, Gradient Boosting. Código Google Colab.	116
Ilustración 34. Captura de caso 1 de interpretabilidad LIME, MLP. Código Google Colab.....	117
Ilustración 35. Captura de caso 2 de interpretabilidad LIME, MLP. Código Google Colab	118
Ilustración 36. Captura de interpretación SHAP, Gradient Boosting. Código Google Colab.	119
Ilustración 37. Captura de interpretación SHAP, MLP. Código Google Colab.	120
Ilustración 38. Tabla de importancias Gradient Boosting. Código Google Colab.	123
Ilustración 39. Importancia LIME, GB.....	123
Ilustración 40. Importancia SHAP, GB.	124
Ilustración 41. Consenso de importancias GB.....	124
Ilustración 42. Tabla de importancias MLP. Código Google Colab.	125
Ilustración 43. Importancia LIME, MLP.....	125
Ilustración 44. Importancia SHAP, MLP.	126
Ilustración 45. Consenso de importancias MLP.	126
Ilustración 46. Gráfica de consenso global. Código Google Colab.....	128
Ilustración 47. Comparación de horas planificadas vs. horas reales por Sprint. Elaboración propia	147
Ilustración 48. Satisfacción del Sprint (%). Elaboración propia	147
Ilustración 49. Tareas pendientes por Sprint. Elaboración propia	148
Ilustración 50. Captura página web del proyecto Bruselas.....	149
Ilustración 51. Pairplot atributos numéricos explicación. Código Google Colab.	151
Ilustración 52. Explicación matriz de correlación. Código Google Colab.....	152

RESUMEN

El presente Trabajo de Fin de Grado se enfoca en el desarrollo de modelos de Machine Learning (ML) y Deep Learning (DL) para predecir el cáncer de mama, con el propósito de mejorar los métodos de diagnóstico y detección de esta enfermedad que se aplican hoy en día.

El objetivo principal del trabajo es crear modelos precisos y fiables que permitan la detección de esta enfermedad en una etapa temprana o antes de que se presente. Para lograrlo, se ha llevado a cabo un proceso que engloba la recolección de datos clínicos, la creación y el entrenamiento de los modelos, y su posterior interpretación, lo que permitirá obtener los factores más determinantes en la predicción. Además, como una manera de fundamentar los resultados obtenidos y dar un enfoque diferente e interesante al trabajo, se ha utilizado la fusión de datos, lo que permite obtener una explicación global de los modelos utilizados.

Los datos utilizados en este trabajo fueron obtenidos de The Cancer Imaging Archive. Concretamente, el repositorio elegido es UCTH Breast Cancer Dataset, que contiene 11 atributos, incluyendo el tipo de tumor de 213 pacientes.

Con el propósito de garantizar que el proyecto siguiera un progreso organizado, se implementó la metodología ágil Scrum, la cual ha proporcionado la flexibilidad necesaria ante cambios repentinos durante el desarrollo. Es una metodología basada en iteraciones, por lo que simplifica el ajuste del proyecto.

En síntesis, este proyecto se centró en la creación de diferentes modelos de Inteligencia Artificial (IA) y, con la eficaz ayuda de Scrum, la optimización del tiempo y una planificación organizada fueron fundamentales para alcanzar los objetivos planteados.

Palabras claves: Inteligencia Artificial (IA), Machine Learning (ML), Deep Learning (DL), Cáncer de mama, Interpretabilidad.

ABSTRACT

This final degree project has been focused on the development of Machine Learning (ML) and Deep Learning (DL) models to predict breast cancer, with the purpose of improving the methods of diagnosis and detection of this disease that are applied today.

The purpose of the project is to create precise and reliable models that allow the detection of this disease at an early stage or before it occurs. To achieve this, a process has been carried out that includes the collection of clinical data, the creation and training of the models and their subsequent analysis. Likewise, as a way to substantiate the results obtained and give the work a different and interesting approach, consensus functions have been used that could allow us to generate a global explanation about the models.

In order to ensure that the project followed an organized course, the agile Scrum methodology was implemented, which has provided the necessary flexibility in the face of sudden changes during development. The methodology is based on iterations, which simplifies the adjustment of the project.

In summary, this project focused on the creation of different Artificial Intelligence (AI) models and with the effective help of Scrum, optimizing time and carrying out organized planning was essential to achieve the objectives we set.

Keywords: Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Breast cancer, Interpretability

1. INTRODUCCIÓN

El cáncer de mama es uno de los tipos de cáncer que mayor tasa de morbilidad y mortalidad tiene (Instituto Nacional del Cáncer, 2020), y aunque estos números han ido bajando en las últimas dos décadas, es preocupante la falta de soluciones efectivas que se faciliten, siendo medicamentos o métodos preventivos.

Dado que el origen exacto aún no es del conocimiento, es importante explorar diferentes caminos para mejorar la detección y aumentar la tasa de supervivencia. En este contexto, el incremento de recursos de investigación y la constante evolución de la Inteligencia Artificial (IA) ofrecen una gran cantidad de oportunidades para contribuir de manera beneficiosa a la problemática expuesta.

En el 2020, al menos el 30% de las personas a quienes se les diagnosticó cáncer de mama no lograron sobrevivir (Organización Mundial de la Salud, 2022). Este dato resalta la necesidad de encontrar soluciones efectivas puesto que, a diferencia de otros tipos de cáncer con causas conocidas y definidas, el cáncer de mama carece de ello. Esto se debe a que el origen y los síntomas varían según la persona y su historial de vida.

Además, la prometedora integración de la IA en los pronósticos de enfermedades tiene el potencial de cambiar completamente la manera de pronóstico y manejo de estas. Por ello, este proyecto pretende aprovechar todas las herramientas para abordar la identificación de patrones, señales tempranas o indicios comunes de cáncer de mama mediante modelos de Machine Learning (ML) y Deep Learning (DL).

1.1. Motivación

Se encuentra como motivación para realizar este trabajo las impactantes cifras relacionadas con la morbilidad y mortalidad del cáncer de mama. Decidimos tomar acción frente a esta problemática, conscientes de la necesidad de mejorar los métodos de detección y tratamiento de esta enfermedad.

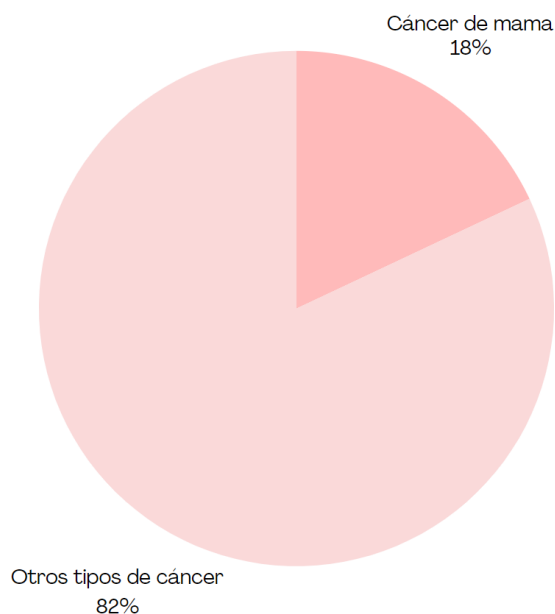
Según las investigaciones, en Estados Unidos, el 91,2% de las personas diagnosticadas con cáncer de mama sobreviven al menos cinco años después del diagnóstico (National Cancer Institute, 2022). Este dato enfatiza la importancia de la detección temprana y el tratamiento adecuado de esta enfermedad. Sin embargo, cabe resaltar que es una estimación; como se mencionó en la introducción, cada paciente es diferente y por ello la supervivencia es relativa.

Enfocados en España, se estima que, en 2024, 36.395 mujeres serán diagnosticadas con cáncer de mama, según la Sociedad Española de Oncología Médica (Sociedad Española de Oncología Médica “SEOM”, 2024). Esta cifra representa la incidencia, es decir, el número de nuevos casos esperados para ese año.

Por otro lado, la prevalencia del cáncer de mama en 2020 fue de 144,233 pacientes, reflejando el total de mujeres vivas en ese momento que habían sido diagnosticadas con la enfermedad, independientemente de cuándo recibieron el diagnóstico. La diferencia significativa entre la incidencia anual y la prevalencia indica una baja mortalidad para este tipo de cáncer, lo que significa que muchas mujeres diagnosticadas con cáncer de mama continúan viviendo varios años después del diagnóstico.

A nivel mundial, el cáncer de mama es uno de los tipos más prevalentes, con una estimación de 7.924.568 casos a cinco años, representando aproximadamente el 18% del total de casos prevalentes de cáncer.

Prevalencia del Cáncer de Mama a nivel mundial en comparación de otros tipos de cáncer



Fuente: Sociedad Española de Oncología Médica, 2024

Ilustración 1. Prevalencia de cáncer de mama a nivel mundial. Elaboración propia.

1.2. Definición

Este Trabajo de Fin de Grado se enfoca en utilizar técnicas y herramientas de ML y DL para detectar el cáncer de mama de manera anticipada, precisa y fiable. Como se ha mencionado anteriormente, el objetivo principal es desarrollar modelos de IA que puedan identificar patrones y señales tempranas utilizando datos clínicos.

Por ello, se llevará a cabo una búsqueda de datos para utilizar en la creación y entrenamiento de los modelos. Posteriormente, se emplearán algoritmos de interpretabilidad para comprender e interpretar los resultados obtenidos. Por último, el uso de funciones de consenso nos permitirá obtener una explicación global de los modelos.

1.3. Objetivos propuestos

En este apartado se presentan los objetivos generales y específicos del proyecto. Además, se discute la relación entre estos objetivos:

1.3.1. *Objetivos generales*

El objetivo principal del proyecto es realizar modelos de ML y DL, que puedan predecir el cáncer de mama en una etapa temprana, haciendo uso de diferentes conjuntos de datos para la predicción.

1.3.2. *Objetivos específicos*

[OG.1] Desarrollo de modelos de ML y DL con el fin de observar y analizar las predicciones.

[OG.2] Implementación de algoritmos de interpretabilidad en los modelos de ML para elaborar explicaciones fundamentadas y comprensibles sobre las predicciones. Asimismo, poder identificar los marcadores importantes que cada algoritmos identifica.

[OG.3] Establecimiento de un sistema de consenso con el propósito de que se alcance una explicación global de los modelos de IA utilizados.

2. ESTUDIO DEL MERCADO (O ESTADO DEL ARTE)

2.1. Conceptos relevantes del dominio de aplicación.

Para comprender mejor el contexto de nuestro proyecto, hay varios términos importantes que se deben explicar para conocer mejor el conjunto de datos con el que estamos trabajando.

- Cuadrante de la Mama: describe la ubicación del tumor dentro del seno de la paciente. Pues el saber dónde se localiza específicamente el tumor que puede ser: cuadrante superior interno, cuadrante superior externo, cuadrante inferior interno y cuadrante inferior externo. En el enfoque de predicción e interpretabilidad, es interesante conocer en que cuadrante aparece el tumor, ya que puede ocurrir que exista una tendencia a que cuando hay un tumor en cierto cuadrante es más probable que sea maligno que benigno.
- Tamaño del tumor: indica el tamaño dimensional del tumor, esto proporciona información sobre la extensión del cáncer de mama e influye en la clasificación del estadio tumoral. Suele ser recurrente que los tumores más grandes tienden a presentar un peor pronóstico.
- Metástasis: se refiere a la propagación del cáncer desde el sitio original a otras partes del cuerpo, es crítico para la evaluación del progreso del cáncer.
- Nodos invadidos: se refiere a los ganglios linfáticos axilares que están afectados por la presencia de células cancerosas, es un factor importante, ya que, se indica que cuanto más afectados estén estos hay una mayor probabilidad de diseminación tumoral.
 - Vasos linfáticos: son vasos sanguíneos que recorren todo el cuerpo, transportando líquido linfático y células. (Komen.org, 2016)
 - Ganglios linfáticos axilares: son pequeñas acumulaciones de células inmunes que actúan como filtros para el sistema linfático, son el primer sitio a donde el cáncer de seno tiene mayor probabilidad de extenderse. (Komen.org, 2016)

- Diseminación tumoral: se refiere a la propagación del cáncer desde su sitio original a otras partes del cuerpo, puede ocurrir a través del sistema linfático o del torrente sanguíneo.
- Transcriptoma: Colección de todas las lecturas de genes presentes en una célula (ByGENS, extraído 2024).
- Multiómicos: Integración de datos genómicos, transcriptómicos, proteómicos y otros tipos de datos biológicos (ByGENS, extraído 2024).
- Genómica: Estudio del genoma completo de un organismo, que incluye todos los genes, así como sus interacciones y variaciones genéticas (ByGENS, extraído 2024).
- Transcriptómica: Estudio de los ARN mensajeros producidos en una célula o tejido en un momento dado (ByGENS, extraído 2024).
- Proteómica: Estudio de todas las proteínas producidas en una célula o tejido en un momento dado (ByGENS, extraído 2024).

Se han explicado todos estos conceptos, ya que, la gran mayoría los encontramos como atributos de nuestro dataset. Es importante conocerlos e identificarlos, ya que cada uno es relevante en el entrenamiento y predicción de nuestros modelos realizados, al igual que para la interpretación de resultados.

2.2.Relación con proyectos con la misma funcionalidad

En este apartado se comentará sobre algunos proyectos relacionados al que se quiere implementar, destacando sus características y contribuciones de cada uno. Hay muchísima variedad de este tipo de proyectos, sin embargo, se han seleccionado tres a comentar.

2.2.1. Breast cancer prediction with transcriptome profiling using feature selection and machine learning methods.

El proyecto “Breast cancer prediction with transcriptome profiling using feature selection and machine learning methods” está enfocado en la predicción del cáncer utilizando análisis de perfiles de transcriptoma implementando técnicas de aprendizaje automático. Su objetivo principal es identificar patrones genéticos que puedan servir como biomarcadores para la detección temprana de la

enfermedad. Para ello se ha empleado modelos de regresión logística, máquinas de vectores de soporte (Support Vector Machine (SVM)), entre otros modelos de ML y redes neuronales: Multilayer Perceptron (MLP) y Deep Reinforcement Learning (DRL).

Los hallazgos de este proyecto proporcionan información importante sobre qué genes o transcripciones son más relevantes, contribuyendo al campo de la oncología molecular y la medicina personalizada.

Para la evaluación de los modelos implementados se hizo uso de métricas como precisión balanceada y área bajo la curva (AUC).

- Modelos: El proyecto implementa una variedad de modelos, incluyendo DL, ML con Logistic Regression for Feature Selection (LGR) y el clasificador Multilayer Perceptron (MLP), así como Redes Neuronales Artificiales como MLP y Decision Tree (DRL).
- Métricas utilizadas: Las métricas empleadas incluyen Precisión Balanceada y Área Bajo la Curva (AUC).
- Resultados Clave: Destaca la eficiencia en la predicción, logrando una precisión balanceada de 0.86 y un AUC de 0.94. Se resalta especialmente la combinación de LGR y MLP, que demostró ser altamente efectiva en este estudio.

2.2.2. Deep learning-based feature-level integration of multi-omics data for breast cancer patients survival analysis

En este proyecto “Deep learning-based feature-level integration of multi-omics data for breast cancer patients survival analysis” se ha investigado la integración de datos multiómicos para analizar la supervivencia de pacientes con cáncer de mamá, con el uso de una gran variedad de datos biológicos (genómica, transcriptómica y proteómica) para poder comprender los factores que influyen en la progresión de la enfermedad.

Asimismo, se aplicaron técnicas de aprendizaje DL, específicamente, redes neuronales profundas (DNN) para integrar y analizar estos datos. Por otro lado, para la evaluación de los modelos se utilizó principalmente el AUC, precisión global, entre otros.

Aunque no se hizo uso explícito de algoritmos de interpretabilidad, la capacidad de identificación de biomarcadores pronósticos puede considerarse como un elemento de interpretabilidad.

- Modelos: Utilizó DNN para integrar y analizar datos, junto con el uso de MLP.
- Métricas: Se enfocó principalmente en el AUC.
- Resultados: Destaca una mejora en la predicción de supervivencia del cáncer de mama con un AUC de 0.641 al integrar datos multiómicos, especialmente la metilación del ADN y la expresión de ARNm.

2.2.3. Explainable AI for Breast Cancer Detection: A LIME-Driven Approach

El proyecto "Explainable AI for Breast Cancer Detection: A LIME-Driven Approach" se centra en la detección de cáncer de mama utilizando métodos de aprendizaje automático explicables. Emplea la técnica de Local Interpretable Model-Agnostic Explanations (LIME) para entender y visualizar cómo los modelos clasifican los casos de cáncer de mama como benignos o malignos. El estudio demuestra la eficacia de LIME al proporcionar gráficos interpretativos que desglosan las contribuciones de diferentes características a las predicciones del modelo.

- Modelos: Se utilizan modelos de aprendizaje automático como redes neuronales convolucionales (CNNs) y algoritmos de ML.
- Métricas utilizadas: Se evalúan los modelos utilizando precisión, sensibilidad y especificidad.
- Resultados clave: LIME ayudó a identificar las características más influyentes en las predicciones, mejorando la confianza en los modelos ML utilizados para el diagnóstico del cáncer de mama.

2.2.4. Aspectos de Interpretabilidad

2.2.4.1. *Breast cancer prediction with transcriptome profiling using feature selection and machine learning methods.*

Este proyecto explora cómo la interpretabilidad de los modelos mejora a través de diversas métricas y técnicas.

- **Uso de Métricas de Evaluación Transparentes:**

Balanced Accuracy y AUC: Son fundamentales para interpretar la capacidad del modelo de distinguir entre casos cáncer y no cáncer. La precisión y el AUC son indicadores clave para evaluar el rendimiento de los modelos ofreciendo fiabilidad y calidad en las predicciones.

- **Técnicas Avanzadas de Selección de Características:**

La selección y extracción de características efectivas son importantes para mejorar la interpretación de modelos. En el proyecto han incluido algunas como ANOVA, Mutual Information y clasificadores como Extra Trees y Regresión Logística.

El enfoque que se le da mejora la interpretabilidad al identificar las características más significativas y al mismo tiempo, optimizar el rendimiento del modelo ya que se concentra en biomarcadores más significativos para el diagnóstico.

2.2.4.2. *Deep learning-based feature-level integration of multi-omics data for breast cancer patients survival analysis*

En sí, el proyecto no hace referencia a los modelos de interpretación empleados, se menciona la necesidad de interpretar el modelo y la validación experimental, pero no se detalla el uso de técnicas específicas de interpretación.

- **Interpretación y Mejora de Modelos:**

Los modelos como ConcatAE¹ y CrossAE² son herramientas que ayudan a combinar datos de diferentes fuentes para predecir cómo evolucionará el cáncer en un paciente. No se entienden profundamente, en el proyecto se sugiere estudiarlos más a fondo para entender el funcionamiento y asegurar las predicciones hechas.

- **Validación de Modelos:**

Para confiar plenamente en estos modelos, necesitamos probarlos con datos muy detallados y completos. Actualmente, faltan estos datos completos, lo que hace difícil probar completamente la efectividad de los modelos.

En la documentación del proyecto se señala que la mejora de la colección y calidad de los datos genómicos ayudaría a que las predicciones del modelo sean más confiables.

2.2.4.3. Explainable AI for Breast Cancer Detection: A LIME-Driven Approach

Este proyecto explora cómo la interpretabilidad de los modelos se mejora mediante el uso de técnicas avanzadas como LIME y SHAP.

- **LIME:** Genera explicaciones localmente precisas y comprensibles para predicciones individuales, simulando el límite de decisión del modelo como lineal cerca de la instancia explicada. Ayuda a visualizar la contribución de diferentes características a las predicciones del modelo, mejorando la confianza en sus resultados.
- **SHAP** (SHapley Additive exPlanations): SHAP proporciona un marco coherente para la interpretación de predicciones del modelo, atribuyendo significancia a cada característica en el contexto de una predicción específica. Esto permite entender cómo cada característica influye en el resultado del modelo.

¹ Modelo que integra información complementaria de múltiples modalidades de datos (Deep learning based feature-level integration of multi-omics data for Breast cancer patients survival analysis, 2020)

² Modelo que está diseñado para integrar la información de consenso entre diferentes modalidades de datos (Deep learning based feature-level integration of multi-omics data for Breast cancer patients survival analysis, 2020)

- **Uso de Métricas de Evaluación Transparentes:** La precisión, sensibilidad y especificidad son fundamentales para interpretar la capacidad del modelo de distinguir entre casos de cáncer y no cáncer. Estas métricas ofrecen fiabilidad y calidad en las predicciones.

2.2.5. Comparación

Se ha explicado y detallado sobre el funcionamiento de dos proyectos muy interesantes que ofrecen una gran contribución al campo del análisis de cáncer de mama y su detección.

En comparación con estos proyectos, el presente Trabajo de Fin de Grado, se distingue por su enfoque en identificar la presencia de la enfermedad de manera anticipada, combinando técnicas de ML y DL para crear, entrenar e interpretar modelos que sean precisos y fiables. La implementación de la interpretabilidad y la fusión de datos sobre los resultados permite una comprensión más completa de los factores determinantes.

Se considera que el enfoque dado a este proyecto realiza una contribución interesante ya que se apunta hacia una detección más temprana y precisa del cáncer de mama.

2.3. Estudio de viabilidad

Este análisis de viabilidad tiene como objetivo evaluar la factibilidad del proyecto de desarrollo de modelos de ML y DL para la detección anticipada del cáncer de mama. La decisión de avanzar con el proyecto se basa en una evaluación completa que incluye un estudio del mercado (estado del arte), la situación actual del sistema que se desea mejorar (en este caso, los métodos de diagnóstico del cáncer de mama), y los resultados de un estudio técnico que explora las alternativas y tecnologías disponibles.

2.3.1. Alcance del proyecto

Este proyecto tiene como objetivo principal desarrollar modelos de IA que puedan identificar patrones y señales tempranas de cáncer de mama utilizando datos clínicos. Los modelos que se implementen buscarán mejorar la precisión y la fiabilidad en la detección de esta enfermedad, con el fin de dictar diagnósticos más tempranos y efectivos.

El proyecto incluye el uso de un dataset adecuado, en este caso, el UCTH Breast Cancer Dataset, que provee información interesante para el entrenamiento y prueba de los modelos. Además, implica el desarrollo y la evaluación de modelos de ML y DL, los cuales serán entrenados, evaluados y optimizados con el fin de identificar indicadores relevantes de cáncer de mama. Asimismo, aborda la interpretación de estos modelos, mediante la aplicación de técnicas de interpretabilidad como LIME, con el propósito de ofrecer explicaciones comprensibles y bien fundamentadas sobre las predicciones realizadas por los modelos.

Por otro lado, el proyecto no incluye elementos como, la optimización en GPU, ya que no se enfocará en mejorar los modelos en plataformas de GPU, aunque esto podría potenciar el rendimiento, no es esencial para los objetivos de validación del modelo. Además, no se contempla una implementación clínica inmediata, ya que el proyecto se concentra en la fase de investigación y desarrollo; la implementación directa en entornos clínicos se abordará una vez se haya completado la validación necesaria y se haya garantizado la conformidad con los estándares médicos. Por último, no se incluirá la

actualización de datos en tiempo real, dado que el proyecto se basará en un conjunto de datos estático para el desarrollo y validación de los modelos.

2.3.2. Estudio de la situación actual

Los actuales métodos de detección del cáncer de mama se basan principalmente en métodos de diagnóstico como mamografías y biopsias, los cuales pueden ser invasivos y a menudo requieren de interpretación experta (Instituto Nacional del Cáncer, 2024) El uso de ML y DL en este campo es relativamente reciente pero prometedor, mostrando mejoras significativas en la capacidad para detectar el cáncer en etapas tempranas.

Los modelos DL particularmente aquellos basados en redes neuronales convolucionales, han demostrado la mejora de la precisión en la detección de cáncer de mama a partir de imágenes médicas. Asimismo, se han identificado biomarcadores genéticos y moleculares prometedores para una detección temprana más precisa, su integración en la implementación de modelos podría mejorar la precisión de las predicciones. Por otro lado, aunque se están creando nuevos modelos de interpretabilidad para comprender mejor los modelos de IA, aún es un desafío el explicar de manera clara su funcionamiento de estos a profesionales médicos que no tienen conocimientos técnicos.

Como se ha comentado, a pesar de la existencia de desafíos, el proyecto a desarrollar tiene potencial de apoyar a la innovación del diagnóstico anticipado del cáncer de mama, utilizando herramientas novedosas y eficientes de la IA.

2.3.3. Estudio y valoración de las alternativas de solución

Para el desarrollo del proyecto se han explorado diferentes alternativas disponibles que son ideales para el trabajo, en este apartado explicaremos las opciones que consideramos tanto en términos de modelos, de algoritmo de interpretabilidad como aspectos técnicos.

2.3.3.1. Creación de modelos

Para la creación de los modelos, se consideraron tanto los enfoques de ML como de DL, ya que ambos son aplicables al tema de la predicción e interpretación en un proyecto de clasificación binaria. La elección de los modelos ha dependido de su idoneidad y eficiencia para los objetivos del proyecto, detalles que se especifican en la sección 6.2.4.1.

Si bien el ML ofrece una variedad de modelos que son más simples de implementar y pueden proporcionar resultados satisfactorios, la opción de utilizar modelos de DL también resultó atractiva. A pesar de ser más complejos y requerir una implementación más tediosa, los modelos de DL ofrecen respuestas más fundamentadas que enriquecen la perspectiva del proyecto.

2.3.3.2. Aspectos técnicos:

En el contexto del proyecto que estamos elaborando, hay muchísimas herramientas que podrían ayudarnos a lograr nuestros objetivos, tanto para la creación de los modelos, su entrenamiento, obtención de resultados y el entorno a trabajar las opciones disponibles son muchas. Sin embargo, hay ciertos criterios que se deben tener en cuenta al elegir, como la comodidad de uso y la experiencia previa, entre otros. Se han analizado las herramientas que nos resultaban más atractivas y en este apartado lo comentamos.

Entorno de desarrollo:

- Google Colab: Es un entorno de desarrollo en la nube que permite escribir y ejecutar código en Python, proporcionando acceso gratuito a GPU y TPU, lo que es ideal para proyectos de ML y DL.
- Jupyter Notebook: Es un entorno de desarrollo como aplicación web que permite crear y desarrollar código Python, muy utilizado para el uso de prototipar y colaborar en proyectos.

Herramientas de Visualización:

- Matplotlib y Seaborn: Son bibliotecas de visualización estática de datos en Python, muy conocidas y básicas a implementar en proyectos de este tipo, ya que proporcionan flexibilidad y simplicidad en la creación de gráficos estadísticos e informativos.
- Plotly: Es una biblioteca de visualización interactiva de datos, utilizada para crear gráficos dinámicos, visualizaciones en línea y más. Permite una mayor interactividad y personalización en comparación con Matplotlib y Seaborn.

Herramienta de Hiper parametrización para DL:

- KerasTuner: Es una biblioteca de ajuste de hiper parámetros para Keras, que proporciona una búsqueda automática para encontrar la mejor combinación de hiper parámetros para los modelos de DL.
- Optuna: Es un marco de optimización de hiper parámetros automatizado que permite una búsqueda más eficiente y flexible de estos en comparación con KerasTuner.

Técnicas de Hiper parametrización para ML:

- GridSearchCV: Es una técnica que se utiliza para encontrar los mejores hiper parámetros para un modelo, evalúa todas las combinaciones de hiper parámetros especificadas para obtener la mejor combinación que permita la mejor predicción.
- RandomizedSearchCV: Es similar a GridSearchCV, pero en lugar de evaluar todas las combinaciones posibles, busca aleatoriamente un subconjunto de las combinaciones de hiper parámetros.

2.3.3.3. Algoritmos de interpretabilidad:

Hay una gran variedad de algoritmos de interpretabilidad que resultan muy interesantes a implementar y que podrían ser beneficiosos para nuestro proyecto. En este apartado comentamos sobre algunos que hemos identificados como ideales para proporcionar una comprensión profunda y valiosa sobre el futuro entrenamiento de nuestros modelos.

- LIME (Local Interpretable Model-agnostic Explanations): Este método ofrece una explicación individual de las predicciones de un modelo ML. Trabaja mediante la toma de una instancia específica generando variaciones ligeras de ella y utiliza un modelo simple para proporcionar una explicación comprensible de cómo las características influyen en la predicción (Ribeiro et al., 2016).
- SHAP (SHapley Additive exPlanations): Este método está basado en la teoría de juegos, ya que asigna una contribución a cada característica en la predicción de un modelo. Esto permite entender cómo cada característica contribuye a la predicción final (Lundberg et al., 2017).
- PDP (Partial Dependence Plots): Este método permite visualizar la relación entre una característica específica y la predicción del modelo y va mostrando cómo cambia la predicción promedio del modelo al variar esa característica. (Friedman, 2001).
- DALEX (Descriptive mACHINE Learning EXplanations): Es un marco que utiliza varios métodos de descomposición de predicciones y permutaciones de características para cuantificar cuánto influye en los modelos (Biecek, 2018).

2.3.4. Selección de la solución

2.3.4.1. Creación de modelos:

En el apartado anterior se comenta que ambos tipos de modelos son bastante atractivos e ideales para la ejecución de nuestro proyecto, por lo que se decidió implementar ambos tipos, tres modelos de ML y dos de DL, con el fin de poder experimentar e interpretar cómo se realizan las predicciones y entender los índices principales de esta enfermedad mediante el uso de la IA.

2.3.4.2. Aspectos técnicos:

En este apartado comentaremos qué aspectos técnicos y herramientas hemos optado por utilizar y por qué, esto se ha basado en ser realistas respecto a la experiencia pasada y cómo lograr que el proyecto siga un curso fluido para evitar contratiempos, ya que hemos considerado que al conocer unas herramientas de antes puede hacer el desarrollo más ameno y manejable.

Entorno de desarrollo:

En este caso, se tiene experiencia de trabajo en ambos entornos, ya que en el pasado se han implementado modelos de aprendizaje automático en ambos sitios. Teniendo en cuenta la fluidez de cada proyecto, el tiempo ahorrado, facilidad de comprensión del entorno y experiencia pasada, se decidió en este proyecto seguir trabajando en Google Colab.

Google Colab desde el primer contacto es muy fácil y entendible de manejar, tiene una interfaz muy sencilla y permite una organización de código que objetivamente, es muy cómoda. Jupyter, por otro lado, también es una muy buena opción, pero puede llegar a ser más confusa y tediosa de iniciar. Ambas plataformas ofrecen un buen desarrollo del proyecto, pero Google Colab da un plus de comodidad y eficiencia.

Herramientas de Visualización:

En trabajos pasados, siempre se ha aplicado Matplotlib y Seaborn para la implementación de distintos gráficos estadísticos y más, y en realidad nunca se ha encontrado razón para no seguir utilizándolo. Sin embargo, en esta oportunidad se quería experimentar con gráficas diferentes, que sean más interactivas, con esta misión se llegó a conocer Plotly. Luego de investigar y ver las diversas opciones que ofrecen, se llegó a la conclusión de mejor seguir trabajando con lo conocido, si bien es cierto, se tenía toda la intención de experimentar con herramientas nuevas, el tiempo iba a ser crucial en este caso.

Ambas bibliotecas cumplen su objetivo, generar gráficos para visualizar los datos y su comportamiento, y aunque Plotly lo haga de una manera diferente, el conocer ya el comportamiento de las otras dos era un plus. Consideramos que tener que aprender aspectos o funcionamiento nuevos durante el desarrollo del

proyecto puede llegar a ser un problema, si hay suerte se puede lograr a realizar todo a tiempo y aunque aprender a usar una nueva librería no nos va a generar un gran contratiempo, estaríamos arriesgando a que ese tiempo invertido en aprendizaje pudo haber utilizado de otra manera.

Es importante aclarar que no se descarta su uso en algún punto del desarrollo, solo que dependerá del factor tiempo ya que se considera que su empleo podría aportar un toque diferente a la generación de gráficas que es una parte muy importante de ese proyecto.

Herramienta de Hiper parametrización para DL:

En este caso, se consideraron las dos opciones expuestas en la anterior sección, tanto KerasTuner como Optuna nos parecen bibliotecas ideales de utilizar para la búsqueda de hiper parámetros, en este caso se decidió utilizar KerasTuner. Optuna la conocimos en el proceso de encontrar una herramienta que agilice la hiper parametrización, nos parece poderosa y personalizable, siendo flexible y proporcionando una curva de aprendizaje más pronunciado, pero esto mismo necesita un tiempo de implementación y configuración mayor, tiempo que no podríamos garantizar tener.

Por otro lado, KerasTuner, está ya integrado con Keras, una de las bibliotecas principales utilizadas en este proyecto, simplificando de esta manera, la implementación e integración. Además, que es una interfaz fácil de usar e intuitiva lo que es beneficioso al no tener una gran experiencia previa en la optimización de hiper parámetros.

Herramienta de Hiper parametrización para ML:

Para la búsqueda y selección de hiper parámetros en modelos ML se consideraron las opciones GridSearchCV y RandomizedSearchCV. Ambas opciones son muy utilizadas para encontrar una combinación de hiper parámetros que garantice un buen rendimiento del modelo.

En este caso, se decidió utilizar GridSearchCV por varias razones, por ejemplo, que evalúa todas las combinaciones posibles de hiper parámetros especificados, lo que garantiza encontrar la mejor combinación sin dejar ninguna opción por

explorar. Además, es adecuado cuando el espacio de búsqueda no es demasiado grande, como es el caso de nuestro proyecto.

Por otro lado, se tiene RandomizedSearchCV, el cual es una excelente opción porque puede ser más eficiente en términos de tiempo computacional puesto que explora un espacio de búsqueda más grande, ya que esta herramienta funciona seleccionando aleatoriamente combinaciones de hiper parámetros.

Sin embargo, se ha considerado que la exhaustividad de GridSearchCV va a garantizar una búsqueda completa y precisa, dos factores importantes a tener en cuenta en su selección, pero no se descarta intentar utilizar el RandomizedSearchCV, en caso sea necesario.

2.3.4.3. Algoritmos de interpretabilidad:

En el apartado anterior identificamos y detallamos sobre los algoritmos que consideramos que serían interesantes y adecuados a implementar, en este explicaremos los que finalmente se eligieron.

- **LIME:** Se ha elegido este método por su capacidad de explicar cómo diversas características influyen en las predicciones de manera específica. En el contexto de este proyecto, al ser modelos que analizan casos de pacientes individuales, LIME ayuda a comprender determinado resultado de la predicción.
- **SHAP:** Se ha elegido este método que, aunque realiza al igual que LIME interpretaciones locales, tiene un enfoque que permite comprender la importancia general de las características de todo el modelo. SHAP descompone la contribución de cada característica a una predicción específica, pero también permite conocer una visión más amplia de cómo todas las características impactan en general.

Al aplicar ambos modelos, se obtienen diferentes resultados sobre la interpretación de los modelos, permitiendo realizar conclusiones fundamentadas, se aprovechan ambas visiones de los métodos para conseguir una visión a detalle sobre la influencia de las características, pero también, general.

Respecto a PDP y DALEX, es cierto que son herramientas muy poderosas en la interpretación de modelos, pero en esta ocasión, realmente no cumplen las

necesidades que se tiene en este proyecto. PDP, se centra en la visualización de la relación entre una característica específica y la presión de un modelo, que es útil, pero necesitamos más profundidad de interpretación.

Por otro lado, DALEX nos va a ofrecer un marco general para la interpretación del modelo, sin embargo, consideramos que necesitamos un enfoque más centrado en las características que es lo que nos permitirá comprender más el modelo y las predicciones brindadas.

3. METODOLOGÍAS USADAS

En este apartado, se selecciona la metodología que mejor se adecua al proyecto. La metodología es crucial para el desarrollo del proyecto, ya que proporciona una estructura para su avance.

Este proyecto se concibe como incremental, por lo que desde el principio estaba claro que se necesitaba una metodología de este tipo. Sin embargo, también se reconocía la posibilidad de encontrar obstáculos en el camino, lo que descartaba la opción de una metodología tradicional como la en cascada.

Recordemos que una metodología tradicional es aquella que lleva una planificación rígida y pasa varios ciclos de vida (Ilustración 1). Sin embargo, no es flexible, no puede haber modificaciones inesperadas, esto conlleva a una variación de costes y tiempo, lo que es un problema.

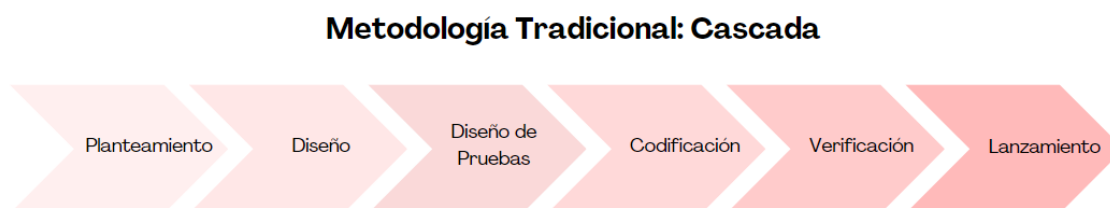


Ilustración 2. Metodología Tradicional "Cascada". Elaboración propia

Volviendo al proyecto, se tenía claro que debía ser flexible. Lo óptimo en este caso es elegir una metodología ágil (Ilustración 2), ya que esta es todo lo contrario a una metodología tradicional, la flexibilidad es el punto fuerte aquí, es iterativo, los requisitos pueden variar y hay un continuo feedback.

Algo muy importante si tenemos en cuenta que las predicciones que se obtengan se tienen que comentar con el cliente, que, por cierto, en esta metodología se encuentra siempre presente.

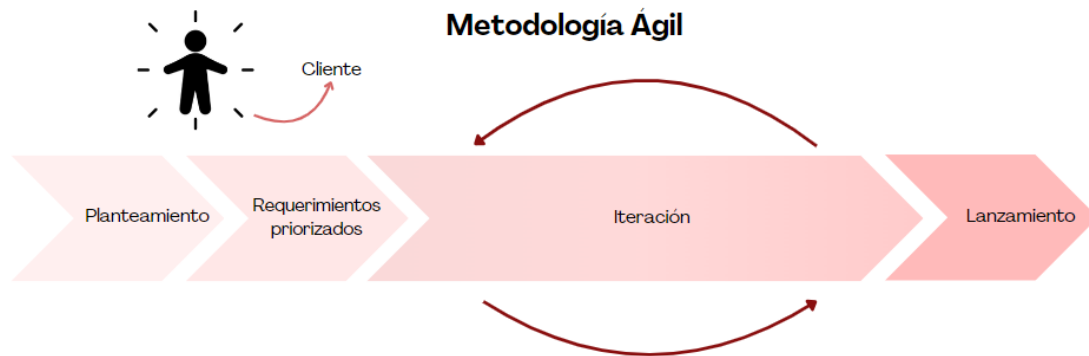


Ilustración 3. Metodología ágil. Elaboración propia.

3.1. Estudio de metodologías ágiles

Luego de decidir el tipo de metodología que se utilizará en el proyecto, se tenía que decidir cuál de las tantas alternativas utilizar, se tuvieron cuatro en cuenta: Scrum, Kanban, Extreme Programming (XP) y Adaptive Software Development (ASD). A continuación, se comentará un poco sobre cada una de estas metodologías.

3.1.1. Scrum

Es una metodología que se enfoca en el desarrollo iterativo e incremental, lo que permite la adaptación en entornos medianamente complejos. El proceso se estructura alrededor del PB, una lista que incluye todas las funcionalidades, requisitos y mejoras esperadas del producto.

Scrum está basado en ciclos de trabajo llamados Sprints y utilizan el PB, donde se establecen y revisan incrementos de valor según la retroalimentación recibida. Cada Sprint culmina con entregas de funcionalidades comprometidas y se caracteriza por reuniones diarias breves y una reunión final de revisión y retrospectiva para evaluar los progresos (Scrum: A Systematic Literature Review, 2023)

3.1.2. *Kanban*

Esta es una metodología visual ya que se basa en la optimización del flujo de trabajo y reducir el tiempo de entrega mediante el uso de tarjetas visuales para representar el progreso de las tareas, de esta manera, se evitan sobrecargas y excesos.

El método de uso se basa en la presentación de tareas por tarjetas y moverlas por medio de las columnas a medida que se avanza. No tiene sprints definidos por lo que el trabajo es realizado continuamente (dos Santos et al. Journal of Software Engineering Research and Development, 2018).

3.1.3. *Extreme Programming (XP)*

Para esta metodología es prioridad la calidad, la buena comunicación y simplicidad, puesto que se basa en pruebas unitarias que se integran continuamente. Asimismo, apuesta por una programación en parejas que permiten un avance más veloz.

Su método de uso implica una definición de valor de negocio por parte del cliente, estimaciones de esfuerzo por parte de los programadores y un ciclo iterativo de desarrollo, el cual incluye fases como la exploración, planificación, iteraciones, producción, entre otros. Esta metodología está diseñada para proyectos donde los requisitos pueden cambiar (Exploring Extreme Programming in Context: An Industrial Case Study, 2004).

3.1.4. *Adaptive Software Development (ASD)*

ASD está centrado en la adaptabilidad y gestión efectiva de riesgos en proyectos enfocados al desarrollo de software. Es interesante porque se basa en la identificación temprana de riesgos y por lo mismo, en la toma de decisiones espontáneas para así responder a los cambios del entorno y del proyecto.

El método de uso implica un ciclo de vida que incorpora tres fases: Especulación, colaboración y aprendizaje. Sin ahondar mucho, en la primera se inicia el proyecto con una planificación que define los ciclos de entrega. En la segunda, el equipo trabaja conjuntamente en el desarrollo. Finalmente, en la tercera, el equipo reflexiona sobre el proceso y aplica las lecciones aprendidas en ciclos

futuros con el fin de mejorar (Adaptive Software Development for developing safety critical software, 2015).

3.2. Elección de metodología

En este apartado comentaremos la metodología escogida, pero también las que se descartaron que en este caso fueron la metodología Kanban y XP.

Kanban se descartó porque, aunque ofrece flexibilidad, que definimos como un requisito fundamental, no proporciona una estructura clara de planificación, algo que es crucial en este proyecto siendo de investigación con objetivos definidos.

XP se descartó como opción porque, si es cierto se centra en la calidad del software y en la entrega continua, su enfoque en el desarrollo no es tan adaptable a un entorno de investigación que está en constante evolución. Además, al involucrar pruebas unitarias no resulta relevante al tener un proyecto cuyo enfoque principal es la experimentación y exploración de modelos ML y DL.

Con eso explicado, nos enfocamos en las dos metodologías que sí consideramos las más adecuadas para el proyecto, Scrum y ASD, para ello presentamos una tabla de ventajas (Tabla 1) y desventajas (Tabla 2) de cada uno.

Tabla 1. Ventajas de Scrum y ASD. Elaboración propia.

Ventajas Scrum	Ventajas ASD
Estructura para desarrollo iterativo: esta es clara ya que se proporcionan roles definidos y sprints iterativos.	Estructura flexible: no está tan estructurado lo que permite explorar diferentes enfoques y soluciones.
Flexibilidad de adaptación a cambios: la existencia de sprints permite adaptaciones rápidas o cambios de requisitos.	Enfoque en adaptabilidad y gestión de riesgos: se centra en los cambios y gestionar riesgos, muy apropiado para este proyecto.
Colaboración y comunicación continua: se fomenta un trabajo en equipo, incluyendo al cliente y así tener una comunicación constante.	Colaboración y comunicación: sigue la misma línea de Scrum, fomenta la comunicación fluida.
Documentación reducida: no es necesario perder mucho tiempo en documentar el proyecto, lo cual nos	

permite disponer de más tiempo para la parte experimental.	
--	--

Tabla 2. Desventajas de Scrum y ASD. Elaboración propia

Desventajas Scrum	Desventajas ASD
Necesidad de planificación detallada: esto es preciso al inicio de cada sprint, lo que puede ser desafiante en casos de constante cambio.	Gestión cuidadosa de cambios: su flexibilidad puede llevar a desviaciones por no tener un camino claro.
Entregas incrementales pueden ser limitantes: al ser un proyecto de investigación, se pueden requerir cambios que se necesite comunicar en cualquier momento y no esperar hasta el siguiente sprint.	Requerimiento de toma de decisiones rápidas: en caso el equipo no esté preparado para este tipo de evento, puede ser desafiante de afrontar.
	Falta de estructura: muy relacionado con la primera desventaja, puede que sea difícil de mantener el enfoque.

Sabiendo las ventajas y desventajas de ambas metodologías y cómo debería ser la organización del proyecto, se terminó por decidir que Scrum sería la más adecuada para el desarrollo, contando con incrementos graduales y flexibles, al igual que una comunicación fluida con el equipo y cliente.

3.3. Uso de Scrum en el proyecto

En este trabajo hemos aplicado la metodología ágil Scrum, la cual se basa en un incremento en el desarrollo del proyecto de manera iterativa y gradual, mediante Sprints.

Para implementar esta metodología es muy importante tener objetivos claros, transparencia respecto a los avances y aceptar la adaptación del proyecto. En un trabajo como este es muy común tener que realizar cambios o replantear el tiempo de dedicación a cierto objetivo, esta es la razón principal por la cual se eligió esta metodología como medio organización.

3.3.1. Roles en el equipo Scrum

En esta metodología encontramos tres roles definidos:

- **Product Owner (PO):** Este es el responsable del avance del trabajo del equipo de desarrollo (Scrum Team), encargado de gestionar el Product Backlog, la cual es una lista de todas las funcionalidades y requisitos que se deben abordar en el proyecto. Por otro lado, también ayuda a definir los objetivos y requisitos y comunicarlo al equipo de desarrollo.
- **Scrum Master:** Es responsable de garantizar que se siga, por parte del equipo de desarrollo, las tareas definidas. Además, se encarga de la planificación, revisión y retrospectiva de los Sprint.
- **Scrum Team:** Es el equipo de desarrollo, el cual es responsable de convertir lo definido en el Product Backlog en avances graduales con el fin de ser entregados al final de cada Sprint, en los cuales participa activamente.

Sabiendo esto y adaptándolo al proyecto, los roles son declarados de la siguiente manera:

Tabla 3. Roles definidos en el proyecto. Elaboración propia.

Rol	Persona
Product Owner (PO)	Antonio Jesús Banegas Luna
Scrum Master	María Isabel Cornejo Rojas
Scrum Team	María Isabel Cornejo Rojas

4. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO

4.1. Tecnologías

4.1.1. *Análisis de datos y modelado*

- **Python:** Este lenguaje de programación se ha utilizado para el desarrollo del código, ya que, no solo se tiene experiencia con este lenguaje, sino porque es el óptimo a implementar por la variedad de bibliotecas que dispone para el análisis de datos y aprendizaje automático.
- **NumPy:** Es la base de bibliotecas necesarias como Pandas, Scikit-learn y TensorFlow. Por ello se ha implementado ya que ayudará a realizar cálculos numéricos como para la creación de matrices, que es un tipo de cálculo muy importante para este tipo de proyecto.
- **Pandas:** Esta biblioteca se utilizará para la manipulación y análisis de los datos, al igual que en la implementación del preprocesamiento y limpieza de estos.
- **Kerastuner:** Se utilizó para la optimización de hiper parámetros en modelos de Keras. Esta biblioteca fue un descubrimiento que se decidió implementar, ya que, es ideal para la optimización de tiempo en la configuración de hiper parámetros en el entrenamiento de los modelos DL.

4.1.2. *Frameworks de aprendizaje automático*

- **Scikit-learn:** Framework para el desarrollo de modelos ML que ofrece una amplia variedad de algoritmos, al igual que herramientas para evaluar y ajustar modelos.
- **TensorFlow y/o Keras:** Frameworks para el desarrollo de modelos DL, proporcionando eficiencia en la implementación y entrenamiento de nuestros modelos.

4.2. Herramientas

4.2.1. Entorno de desarrollo

- **Google Colab:** Utilizado como entorno de ejecución ya que por cómo se organiza es muy cómodo y fácil de entender, y explicar su funcionamiento en caso sea necesario.
- **Google Drive:** Utilizado para el almacenamiento de archivos relacionados con el proyecto como los datasets, memoria sobre el desarrollo y el mismo proyecto de programación.

4.2.2. Gestión de versiones y colaboración

- **Git Hub:** Se utilizó Git como como repositorio de varias bibliotecas que se necesitaron, había un problema con Google Colab que las versiones que tenían no eran compatibles con ciertas bibliotecas, desde Git Hub se exportaron.

4.2.3. Visualización de datos

- **Matplotlib y Seaborn:** Librerías de visualización de datos en Python que se emplearon para crear gráficos y visualizaciones informativas.

4.2.4. Herramientas de documentación y presentación

- **Microsoft Word y PowerPoint:** Utilizado para la redacción del desarrollo del proyecto y apuntes de notas sobre resultados obtenidos.
- **Scribbr:** Generador de citas utilizado para gestionar las referencias y citas bibliográficas del proyecto.
- **Google Scholar:** Plataforma utilizada para la búsqueda de documentos académica y referencias relevantes para el proyecto.
- **Canva:** Herramienta de diseño donde se crearon varias figuras y gráficos que complementaron el documento.

5. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN

5.1. Organización de Sprints

En la ilustración 4 se muestra una línea de tiempo de los Sprints realizados, que en total han sido 6 y sus fechas de inicio. Como se detalla más adelante en el informe de cada sprint, se ha respetado el tiempo de 15 días (2 semanas) por Sprint, excepto en el último sprint que solo ha durado 11 días.



Ilustración 4. Línea de tiempo de cada Sprint. Elaboración propia

Se observa que se han llevado a cabo Sprints en los que el PO no ha estado presente físicamente, pero se le ha informado sobre el progreso o los retrasos del proyecto a través de otros medios. También se han tenido sprints en los que el PO ha estado presente y ha aportado nuevas necesidades u opiniones que han ayudado a confirmar el buen camino del proyecto.

Se decidió organizar cada sprint basándonos en las especificaciones del Product Backlog, que principalmente son 11, aunque más adelante se definen sus subtarefas técnicas. Al organizar los sprints de esta manera, cada iteración puede variar entre 2 y 3 tareas por realizar.

5.2. Especificación de Product Backlog

En este apartado se presenta la lista de tareas definidas, cada una con un identificador propio, siendo "US" de User y el número de la tarea (Tabla 4).

Tabla 4. Product Backlog. Elaboración propia.

ID	Nombre de requisito de Product Backlog
US_1	Búsqueda y elección del dataset.
US_2	Implementación de la GAN.
US_3	Elección de métricas.
US_4	Implementación de Análisis Exploratorio de Datos.
US_5	Comparación de dataset original y sintético.
US_6	Preparación de dataset para la aplicación de IA (separación de variables).
US_7	Elección e implementación de modelos ML.
US_8	Elección e implementación de modelos DL.
US_9	Evaluación en base a métricas sobre los resultados obtenidos.
US_10	Implementación de algoritmos de interpretabilidad.
US_11	Implementación de función de consenso.

Habiendo definido estos requisitos principales con el PO, con el equipo desarrollador se definieron las subtarefas, creando la Tabla 5 que no solo contiene todas las tareas a realizar, sino también la prioridad, esfuerzo y tiempo estimado. El cómo se definió cada uno se explica en el siguiente apartado.

5.3. Estimaciones utilizadas

5.3.1. Estimación de prioridad

Teniendo los roles definidos, se utilizará el método MoSCoW para especificar la prioridad de cada historia de usuario. Este se compone por una nemotecnia: Must, Should, Could, Won't. El primero, Must (M), se refiere a que es una historia de máxima prioridad y de realización urgente. Should (S), una historia que debería hacerse a largo plazo. Could (C), significa que la historia no es necesario elaborarla y Won't (W), que no se usará en esta ocasión.

5.3.2. Estimación de esfuerzo

Para estimar el esfuerzo de cada tarea se utiliza la técnica de Puntos de Historia apoyándonos en un grupo de valores de la secuencia de Fibonacci. Básicamente, la implementación de este método medirá cuánto trabajo se requiere para una tarea en comparación con otras. A la hora de evaluar la complejidad de cada tarea se tienen en cuenta factores como la incertidumbre, la experiencia previa y detalles que pueden afectar a la estimación del esfuerzo.

La ayuda de la secuencia de Fibonacci, la cual es una serie de números en la que cada uno es la suma de los dos anteriores. Con estos valores se asignarán puntos a las tareas, normalmente se utilizan 1, 2, 3, 5, 8 e incluso 13. Se suelen evaluar las tareas más fáciles con 1, 2 o 3 puntos, aquellas más complicadas con 5, 8 o 13.

5.4. Tabla definitiva del Product Backlog con las estimaciones de prioridad y esfuerzo

Luego de haber definido la manera de puntuar los requisitos definidos, en la Tabla 5 se detalla todo de una manera visual, para luego explicar la justificación de las puntuaciones.

* *Tiempo estimado* = *PH* * 2

Tabla 5. Requisitos técnicos específicos. Elaboración propia.

ID	Nombre de requisito de Product Backlog	ID	Requisito específico	Prioridad	Esfuerzo (PH)	Tiempo estimado (Horas)
US_1	Búsqueda y elección del dataset.			M	3	6
					3 PH	6 hora
US_2	Implementación de la GAN.	US_2.1	Limpieza de dataset.	M	3	6
		US_2.2	Aplicación de codificadores.	S	3	6
		US_2.3	Entrenamiento de la GAN	M	13	26
		US_2.4	Generación y exportación de nuevo dataset.	M	1	2
					20 PH	40 horas
US_3	Elección de métricas.			C	3	6
					3 PH	6 horas
US_4	Implementación de EDA	US_4.1	Implementación de EDA pre-GAN (dataset original)	M	3	6

		US_4.2	Implementación de EDA post GAN (dataset sintético)	M	3	6
					6 PH	12 horas
US_5	Comparación de dataset original y sintético.			M	2	4
					2 PH	4 horas
US_6	Preparación de datos para IA	US_6.1	Separación de variables features y target para ML.	S	1	2
		US_6.2	Separación train, validation y test para ML.	S	2	4
		US_6.3	Normalización de variables.	C	1	2
					4 PH	8 horas
US_7	Elección e implementación de modelos ML.	US_7.1	Entrenamiento y evaluación modelo 1 (Regresión Logística).	M	3	6
		US_7.2	Entrenamiento y evaluación modelo 2 (Gradient Boosting).	M	3	6

		US_7.3	Entrenamiento y evaluación modelo 3 (Random Forest).	M	3	6
		US_7.4	Comparación resultados entre modelos ML.	S	1	2
					10 PH	20 horas
US_8	Elección e implementación de modelos DL.	US_8.1	Entrenamiento y evaluación modelo 1 (ANN).	M	13	26
		US_8.2	Entrenamiento y evaluación modelo 2 (MLP).	M	8	16
		US_8.3	Comparación resultados entre modelos DL.	S	1	2
		US_8.4	Comparación de modelos ML y DL.	S	1	2
					23 PH	46 horas
US_9	Evaluación métricas sobre resultados.			M	1	2
					1 PH	2 horas
US_10	Implementación algoritmos de interpretabilidad.	US_10.1	Implementación de LIME	M	8	16
		US_10.2	Implementación de SHAP	M	8	16

					16 PH	32 horas
US_11	Implementación de función de consenso.			S	5	10
					5 PH	10 horas

Total, horas estimadas: 186 horas

Total, puntos de historia: 93 puntos de historia

Explicación de valores dados a cada subtask del Product Backlog:

[US_1]

- [US_1] Búsqueda y elección del dataset.
 - Prioridad: Must Have (M). Una tarea relativamente sencilla pero crucial y puede requerir tiempo para encontrar el dataset adecuado.

[US_2]

- [US_2.1] Limpieza de dataset.
 - Prioridad: Must Have (M). La limpieza del dataset es importante para garantizar la calidad y consistencia de los datos utilizados en el proyecto.
- [US_2.2] Aplicación de codificadores
 - Prioridad: Should Have (S). La aplicación de codificadores es importante para la preparación de datos y la extracción de características relevantes antes de su procesamiento por la GAN.
- [US_2.3] Entrenamiento de la GAN
 - Prioridad: Must Have (M). El entrenamiento de la GAN es una etapa crítica para lograr que el modelo aprenda a generar datos sintéticos que sean representativos del conjunto de datos original.
- [US_2.4] Generación y exportación de nuevo dataset
 - Prioridad: Must Have (M). La generación y exportación del nuevo dataset sintético es crítico en el proyecto para disponer dataset más grande pero representativo del original.

[US_3]

- [US_3] Evaluación métricas sobre resultados
 - Prioridad: Must Have (M). La evaluación de métricas sobre los resultados es crucial para medir el rendimiento y la eficacia de los modelos implementados.

[US_4]

- [US_4.1] Implementación de EDA pre-GAN (dataset original)
 - Prioridad: Must Have (M). Fundamental para comprender la estructura y características del dataset antes de aplicar la GAN, lo que ayuda a identificar posibles problemas.
- [US_4.2] Implementación de EDA post GAN (dataset sintético)
 - Prioridad: Must Have (M). Fundamental para evaluar la calidad y compararlo con el dataset original, lo que proporciona información importante sobre la efectividad de la generación de datos sintéticos.

[US_5]

- [US_5] Evaluación métricas sobre resultados
 - Prioridad: Must Have (M). La evaluación de métricas sobre los resultados es muy importante para medir el rendimiento y la eficacia de los modelos implementados. Es esencial para validar y garantizar que los modelos cumplan con los requisitos y objetivos del proyecto.

[US_6]

- [US_6.1] Separación de variables features y target para ML
 - Prioridad: Should Have (S). La separación adecuada de las variables features y target sirve para preparar los datos para el entrenamiento de modelos de ML, garantizando que las variables predictoras y la variable objetivo estén correctamente definidas y de una manera organizada.
- [US_6.2] Separación train, validation y test para ML
 - Prioridad: Should Have (S). La separación de los datos en conjuntos de entrenamiento, validación y prueba sirve para evaluar el rendimiento de los modelos y evitar el sobreajuste.

- [US_6.3] Normalización de variables
 - Prioridad: Could Have (C). La normalización de variables ayudará a mejorar el rendimiento de los modelos ML, asegurando que todas las características tengan la misma escala.

[US_7]

- [US_7.1] Entrenamiento y evaluación modelo 1 (Regresión Logística)
 - Prioridad: Must Have (M). Entrenar y evaluar el modelo de Regresión Logística es crucial para entender cómo los datos se ajustan a un modelo lineal, lo que proporciona información sobre las relaciones entre las variables y su capacidad predictiva.
- [US_7.2] Entrenamiento y evaluación modelo 2 (Gradient Boosting)
 - Prioridad: Must Have (M). Entrenar y evaluar el modelo de Gradient Boosting es esencial para explorar su capacidad para mejorar la precisión del modelo mediante la combinación de múltiples modelos débiles, lo que ayuda a obtener resultados más precisos.
- [US_7.3] Entrenamiento y evaluación modelo 3 (Random Forest)
 - Prioridad: Must Have (M). Entrenar y evaluar el modelo de Random Forest es fundamental ya que proporciona información sobre la importancia de las variables y la capacidad de generalización del modelo.
- [US_7.4] Comparación resultados entre modelos ML
 - Prioridad: Should Have (S). Comparar los resultados de los modelos es importante para identificar el mejor modelo, ya que luego este se comparará con el mejor de DL.

[US_8]

- [US_8.1] Entrenamiento y evaluación modelo 1 (ANN)
 - Prioridad: Must Have (M). Entrenar y evaluar el modelo de Red Neuronal Artificial (ANN) es crucial para explorar su capacidad para modelar relaciones complejas en los datos.

- [US_8.2] Entrenamiento y evaluación modelo 2 (MLP)
 - Prioridad: Must Have (M). Entrenar y evaluar el modelo de Perceptrón Multicapa (MLP) es esencial para comprender cómo este tipo de red neuronal puede manejar tareas complejas de clasificación, lo que proporciona información valiosa sobre su rendimiento en comparación con otros modelos.
- [US_8.3] Comparación resultados entre modelos DL
 - Prioridad: Should Have (S). Comparar los resultados entre los modelos de DL es importante para seleccionar el mejor modelo.
- [US_8.4] Comparación de modelos ML y DL
 - Prioridad: Should Have (S). Comparar los resultados entre el mejor modelo ML y DL es crucial para entender las diferencias en su rendimiento y determinar cuál es rinde mejor, para luego interpretar y realizar un consenso.

[US_9]

- [US_9] Evaluación métricas sobre resultados
 - Prioridad: Must Have (M). La evaluación de métricas sobre los resultados es crucial para medir el rendimiento y la eficacia de los modelos implementados. Proporciona información objetiva sobre la calidad de las predicciones y ayuda a identificar áreas de mejora.

[US_10]

- [US_10.1] Implementación de LIME
 - Prioridad: Must Have (M). Implementar LIME (Local Interpretable Model-agnostic Explanations) es esencial para comprender cómo los modelos de ML y DL toman decisiones a nivel local, lo que ayuda a comprender sore cómo se generan las predicciones.
- [US_10.2] Implementación de SHAP
 - Prioridad: Must Have (M). Implementar SHAP (SHapley Additive exPlanations) es crucial para comprender la importancia de cada característica en la predicción del modelo, lo que proporciona

explicaciones globales y locales sobre cómo las características afectan las predicciones de los modelos.

[US_11]

- [US_11] Implementación de función de consenso
 - Prioridad: Should Have (S). La implementación de una función de consenso es de gran ayuda para combinar las predicciones de varios modelos en un solo resultado, lo que puede mejorar la robustez y la precisión del sistema final al tener en cuenta múltiples perspectivas.

5.5. Realizar una planificación temporal del proyecto

En este apartado, continuaremos con la explicación de cómo se ha definido la estimación de prioridad y esfuerzo de las subtarear definidas en el Product Backlog anteriormente. Estas estimaciones se han realizado teniendo conocimiento previo de su implementación y sabiendo que es relativo, ya que, al trabajar con una metodología ágil, nos encontramos en una posición flexible que permite ajustar lo organizado en el momento que sea conveniente.

5.5.1. Estimación de tiempo

El objetivo principal de esta sección es detallar cómo los puntos de historia y la priorización MoSCoW, discutidos previamente en la sección 5.3, se utilizan para calcular el tiempo necesario para completar cada tarea y el proyecto en su totalidad. Se busca proporcionar un marco temporal detallado que refleje la complejidad de las tareas y la capacidad del equipo.

Varios detalles se tienen en cuenta para la organización del tiempo. Uno de ellos es la duración del proyecto. Se estableció como fecha de fin el 5 de mayo y una duración de 5 meses, con el inicio programado para el 5 de enero. Sin embargo, debido a problemas de organización y disponibilidad, se aplazó un mes, como se indica en el sprint 0. La primera reunión tuvo lugar el 6 de febrero.

Sabiendo esto, el número total de semanas para el desarrollo del proyecto ha sido de 16 semanas, considerando el desarrollo solo durante los días laborales (lunes a viernes) y descontando los días festivos. Sin embargo, durante el último sprint del 25/04 al 06/05, se trabajaron los dos fines de semana y, a diferencia de los otros sprints donde solo se trabajaba 5 horas, en este se trabajó hasta 9 horas al día. A continuación, se presenta la información resumida.

Días Laborales Ajustados:

- Total de semanas: 16
- Días laborales por semana: 5
- Total de días laborales sin ajustar:

$$16 \text{ semanas} \times 5 \frac{\text{días}}{\text{semana}} = 80 \text{ días}$$

- Días festivos: 4 días
- Días laborales ajustados: $80 - 4 = 76 \text{ días}$

Horas de trabajo durante días laborales normales:

- Horas de trabajo por día: 5
- Horas por día durante este periodo: 9
- Tiempo total de trabajo disponible (sin fines de semana extra):

$$76 \text{ días} \times 5 \frac{\text{horas}}{\text{día}} = 380 \text{ horas}$$

Trabajo extra durante el último sprint (incluyendo fines de semana):

- Días adicionales trabajados: 5 (contando fines de semana del 25 de abril al 5 de mayo)
- Horas por día durante este periodo: 9
- Tiempo de trabajo extra:

$$5 \text{ días} \times 9 \frac{\text{horas}}{\text{día}} = 45 \text{ horas}$$

Tiempo total de trabajo disponible:

$$\text{Tiempo total de trabajo} = 380 \text{ horas} + 45 \text{ horas} = 425 \text{ horas}$$

Es importante destacar que, aunque la suma de todas las tareas en la Tabla 5 indica 186 horas, esta cifra no contempla el tiempo asignado a la resolución de imprevistos, la definición y ajustes de la organización del proyecto, etapas de pruebas, reuniones, y la investigación necesaria para el desarrollo.

Estos elementos adicionales son cruciales para una comprensión completa del esfuerzo requerido y están detalladamente explicados en la sección 5.3, donde se discuten las actividades post ejecución de cada tarea.

5.5.1.1. Metodología de Estimación

- Asignación de Puntos de Historia: Cada subtarea del proyecto ha sido evaluada y se le han asignado puntos de historia utilizando la secuencia de Fibonacci para reflejar la complejidad y el esfuerzo requerido de manera proporcional.

- Cálculo de la Velocidad del Equipo: Basado en el total de 425 horas de trabajo disponibles para el proyecto y los 93 puntos de historia establecidos, la velocidad inicial se calcula de la siguiente manera:

$$Velocidad\ del\ equipo = \frac{93\ PH}{425\ horas} = 0.219\ \frac{PH}{hora}$$

5.5.2. Estimación de escenarios

Este valor se ajustará para crear escenarios de estimación optimista, realista y pesimista mediante la aplicación de modificaciones del 15%. En realidad, no hay un número definitivo a usar para el rango de los escenarios de estimación, pero en este caso se utilizará el 15%, ya que al ser un proyecto con mucha similitud al realizado en el pasado (Proyecto de alumna interna), la dirección y el cómo encaminar el proyecto ya son conocidos.

Se decidió implementar esta estimación ya que al trabajar con una metodología ágil donde aplicamos la flexibilidad se necesita tener unos límites. Por ello, es ideal tener una perspectiva de la mejor o peor situación que se puede llegar a atravesar.

Según Mike Cohn en su libro “Agile Estimating and Planning” (2005), la práctica de este tipo de estimación es crucial para mejorar la planificación del proyecto o la incertidumbre, obteniendo así éxito en entornos donde se aplican metodologías ágiles.

Los escenarios se calculan de la siguiente manera:

- Optimista: Se aumenta la velocidad en un 15%, lo que supone una ejecución más eficiente. $0.219 \times 1.15 \approx 0.252\ PH/hora$
- Realista: Se mantiene la velocidad calculada inicialmente.
- Pesimista: Se reduce la velocidad en un 15% debido a posibles inconvenientes. $0.219 \times 0.85 \approx 0.186\ PH/hora$

El tiempo estimado varía según el escenario:

- Optimista: $Tiempo\ estimado = \frac{93\ PH}{0.252\ PH/hora} \approx 370\ horas$
- Realista: Mismo tiempo estimado al inicio.

- Pesimista: $Tiempo\ estimado = \frac{93\ PH}{0.186\ PH/hora} \approx 500\ horas$

Tabla 6. Tabla de velocidad y tiempo estimado. Elaboración propia.

Visión	Velocidad (PH/h)	Tiempo Estimado (h)
Optimista	0.252	370
Realista	0.219	425
Pesimista	0.186	500

La tabla demuestra cómo la variación en la velocidad afecta al tiempo total necesario para completar el proyecto dependiendo del escenario. Se usa un 15% de ajuste ya que se debe tener en cuenta un cierto grado de incertidumbre adecuado.

5.5.3. Planificación temporal

En este apartado se mostrará la planificación temporal utilizando un diagrama de Gantt simplificado. La planificación se ha separado por sprint, y cada cuadrado representa un día de trabajo de 5 horas. Además, cada tarea del sprint tiene una duración planificada de acuerdo con lo indicado en la Tabla 5.

A continuación, se muestra la leyenda que se utiliza a lo largo de la planificación:

Ilustración 5. Leyenda planificación temporal. Elaboración propia.

	Reunión con el cliente	
	Envío reporte	
	Entrega final	
	Desarrollo de tareas	
	Margen por emergencia	

Esta tabla ayuda a entender mejor las siguientes ilustraciones, aunque son muy sencillas, están complementadas por colores.

Sprint 0		Semana 0	Semana 1	Semana 2
US_1	US_1			

Ilustración 6. Planificación temporal del Sprint 0. Elaboración propia.

En la ilustración, la semana 0 comienza el 5 de febrero, la primera reunión sería el 6 de febrero, tal como se planificó, con duración de 11 días hasta la segunda reunión que daría inicio al sprint 1.

Sprint 1		Semana 2	Semana 3	Semana 4
US_2	US_2.1			
	US_2.2			
	US_2.3			
	US_2.4			
US_3	US_3			

Ilustración 7. Planificación temporal Sprint 1. Elaboración propia.

La semana 2 es la continuación de la ilustración anterior, teniendo la segunda reunión el miércoles 21 de febrero. Ese mismo día se iniciaría la investigación sobre la tarea US_3 relacionada con la definición de las métricas a usar. Al día siguiente, se seguiría con la tarea 2.1 y así sucesivamente hasta la tercera reunión el viernes 8 de marzo, dando inicio al sprint 2.

Sprint 2		Semana 5	Semana 6
US_4	US_4.1		
	US_4.2		
US_5	US_5		
US_6	US_6.1		
	US_6.2		
	US_6.3		
US_7	US_7.1		
	US_7.2		
	US_7.3		
	US_7.4		

Ilustración 8. Planificación temporal Sprint 2. Elaboración propia.

Como se ha mencionado, se tendría la reunión el viernes y el lunes siguiente (11 de marzo) comenzarían las tareas del sprint 2, avanzando de manera sucesiva

según lo planificado y estimado, hasta la cuarta reunión el viernes 22 de marzo, que iniciaría el sprint 3.

Sprint 3		Semana 9	Semana 10	Semana 11
US_8	US_8.1			
	US_8.2			
	US_8.3			
	US_8.4			

Ilustración 9. Planificación temporal Sprint 3. Elaboración propia.

En este sprint no se tendría reunión con el PO debido a días festivos, por lo que se seguiría trabajando hasta la siguiente reunión. Por ello, se han asignado tareas de mayor dificultad y duración. Además, el cuadrado azul, como indica la Tabla 14, representa el envío de reportes para mantener informado al PO sobre los avances o contratiempos.

Sprint 4		Semana 11	Semana 12	Semana 13
US_9	US_9			
US_10	US_10.1			
	US_10.2			
US_11	US_11			

Ilustración 10. Planificación temporal Sprint 4. Elaboración propia.

Siguiendo con el sprint 4, se estima de que el avance sea dé sin problema y fluido, igual hay cuatro días de margen en caso hubiese percances, hasta la quinta reunión que daría inicio al último sprint antes de la entrega final.

Sprint 5		Semana 13	Semana 14	Semana 15
Modificaciones				

Ilustración 11. Planificación temporal Sprint 5. Elaboración propia.

Finalmente, el sprint 5 se estima un esfuerzo máximo por terminar todas las tareas en caso haya alguna pendiente y entregar todo a tiempo el lunes 6 de mayo.

5.6. Realizar una valoración de la dedicación y el coste económico

En este apartado, se analizan los costes del proyecto, los cuales se determinaron teniendo en cuenta diferentes aspectos como los recursos humanos o los materiales necesarios para el desarrollo. Es importante destacar que todas las estimaciones se basaron en la valoración de la dedicación fundamentada en la determinación de las horas de trabajo necesarias para alcanzar los objetivos planificados en el tiempo estimado.

Tabla 7. Estimación económica del proyecto. Elaboración propia.

Concepto	Coste €
Salario del Desarrollador	5.819,25
Coste de la Seguridad Social	1.644,75
Costes de Recursos y Equipamiento	500,00
Costes Indirectos (10% del salario bruto)	582,25
IVA (21% sobre el total)	1.733,57
Total	10.279,82

A continuación, la justificación cada elemento que se muestra en la Tabla 8.

Salario del desarrollador (425 horas): El coste del salario del desarrollador por el tiempo del proyecto se obtiene del salario bruto anual en España, según lo establecido en el XVII convenio de empresas de consultoría y estudios de mercado y de la opinión pública (BOE, 2019). Siguiendo esta línea, el salario bruto anual de un Analista Programador es de 24.640,37 €, lo que equivale a 13,69 € por hora. Por lo tanto, se calculó un coste total del salario del desarrollador de 5.819,25 €.

Seguridad Social: El último dato que tenemos sobre el porcentaje de la Seguridad Social fue del 2022 y se situaba en un 28,3% del salario bruto del trabajador (Web Oficial de la Seguridad Social, 2022). Realizando los cálculos y aplicando este porcentaje sobre el salario que calculamos del desarrollador por 425 horas de trabajo, obtenemos un total de 1.644,75 €.

Costes de Recursos y Equipamiento: Se definieron los gastos relacionados a la adquisición de recursos y equipamiento necesarios para el desarrollo del proyecto, como la conexión a Internet, equipos informáticos, electricidad. Se estimó un coste fijo de 500€ para cubrir estos gastos.

Costes Indirectos: Los costes indirectos son aquellos asociados a la gestión y administración del proyecto, al igual como a otros gastos no específicos. Se calcula aplicando un 10% sobre el salario bruto del desarrollador, resultando un 582,25 €.

IVA (21% sobre el total): El Impuesto sobre el Valor Añadido (IVA) es un atributo indirecto que grava el consumo y se aplica en cada etapa del proceso de producción. En este proyecto, se calcula aplicando una tasa del 21% sobre el total de los costes directos e indirectos descritos anteriormente, calculado así el importe de 1.733,57 €.

En resumen, todas las estimaciones económicas realizadas han sido basadas dentro de requisitos legales y fiscales en España, teniendo en cuenta al mismo tiempo, los aspectos laborales y la gestión de recursos durante el desarrollo del proyecto. En la Tabla 8, se visualiza todo lo que se ha comentado en este apartado, pero de manera más simplificada y obteniendo el coste total

6. DESARROLLO DEL CONTENIDO DEL PROYECTO

6.1. Diseño y organización de código

Como se mencionó en la sección 4, la herramienta utilizada para el desarrollo del proyecto es Google Colab y la estructura que representa cómo se ha organizado el código se muestra en la ilustración 10. Se observan 6 grandes apartados, en los cuales se reflejan las grandes tareas definidas en el Product Backlog y la organización de sprints. Por ejemplo, la parte de la GAN está representada por un solo apartado en la estructura del código porque también es una sola gran tarea con subtareas en la organización.

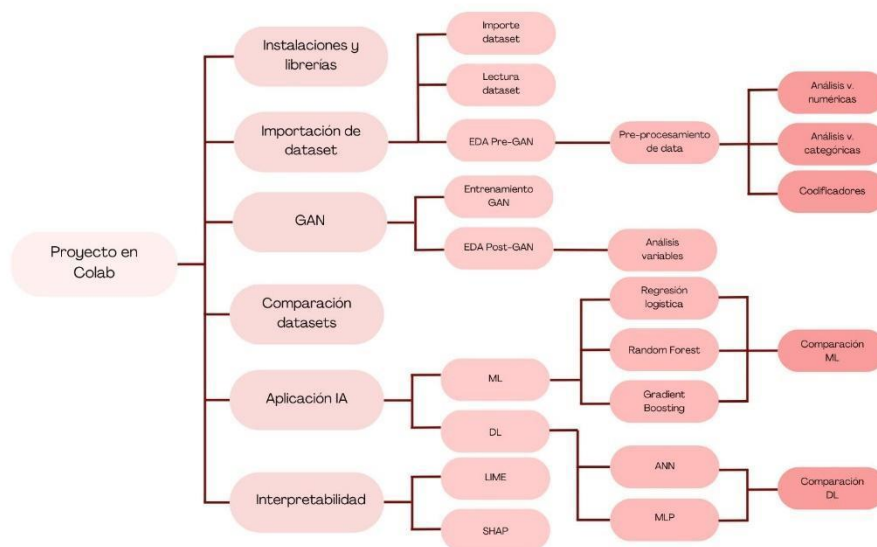


Ilustración 12. Esquema de organización código. Elaboración propia.

Cómo se observa en el esquema, el trabajo se divide en seis secciones principales:

1. **Instalación y Actualización de Bibliotecas:** Se instalan y actualizan las librerías necesarias para el proyecto.
2. **Preparación y Análisis de Datos Iniciales (EDA Pre-GAN):** Esta sección se subdivide en tres partes: importación del dataset, lectura de datos y Análisis Exploratorio de Datos (EDA) antes de la Generative Adversarial Network (GAN). El EDA incluye preprocesamiento de datos, análisis de variables numéricas y categóricas, y la implementación de codificadores.

3. **Entrenamiento de la GAN y Análisis de Datos Generados (EDA Post-GAN):** Aquí se realiza el entrenamiento de la GAN y se analizan las nuevas variables generadas por el proceso.
4. **Comparación entre Datasets Original y Sintético:** Se comparan los datasets original y sintético, generando figuras que facilitan la comprensión de las diferencias y similitudes entre ellos.
5. **Implementación y Comparación de Modelos de IA:** Se implementan y comparan modelos de ML como Regresión Logística, Random Forest y Gradient Boosting, así como modelos de DL como Artificial Neural Network (ANN) y Multi-Layer Perceptron (MLP).
6. **Aplicación de Técnicas de Interpretabilidad:** Se utilizan técnicas como LIME y SHAP para interpretar los resultados de los mejores modelos de ML y DL respectivamente.

Cada sección aborda aspectos específicos del proyecto, desde la preparación de datos hasta la evaluación y comprensión de los modelos generados.

6.2. Especificación Sprints

En este subapartado, se explicará a detalle los sucesos de cada sprint, lo que nos propusimos a realizar, problemas que se tuvieron, manejo de retrasos decisiones tomadas y más.

6.2.1. Sprint 0

Tabla 8. Informe Sprint 0. Elaboración propia.

Nombre: Sprint 0			
Inicio: 06/02	Fin: 21/02	Número de días: 11 días	Número de PH: 3 PH
Informe sobre el Sprint: Se llevó a cabo la primera reunión (06/02) con el Product Owner (PO). En este sprint se estableció el Product Backlog (ver sección 5.2) que abarca el desarrollo del proyecto. Asimismo, se definieron las especificaciones técnicas (ver sección 5.3.) Se encomendó como primera tarea, la búsqueda y pronta elección del dataset más adecuado para el proyecto.			
Tareas superadas del anterior Sprint: -			
Tareas planificadas para el Sprint actual: <ul style="list-style-type: none"> • [US_1] Búsqueda y elección del dataset. 			

Especificación de Requisitos:

- [US_1] Búsqueda y elección del dataset: Se debe investigar y encontrar un dataset adecuado para utilizarlo en el entrenamiento de los modelos ML y DL. El propósito de este requisito es hacer una investigación medianamente profunda en fuentes confiables para encontrar un modelo que tenga todos los detalles necesarios para que las predicciones puedan ser fiables. Asimismo, debe tener las características de no ser muy antiguo, una cantidad aceptable de atributos y filas, al igual que los atributos deben estar bien documentados.

Sobre el Sprint:

Aunque este sprint no se reconoce explícitamente como uno, ya que generalmente contiene información más útil para los desarrolladores que para el Product Owner, es importante destacar los factores definidos en este periodo. Estos se incluyen los métodos de estimación esfuerzo y prioridad detallados en la sección 5.3 y luego aplicados al Product Backlog, explicada también en la sección. Debido a la estructura del documento se prefirió colocar estos aspectos primero para mantener la coherencia.

En la primera reunión, se estableció la primera tarea a desarrollar y finalizar para utilizar en el siguiente sprint, esta fue la [US_1] Búsqueda y elección del dataset, por lo que a lo largo del sprint se realizó una continua investigación de datasets fiables y que contengan las características necesarias. En la tabla 10 se detallan los diferentes datasets que se consideraron en su momento.

Tabla 9. Conjunto de datos considerados. Elaboración propia

Nombre de dataset	N.º de Filas	N.º de Atributos	Año	Fuente	Comentarios
UCTH Breast Cancer Dataset	213	11	2023	https://data.mendeley.com/datasets/63fpbc9cm4/2	No hay muchos datos, se tendría que implementar GAN.
Breast Cancer (MSK, Cancer Cell 2018)	1918	23	2018	https://www.cbioportal.org/study/cnSegmentations?id=breast_msk_2018	No predice si el paciente padece de cáncer o no, sino si sobrevive o no.
Breast Lesions USG	256	21	2024	https://www.cancerimagingarchive.net/collection/breast-lesions-usg/	Sería interesante trabajar con la data por los atributos y es muy actual.
Breast Cancer Prediction	39998	13	2008	https://www.kaggle.com/datasets/haithemhermessi/breast-cancer-screening-data-set?select=dataset.csv	Tiene muchos datos, pero es un conjunto antiguo.

Estos conjuntos de datos serían los que se mostrarían para la siguiente reunión que daría inicio al sprint 1.

6.2.1.1. Retrospectiva Sprint 0

6.2.1.1.1. Lo que se hizo bien:

- Realización de PB para un desarrollo claro del proyecto.
- Definición de las especificaciones técnicas para evitar desviaciones.
- Finalización de la primera tarea pedida.

6.2.1.1.2. Áreas de Mejora:

Posibilidad de mejorar la fluidez de comunicación con el PO para asegurar una comprensión clara y entregar un trabajo satisfactorio.

6.2.1.1.3. Próximos Pasos:

Inicio de un nuevo sprint continuando con los PB definidos.

6.2.2. Sprint 1

Tabla 10. Informe Sprint 1. Elaboración propia.

Nombre: Sprint 1			
Inicio: 22/02	Fin: 08/03	Número de días: 11 días	Número de PH: 23 PH
Informe sobre el Sprint: Se llevó a cabo la segunda reunión (21/02) con el PO. En la reunión se presentó una lista de datasets que fueron considerados para el desarrollo del trabajo, junto al PO se eligió el más adecuado. Por otro lado, surgió una nueva tarea, el desarrollo de una Generative Adversarial Networks (GAN), agregando como una nueva tarea [US_2]			
Tareas superadas del anterior Sprint: <ul style="list-style-type: none"> • [US_1] Búsqueda y elección del dataset. 			
Tareas planificadas para el Sprint actual: <ul style="list-style-type: none"> • [US_2] Implementación de la GAN. • [US_2.1] Limpieza de dataset. • [US_2.2] Aplicación de codificadores. • [US_2.3] Entrenamiento de la GAN. • [US_2.4] Generación y exportación de nuevo dataset. • [US_3] Elección de métricas. 			

Especificación de Requisitos:

- [US_2.1] Limpieza de dataset: en este requisito lo que se pide es limpiar valores basura que no van a servir en el futuro. Por ejemplo, enfocados al dataset que tenemos se tuvieron que limpiar las filas que tenían valores nulos, representados por “#” ya que en un principio se consideró reemplazarlo por “nan” o “0”, pero realmente los casos que llevaban ese valor tenían varios “#” en diferentes atributos.
- [US_2.2] Aplicación de codificadores: este requisito en realidad no es obligatorio, pero se optaría por hacerlo en caso sea necesario para la implementación de la GAN.
- [US_2.3] Entrenamiento de la GAN: este requisito aborda los diferentes pasos que se implementan para crear la GAN, desde su codificación, dividir los datos, las declaraciones de variables, la implementación del

discriminador y generador, y luego la definición de la GAN es lo que se debe hacer para obtener un modelo representativo del dataset original.

- [US_2.4] Generación y exportación de nuevo dataset: este requisito se refiere a que luego del entrenamiento de la GAN, se debe generar el nuevo dataset y verificar que las variables sintéticas sean derivadas del original. Es decir, si normalmente se tiene un límite entre 2 y 12 cm en el en la medida del tumor el nuevo dataset que se genere debería oscilar entre las mismas medidas y no crear un dato atípico que sea un tumor de 28 cm.
- [US_3] Elección de métricas: este requisito se refiere a la investigación de las métricas que se pueden utilizar para los modelos ML y DL que se van a implementar, con el fin de utilizar las métricas más adecuadas para tener una información beneficiosa de lo que estamos entrenando.

Sobre el Sprint:

En la segunda reunión que dio inicio al sprint 1, se mostró la lista de datasets mostradas en el sprint 0 y luego de evaluar con el PO las que se consideraban más adecuadas, se terminó por elegir el dataset UCTH Breast Cancer Dataset, pero al tener solo 213 filas, surgió el primer inconveniente. En principio el proyecto sólo tenía planificado abordar la implementación de modelos ML y DL, luego aplicar algoritmos de interpretabilidad y utilizar una función de consenso para explicar con un buen fundamento el porqué de los resultados.

Sin embargo, con el PO, se acordó que la mejor manera de abordar el inconveniente sería implementando una GAN para poder generar un dataset sintético a partir del original encontrado.

Aunque esta tarea se encuentra ya definida en el listado de PB (ver sección 5.2) este sprint fue en el que se decidió añadir el requisito.

Como observamos en la tabla, todas las tareas planificadas para este ciclo están relacionadas con la implementación de la GAN y la elaboración de un documento de justificación sobre las métricas que se seleccionarán para la futura evaluación de los modelos.

El desarrollo de la GAN fue normal y fluido, se consultaron trabajos pasados y fuentes de información para hacerlo adecuadamente. El único inconveniente que se encontró fue al aplicar los encoders, en principio se iba a utilizar sólo Label Encoder, pero al momento de generar el nuevo dataset se erraba mucho en la generación de variables numéricas como año o edad, dando resultados muy fuera del rango que se encontraba el dataset original.

Year	Age	Menopause	Tumor Size (cm)	Inv-Nodes	Breast	Metastasis	Breast Quadrant	History	Diagnosis Result
3030	919	1	53	1	1	1	6	1	1
3029	1094	1	41	1	1	1	6	1	1
3029	1139	1	30	1	1	1	5	1	1
3030	837	1	16	1	1	0	3	1	0
3029	745	1	25	1	1	1	6	0	1
3029	1745	1	30	1	1	1	6	1	1

Ilustración 13. Captura de una parte del dataset sintético usando Label Encoder como codificador.

En la ilustración, se puede observar lo que se comentó anteriormente, el año de la prueba o la edad de la persona está muy por fuera de lo aceptable, igualmente el tamaño del tumor.

Para resolver esto se decidió solo utilizar One-Hot Encoder, por probar un método diferente y visualizar el resultado, en este caso sí se consiguió un dataset con sentido.

Year	Age	Menopause	Tumor Size (cm)	Breast_Left	Breast_Right	Breast Quadrant_Lov	Breast Quadrant_Lov	Breast Quadrant_Up
2019	49	0	2	0	1	0	0	1
2020	59	0	10	1	1	0	0	0
2020	19	1	2	0	1	0	0	1
2020	55	0	5	1	0	0	0	0
2020	44	1	4	1	0	0	0	0
2020	29	1	8	0	1	0	0	0

Ilustración 14. Captura de una parte del dataset sintético usando One-Hot Encoder como codificador.

El problema estaba solucionado, se consiguió un dataset que tenga datos con sentido y derivados de los reales, el único “problema” en este momento era que se tenían muchas columnas. Al experimentar con el uso de ambos codificadores, nos dimos cuenta de que Label Encoder trabajaba bien con los atributos binarios, y One-Hot Encoder, trabajaba bien con las numéricas, entonces se decidió utilizar ambos para evitar generar más columnas de las necesarias y tener datos acordes al contexto.

Year	Age	Menopause	Tumor Size (cm)	Breast	Metastasis	History	Diagnosis Result	Breast Quadrant_Lov	Breast Quadrant_Lov	Breast Quadrant_Up	Breast Quadrant_Up	Breast Quadrant_Up	Breast Quadrant_Up	Inv-Nodes_0.0	Inv-Nodes_1.0	Inv-Nodes_3.0
2020	46	1	5	1	1	0	1	0	0	0	1	0	0	0	1	0
2020	34	1	3	1	0	0	0	0	1	0	0	0	0	1	0	0
2020	51	1	9	1	1	0	1	0	0	0	0	1	0	0	1	0
2020	20	1	2	0	0	0	1	0	1	0	0	0	1	0	0	0
2019	23	1	2	0	0	0	0	1	0	0	0	0	0	1	0	0
2020	29	1	4	1	0	0	0	1	0	0	0	0	0	1	0	0
2020	35	1	3	0	0	1	0	0	0	1	0	0	0	1	0	0

Ilustración 15. Captura de dataset sintético con el uso de ambos codificadores

De esta manera resolvimos los inconvenientes y se consiguió un dataset óptimo para el desarrollo del proyecto, este siendo compuesto por 16 atributos y 10000 filas.

Por otro lado, la búsqueda y decisión de las métricas tenía que hacerse teniendo en cuenta que se aplicarían a ambos modelos, deberían dar resultados que realmente sean apropiados para el tipo de predicción que se realizaría y nos proporcione información sobre el rendimiento de los modelos.

Siendo muy breves se decidió utilizar:

- Precisión: Indica la proporción de identificaciones positivas que son correctas. La alta precisión refleja una baja tasa de falsos positivos.
- Recall: Muestra el porcentaje del modelo para identificar correctamente las instancias positivas reales.
- F1-Score: Su porcentaje representa un balance entre las dos métricas anteriores, si se obtiene un porcentaje alto significa que se logra identificar satisfactoriamente los verdaderos positivos y la minimización de falsos positivos y negativos.
- Área Bajo la Curva (AUC): Evalúa el rendimiento del modelo en todas las tasas de clasificación. Un porcentaje alto significa mejor discriminación entre las clases positivas y negativas.
- Especificidad: Identifica correctamente las instancias negativas reales, importante para evitar falsos positivos.

En resumen, estas métricas fueron escogidas sabiendo que tenemos un proyecto de predicción binaria, donde se está clasificando el tipo de un tumor, es muy importante tener buenos resultados para evitar falsos positivos que, en este caso, es crucial.

6.2.2.1. Retrospectiva Sprint 1

6.2.2.1.1. Lo que se hizo bien:

- Conversación fluida con el cliente para comunicar inconvenientes.
- Buen abordaje y solución de problemas presentados.
- Dedicación en la entrega satisfactoria de este scrum.

6.2.2.1.2. Áreas de Mejora:

Organización de tiempo, si bien es cierto se entregó a tiempo, se pudo mejorar la gestión de este para no llevar prisas con el desarrollo.

6.2.2.1.3. Próximos Pasos:

Tener una nueva reunión con el PO para mostrar los resultados obtenidos, comentar las dificultades y trazar el camino hacia las siguientes tareas.

6.2.3. Sprint 2

Tabla 11. Informe Sprint 2. Elaboración propia.

Nombre: Sprint 2			
Inicio: 11/03	Fin: 22/03	Número de días: 11 días	Número de PH: 22 PH
Informe sobre el Sprint: Se llevó a cabo la tercera reunión (08/03) con el PO, mostrándole todos los avances obtenidos ya que se superaron todas las tareas propuestas, logrando así a poder crear un dataset sintético en base del original elegido en la reunión previa.			
Tareas superadas del anterior Sprint: <ul style="list-style-type: none"> • [US_2] Implementación de la GAN. • [US_2.1] Limpieza de dataset. • [US_2.2] Aplicación de codificadores. • [US_2.3] Entrenamiento de la GAN. • [US_2.4] Generación y exportación de nuevo dataset. • [US_3] Elección de métricas. 			
Tareas planificadas para el Sprint actual: <ul style="list-style-type: none"> • [US_4] Implementación de EDA • [US_4.1] Implementación de EDA pre-GAN (dataset original) • [US_4.2] Implementación de EDA post GAN (dataset sintético) • [US_5] Comparación de dataset original y sintético. • [US_6] Preparación de datos para aplicación de IA. • [US_6.1] Separación de variables features y target para ML. • [US_6.2] Separación de variables train, validation y test para ML. • [US_6.3] Normalización de variables. • [US_7] Elección e implementación de modelos ML. • [US_7.1] Entrenamiento y evaluación del modelo 1 (Regresión Logística) • [US_7.2] Entrenamiento y evaluación del modelo 2 (Gradient Boosting) • [US_7.3] Entrenamiento y evaluación del modelo 3 (Random Forest) • [US_7.4] Comparación de resultados entre los modelos ML. 			

Especificación de Requisitos:

- [US_4.1] Implementación de EDA pre-GAN (dataset original): este requisito consta de la implementación de diferentes técnicas de análisis exploratorio del dataset para poder conocer más detalles sobre cuál es su comportamiento y comprender su estructura.

- [US_4.2] Implementación de EDA post GAN (dataset sintético): este requisito al igual que el anterior se trata de un análisis exploratorio del dataset para poder conocer su comportamiento y estructura, y también verificar que los nuevos valores sintéticos sean representativos del dataset original. Anteriormente se planteó un pequeño ejemplo sobre el tamaño del tumor y la posibilidad de un valor atípico, pues este requisito es para una confirmación fundamentada de ello.
- [US_5] Comparación de dataset original y sintético: este requisito es una mezcla de los dos anteriores y lo que se pide es comparar los dos datasets para garantizar su similitud en correlación, estadísticas y otros.
- [US_6.1] Separación de variables features y target para ML: implica la separación de los conjuntos de datos en variables de entrenamiento y la variable objetivo, para luego hacer uso de éste en los modelos.
- [US_6.2] Separación de variables train, validation y test para ML: al igual que el anterior requisito, implica dividir el conjunto de datos en conjuntos de entrenamiento, validación y prueba, para luego en los modelos evaluar el rendimiento de manera efectiva.
- [US_6.3] Normalización de variables: este requisito es necesario para garantizar que todas las características tengan una escala similar y por ello lo aplicamos, ya que, esto va a facilitar el entrenamiento de los modelos.
- [US_7.1] Entrenamiento y evaluación del modelo 1 (Regresión Logística): este requisito implica el proceso de entrenamiento y evaluación de un modelo de regresión logística utilizando el dataset sintético.
- [US_7.2] Entrenamiento y evaluación del modelo 2 (Gradient Boosting): este requisito implica el proceso de entrenamiento y evaluación de un modelo de Gradient Boosting utilizando el dataset sintético.
- [US_7.3] Entrenamiento y evaluación del modelo 3 (Random Forest): este requisito implica el proceso de entrenamiento y evaluación de un modelo de Random Forest utilizando el dataset sintético.
- [US_7.4] Comparación de resultados entre los modelos ML: este requisito se implementa para determinar cuál modelo proporciona un mejor rendimiento bajo la evaluación de las métricas seleccionadas.

Sobre el Sprint:

En la tercera reunión que dio inicio al sprint 2 se comentaron todos lo comentado en el apartado anterior, teniendo al PO al tanto de todo lo sucedido se decidió avanzar a las siguientes tareas [US_4], [US_5], [US_6] y [US_7].

Este sprint se nos hizo sencillo por la experiencia previa que se tenía sobre la implementación de estas herramientas, se ahondará más a fondo en los siguientes capítulos respecto a la implementación de la EDA, pero a modo de resumen, se usaron gráficas como mapas de correlación, gráficos de barras, entre otros.

Por otro lado, la comparación entre los datasets se realizó con el propósito de confirmar que los datos sintéticos y los originales no tenían mucha variación para evitar problemas por datos mal generados en el futuro y evitar retrasos.

Asimismo, habiendo asegurado que tenemos un dataset listo para ser utilizado, se comenzó a preparar para la implementación de modelos IA, por ejemplo, separando los features, el target, normalizando y más, tareas incluidas dentro del [US_6].

Resumen [US_4.1] EDA Pre-GAN:

Lo primero que se hizo fue la limpieza del dataset original realizada y explicada en el anterior sprint, pero se complementa con este sprint. Luego se separaron las variables de análisis numéricas y categóricas.

En las variables numéricas se guardan atributos como la edad, el tamaño del tumor y los nodos invadidos, y para poder estudiar mejor estos datos y su comportamiento, se realizaron gráficas que se explican a continuación. (Se mostrará solo un gráfico por tipo, para la explicación de todos los gráficos generados se encuentran en el notebook).

- **Histograma:**

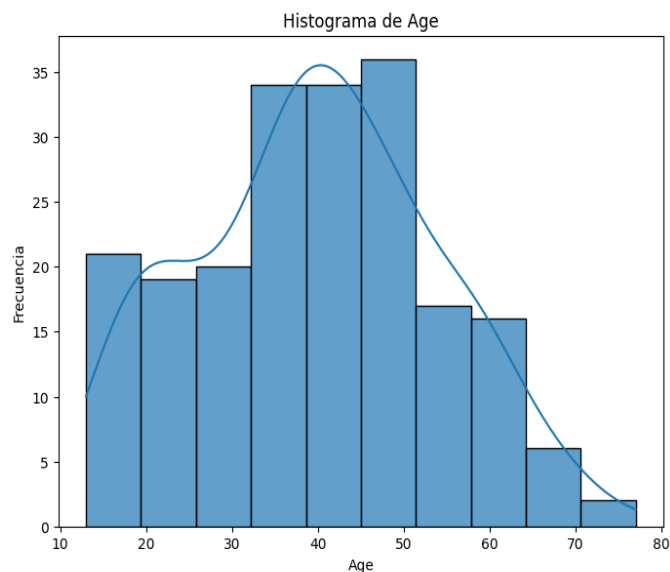


Ilustración 16. Histograma de Edad. Código Google Colab.

Este gráfico muestra la distribución de edades en el dataset. Observamos que la mayoría de las personas tienen entre 40 y 50 años, siendo este el rango de edad más común.

- **Pairplot:**

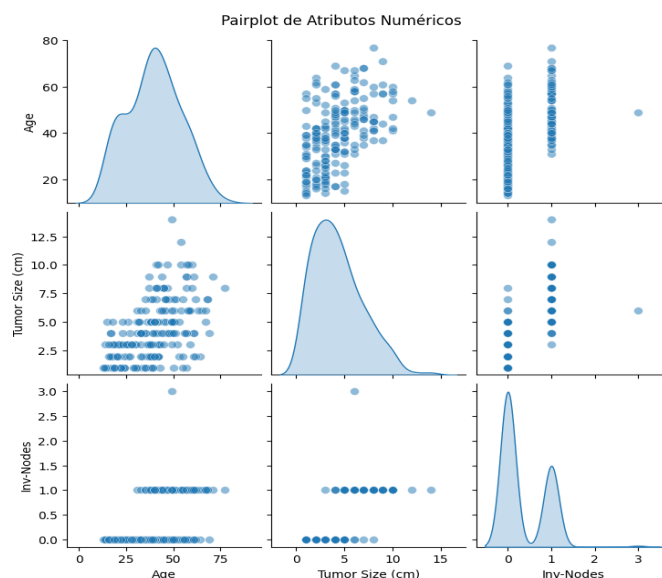


Ilustración 17. Pairplot de Atributos Numéricos. Código Google Colab.

Este gráfico ilustra las relaciones entre múltiples variables numéricas en el dataset.

En la diagonal (distribuciones univariadas), la primera fila y columna muestran la distribución de la edad, la cual se concentra alrededor de los 50 años. En la segunda fila y columna, se exhibe la distribución del tamaño del tumor (cm), destacando una concentración de valores más pequeños, con algunos valores que se extienden hasta alrededor de 12 cm. Por último, la tercera fila y columna representan la distribución de los nodos invadidos, siendo la mayoría de los valores iguales a 0.

Fuera de la diagonal (relaciones bivariadas), se observa lo siguiente:

- Entre Edad y Tamaño del Tumor: Los puntos muestran una distribución dispersa sin una clara relación entre la edad y el tamaño del tumor.
- Entre Edad y Nodos Invadidos: Similarmente, no parece haber una relación lineal clara entre la edad y la cantidad de nodos invadidos.
- Entre Tamaño del Tumor y Nodos Invadidos: Los datos son algo dispersos y no se observa una correlación evidente. Aunque los casos con mayor cantidad de nodos invadidos parecen tener, en su mayoría, tumores de tamaño pequeño a mediano.
- Los gráficos de dispersión entre Tamaño del Tumor y Edad, Nodos Invadidos y Edad, y Nodos Invadidos y Tamaño del Tumor repiten las combinaciones mencionadas anteriormente desde diferentes perspectivas.
- **Gráfico de dispersión entre Tumor y Edad:**

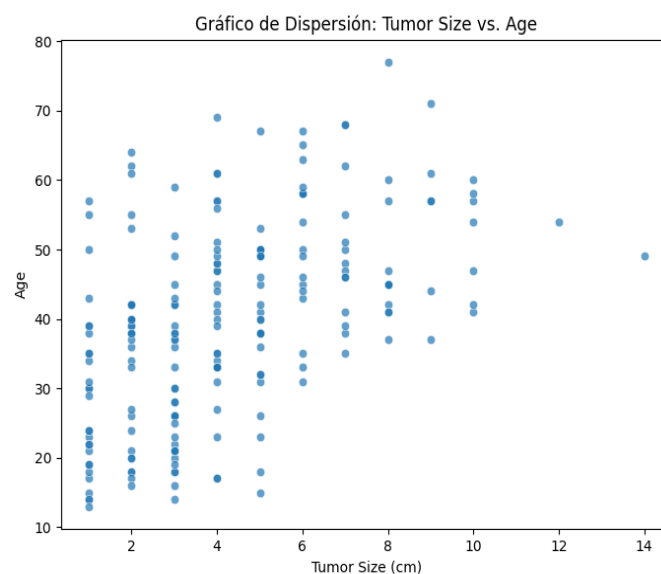


Ilustración 18. Gráfico de Dispersión. Código Google Colab.

Este gráfico de dispersión muestra la relación entre el tamaño del tumor y la edad de los pacientes.

Se observa que la distribución de los puntos muestra una dispersión amplia sin una tendencia clara, lo que sugiere una falta de relación directa entre la edad y el tamaño del tumor. Los pacientes representados en el gráfico abarcan un rango de edad desde los 20 hasta los 70 años aproximadamente, con tamaños de tumor variando desde cerca de 1 cm hasta aproximadamente 14 cm. A pesar de la dispersión de los datos, se observa una concentración moderada de casos con tumores entre 2 y 6 cm en un amplio rango de edades.

Luego se separaron las variables categóricas y se realizó su respectivo análisis, los atributos son: Breast, Cuadrante de mama y Resultado del Diagnóstico.

- **Distribución de casos de cáncer de mama por cuadrante:**

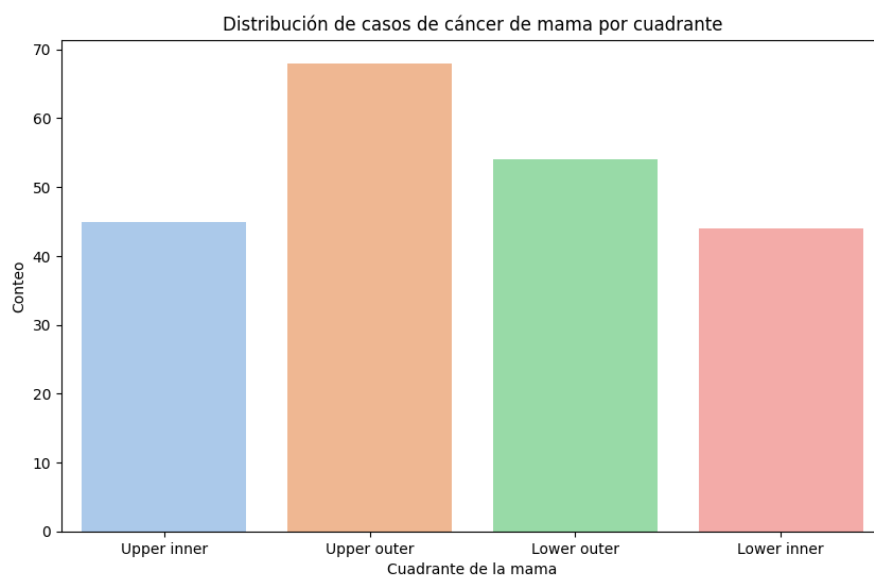


Ilustración 19. Frecuencia de Breast Quadrant. Código Google Colab.

La ilustración 19, muestra un gráfico de barras que contiene la cantidad de casos de cáncer de mama distribuidos entre los cuatro cuadrantes de la mama: superior interno, superior externo, inferior externo e inferior interno.

Observamos que el cuadrante superior externo tiene más casos en comparación con los otros cuadrantes, mientras que los cuadrantes inferior externo e inferior interno presentan menos casos. Esto podría indicar que ciertas regiones de la

mama son más propensas a desarrollar cáncer, siendo el cuadrante superior externo el más afectado.

- **Distribución de resultados de diagnóstico por cuadrante de la mama:**

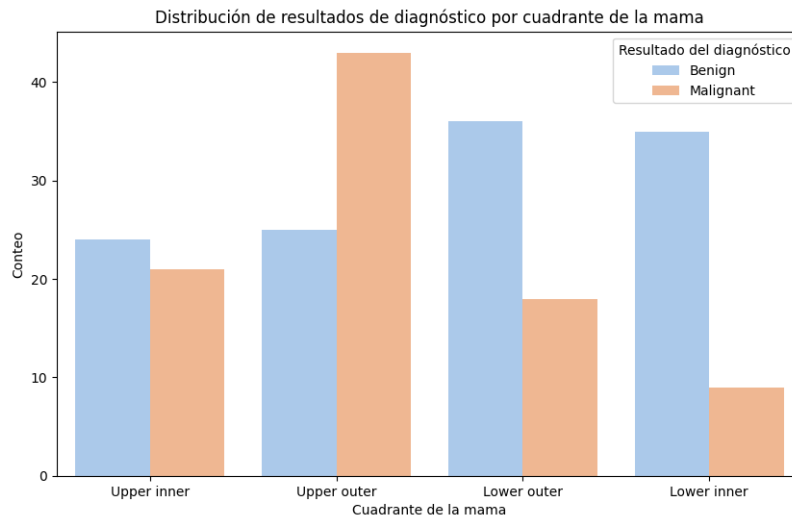


Ilustración 20. Distribución de Breast Quadrant por diagnóstico. Código Google Colab.

Este gráfico de barras compara la cantidad de diagnósticos benignos y malignos por cuadrante de la mama:

En todos los cuadrantes, se observa una considerable cantidad de diagnósticos malignos, siendo el cuadrante superior externo el que muestra la mayor cantidad tanto de casos benignos como malignos. Los casos en el cuadrante superior interno están casi igualmente divididos entre benignos y malignos. Esto sugiere que mientras ciertos cuadrantes pueden tener más incidencias de tumores, no necesariamente significa que resulten en diagnósticos más malignos.

- **Gráfico de dispersión: Edad vs Cuadrante de la mama**

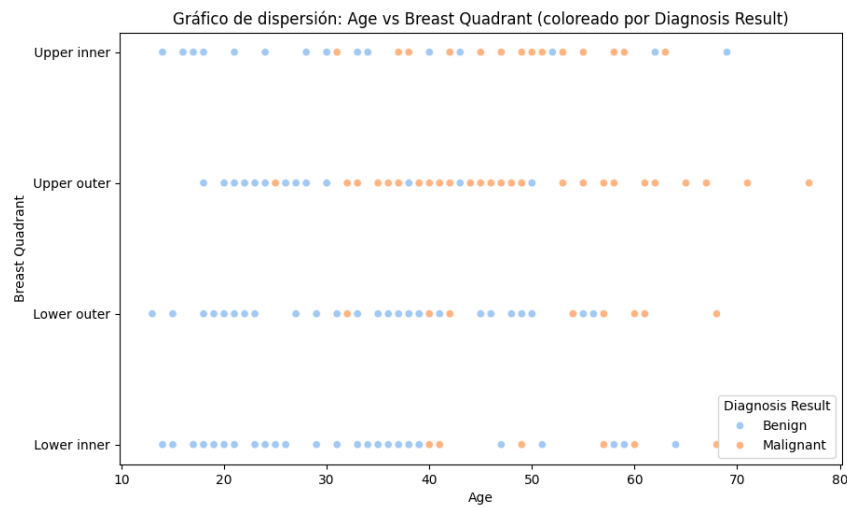


Ilustración 21. Gráfico de dispersión Código Google Colab.

Este gráfico de dispersión muestra la relación entre la edad de los pacientes, el cuadrante de la mama donde se detectó el cáncer, y el resultado del diagnóstico (benigno o maligno):

Los puntos están distribuidos a lo largo de los rangos de edad para cada cuadrante, con colores que distinguen entre resultados benignos y malignos.

No parece haber una tendencia clara que relacione la edad con la probabilidad de que el cáncer sea benigno o maligno en ningún cuadrante específico.

Resumen [US_4.2] EDA Post-GAN:

- **Análisis de variables:**

Luego de la creación y la exportación del nuevo dataset se realiza un análisis exploratorio de los datos para asegurar que los nuevos datos son representantes del original. Primero se mostrará el detalle del nuevo dataset y luego las gráficas generadas.

```

Data columns (total 14 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Year                                       10000 non-null  float64
1   Age                                       10000 non-null  float64
2   Menopause                                10000 non-null  float64
3   Tumor Size (cm)                         10000 non-null  float64
4   Breast                                   10000 non-null  float64
5   Metastasis                              10000 non-null  float64
6   History                                  10000 non-null  float64
7   Diagnosis Result                         10000 non-null  float64
8   Breast Quadrant_Lower inner             10000 non-null  float64
9   Breast Quadrant_Lower outer             10000 non-null  float64
10  Breast Quadrant_Upper inner             10000 non-null  float64
11  Breast Quadrant_Upper outer             10000 non-null  float64
12  Inv-Nodes_0.0                           10000 non-null  float64
13  Inv-Nodes_1.0                           10000 non-null  float64
dtypes: float64(14)

```

Ilustración 22. Detalle dataset sintético. Código Google Colab.

El dataset sintético consta de 15 columnas incluyendo el año que posteriormente, para la para el entrenamiento de los modelos de aprendizaje profundo, es cierto que antes eran 16, pero a raíz del EDA Pre-GAN, se descubrió que había un dato que se era errado, ya que, al observar la dispersión, no quedaba claro cómo había uno de esos datos, es el Inv-Nodes_3.0 (ya eliminado en la Ilustración 22), este es un fallo de la recopilación original de datos.

- Mapa de correlación:

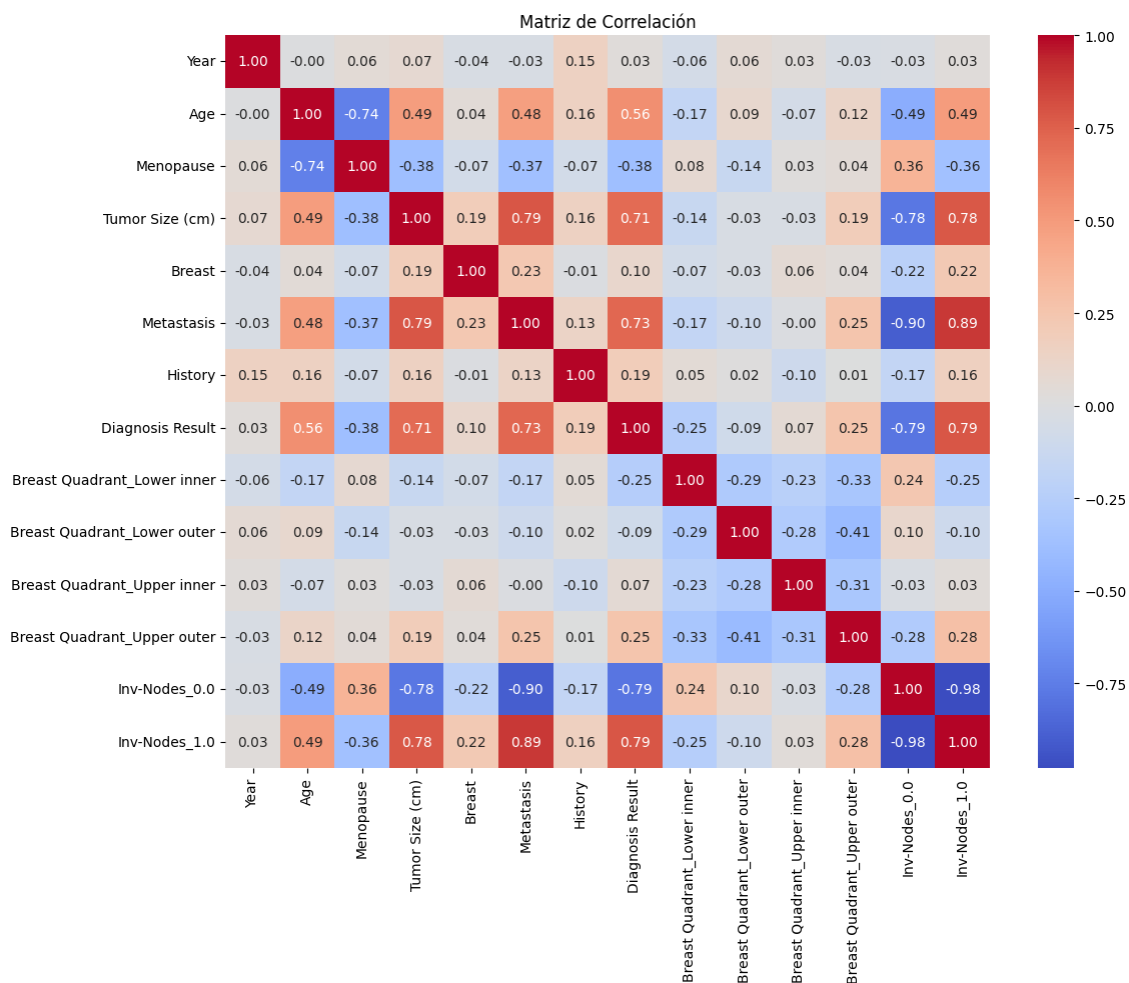


Ilustración 23. Matriz de Correlación. Código Google Colab

Esta matriz de correlación muestra las relaciones entre diferentes variables en el dataset, las celdas de la matriz están coloreadas según el valor de correlación entre cada par de variables: los valores cercanos a 1.00 (color rojo) indican una correlación positiva fuerte, mientras que los valores cercanos a -1.00 (color azul) indican una correlación negativa fuerte. Los valores cercanos a 0 (color blanco) indican falta de correlación.

Algunas de las correlaciones más destacadas son:

- Edad y Tamaño del Tumor: Existe una correlación moderada positiva (0.49) entre la edad y el tamaño del tumor, lo que sugiere que a mayor edad, el tamaño del tumor tiende a ser mayor.

- Edad y Menopausia: Hay una fuerte correlación negativa (-0.74) entre la edad y la menopausia, indicando que a medida que aumenta la edad, es más probable que las mujeres estén en la menopausia.
- Tamaño del Tumor y Metástasis: Existe una alta correlación positiva (0.79) entre el tamaño del tumor y la metástasis, sugiriendo que los tumores más grandes están más asociados con la presencia de metástasis.
- Resultado del Diagnóstico y Tamaño del Tumor: La correlación positiva (0.71) entre el tamaño del tumor y el resultado del diagnóstico indica que los tumores más grandes tienden a tener un diagnóstico más severo.
- Resultado del Diagnóstico y Metástasis: Hay una fuerte correlación positiva (0.73) entre el resultado del diagnóstico y la metástasis, lo cual indica que la presencia de metástasis está asociada con peores resultados en el diagnóstico.
- Edad e Inv-Nodos_0.0: Existe una correlación negativa moderada (-0.49) entre la edad y la ausencia de invasión de nodos linfáticos (Inv-Nodos_0.0), lo que sugiere que a mayor edad, es menos probable que no haya invasión de nodos linfáticos.
- Edad e Inv-Nodos_1.0: Hay una correlación positiva moderada (0.49) entre la edad y la presencia de invasión de nodos linfáticos (Inv-Nodos_1.0), indicando que, a mayor edad, es más probable que haya invasión de nodos linfáticos.
- Metástasis e Inv-Nodos_0.0: Existe una fuerte correlación negativa (-0.90) entre la metástasis y la ausencia de invasión de nodos linfáticos, indicando que, si no hay invasión de nodos linfáticos, es muy probable que no haya metástasis.
- Metástasis e Inv-Nodos_1.0: Hay una fuerte correlación positiva (0.89) entre la metástasis y la presencia de invasión de nodos linfáticos, sugiriendo que la invasión de nodos linfáticos está fuertemente asociada con la presencia de metástasis.

En resumen, las correlaciones más fuertes y significativas se observan entre la edad y la menopausia, el tamaño del tumor y la metástasis, y entre la metástasis y la invasión de nodos linfáticos, así como entre el tamaño del tumor y el

resultado del diagnóstico. Estas relaciones destacan la importancia de estos factores en el diagnóstico y la progresión del cáncer de mama.

- **Gráficos de barras:**

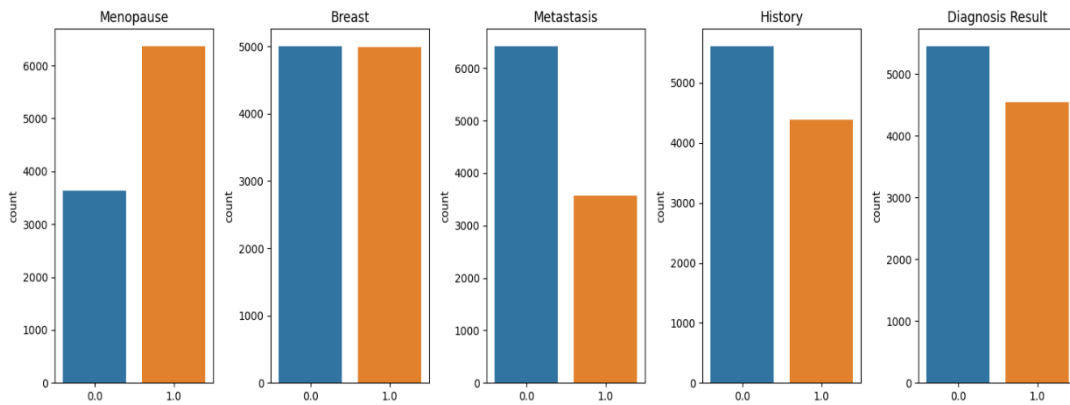


Ilustración 24. Gráficos de barras de variables. Código Google Colab.

La ilustración 24 es una serie de gráficos de barras que representan las cuentas de las variables categóricas del dataset. Cada gráfico de barras tiene dos categorías, 0.0 (no o izquierda) y 1.0 (sí o derecha).

El eje Y muestra la cantidad de casos (count) en cada categoría de la variable respectiva.

Aquí están las observaciones clave para cada variable:

- **Menopausia:** Hay más casos de mujeres que están en menopausia (1.0) que de mujeres que no lo están (0.0). Esto puede reflejar la distribución de edad de la población del dataset.
- **Breast (Pecho):** Las barras son casi iguales, indicando una distribución uniforme entre los dos lados. Esto sugiere que el cáncer de mama se distribuye casi equitativamente entre la mama izquierda (0.0) y la derecha (1.0).
- **Metastasis:** Claramente hay menos casos con metástasis (1.0) comparados con los sin metástasis (0.0). Esta es una observación importante que resalta la gravedad y progresión de la enfermedad en el dataset, ya que la mayoría de los casos no han desarrollado metástasis.
- **History (Historia):** Hay más casos sin un historial de cáncer de mama (0.0) que los que sí tienen (1.0). Esto podría indicar que, en la mayoría de los

casos, el cáncer de mama ocurre de nuevo, sin una historia previa personal o familiar de la enfermedad.

- **Diagnosis Result (Resultado del Diagnóstico):** Hay más diagnósticos benignos (0.0) que malignos (1.0), lo cual es positivo desde el punto de vista de la salud. Esto sugiere que una mayor proporción de los casos evaluados no resultan en un diagnóstico de cáncer.

Estas observaciones proporcionan una visión general de las distribuciones de las características categóricas en el dataset y pueden ser útiles para entender la prevalencia y la distribución de diferentes factores relacionados con el cáncer de mama.

- **Gráfico de Dispersión:**

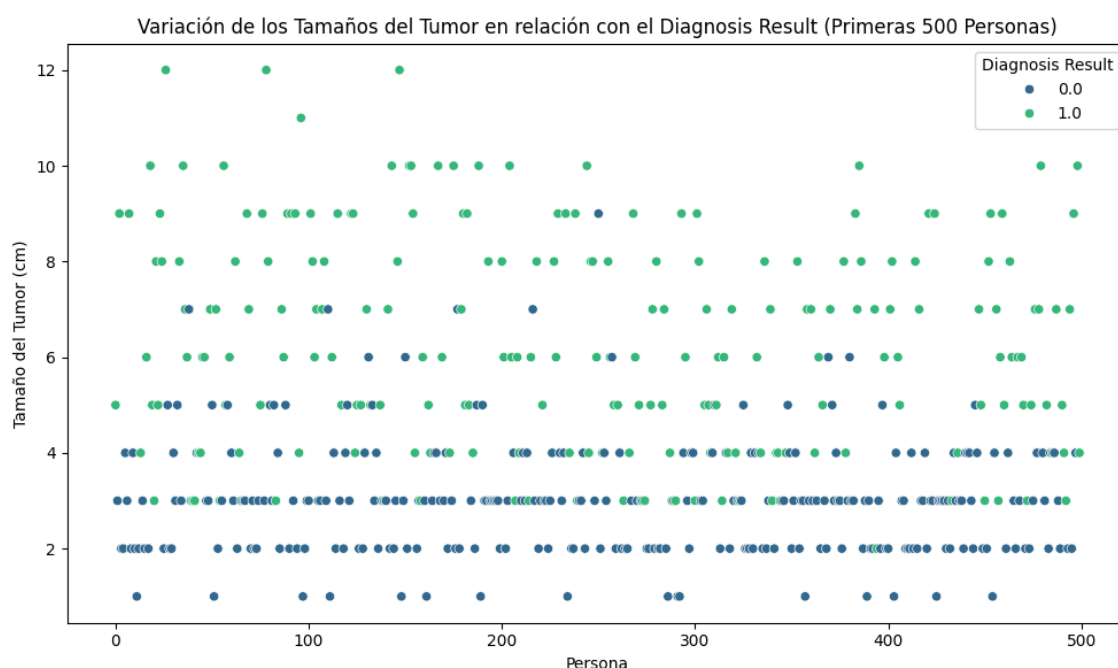


Ilustración 25. Variación de tamaños del tumor en relación con el diagnóstico. Código Google Colab.

La Ilustración 25 es un diagrama de dispersión que muestra la variación de los tamaños de tumor en relación con el resultado del diagnóstico para las primeras 500 personas del dataset.

Los puntos azules representan diagnósticos benignos (0.0) y los puntos verdes representan diagnósticos malignos (1.0). Algunas observaciones clave del gráfico son las siguientes:

En cuanto a la distribución de tamaños de tumor:

- Los tumores diagnosticados como benignos (puntos azules) tienden a ser más pequeños, agrupándose mayoritariamente en el rango de 2 a 4 cm.
- Los tumores diagnosticados como malignos (puntos verdes) presentan una distribución más grande de tamaños, abarcando desde aproximadamente 2 cm hasta más de 10 cm.

En relación a la relación entre tamaño y diagnóstico:

- Existe una tendencia visual de que los tumores más grandes están asociados con diagnósticos malignos, mientras que los tumores más pequeños suelen ser benignos.

Resumen [US_5] Comparación de dataset original y sintético:

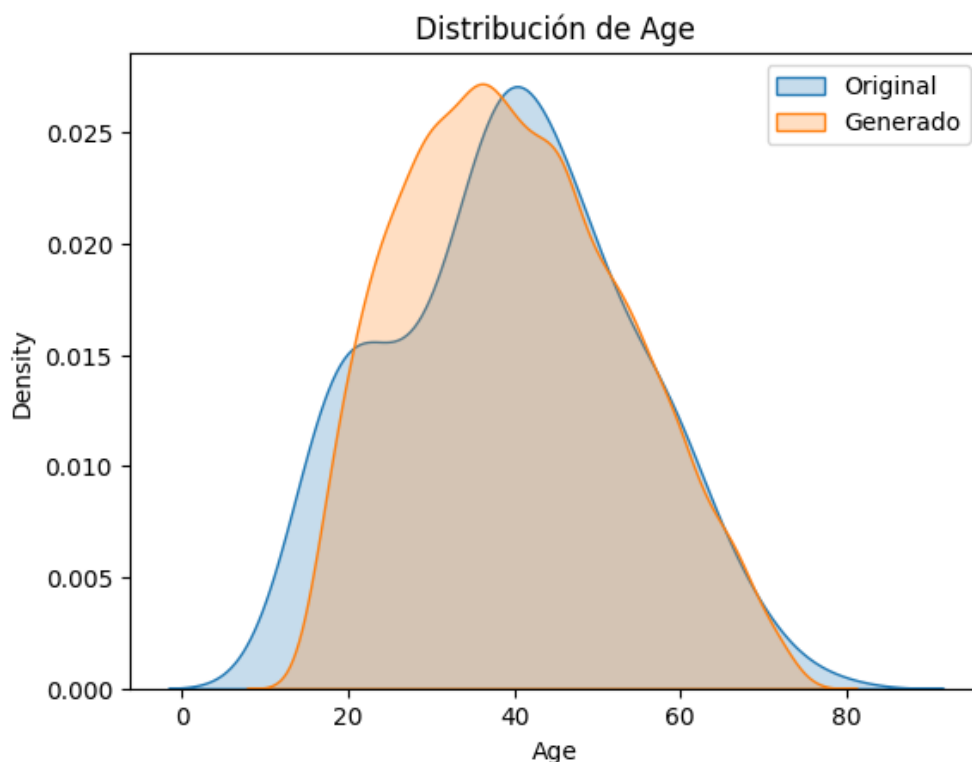


Ilustración 26. Distribución de la edad entre dataset original y sintético. Código Google Colab.

El gráfico compara las distribuciones de edad entre el dataset original y el sintético.

Se observa que la curva azul representa la distribución original de las edades, con un pico alrededor de los 60 años y otro más hacia la derecha, indicando una mayor concentración de individuos en edades avanzadas.

Por otro lado, la curva naranja, que representa la distribución generada, también muestra un pico cerca de los 60 años, pero con una distribución más amplia en comparación con la original.

Finalmente, la comparación de ambas distribuciones revela que tienen sus picos en la misma región de edad, lo que sugiere que el modelo generado está capturando con precisión el centro de la distribución de edad.

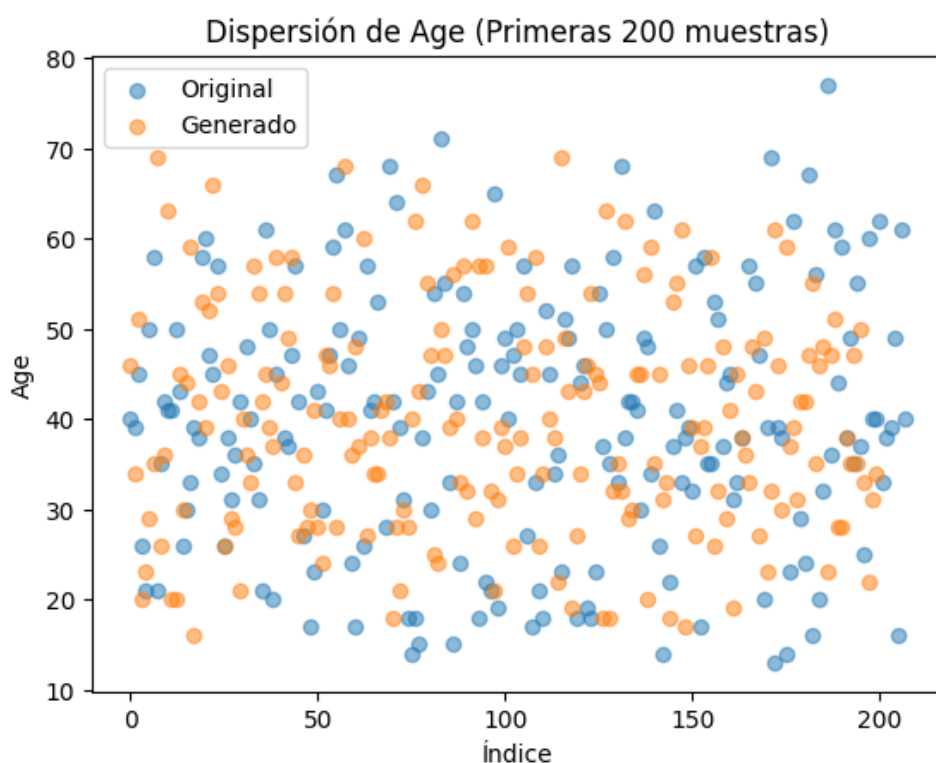


Ilustración 27. Dispersión de la edad entre dataset original y sintético. Código Google Colab.

Este gráfico compara la distribución de las edades entre los datos originales y los generados para las primeras 200 muestras.

Se observa que el rango de edad abarcado por ambos conjuntos va desde los 20 hasta los 70 años, con una concentración notable en el intervalo de 40 a 60 años. Este patrón sugiere similitudes demográficas entre los datos originales y los generados.

En cuanto a la distribución, no se observa un patrón discernible de agrupación o secuencia en las edades a lo largo del índice. Esto sugiere que las edades están probablemente distribuidas al azar entre las muestras. Además, la dispersión de los datos generados refleja de cerca la de los originales, lo que sugiere una buena representación por parte del dataset sintético.

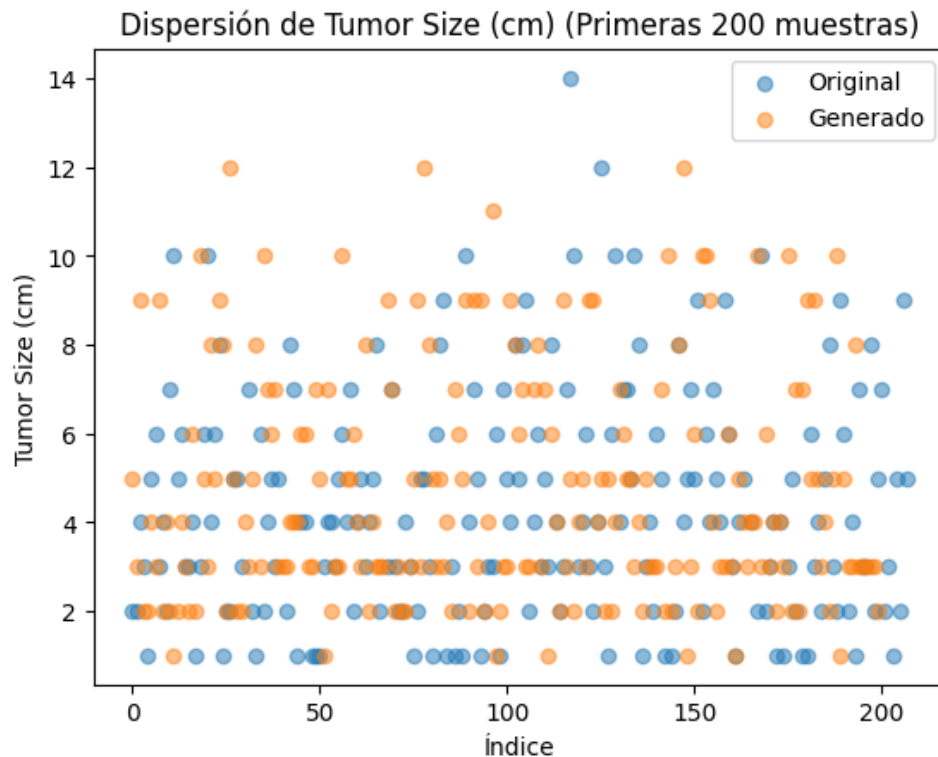


Ilustración 28. Dispersión de Tumor Size (cm). Código Google Colab

Este gráfico muestra la comparación del tamaño de los tumores en centímetros entre los datos originales y los generados.

Se observa que el rango de tamaño del tumor en ambos conjuntos de datos abarca desde aproximadamente 2 cm hasta un poco más de 12 cm, con la mayoría de los puntos concentrados entre 2 cm y 8 cm.

En cuanto a la distribución de los tamaños, se observa una variedad considerable a lo largo del índice de muestras. Los datos generados reflejan esta distribución con alta fidelidad, lo que sugiere que el modelo puede simular con precisión la variabilidad observada en el tamaño del tumor en los datos originales.

Resumen [US_6.1, 2 y 3] Separación de variables features, target, validation, train y test para ML y normalización:

Inicialmente, se definieron las características “features” (X) y el objetivo “target” (y) del conjunto de datos, donde X tendrá 10000 muestras con 13 características cada una, mientras que y tendrá 10000 valores objetivo-asociados. Luego, se eliminó la columna de año “year” del conjunto de características, lo que redujo la dimensión de X a 12 características.

```
Train - Features:  
(6000, 12)  
Train - Target:  
(6000,)  
  
Validation - Features:  
(2000, 12)  
Validation - Target:  
(2000,)  
  
Test - Features:  
(2000, 12)  
Test - Target:  
(2000,)
```

Ilustración 29. División de validación, entrenamiento y prueba. Código Google Colab

Posteriormente, se establecieron las proporciones para dividir el conjunto de datos en entrenamiento “train”, validación “validation” y prueba “test”, asignando el 60% para entrenamiento, el 20% para validación y otro 20% para prueba. Tras la división, el conjunto de entrenamiento contará con 6000 muestras tanto para características como para el objetivo. El conjunto de validación y el conjunto de prueba tendrán 2000 muestras cada uno, manteniendo las mismas dimensiones de características y objetivos.

Finalmente, se normalizaron los conjuntos de datos de validación y prueba utilizando un objeto StandardScaler ajustado a los datos de entrenamiento (X_train). De esta manera, se logró una escala uniforme de las características para preparar los datos para el modelado de ML.

6.2.3.1. Modelos seleccionados ML

Para esta parte del proyecto se eligieron los modelos de Regresión Logística, Gradient Boosting y Random Forest.

- **Regresión Logística:** Este modelo evalúa la relación entre una o más variables independientes una variable categórica binaria, usa la función logística para calcular la probabilidad de que una instancia pertenezca a cierta categoría específica. Es un modelo adecuado por su capacidad para identificar cuán probable es que suceda algo dependiendo de lo que está bajo su conocimiento. Es sencillo de implementar y fácil de interpretar.
- **Gradient Boosting:** Este modelo trabaja mediante el aprendizaje de errores, es decir se crea un modelo y realiza una predicción inicial, y luego construye otro modelo buscando corregir los errores del anterior para mejorar las predicciones gradualmente. Es un gran modelo para implementar porque puede capturar relaciones complejas en los datos.
- **Random Forest:** Este modelo está compuesto por árboles que se entrenan con una muestra de datos aleatoria y utiliza un subconjunto también aleatorio para realizar las predicciones. Al final, todos los árboles se promedian y así se obtiene la predicción final. Definitivamente un modelo muy básico, conocido y sencillo de implementar que nos puede proporcionar buenos resultados.

Resumen [US_7.1] Entrenamiento y evaluación del modelo 1 (Regresión Logística):

Se configuró el modelo especificando el algoritmo de optimización 'lbfgs' y se estableció una semilla aleatoria.

Se procedió a definir los hiper parámetros a explorar mediante la técnica de búsqueda de hiper parámetros. Se consideraron diferentes valores para el parámetro de regularización 'C' y se limitó la penalización al método 'l2'.

La búsqueda de hiper parámetros se llevó a cabo utilizando la validación cruzada con tres pliegues para evaluar la capacidad de generalización del modelo. La

métrica de evaluación seleccionada fue la precisión ('accuracy'), que proporciona una medida general del rendimiento del modelo.

Una vez completada la búsqueda de hiper parámetros, se identificó el mejor modelo encontrado, el cual se utilizó para realizar predicciones en los conjuntos de datos normalizados de validación y prueba.

Se evaluó el rendimiento del mejor modelo tanto en los datos de validación como en los datos de prueba, calculando diversas métricas de evaluación, incluyendo precisión, recall, F1-score y AUC.

Finalmente, se imprimieron las métricas de evaluación obtenidas en los datos de validación y prueba para su análisis y comparación

Tabla 12. Tabla comparación de la evaluación de los datos de Regresión Logística. Elaboración propia.

	Validación	Prueba
Precisión	0.9272	0.9396
Recall	0.8712	0.8668
F1-Score	0.8983	0.9017
AUC	0.9080	0.9098

Resumen [US_7.2] Entrenamiento y evaluación del modelo 2 (Gradient Boosting):

Este modelo se configuró inicialmente sin especificar ningún hiper parámetro adicional y se estableció una semilla aleatoria.

Luego, se definieron de hiper parámetros a explorar mediante la técnica de búsqueda de hiper parámetros. Se consideraron diferentes valores para tres parámetros clave: el número de estimadores ('n_estimators'), la tasa de aprendizaje ('learning_rate') y la profundidad máxima del árbol ('max_depth').

La búsqueda de hiper parámetros se llevó a cabo utilizando la validación cruzada con tres pliegues para evaluar el rendimiento del modelo en distintas configuraciones de hiper parámetros. Se utilizó la métrica de precisión ('accuracy') como medida de evaluación.

Tras completar la búsqueda de hiper parámetros, se identificó el mejor modelo encontrado, el cual se utilizó para realizar predicciones en los conjuntos de datos normalizados de validación y prueba.

El rendimiento del mejor modelo fue evaluado en los datos de validación y prueba, calculando diversas métricas de evaluación, incluyendo precisión, recall, F1-score y AUC.

Finalmente, se imprimieron las métricas de evaluación obtenidas en los datos de validación y prueba para su análisis y comparación.

Tabla 13. Tabla comparación de la evaluación de los datos de Gradient Boosting. Elaboración propia.

	Validación	Prueba
Precisión	0.9272	0.9396
Recall	0.8712	0.8668
F1-Score	0.8983	0.9017
AUC	0.9080	0.9098

Resumen [US_7.3] Entrenamiento y evaluación del modelo 3 (Random Forest):

Este modelo se configuró inicialmente sin especificar ningún hiper parámetro adicional y se estableció una semilla aleatoria.

A continuación, se definieron los hiper parámetros a explorar mediante la técnica de búsqueda de hiper parámetros. Se consideraron diferentes valores para cuatro parámetros importantes: el número de estimadores ('n_estimators'), la profundidad máxima del árbol ('max_depth'), el número mínimo de muestras necesarias para dividir un nodo interno ('min_samples_split') y el número mínimo de muestras necesarias para cada hoja ('min_samples_leaf').

La búsqueda de hiper parámetros se llevó a cabo utilizando la validación cruzada con tres pliegues para evaluar el rendimiento del modelo en diversas configuraciones de hiper parámetros. Se utilizó la métrica de precisión ('accuracy') como medida de evaluación.

Una vez completada la búsqueda de hiper parámetros, se identificó el mejor modelo encontrado, el cual se utilizó para realizar predicciones en los conjuntos de datos normalizados de validación y prueba.

El rendimiento del mejor modelo fue evaluado en los datos de validación y prueba, calculando diversas métricas de evaluación, incluyendo precisión, recall, F1-score y AUC.

Finalmente, las métricas de evaluación obtenidas en los datos de validación y prueba para su análisis y comparación.

Tabla 14. Tabla comparación de la evaluación de los datos de Random Forest. Elaboración propia.

	Validación	Prueba
Precisión	0.9288	0.9433
Recall	0.8622	0.8537
F1-Score	0.8943	0.8962
AUC	0.9044	0.9051

Sobre los hiper parámetros de los 3 modelos:

- Regresión Logística:
 - C: Parámetro de regularización que controla la inversa de la fuerza de regularización. Valores más pequeños significan una regularización más fuerte.
 - penalty: Tipo de penalización a utilizar. En este caso, se utilizó 'l2' que se refiere a la penalización de la norma L2.
- Gradient Boosting:
 - n_estimators: El número de árboles de decisión que se utilizarán en el conjunto. Cuantos más árboles se agreguen, más robusto será el modelo, pero también aumenta el riesgo de sobreajuste.
 - learning_rate: Tasa de aprendizaje que controla la contribución de cada árbol en el conjunto. Valores más pequeños hacen que el modelo sea más robusto al sobreajuste, pero también pueden requerir más árboles.

- max_depth: La profundidad máxima de cada árbol de decisión en el conjunto. Controla la complejidad del modelo y su capacidad para ajustarse a los datos.
- Random Forest:
 - n_estimators: El número de árboles de decisión que se utilizarán en el conjunto.
 - max_depth: La profundidad máxima de cada árbol de decisión en el conjunto.
 - min_samples_split: El número mínimo de muestras necesarias para dividir un nodo interno.
 - min_samples_leaf: El número mínimo de muestras requeridas para que un nodo sea considerado una hoja.

Resumen [US_7.4] Comparación de resultados entre los modelos ML:

Tabla 15. Comparación de evaluación modelos ML. Elaboración propia.

Modelo	Precisión	Recall	F1-Score	AUC
Regresión Logística	0.9396	0.8668	0.9017	0.9098
Gradient Boosting	0.9305	0.8919	0.9108	0.9178
Random Forest	0.9433	0.8537	0.8962	0.9051

- Mejor precisión: El Random Forest presenta la precisión más alta con un valor de 0.9433, lo que significa que es el modelo con la mayor proporción de predicciones positivas correctas. Esto implica que clasifica correctamente una observación como positiva cuando realmente lo es en el conjunto de datos de prueba.
- Mejor recall: Aunque la Regresión Logística tiene un recall de 0.8668, el Gradient Boosting sigue siendo el modelo con el recall más alto (0.8919). Esto indica que el Gradient Boosting es el mejor en identificar los casos positivos reales en el conjunto de datos de prueba, lo que es crucial en el contexto de la detección temprana del cáncer de mama.

- Mejor F1-Score: El Gradient Boosting tiene el F1-Score más alto con un valor de 0.9108, lo que sugiere que logra un buen equilibrio entre precisión y recall.
- Mejor AUC: El Gradient Boosting también lidera en AUC con un valor de 0.9178, lo que indica que es el mejor en clasificar los casos positivos y negativos.

Por lo tanto, si se prioriza la detección efectiva del cáncer de mama, el modelo de Gradient Boosting sería la mejor opción debido a su alto recall.

En este contexto, es fundamental maximizar el recall para asegurar la detección temprana y precisa del cáncer de mama, lo que puede mejorar significativamente las tasas de supervivencia y el resultado del tratamiento.

6.2.3.2. Retrospectiva Sprint 2

6.2.3.2.1. Lo que se hizo bien:

Se mejoró el factor de gestión de tiempo, también es cierto que fueron tareas muy pequeñas y no resultó tedioso el desarrollo.

6.2.3.2.2. Áreas de Mejora:

Se pudo abarcar un poco más de trabajo ya que en realidad hubo tiempo restante en el sprint, o se pudo realizar alguna investigación sobre alguna tarea futura.

6.2.3.2.3. Próximos Pasos:

Se debe reconocer que la línea que se está siguiendo actualmente es ideal. El próximo paso de este Sprint es comenzar con el desarrollo de los modelos DL.

6.2.4. Sprint 3

Tabla 16. Informe Sprint 3. Elaboración propia.

Nombre: Sprint 3			
Inicio: 25/03	Fin: 08/04	Número de días: 15 días	Número de PH: 23 PH
Informe sobre el Sprint: Se llevó a cabo la cuarta reunión (22/03) con el PO e informando sobre todos los avances, resultados obtenidos, decisiones tomadas y percances ocurridos. Luego de dar su visto bueno sobre el desarrollo, se dio comienzo a desarrollar las tareas siguientes.			
Tareas superadas del anterior Sprint: <ul style="list-style-type: none"> • [US_4] Implementación de EDA • [US_4.1] Implementación de EDA pre-GAN (dataset original) • [US_4.2] Implementación de EDA post GAN (dataset sintético) • [US_5] Comparación de dataset original y sintético. • [US_6] Preparación de datos para aplicación de IA. • [US_6.1] Separación de variables features y target para ML. • [US_6.2] Separación de variables train, validation y test para ML. • [US_6.3] Normalización de variables. • [US_7] Elección e implementación de modelos ML. • [US_7.1] Entrenamiento y evaluación del modelo 1 (Regresión Logística) • [US_7.2] Entrenamiento y evaluación del modelo 2 (Gradient Boosting) • [US_7.3] Entrenamiento y evaluación del modelo 3 (Random Forest) • [US_7.4] Comparación de resultados entre los modelos ML. 			
Tareas planificadas para el Sprint actual: <ul style="list-style-type: none"> • [US_8] Elección e implementación de modelos DL. • [US_8.1] Entrenamiento y evaluación del modelo 1 (ANN) • [US_8.2] Entrenamiento y evaluación del modelo 2 (MLP) • [US_8.3] Comparación entre los modelos DL. • [US_8.4] Comparación de modelos ML y DL mediante las métricas. 			

Especificación de Requisitos:

- [US_8.1] Entrenamiento y evaluación del modelo 1 (ANN): este requisito implica el proceso de entrenamiento y evaluación de un modelo de redes neuronales artificiales (ANN) utilizando el dataset sintético.

- [US_8.2] Entrenamiento y evaluación del modelo 2 (MLP): este requisito implica el proceso de entrenamiento y evaluación de un modelo de perceptrón multicapa (MLP) utilizando el dataset sintético.
- [US_8.3] Comparación entre los modelos DL: este requisito se implementa para determinar cuál modelo proporciona un mejor rendimiento bajo la evaluación de las métricas seleccionadas.
- [US_8.4] Comparación de modelos ML y DL mediante las métricas: este requisito busca comparar los resultados obtenidos de los modelos ML y DL utilizando las métricas seleccionadas para determinar cuál tipo de modelo es más efectivo.

Sobre el Sprint:

En este sprint se comenzó con la implementación de los modelos DL, luego de la cuarta reunión con el cliente donde se mostraron los modelos de ML y la aclaración de algunas dudas sobre el futuro del proyecto, se comenzó a implementar nuevas cosas. A continuación, se comentará un poco sobre cada modelo elegido, sobre su implementación y resultados obtenidos.

6.2.3.3. *Modelos seleccionados DL*

Resumen [US_8] Elección e implementación de modelos DL:

- **Redes Neuronales Artificiales (ANN):** Este modelo de aprendizaje utiliza múltiples capas de neuronas artificiales, permitiendo el aprendizaje de datos complejos. Sin embargo, es importante aclarar que las ANN incluyen capas de entrada, capas ocultas (intermedias) y una capa de salida. A través de la adaptabilidad de su arquitectura, las ANN pueden capturar patrones complicados y son flexibles para abordar una variedad de problemas.
- **Multilayer Perceptron (MLP):** El MLP es una variante de las ANN que, al igual que estas, utiliza múltiples capas de neuronas. A diferencia de las ANN, el MLP incluye explícitamente una capa de entrada, una o más capas ocultas y una capa de salida. Se destaca por su capacidad para identificar patrones complejos a través del uso de las capas ocultas. Sin embargo, para obtener buenos resultados

con MLP, es crucial controlar el sobreajuste y realizar una buena hiper parametrización.

Resumen [US_8.1] Entrenamiento y evaluación del modelo 1 (ANN):

Sobre las diferentes configuraciones:

A lo largo del proyecto, se configuraron varios modelos, y con la ayuda de KerasTuner se pudieron obtener diferentes configuraciones con resultados atractivos conforme a lo que se buscaba, un modelo eficiente y tal vez mejor rendimiento al modelo ML evaluado anteriormente.

A continuación, en la tabla 18 se muestran los resultados de evaluación de 10 modelos diferentes con las métricas seleccionadas.

Tabla 17. Comparación de evaluación de diferentes configuraciones ANN. Elaboración propia.

Conf. Modelo	Precision	Recall	F1 Score	AUC
1	0.9452	0.8668	0.9043	0.9730
2	0.9175	0.8985	0.9079	0.9752
3	0.9320	0.8974	0.9143	0.9780
4	0.9241	0.8908	0.9072	0.9770
5	0.9271	0.8886	0.9075	0.9752
6	0.9216	0.8985	0.9099	0.9777
7	0.9320	0.8974	0.9143	0.9780
8	0.9548	0.8526	0.9008	0.9739
9	0.8800	0.9300	0.9000	0.9743
10	0.9340	0.8963	0.9148	0.9778

En la tabla, la Configuración 9 se destaca como la mejor opción a pesar de no tener la mayor precisión. Sus métricas son las siguientes: precisión de 0.8800, recall de 0.9300, F1 score de 0.9000 y AUC de 0.9743. La razón principal para considerar la Configuración 9 como la mejor es su alto valor de recall (0.9300).

Un alto recall es crucial en situaciones donde es más importante minimizar los falsos negativos que los falsos positivos. Esto significa que el modelo es muy efectivo en identificar correctamente los casos positivos, lo cual es esencial en aplicaciones como la detección de enfermedades o fraudes, dónde pasar por alto un caso positivo puede tener graves consecuencias. Aunque la precisión es ligeramente menor, el balance entre el recall y las otras métricas (F1 Score y

AUC) indica que la Configuración 9 ofrece un rendimiento robusto y confiable, priorizando la identificación de todos los casos positivos posibles.

Otras configuraciones interesantes incluyen la Configuración 10, que ofrece una precisión de 0.9340, recall de 0.8963, F1 score de 0.9148 y AUC de 0.9778. Esta configuración proporciona un excelente equilibrio general, destacándose especialmente en precisión y F1 score. Las Configuraciones 3 y 7, con métricas idénticas de precisión (0.9320), recall (0.8974), F1 score (0.9143) y AUC (0.9780), también muestran un buen balance y rendimiento general. La Configuración 8, aunque tiene una alta precisión de 0.9548, presenta un recall más bajo de 0.8526, lo que reduce su efectividad general en comparación con las configuraciones mencionadas anteriormente.

Sobre el modelo elegido:

Se definió una función para construir el modelo ANN (Red Neuronal Artificial), donde se seleccionaron dinámicamente el número de capas y unidades ocultas utilizando Keras Tuner. Esta función permite ajustar hiper parámetros como el número de capas, el número de unidades en cada capa, la tasa de dropout y la tasa de aprendizaje. El modelo se compiló utilizando el optimizador Adam y la función de pérdida de entropía cruzada binaria.

Luego, se configuró la búsqueda aleatoria de hiper parámetros con RandomSearch de Keras Tuner, definiendo cuántas combinaciones de hiper parámetros probar (`max_trials=5`) y cuántas ejecuciones realizar por cada combinación (`executions_per_trial=1`). La búsqueda se llevó a cabo sobre los datos de entrenamiento y validación, con un proceso de entrenamiento de 50 épocas y un tamaño de lote de 64.

Una vez completada la búsqueda, se identificó el mejor modelo y sus hiper parámetros óptimos. Este modelo se evaluó en el conjunto de prueba, calculando varias métricas de evaluación. Además, se generaron predicciones de probabilidad sobre los datos de prueba y se ajustó el umbral de clasificación según fuera necesario. Finalmente, se calculó la matriz de confusión y el reporte de clasificación, proporcionando una visión detallada del rendimiento del modelo

ajustado. También se calculó el AUC (Área Bajo la Curva) para medir la capacidad del modelo en distinguir entre las clases.

Sobre los hiper parámetros del modelo:

- Número de capas ocultas: Se buscó un valor entre 1 y 3 capas.
- Número de unidades en cada capa: Se buscó un valor entre 32 y 256 unidades, en incrementos de 32.
- Tasa de dropout: Se buscó un valor entre 0.2 y 0.5, en incrementos de 0.1.
- Tasa de aprendizaje: Se seleccionó entre 0.01, 0.001 y 0.0001.

Estos hiper parámetros se optimizaron para maximizar la métrica de "val_recall" (recall en el conjunto de validación), asegurando que el modelo tuviera un buen desempeño en términos de la tasa de verdaderos positivos.

En resumen, el proceso de ajuste de hiper parámetros mediante Keras Tuner permitió encontrar la configuración óptima del modelo, mejorando así su capacidad de generalización y rendimiento en el conjunto de pruebas.

Resumen [US_8.2] Entrenamiento y evaluación del modelo 2 (MLP):**Sobre las diferentes configuraciones:**

Se evaluaron varias configuraciones del modelo MLP, a continuación, se muestran y comentan algunos de ellos:

Tabla 18. Comparación de evaluación de diferentes configuraciones MLP. Elaboración propia.

Conf. Modelo	Precisión	Recall	F1 Score	AUC
1	0.9460	0.8799	0.9131	0.977
2	0.9218	0.9007	0.9111	0.977
3	0.9218	0.9007	0.9111	0.977
4	0.9501	0.9773	0.9101	0.9501
5	0.8701	0.8777	0.9112	0.9534
6	0.9491	0.8755	0.9152	0.9774
7	0.9342	0.8775	0.9150	0.9473
8	0.9165	0.8688	0.9106	0.9404
9	0.9412	0.8908	0.9153	0.9772
10	0.9088	0.9138	0.9113	0.9761
11	0.8799	0.9017	0.9056	0.9329
12	0.9456	0.8723	0.9074	0.9768
13	0.9000	0.9200	0.9100	0.9775
14	0.9317	0.8941	0.9125	0.9766
15	0.9578	0.8668	0.9100	0.9768

En la tabla de métricas de evaluación del modelo, la Configuración 13 emerge como una opción destacada en términos de equilibrio general, aunque no es la más alta en todas las métricas individuales. Con una precisión de 0.9000, un recall de 0.9200, un F1 score de 0.9100 y un AUC de 0.9775, esta configuración ofrece un balance sólido entre precisión y recall. El recall de 0.9200 indica que el modelo es efectivo en identificar correctamente los casos positivos, lo cual es crucial en escenarios donde es vital minimizar los falsos negativos, como en la detección de enfermedades o fraudes.

Aunque otras configuraciones muestran métricas superiores en algunos aspectos específicos, como la Configuración 9 que tiene un recall ligeramente mayor y una precisión ligeramente menor, o la Configuración 15 con la mayor precisión, la Configuración 13 destaca por mantener un buen equilibrio en todas las métricas evaluadas. Este equilibrio sugiere que la Configuración 13 es una

opción robusta y confiable para aplicaciones donde es crucial una identificación precisa y balanceada de casos positivos.

Sobre el modelo elegido:

Se definió una función para construir el modelo MLP, en la cual se seleccionaron dinámicamente el número de capas y unidades ocultas utilizando Keras Tuner. Este modelo MLP está configurado con la capacidad de ajustar automáticamente el número de capas ocultas y el número de unidades en cada capa. Luego, se compiló el modelo utilizando el optimizador Adam y la función de pérdida de entropía cruzada binaria. Este modelo MLP se entrenó y evaluó para su capacidad de clasificación binaria.

Sobre los hiper parámetros del modelo:

- Número de capas ocultas: Se buscó un valor entre 1 y 3 capas.
- Número de unidades en cada capa: Se buscó un valor entre 32 y 256 unidades, en incrementos de 32.
- Tasa de dropout: Se buscó un valor entre 0.2 y 0.5, en incrementos de 0.1.
- Tasa de aprendizaje: Se seleccionó entre 0.01, 0.001 y 0.0001.

Además, se cargaron los datos de entrenamiento normalizados desde un archivo NumPy

Resumen [US_8.3] Comparación entre los modelos DL:*Tabla 19. Comparación de evaluación modelos DL. Elaboración propia.*

Modelo	Precisión	Recall	F1-Score	AUC
ANN	0.8800	0.9300	0.9000	0.9743
MLP	0.9000	0.9200	0.9100	0.9775

- **Mejor Precisión:** El MLP tiene la precisión más alta (0.9000), lo que significa que es el modelo con la mayor proporción de predicciones positivas correctas. En otras palabras, el MLP clasifica correctamente una observación como positiva con mayor frecuencia en el conjunto de datos de prueba.
- **Mejor Recall:** La ANN tiene el recall más alto (0.9300), lo que indica que es el modelo que mejor identifica los casos positivos reales. Esto significa que la ANN es más efectiva para detectar todas las instancias verdaderamente positivas en el conjunto de datos de prueba.
- **Mejor F1-Score:** El MLP tiene el F1-Score más alto (0.9100), lo que sugiere que logra un buen equilibrio entre precisión y recall. Este equilibrio es crucial en situaciones donde tanto la correcta identificación de positivos como la minimización de falsos positivos son importantes.
- **Mejor AUC:** El MLP tiene el AUC más alto (0.9775), lo que indica que es el modelo que mejor clasifica los casos positivos y negativos. Un AUC alto refleja una mayor capacidad del modelo para diferenciar entre las clases.

En conclusión, si se prioriza la precisión, el MLP sería la mejor opción debido a su alta precisión. Si se prioriza el Recall, la ANN podría ser preferible ya que tiene el recall más alto.

Para un equilibrio entre precisión y Recall, la MLP sería la mejor opción debido a su alto F1-Score y AUC.

En este caso, tomaremos como referencia el MLP, ya que ofrece un buen equilibrio entre precisión y recall, además de tener el mejor AUC, lo que lo hace superior en términos de clasificación general.

Resumen [US_8.4] Comparación de modelos ML y DL mediante las métricas:

Tabla 20. Comparación mejores modelos ML y DL. Elaboración propia.

Modelo	Precisión	Recall	F1-Score	AUC
Gradient Boosting	0.9305	0.8919	0.9108	0.9178
MLP	0.9000	0.9200	0.9100	0.9775

Al comparar los modelos Gradient Boosting y MLP, observamos diferencias significativas en sus métricas de evaluación. El modelo Gradient Boosting presenta una precisión de 0.9305, lo que indica una mayor proporción de predicciones positivas correctas en comparación con el MLP, que tiene una precisión de 0.9000. Sin embargo, en términos de recall, el MLP tiene ventaja con un valor de 0.9200, en contraste con el 0.8919 del Gradient Boosting, lo que sugiere que el MLP es un poco más efectivo en la identificación de casos positivos reales.

El F1-Score, que equilibra precisión y recall, es casi equivalente para ambos modelos: 0.9108 para Gradient Boosting y 0.9100 para el MLP. Esta similitud indica que ambos modelos logran un equilibrio similar entre precisión y recall. Sin embargo, la métrica AUC, que refleja la capacidad del modelo para distinguir entre clases positivas y negativas, es significativamente más alta para el MLP (0.9775) en comparación con Gradient Boosting (0.9178).

En resumen, aunque Gradient Boosting tiene una mejor precisión, el MLP destaca en recall y AUC, lo que lo hace más adecuado para aplicaciones donde es crucial un equilibrio entre la correcta identificación de casos positivos y la minimización de falsos negativos, además de tener una excelente capacidad de discriminación general entre clases.

6.2.3.4. Retrospectiva Sprint 3

6.2.3.4.1. Lo que se hizo bien:

- Dividir los tiempos para organizar mejor la elaboración de los códigos
- Investigar y definir previamente los modelos a utilizar.

6.2.3.4.2. Áreas de Mejora:

Se pudo aprovechar el tiempo sobrante de anteriores sprint para avanzar ciertos aspectos como investigar o comenzar a revisar el comportamiento de varios modelos para no estancarse al momento de entrenarlos.

6.2.3.4.3. Próximos Pasos:

Intentar organizar el tiempo de la mejor manera ya que las dos siguientes tareas conocemos muy poco su complejidad dado que solo se ha realizado una vez.

6.2.5. Sprint 4

Tabla 21. Informe Sprint 4. Elaboración propia.

Nombre: Sprint 4			
Inicio: 09/04	Fin: 24/04	Número de días: 15 días	Número de PH: 22 PH
Informe sobre el Sprint: No se llevó a cabo reunión física con el PO, pero se le comunicó todos los logros y los retrasos, ya que, en este sprint no se logró acabar con todas las tareas encomendadas, pasando tres tareas del sprint 3 al 4.			
Tareas superadas del anterior Sprint: <ul style="list-style-type: none"> • [US_8] Elección e implementación de modelos DL. • [US_8.1] Entrenamiento y evaluación del modelo 1 (ANN) • [US_8.2] Entrenamiento y evaluación del modelo 2 (MLP) • [US_8.3] Comparación entre los modelos DL. • [US_8.4] Comparación de modelos ML y DL mediante las métricas. 			
Tareas planificadas para el Sprint actual: <ul style="list-style-type: none"> • [US_9] Evaluación en base a métricas sobre los resultados obtenidos. • [US_10] Implementación de algoritmos de interpretabilidad. • [US_11] Implementación de función de consenso. 			

Especificación de Requisitos:

- [US_9] Evaluación en base a métricas sobre los resultados obtenidos: este requisito implica que los resultados obtenidos del entrenamiento de los modelos sean evaluados bajo las métricas seleccionadas y se analice el desempeño para determinar la eficacia en la predicción.
- [US_10] Implementación de algoritmos de interpretabilidad: este requisito implica el desarrollo de algoritmos que permitan interpretar y comprender las decisiones tomadas por los modelos entrenados.
- [US_11] Implementación de función de consenso: este requisito implica el desarrollo y aplicación de una función que permita combinar y analizar las predicciones de los modelos para obtener un resultado final fiable.

Sobre el Sprint:

En este sprint se realizó primero la comparación de los mejores modelos, el mejor de ML y el mejor de DP. Luego se aplicó algoritmos de interpretabilidad LIME y SHAP en ambos modelos, en el caso de ML es el Gradient Boosting, por otro lado, en DL es MLP.

Resumen [US_9] Evaluación en base a métricas sobre los resultados obtenidos:

En esta parte del proyecto solo se compararon los resultados obtenidos de las evaluaciones bajo las métricas de ambos modelos. En la tabla 21, se visualiza los valores de Random Forest y MLP

Tabla 22. Tabla de comparación de los mejores modelos. Elaboración propia.

	Gradient Boosting	MLP
Precisión	0.9305	0.9100
Recall	0.8919	0.9000
F1-Score	0.9108	0.9100
AUC	0.9178	0.9765

A continuación, se muestra la matriz de confusión de ambos modelos.

- Matriz de confusión Gradient Boosting

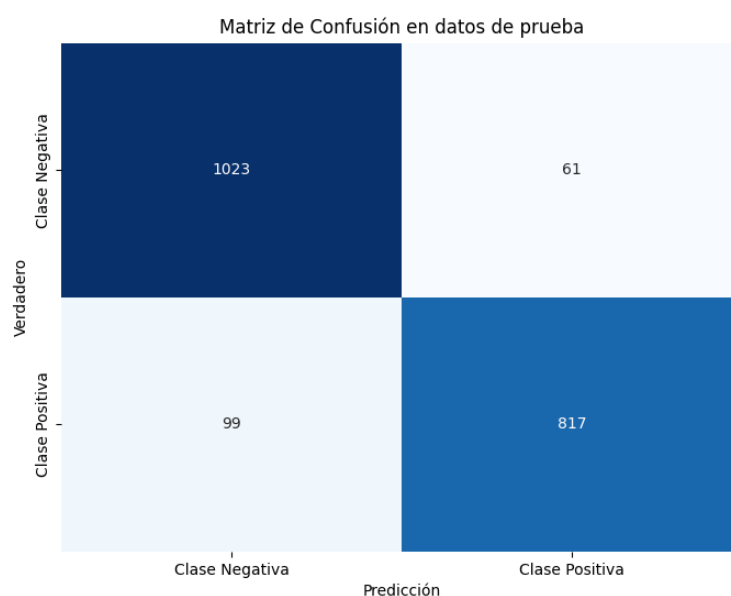


Ilustración 30. Matriz de confusión Gradient Boosting. Código Google Colab.

Se observa en la ilustración 30, la matriz de confusión del modelo Random Forest, representa en primer lugar, los Verdaderos Negativos, que indican las instancias en las que el modelo predijo correctamente la clase negativa. Se observaron 1023 instancias clasificadas correctamente en esta categoría.

Por otro lado, los Falsos Positivos representan las instancias en las que el modelo predijo incorrectamente la clase positiva como negativa. Se identificaron 61 instancias en esta situación.

Asimismo, los Falsos Negativos muestran las instancias en las que el modelo predijo incorrectamente la clase positiva como negativa. En total, se encontraron 99 instancias clasificadas de esta manera.

Por último, los Verdaderos Positivos representan las instancias en las que el modelo predijo correctamente la clase positiva. Se contaron 817 instancias en esta categoría.

En resumen, el modelo exhibe una alta precisión en la predicción de ambas clases, logrando más de 1800 predicciones correctas en total y manteniendo un número relativamente bajo de errores.

- Matriz de confusión MLP

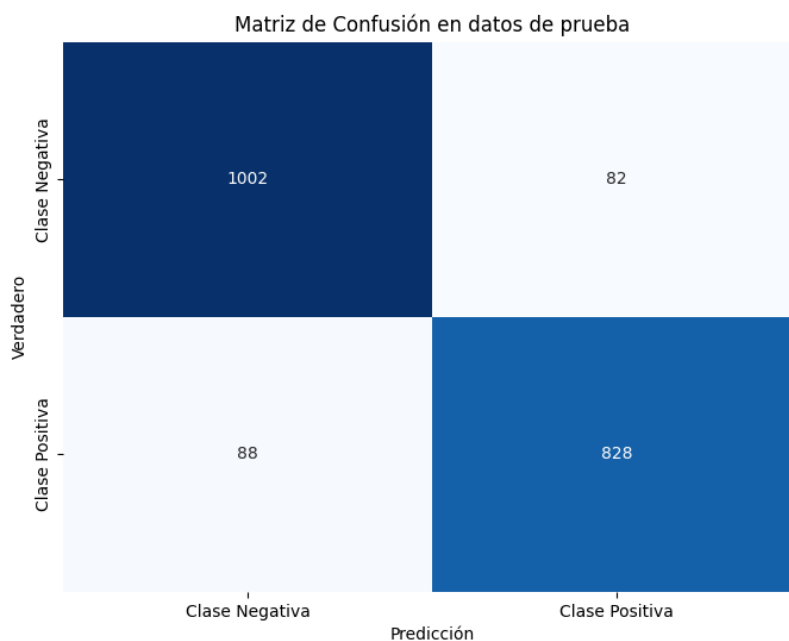


Ilustración 31. Matriz de confusión MLP. Código Google Colab.

La ilustración muestra la matriz de confusión del modelo evaluado. En primer lugar, los Verdaderos Negativos (VN) indican las instancias en las que el modelo predijo correctamente la clase negativa, con un total de 1002 instancias clasificadas correctamente.

Los Falsos Positivos (FP) representan las instancias en las que el modelo predijo incorrectamente la clase positiva como negativa, identificándose 82 instancias en esta situación.

Asimismo, los Falsos Negativos (FN) muestran las instancias en las que el modelo predijo incorrectamente la clase negativa como positiva, con un total de 88 instancias clasificadas de esta manera.

Por último, los Verdaderos Positivos (VP) representan las instancias en las que el modelo predijo correctamente la clase positiva, contabilizando 828 instancias en esta categoría.

En resumen, el modelo exhibe una alta precisión en la predicción de ambas clases, logrando más de 1800 predicciones correctas en total y manteniendo un número relativamente bajo de errores.

Resumen [US_10] Implementación de algoritmos de interpretabilidad:

Los algoritmos de interpretabilidad se le aplicaron a los dos modelos, dándonos un mejor entendimiento de cómo funcionan. Fue un proceso tedioso ya que el modelo de DL daba errores o no se interpretaba bien por lo que se tuvo que volver al entrenamiento un par de veces, pero finalmente funcionó todo de manera óptima.

- LIME (Gradient Boosting)

Se muestran 2 de los 10 casos que se visualizan en el notebook. Tener en cuenta que los datos están normalizados.

Primer Caso:

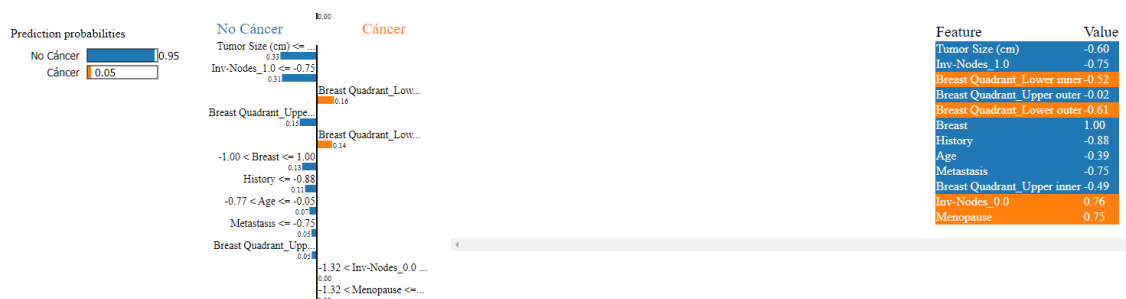


Ilustración 32. Captura de caso 1 de interpretabilidad LIME, Gradient Boosting. Código Google Colab

Se observa en la ilustración 32, una probabilidad de 0.95 para "No Cáncer" y 0.05 para "Cáncer". Cada característica contribuye de manera diferente a la predicción. Por ejemplo, el tamaño del tumor (Tumor Size) con un valor de -0.60 disminuye la probabilidad de cáncer, al igual que la presencia de ganglios linfáticos invadidos (Inv-Nodes_1.0) con un valor de -0.75. Sin embargo, características como la ubicación del tumor en el cuadrante inferior interno del pecho (Breast Quadrant_Lower inner) con un valor de -0.52, aumentan la probabilidad de cáncer. Otros factores y sus valores específicos se muestran en una lista, como la edad (Age), la historia médica (History), y la menopausia (Menopause), cada uno contribuyendo de manera distinta a la predicción final.

Segundo Caso:



Ilustración 33. Captura de caso 2 de interpretabilidad LIME, Gradient Boosting. Código Google Colab.

En la ilustración 33, la probabilidad de predicción es de 0.00 para "No Cáncer" y 1.00 para "Cáncer". De nuevo, cada característica está desglosada con su

influencia en la predicción. Por ejemplo, la edad (Age) con un valor de 1.43 y la presencia de ganglios linfáticos invadidos (Inv-Nodes_1.0) con un valor de 1.34 aumentan significativamente la probabilidad de cáncer. El tamaño del tumor (Tumor Size) con un valor de 1.82 también contribuye positivamente a la probabilidad de cáncer. En contraste, la menopausia (Menopause) con un valor de -1.32 disminuye la probabilidad de cáncer. La lista de características incluye otros factores como la metástasis (Metastasis), la historia médica (History) y la ubicación del tumor en diferentes cuadrantes del pecho, cada uno con sus respectivos valores específicos.

- LIME (MLP):

Se muestran 2 de los 10 casos que se visualizan en el notebook. Tener en cuenta que los datos están normalizados.

Primer Caso:

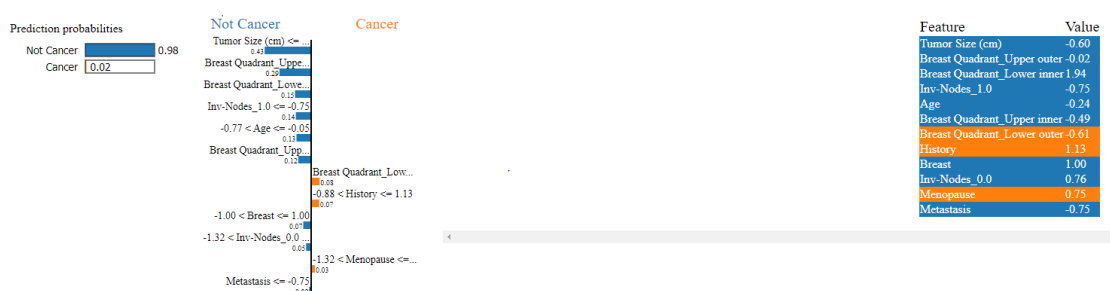


Ilustración 34. Captura de caso 1 de interpretabilidad LIME, MLP. Código Google Colab.

La ilustración tiene una probabilidad de 0.98 para "No Cáncer" y 0.02 para "Cáncer". Cada característica contribuye de manera diferente a la predicción. Por ejemplo, el tamaño del tumor (Tumor Size) con un valor de -0.60 disminuye significativamente la probabilidad de cáncer. La característica "Breast Quadrant_Upper outer" con un valor de -0.02 y "Breast Quadrant_Lower inner" con un valor de -0.52 también contribuyen a reducir la probabilidad de cáncer. Otros factores, como la presencia de ganglios linfáticos invadidos (Inv-Nodes_1.0) con un valor de -0.75 y la edad (Age) con un valor de -0.24, disminuyen aún más la probabilidad. Sin embargo, la historia médica (History) con un valor de 1.13 y la menopausia (Menopause) con un valor de 0.75 aumentan ligeramente la probabilidad de cáncer, pero no lo suficiente como para cambiar la predicción final de "No Cáncer".

Segundo Caso:

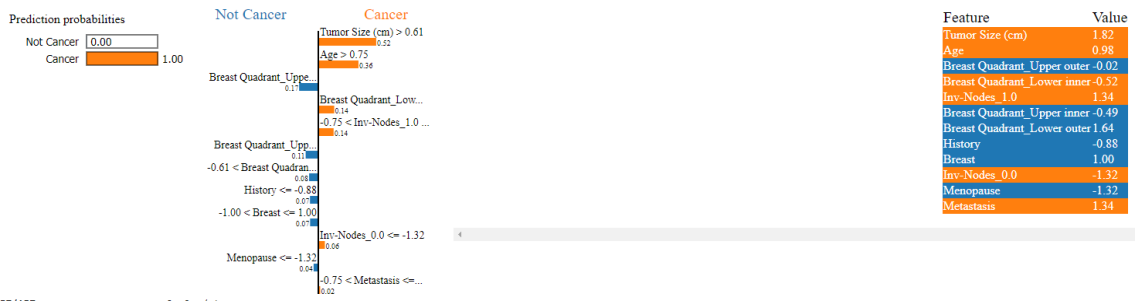


Ilustración 35. Captura de caso 2 de interpretabilidad LIME, MLP. Código Google Colab

Se observa una probabilidad de 1.00 para "Cáncer" y 0.00 para "No Cáncer". Aquí, el tamaño del tumor (Tumor Size) con un valor de 1.82 es un factor significativo que incrementa la probabilidad de cáncer. La edad (Age) con un valor de 0.98 y la ubicación del tumor en el cuadrante inferior interno del pecho (Breast Quadrant_Lower inner) con un valor de -0.52 también contribuyen a aumentar la probabilidad. Otros factores que incrementan la probabilidad de cáncer incluyen la presencia de ganglios linfáticos invadidos (Inv-Nodes_1.0) con un valor de 1.34 y la metástasis (Metastasis) con un valor de 1.34. Factores como la historia médica (History) con un valor de -0.88 y la menopausia (Menopause) con un valor de -1.32 disminuyen la probabilidad, pero no lo suficiente para cambiar la predicción final de "Cáncer".

- SHAP (Gradient Boosting)

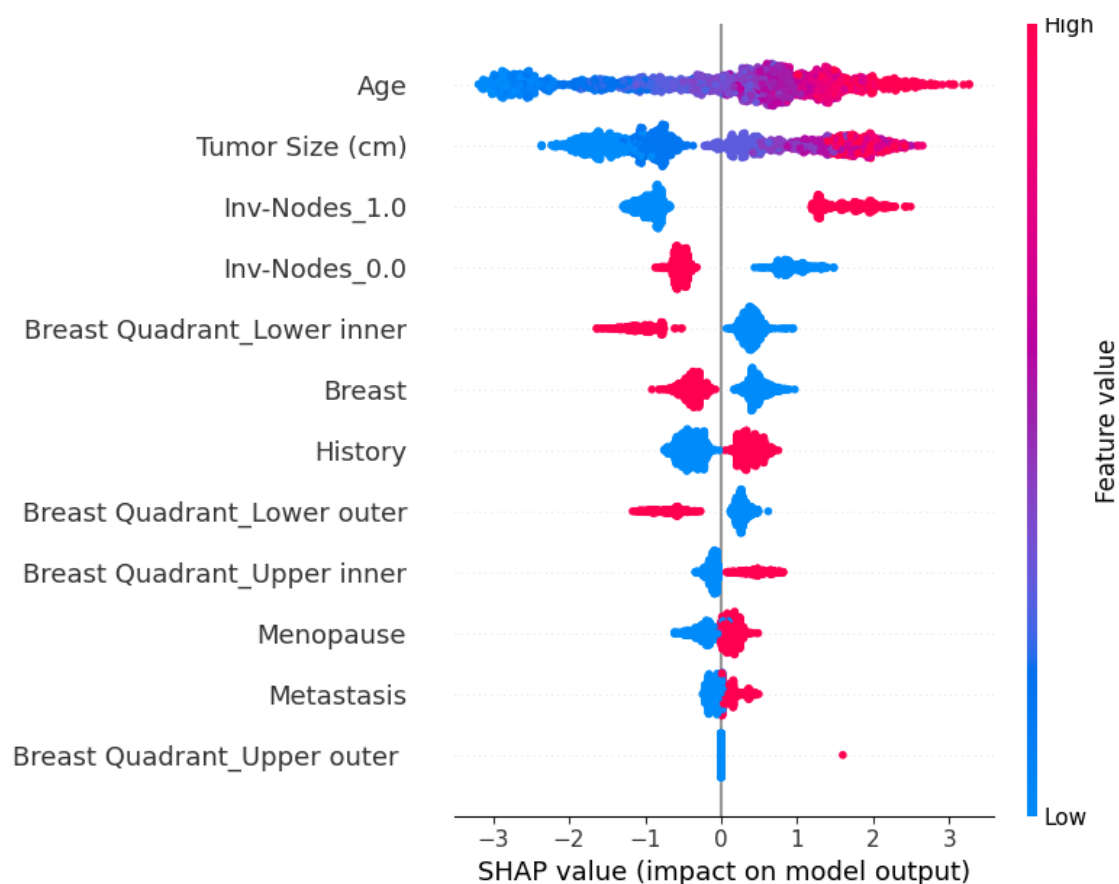


Ilustración 36. Captura de interpretación SHAP, Gradient Boosting. Código Google Colab.

En la ilustración, se ven los valores altos en rojo y valores bajos en azul indican el efecto de las características en la predicción. Por ejemplo, el tamaño del tumor (Tumor Size) tiene un impacto significativo, donde valores más altos tienden a aumentar la probabilidad de cáncer. La edad (Age) también muestra un gran impacto, con edades más avanzadas aumentando la probabilidad de cáncer. La presencia de ganglios linfáticos invadidos (Inv-Nodes_1.0) aumenta la probabilidad de cáncer, mientras que la ausencia de estos (Inv-Nodes_0.0) la disminuye.

Otros factores como la ubicación del tumor en diferentes cuadrantes del pecho (Breast Quadrant) y la presencia de metástasis (Metastasis) también tienen impactos notables en las predicciones del modelo.

- SHAP (MLP)

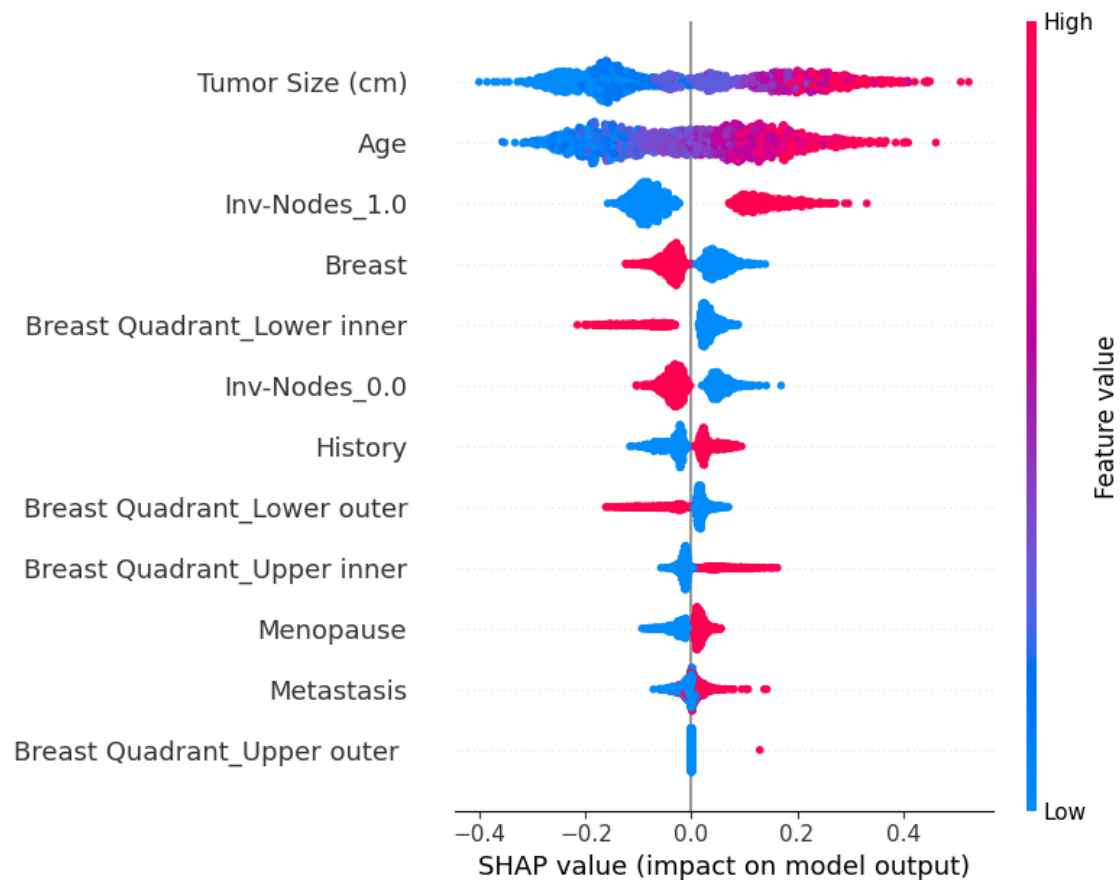


Ilustración 37. Captura de interpretación SHAP, MLP. Código Google Colab.

En esta interpretación el tamaño del tumor (Tumor Size) continúa mostrando un impacto significativo, donde valores más altos aumentan la probabilidad de cáncer. La edad (Age) sigue siendo una característica importante, con edades más avanzadas aumentando la probabilidad de cáncer. La presencia de ganglios linfáticos invadidos (Inv-Nodes_1.0) también incrementa la probabilidad de cáncer, mientras que la ausencia de estos (Inv-Nodes_0.0) la disminuye.

La ubicación del tumor en diferentes cuadrantes del pecho (Breast Quadrant), la historia médica (History) y la presencia de metástasis (Metastasis) son otros factores que afectan notablemente las predicciones del modelo.

6.2.3.5. Retrospectiva Sprint 4

6.2.3.5.1. Lo que se hizo bien:

Avanzar el resto del proyecto, se dejó algún que otro punto sin concluir, pero evitó el estancamiento. Al final, al volver al punto inconcluso, se resolvía fácilmente.

6.2.3.5.2. Áreas de Mejora:

Nos dimos cuenta de que el descanso era muy importante, que había apartados del proyecto que se nos hacía difícil, la mejor decisión era dejarlo e intentar avanzar algo independiente. Por ello, se implementó una dinámica de descanso que ayudó a mejorar el rendimiento.

6.2.3.5.3. Próximos Pasos:

A este punto solo queda una tarea por implementar, ya que lamentablemente no ha dado el tiempo para realizar la fusión de datos, pero confiamos que en el próximo sprint dará tiempo suficiente para elaborarlo y afinar detalles.

6.2.4. Sprint 5

Tabla 22. Informe Sprint 5. Elaboración propia.

Nombre: Sprint 5			
Inicio: 25/04	Fin: 06/05	Número de días: 21 días	Número de PH: 5 PH
Informe sobre el Sprint: Se llevaron a cabo dos reuniones con el PO en las fechas de 24/04 y 03/05. Al ser este el último sprint antes de la fecha de entrega hubo más reuniones para arreglar últimos detalles y terminar las tareas pendientes.			
Tareas superadas del anterior Sprint: <ul style="list-style-type: none">• [US_9] Evaluación en base a métricas sobre los resultados obtenidos.• [US_10] Implementación de algoritmos de interpretabilidad.			
Tareas planificadas para el Sprint actual: <ul style="list-style-type: none">• [US_11] Implementación de función de consenso.			

Especificación de Requisitos:

- [US_11] Implementación de función de consenso: este requisito implica el desarrollo y aplicación de una función que permita combinar y analizar las predicciones de los modelos para obtener un resultado final fiable.

Sobre el Sprint:

En este sprint se implementó la función de consenso que ha permitido tener una visión más fundamentada sobre la interpretación de los modelos, permitiendo entender mejor el entrenamiento y cómo funcionan en su predicción, a continuación, se comenta sobre la fusión de datos realizada y la conclusión llegada.

Se comentó en el sprint anterior sobre el empleo de métodos de interpretación, LIME y SHAP, pues para obtener una visión integral de las importancias de las características y finalmente calcular una medida de consenso se ha realizado una combinación de los métodos, logrando primero una visión local por cada algoritmo y luego una global. A continuación, se detalla el análisis y los

resultados obtenidos para cada modelo, apoyados en las gráficas y tablas generadas.

Gradient Boosting (GB)

	Feature	LIME Importance	SHAP Importance	Consensus Importance
0	Age	-0.013572	1.297920	0.642174
1	Menopause	-0.000154	0.167870	0.083858
2	Tumor Size (cm)	-0.098106	1.215182	0.558538
3	Breast	0.069482	0.438249	0.253865
4	Metastasis	-0.024959	0.112471	0.043756
5	History	-0.062681	0.407281	0.172300
6	Breast Quadrant_Lower inner	0.094915	0.522250	0.308583
7	Breast Quadrant_Lower outer	0.098921	0.380403	0.239662
8	Breast Quadrant_Upper inner	-0.036939	0.187709	0.075385
9	Breast Quadrant_Upper outer	0.000000	0.000800	0.000400
10	Inv-Nodes_0.0	-0.000063	0.679014	0.339476
11	Inv-Nodes_1.0	-0.194331	1.181356	0.493513

Ilustración 38. Tabla de importancias Gradient Boosting. Código Google Colab.

En la tabla se observan las importancias de características según LIME, SHAP y la medida de consenso. Las características más destacadas son Age, Tumor Size (cm), y Inv-Nodes_1.0, con altos valores de consenso.

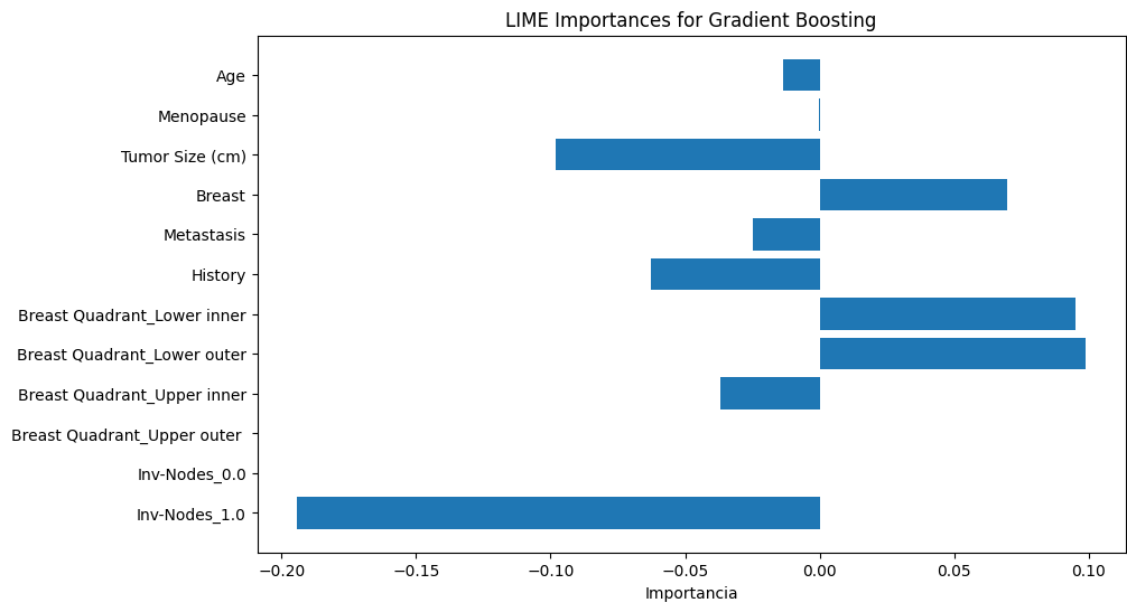


Ilustración 39. Importancia LIME, GB.

La gráfica muestra las importancias de características según LIME. Se observa que Tumor Size (cm) y Inv-Nodes_1.0 tienen valores negativos significativos, indicando su relevancia en las predicciones del modelo.

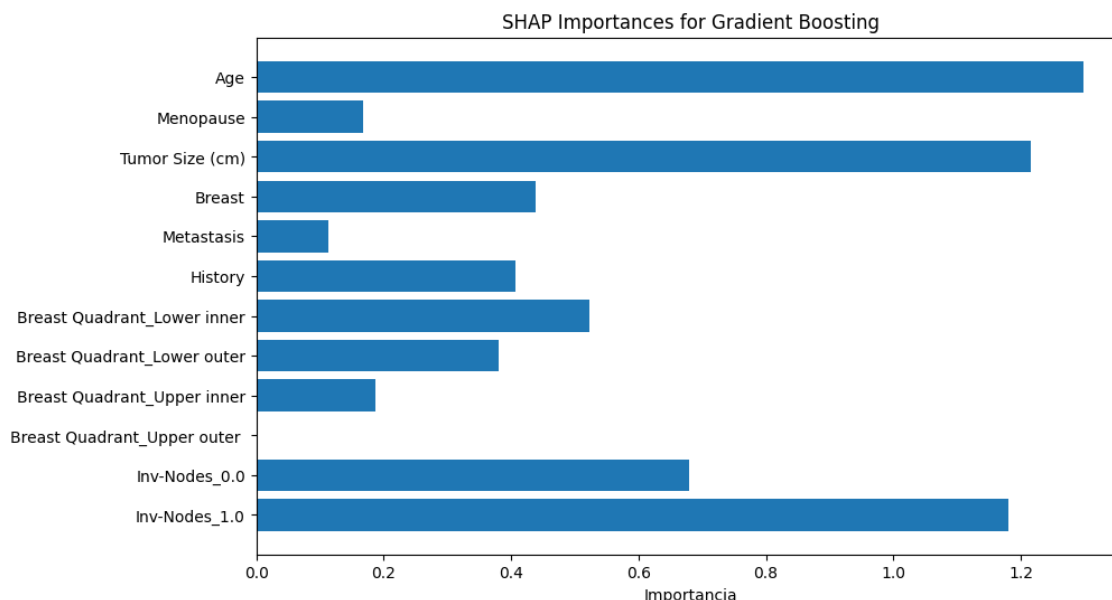


Ilustración 40. Importancia SHAP, GB.

La gráfica de SHAP resalta la alta importancia de Age, Tumor Size (cm), y Inv-Nodes_1.0. SHAP proporciona una interpretación global y muestra cómo cada característica impacta las predicciones del modelo.

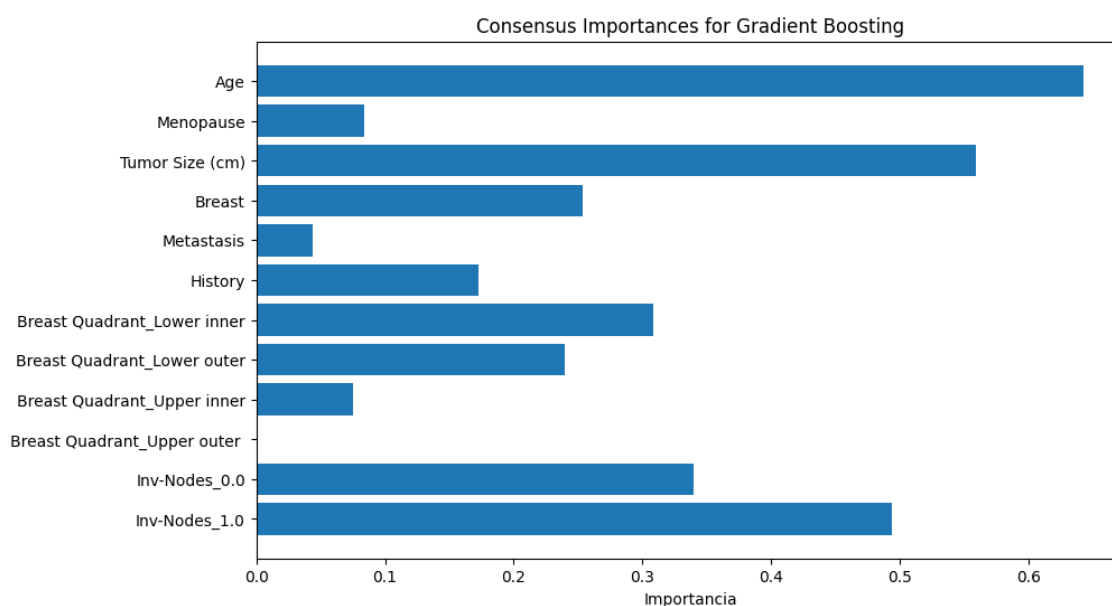


Ilustración 41. Consenso de importancias GB.

La medida de consenso, que combina LIME y SHAP, confirma la relevancia de Age, Tumor Size (cm), y Inv-Nodes_1.0. Esta combinación permite obtener una visión más robusta de las características críticas para el modelo.

	Feature	LIME Importance	SHAP Importance	Consensus Importance
0	Age	-0.011479	0.130415	0.059468
1	Menopause	-0.011849	0.026781	0.007466
2	Tumor Size (cm)	-0.123373	0.171107	0.023867
3	Breast	0.047124	0.053084	0.050104
4	Metastasis	-0.020755	0.012533	-0.004111
5	History	-0.038112	0.046342	0.004115
6	Breast Quadrant_Lower inner	0.060694	0.053080	0.056887
7	Breast Quadrant_Lower outer	0.084577	0.043492	0.064035
8	Breast Quadrant_Upper inner	-0.094755	0.022122	-0.036316
9	Breast Quadrant_Upper outer	0.000000	0.000119	0.000060
10	Inv-Nodes_0.0	0.019163	0.063840	0.041501
11	Inv-Nodes_1.0	-0.096752	0.086251	-0.005251

Ilustración 42. Tabla de importancias MLP. Código Google Colab.

Similar a GB, la tabla muestra las importancias de características según LIME, SHAP y el consenso. Age, Tumor Size (cm), y Breast Quadrant_Lower inner son destacadas por su alta importancia de consenso.

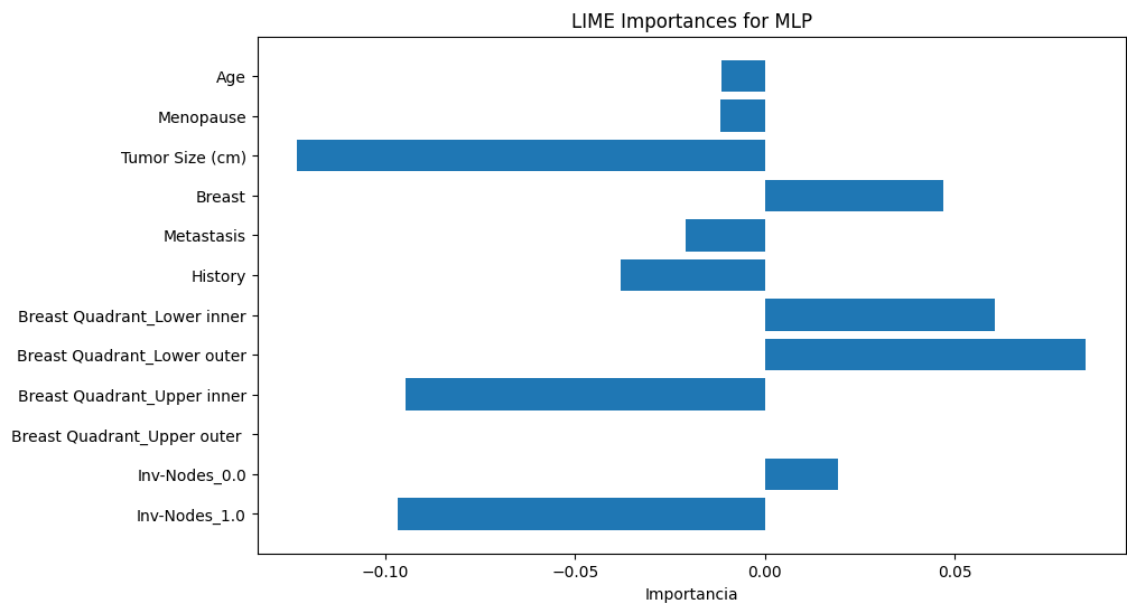


Ilustración 43. Importancia LIME, MLP.

En LIME, Tumor Size (cm) muestra una importancia significativa, indicando su relevancia en la predicción del modelo MLP.

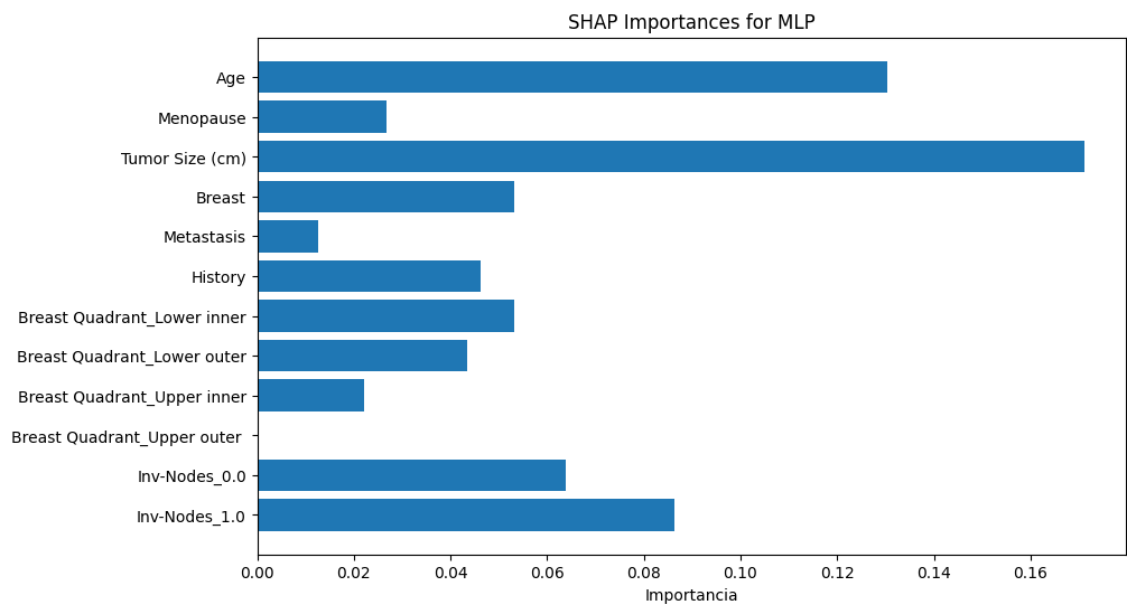


Ilustración 44. Importancia SHAP, MLP.

SHAP destaca Age y Tumor Size (cm) como las características más importantes para el modelo MLP, proporcionando una visión global de su impacto en las predicciones.

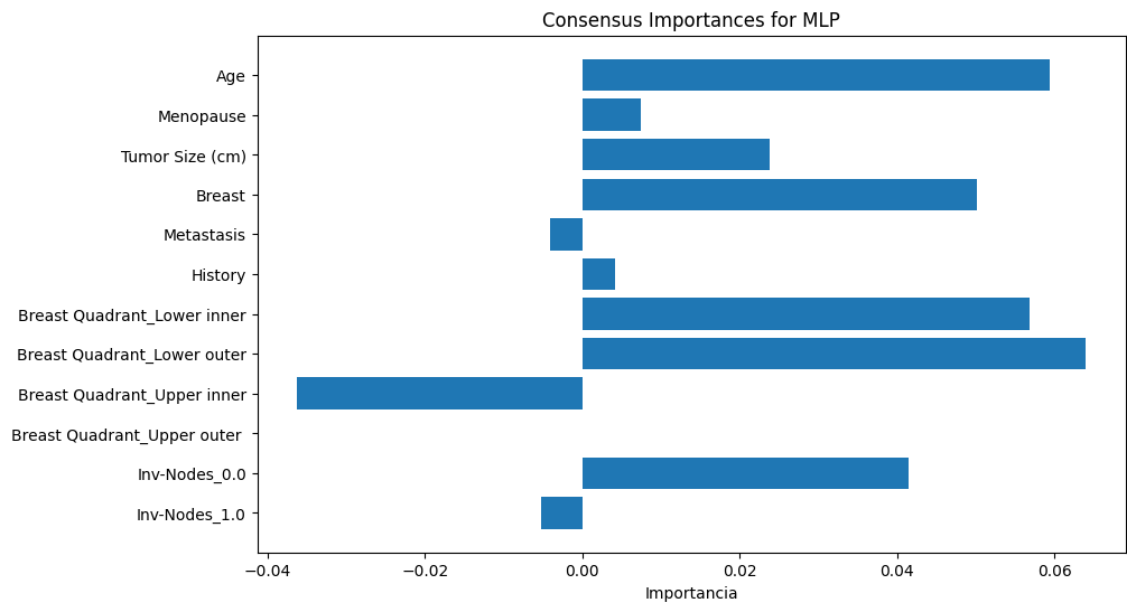


Ilustración 45. Consenso de importancias MLP.

La gráfica de consenso muestra que Age, Tumor Size (cm), y Breast Quadrant_Lower inner son características cruciales, reafirmando los hallazgos de LIME y SHAP.

Respecto al consenso global de las importancias de las características combinando los resultados obtenidos de los modelos Gradient Boosting (GB) y Multi-Layer Perceptron (MLP), se muestran los siguientes datos y gráficas.

Tabla 23. Tabla de Consenso Global. Elaboración propia.

Feature	Consensus Importance GB	Consensus Importance MLP	Consensus Importance Global
Age	0.6421	0.0594	0.3508
Menopause	0.0838	0.0074	0.0456
Tumor Size (cm)	0.5585	0.0238	0.2912
Breast	0.2538	0.0501	0.1519
Metastasis	0.0437	-0.0041	0.0198
History	0.1723	0.0041	0.0882
Breast Quadrant_Lower inner	0.3085	0.0568	0.1827
Breast Quadrant_Lower outer	0.2396	0.0640	0.1518
Breast Quadrant_Upper inner	0.0753	-0.0363	0.0195
Breast Quadrant_Upper outer	0.0004	0.00006	0.0002
Inv-Nodes_0.0	0.3394	0.0415	0.1904
Inv-Nodes_1.0	0.4935	-0.0052	0.2441

La tabla muestra las importancias de consenso de características para ambos modelos (GB y MLP) y el consenso global calculado como el promedio de estas importancias. A continuación, la gráfica muestra las mismas características, pero de manera más visual.

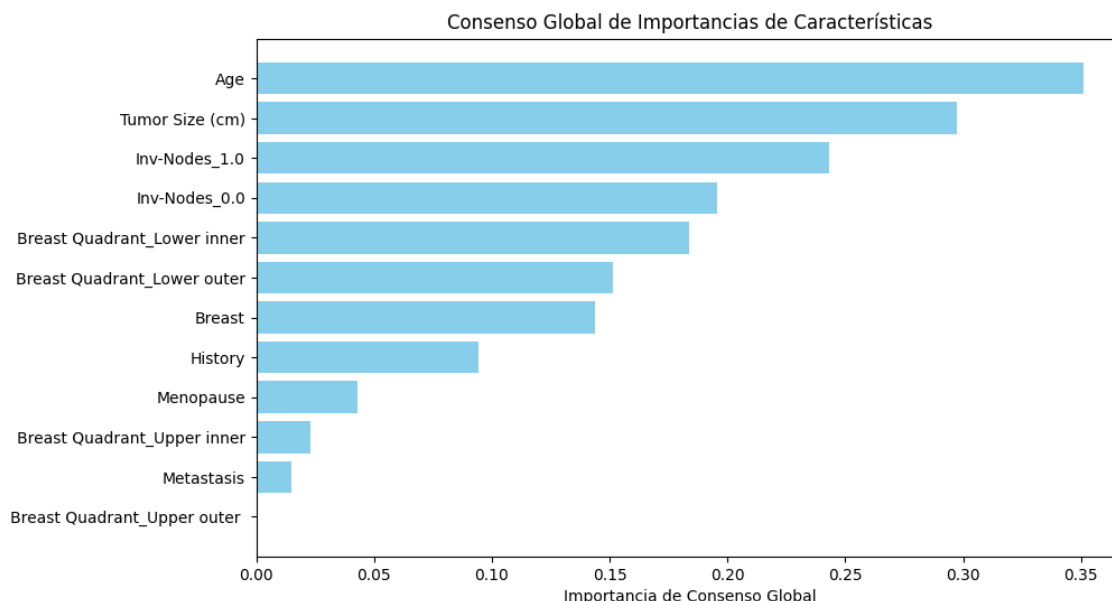


Ilustración 46. Gráfica de consenso global. Código Google Colab.

La edad presenta una importancia de consenso global de 0.3508, esta característica es crucial en el diagnóstico y pronóstico del cáncer de mama, ya que el riesgo de desarrollar la enfermedad aumenta con la edad. La alta importancia de esta característica en el consenso global reafirma su papel como un factor de riesgo significativo.

El tamaño del tumor (cm) tiene una importancia de consenso global de 0.2912, lo que es un indicador directo de la progresión del cáncer. Tumores más grandes suelen estar asociados con una mayor agresividad y un peor pronóstico, lo que explica su alta relevancia en el consenso global.

La característica Inv-Nodes_1.0 muestra una importancia de consenso global de 0.2441. La invasión de nodos linfáticos es un fuerte indicador de diseminación del cáncer, la presencia de nodos linfáticos invadidos (Inv-Nodes_1.0) sugiere un mayor riesgo de metástasis, lo que la convierte en una característica clave.

Inv-Nodes_0.0 presenta una importancia de consenso global de 0.1904. La ausencia de invasión de nodos linfáticos (Inv-Nodes_0.0) también es importante, ya que sugiere una menor probabilidad de diseminación del cáncer. Esta característica es relevante para evaluar la extensión del cáncer y el pronóstico.

La localización del tumor en el cuadrante inferior interno del pecho (Breast Quadrant_Lower inner) tiene una importancia de consenso global de 0.1827. Esta característica tiene relevancia clínica, ya que puede influir en las opciones de tratamiento y en el pronóstico del paciente.

La característica Breast tiene una importancia de consenso global de 0.1519. La implicación del pecho como característica general puede referirse a aspectos específicos de la anatomía y patología del tumor, influyendo en el diagnóstico y tratamiento.

La localización del tumor en el cuadrante inferior externo del pecho (Breast Quadrant_Lower outer) muestra una importancia de consenso global de 0.1518. Similar al cuadrante inferior interno, la localización del tumor en el cuadrante inferior externo también afecta el pronóstico y las opciones de tratamiento.

Los antecedentes médicos y familiares (History) presentan una importancia de consenso global de 0.0882. Estos factores son importantes en el riesgo de cáncer de mama. La historia clínica puede proporcionar información sobre predisposiciones genéticas y exposiciones previas.

La menopausia tiene una importancia de consenso global de 0.0456. La menopausia afecta los niveles hormonales y puede influir en el riesgo de cáncer de mama, especialmente en tipos de cáncer sensibles a hormonas.

La característica Metastasis muestra una importancia de consenso global de 0.0198. La presencia de metástasis es un indicador directo de la gravedad del cáncer. Aunque su importancia es relativamente menor en comparación con otras características, sigue siendo un factor crucial en el pronóstico.

La localización del tumor en el cuadrante superior interno del pecho (Breast Quadrant_Upper inner) presenta una importancia de consenso global de 0.0195. Esta característica tiene un impacto menor en comparación con otros cuadrantes, pero aún es relevante para el diagnóstico y tratamiento.

Finalmente, la localización del tumor en el cuadrante superior externo del pecho (Breast Quadrant_Upper outer) tiene una importancia de consenso global de 0.0002. Similar al cuadrante superior interno, la importancia de esta

característica es menor, pero puede influir en el manejo clínico del cáncer de mama.

El consenso global de importancias de características proporciona una evaluación equilibrada y robusta de las características más relevantes para la predicción del cáncer de mama.

Al combinar los resultados de los modelos Gradient Boosting (GB) y Multi-Layer Perceptron (MLP), se obtiene una visión más completa y confiable de los factores críticos, mejorando la interpretación y la aplicabilidad de los resultados en contextos clínicos.

6.2.4.1. *Retrospectiva Sprint 5*

6.2.4.1.1. Lo que se hizo bien:

- Se fue consistente durante el sprint, se consultó con el PO por los resultados obtenidos y estar de acuerdo en la lógica de los resultados obtenidos.

6.2.4.1.2. Próximos Pasos:

- Redacción del documento de apoyo del proyecto y entrega final del mismo.

El día 6 de mayo se dio fin a los Sprint, siendo este el día de la entrega del proyecto al PO.

7. DESPLIEGUE Y PRUEBA DE LA SOLUCIÓN

En este apartado se expondrá el plan de pruebas planteado y desarrollado para la implementación de este proyecto.

Para introducir a todo ello, la manera de poder ingresar y visualizar este proyecto, este a través del siguiente link:

<https://colab.research.google.com/drive/1aqWwAyFFQyWthocZQnrV27VZdkWJchLB?usp=sharing>

7.1. Plan de pruebas

Al final de cada interacción de Scrum, es decir, los sprints, se realizan pruebas de lo desarrollado, de esta manera garantizamos un incremento eficiente al cliente.

En este apartado, se presentará el plan de pruebas elaborado que aborda todas los requisitos definidos en el Product Backlog, se organiza en la Tabla x, donde encontramos varias columnas: el número de del Product Backlog, id y nombre del requisito específico, número de sprint y resultado.

Tabla 24. Requisitos técnicos específicos. Elaboración propia.

ID	Nombre de requisito de Product Backlog	ID	Requisito específico	Sprint	Resultado
US_1	Búsqueda y elección del dataset			0	
US_2	Implementación de la Gan	US_2.1	Limpieza de dataset	1	
		US_2.2	Aplicación de codificadores	1	
		US_2.3	Entrenamiento de la GAN	1	
		US_2.4	Generación y exportación de nuevo dataset.	1	
US_3	Elección de métricas.			1	
US_4	Implementación de EDA	US_4.1	Implementación de EDA pre-GAN (dataset original)	2	
		US_4.2	Implementación de EDA post GAN (dataset sintético)	2	

US_5	Comparación de dataset original y sintético.			2	
US_6	Preparación de datos para IA	US_6.1	Separación de variables features y target para ML.	2	
		US_6.2	Separación train, validation y test para ML.	2	
		US_6.3	Normalización de variables.	2	
US_7	Elección e implementación de modelos ML.	US_7.1	Entrenamiento y evaluación modelo 1 (Regresión Logística).	2	
		US_7.2	Entrenamiento y evaluación modelo 2 (Gradient Boosting).	2	
		US_7.3	Entrenamiento y evaluación modelo 3 (Random Forest).	2	
		US_7.4	Comparación resultados entre modelos ML.	2	
		US_8.1	Entrenamiento y evaluación modelo 1 (ANN).	3	

US_8	Elección e implementación de modelos DL.	US_8.2	Entrenamiento y evaluación modelo 2 (MLP).	3	
		US_8.3	Comparación resultados entre modelos DL.	3	
		US_8.4	Comparación de modelos ML y DL.	3	
US_9	Evaluación métricas sobre resultados.			4	
US_10	Implementación algoritmos de interpretabilidad.	US_10.1	Implementación de LIME	4	
		US_10.2	Implementación de SHAP	4	
US_11	Implementación de función de consenso.			4	

A continuación, se comentará a detalle cada caso de prueba:

Tabla 25. Caso de Prueba [US_1]

ID: CP_1	Requisito específico: US_1	
Caso de prueba: Validación de dataset		Sprint: 0
Descripción: Verificar que el dataset cumple con los requisitos y la calidad necesaria para el proyecto.		
Precondiciones: El dataset debe cumplir ciertas especificaciones necesarias para que se puedan utilizar en el entrenamiento del modelo, como que sea de predicción binaria y actual.		
Resultado esperado: El dataset es validado cumple con lo especificado y puede utilizarse.		

Tabla 26. Caso de Prueba [US_2.1]

ID: CP_2	Requisito específico: US_2.1	
Caso de prueba: Validación de dataset		Sprint: 1
Descripción: Verificar la correcta limpieza y formato del dataset.		
Precondiciones: Dataset inicial debe estar disponible.		
Resultado esperado: Dataset limpio y formateado correctamente.		

Tabla 27. Caso de Prueba [US_2.2]

ID: CP_3	Requisito específico: US_2.2	
Caso de prueba: Funcionamiento de Codificadores		Sprint: 1
Descripción: Test para validar que los codificadores trabajan adecuadamente en el dataset.		
Precondiciones: Codificadores configurados correctamente.		
Resultado esperado: Codificadores funcionan sin errores.		

Tabla 28. Caso de Prueba [US_2.3]

ID: CP_4	Requisito específico: US_2.3	
Caso de prueba: Entrenamiento de la GAN		Sprint: 1
Descripción: Probar la efectividad del entrenamiento de la GAN.		
Precondiciones: Dataset limpio y codificadores en función.		
Resultado esperado: Entrenamiento estable y efectivo.		

Tabla 29. Caso de Prueba [US_2.4]

ID: CP_5	Requisito específico: US_2.4	
Caso de prueba: Generación y Exportación de Datos		Sprint: 1
Descripción: Asegurar que la GAN genera y exporta los datos correctamente.		
Precondiciones: GAN entrenada y funcionando.		
Resultado esperado: Datos generados y exportados correctamente.		
Errores: Errores en la generación de los datos, ya que no eran representativos del dataset original		

Tabla 30. Caso de Prueba [US_3]

ID: CP_6	Requisito específico: US_3	
Caso de prueba: Elección y Aplicación de Métricas		Sprint: 1
Descripción: Verificar que las métricas elegidas son aplicadas correctamente para evaluar los futuros modelos.		
Precondiciones: Tener claro los objetivos del proyecto y que métricas podrían ser ideales a implementar.		
Resultado esperado: Métricas elegidas correctamente para su futuro uso.		

Tabla 31. Caso de Prueba [US_4.1]

ID: CP_7	Requisito específico: US_4.1	
Caso de prueba: Análisis Exploratorio de Datos Original		Sprint: 2
Descripción: Realizar un análisis exploratorio detallado del dataset original.		
Precondiciones: Dataset original completo y herramientas de EDA listas.		
Resultado esperado: EDA completado con detalles que permiten comprender mejor el dataset.		

Tabla 32. Caso de Prueba [US_4.2]

ID: CP_8	Requisito específico: US_4.2	
Caso de prueba: Análisis Exploratorio de Datos Sintético		Sprint: 2
Descripción: Evaluar el dataset sintético para confirmar su calidad y utilidad.		
Precondiciones: Dataset sintético generado y herramientas de EDA listas.		
Resultado esperado: Confirmación de la calidad y utilidad del dataset sintético.		

Tabla 33. Caso de Prueba [US_5]

ID: CP_9	Requisito específico: US_5	
Caso de prueba: Comparación de Datasets		Sprint: 2
Descripción: Comparar las características y la distribución de datos entre los dos datasets.		
Precondiciones: EDA completado para ambos datasets.		
Resultado esperado: Comparación detallada realizada para garantizar la representación del dataset sintético del original.		

Tabla 34. Caso de Prueba [US_6.1]

ID: CP_10	Requisito específico: US_6.1	
Caso de prueba: Separación de Features y Target		Sprint: 2
Descripción: Separar las variables de entrada (features) y de salida (target) para ML.		
Precondiciones: Datasets limpios y disponibles.		
Resultado esperado: Variables correctamente separadas sin pérdida de datos.		

Tabla 35. Caso de Prueba [US_6.2]

ID: CP_11	Requisito específico: US_6.2	
Caso de prueba: División en Conjuntos de Entrenamiento		Sprint: 2
Descripción: Dividir los datos en conjuntos de entrenamiento, validación y prueba.		
Precondiciones: Variables features y target correctamente separadas.		
Resultado esperado: Conjuntos de datos correctamente divididos y balanceados.		

Tabla 36. Caso de Prueba [US_6.3]

ID: CP_12	Requisito específico: US_6.3	
Caso de prueba: Normalización de Datos		Sprint: 2
Descripción: Normalizar las variables para que estén en la misma escala.		
Precondiciones: Datos divididos en conjuntos adecuados.		
Resultado esperado: Datos normalizados listos para el modelado.		

Tabla 37. Caso de Prueba [US_7.1]

ID: CP_13	Requisito específico: US_7.1	
Caso de prueba: Entrenamiento de Regresión Logística		Sprint: 2
Descripción: Entrenar y evaluar un modelo de Regresión Logística.		
Precondiciones: Datos preparados y normalizados.		
Resultado esperado: Modelo entrenado y evaluado con las métricas elegidas, y obtener un rendimiento bueno, que supere el 0.9 en precisión.		
Errores: Rendimiento que no llenaba nuestras expectativas.		

Tabla 38. Caso de Prueba [US_7.2]

ID: CP_14	Requisito específico: US_7.2	
Caso de prueba: Entrenamiento de Gradient Boosting		Sprint: 2
Descripción: Entrenar y evaluar un modelo de Gradient Boosting.		
Precondiciones: Datos preparados y normalizados.		
Resultado esperado: Modelo entrenado y evaluado, rendimiento bueno y mejor que el modelo 1.		

Tabla 39. Caso de Prueba [US_7.3]

ID: CP_15	Requisito específico: US_7.3	
Caso de prueba: Entrenamiento de Random Forest		Sprint: 2
Descripción: Entrenar y evaluar un modelo de Random Forest.		
Precondiciones: Datos preparados y normalizados.		
Resultado esperado: Modelo entrenado y evaluado, obtener un buen rendimiento.		

Tabla 40. Caso de Prueba [US_7.4]

ID: CP_16	Requisito específico: US_7.4	
Caso de prueba: Comparación de Modelos ML		Sprint: 2
Descripción: Comparar el rendimiento de los modelos de ML implementados.		
Precondiciones: Modelos ML entrenados y evaluados.		
Resultado esperado: Realizar una comparación detallada que permita identificar el mejor modelo.		

Tabla 41. Caso de Prueba [US_8.1]

ID: CP_17	Requisito específico: US_8.1	
Caso de prueba: Entrenamiento de ANN		Sprint: 3
Descripción: Entrenar y evaluar un modelo de Red Neuronal Artificial (ANN).		
Precondiciones: Datos preparados y normalizados.		
Resultado esperado: Modelo entrenado y evaluado, obtener un buen rendimiento en las métricas elegidas, especialmente AUC y Recall.		
Errores: Modelo con bajo rendimiento, no llenaba para nada las expectativas.		

Tabla 42. Caso de Prueba [US_8.2]

ID: CP_18	Requisito específico: US_8.2	
Caso de prueba: Entrenamiento de MLP		Sprint: 3
Descripción: Entrenar y evaluar un modelo de Perceptrón Multicapa (MLP).		
Precondiciones: Datos preparados y normalizados.		
Resultado esperado: Modelo entrenado y evaluado, obtener un rendimiento que sea competitivo a comparación con los otros resultados obtenidos.		

Errores: Modelo con un rendimiento medio, no eran números suficientemente buenos.

Tabla 43. Caso de Prueba [US_8.3]

ID: CP_19	Requisito específico: US_8.3	
Caso de prueba: Comparación de Modelos DL		Sprint: 3
Descripción: Comparar el rendimiento de los modelos de DL implementados.		
Precondiciones: Modelos DL entrenados y evaluados.		
Resultado esperado: Comparación detallada realizada, mejores modelos identificados.		

Tabla 44. Caso de Prueba [US_8.4]

ID: CP_20	Requisito específico: US_8.4	
Caso de prueba: Comparación ML vs. DL		Sprint: 3
Descripción: Comparar el rendimiento entre los modelos de ML y DL.		
Precondiciones: Modelos ML y DL entrenados.		
Resultado esperado: Obtener detalles de los modelos e identificar en los que mejor rendimiento tienen de ambos tipos.		

Tabla 45. Caso de Prueba [US_9]

ID: CP_21	Requisito específico: US_9	
Caso de prueba: Evaluación de Modelos con Métricas		Sprint: 4
Descripción: Evaluación y comparación de los dos mejores modelos bajo las métricas seleccionadas anteriormente.		
Precondiciones: Modelos ML y DL entrenados y listos para evaluación.		
Resultado esperado: Métricas aplicadas correctamente y resultados claramente documentados.		

Tabla 46. Caso de Prueba [US_10.1]

ID: CP_22	Requisito específico: US_10.1	
Caso de prueba: Implementación y Evaluación de LIME		Sprint: 4
Descripción: Implementar y evaluar LIME para explicar las predicciones de los dos mejores modelos.		
Precondiciones: Modelos entrenados.		
Resultado esperado: LIME implementado correctamente, explicaciones efectivas y coherentes.		
Errores: Errores en la generación de explicaciones.		

Tabla 47. Caso de Prueba [US_10.2]

ID: CP_23	Requisito específico: US_10.2	
Caso de prueba: Implementación y Evaluación de SHAP		Sprint: 4
Descripción: Implementar y evaluar SHAP para explicar las predicciones de los dos mejores modelos.		
Precondiciones: Modelos entrenados.		
Resultado esperado: SHAP implementado correctamente, explicaciones generadas y coherentes.		

Tabla 48. Caso de Prueba [US_11]

ID: CP_24	Requisito específico: US_11	
Caso de prueba: Función de Consenso		Sprint: 4
Descripción: Implementar y validar una función de consenso para decisiones finales basadas en los modelos evaluados.		
Precondiciones: Todos los modelos deben estar completamente evaluados y comparados.		
Resultado esperado: Función de consenso operativa y fiable.		

7.2. Objetivos alcanzados

Se ha alcanzado y concluido el objetivo principal que se planteó al inicio del proyecto. Desde la búsqueda de un buen conjunto de datos, implementación de los modelos y su interpretación, se ha logrado obtener una mejor comprensión de los modelos y de cómo se interpretan, al igual de comprender cómo funcionan los entrenamientos y qué características de los datos son más importantes para cada modelo. Lo mencionado, concluye que el proyecto ha sido exitoso dentro de los parámetros de los objetivos generales definidos.

Respecto a los objetivos específicos, también se han completado, aunque ha requerido de esfuerzo y constancia en el desarrollo por los desafíos que se han presentado. Se han invertido muchas horas de desarrollo y compromiso con el proyecto que fue necesario para lograr lo propuesto y cumplir no solo las expectativas del Product Owner, sino también la del equipo.

Por otro lado, sobre los aspectos técnicos, el proyecto ha permitido profundizar en el Machine y DL, posibilitando el comprender mejor cómo funcionan los modelos al interpretarlos. Asimismo, se ha tenido primeros contactos con diferentes técnicas y tecnologías, como la regresión logística o la MLP, además, se ha podido explorar nuevas técnicas, librerías y formas de implementar lo que se quería desarrollar.

Ahondando en los objetivos alcanzados técnicamente, los cinco modelos que se han desarrollado muestran un gran potencial para futuros desarrollos. El modelo Random Forst ha dado buenos resultados en términos de precisión y AUC, lo que puede ser prometedor profundizar en su entrenamiento. De igual manera, el modelo Regresión Logística, aunque no se haya invertido mucho en su entrenamiento, puede ser interesante y prometedor conocerlo más. La comparación realizada entre los modelos ML desde un principio colocó como el modelo más sólido, Gradient Boosting que mostró resultados destacables bajo las métricas más importantes, entre ellas Recall.

Finalmente, los modelos de DL, tanto la MLP como la ANN han requerido de un gran esfuerzo para estudiarlos y entrenarlos de manera eficiente, de manera que se logró obtener mejores tasas de predicción que los modelos ML, concluyendo

la MLP como el mejor modelo global. Luego con la implementación de LIME y SHAP, se ha podido interpretar y de esta manera comprender mejor los modelos y las características que influyen más, lo que ha proporcionado una visión más amplia sobre estos. La fusión de datos ha permitido tener una interpretación más global y local de los algoritmos y de los modelos entrenados, al igual de proveer información e identificar marcadores interesantes e importantes resultantes del entrenamiento de los modelos.

En resumen, se ha logrado un proyecto exitoso que ha estado lleno de aprendizajes, experiencias y ha puesto a prueba las habilidades del equipo dentro del campo de la IA.

7.3. Indicadores del desempeño y/o utilidad.

Para la elaboración del proyecto ha sido esencial garantizar el logro de los objetivos establecidos, por ello los indicadores de desempeño y utilidad han proporcionado una medida objetiva de cómo el esfuerzo del equipo ha sido plasmado en los resultados obtenidos, permitiendo realizar ajustes en la gestión del proyecto.

En este apartado se evaluará el progreso del proyecto, para identificar áreas de mejoras y satisfacer las expectativas del Product Owner.

7.3.1. Metodología de Medición

Los datos que se han utilizado para realizar los gráficos de indicadores de desempeño han sido a través de la captura de las retrospectivas descritas al final de cada sprint del proyecto. Se incluyen horas de trabajo planificadas versus las horas reales, nivel de satisfacción del equipo y número de tareas pendientes al final de cada sprint.

Se han utilizado herramientas de gestión de proyectos y hojas de cálculo para registrar y analizar estos datos, permitiendo visualizar tendencias y realizar comparaciones efectivas a lo largo del proyecto.

Tabla 49. Tabla de Indicadores clave de Rendimiento

Sprint	Fecha de Inicio	Fecha de Fin	Horas Planificadas	Horas Reales	Satisfacción del Sprint (%)	Tareas Pendientes
Sprint 0	06/02	21/02	6	4	80	0
Sprint 1	22/02	08/03	46	50	100	0
Sprint 2	11/03	22/03	44	48	85	0
Sprint 3	25/03	08/04	46	55	70	1
Sprint 4	09/04	24/04	44	46	90	1
Sprint 5	25/04	06/05	10	12	90	0

7.3.2. Indicadores Clave de Rendimiento (KPIs)

Horas Planificadas vs. Horas Reales: Este KPI mide la eficacia en la planificación frente a la ejecución real. Al no haber una discrepancia significativa entre los datos, no es necesario el ajuste de las estimaciones de tiempo o la mejora de la gestión de recursos.

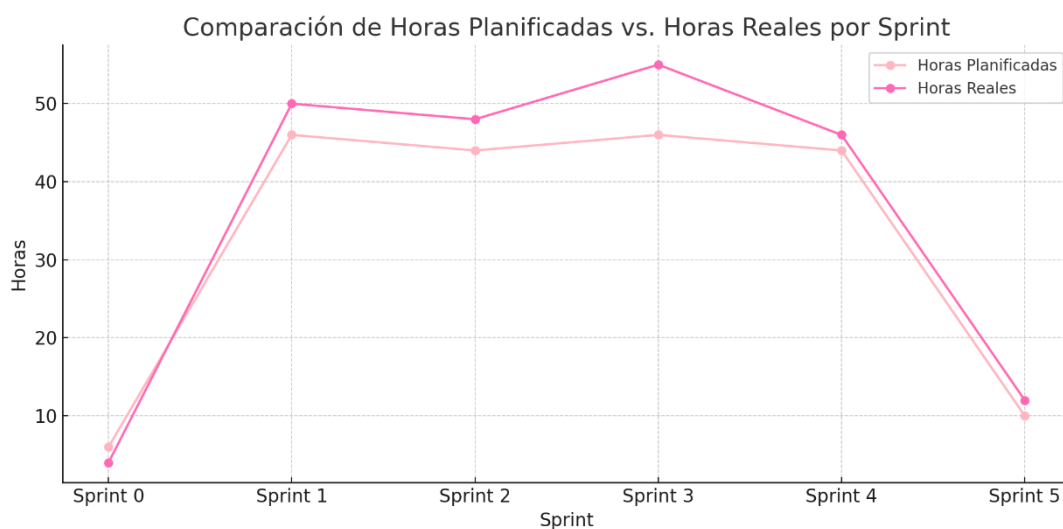


Ilustración 47. Comparación de horas planificadas vs. horas reales por Sprint. Elaboración propia

Satisfacción del Sprint (%): Se evalúa la satisfacción del equipo al final de cada sprint mediante un feedback. Este indicador es de suma importancia para medir el bienestar y la motivación del equipo.

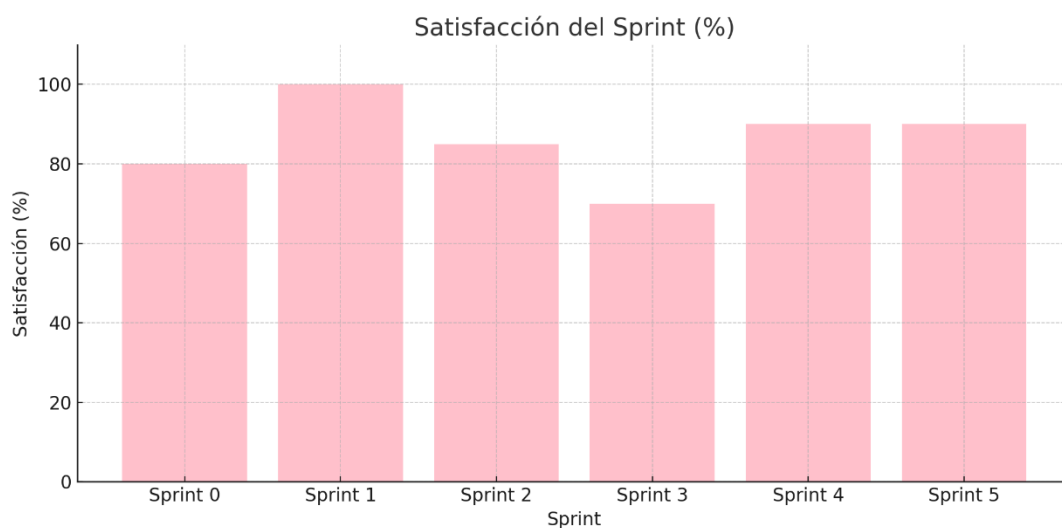


Ilustración 48. Satisfacción del Sprint (%). Elaboración propia

Tareas Pendientes: El número de tareas no completadas al final de cada sprint proporciona una visión de la carga de trabajo y la eficacia del equipo para gestionar sus responsabilidades dentro de los plazos planificados.

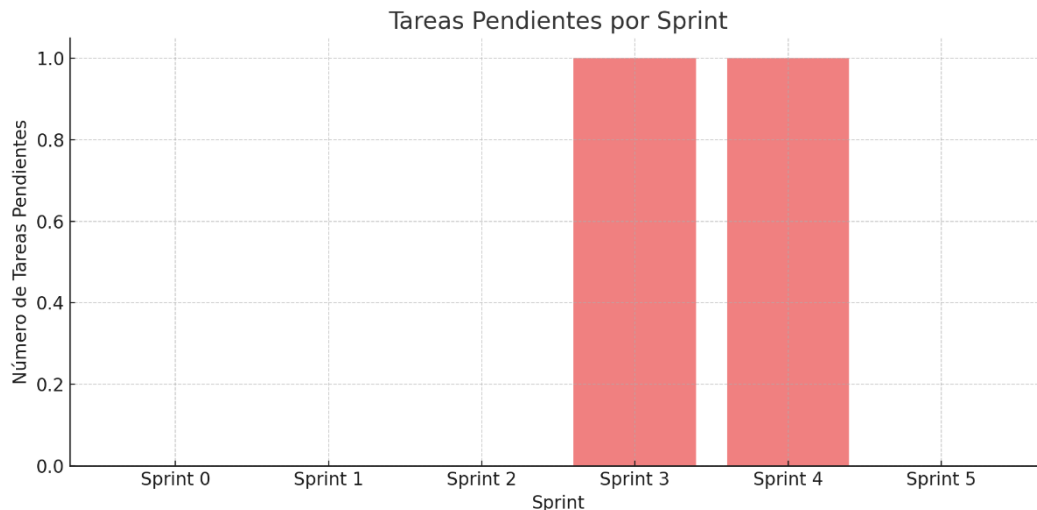


Ilustración 49. Tareas pendientes por Sprint. Elaboración propia

7.3.3. Análisis de Tendencias

Los gráficos presentados se han utilizado para visualizar las tendencias y variaciones en los KPIs a lo largo de los sprints, estos ayudan a identificar patrones como la variación de la satisfacción de cada sprint o la eficiencia de la planificación realizada al inicio del proyecto.

- **Gráfico de Horas Planificadas vs. Horas Reales:** Permite visualizar la precisión de la planificación y cómo se ajusta a lo largo del proyecto.
- **Gráfico de Satisfacción del Sprint:** Demuestra la relación entre la carga de trabajo y la satisfacción del equipo, ayudando a gestionar mejor las expectativas y el bienestar del equipo.
- **Gráfico de Tareas Pendientes:** Indica la eficiencia en la gestión del tiempo y la asignación de tareas.

Se concluye que estos indicadores no solo ayudan a mantener un curso fluido del proyecto, sino que también facilitan la implementación de mejoras durante el desarrollo del trabajo. Estas comparaciones complementan la experiencia del desarrollo, garantizando una mejora de los métodos y metodologías a implementar en futuros proyectos.

7.4. Escalabilidad / extensión / mantenimiento / Soporte

7.4.1. Escalabilidad y extensión

Debido a que el proyecto está relacionado principalmente con la investigación y no diseñado para un despliegue a gran escala refiriéndonos en términos como hardware o infraestructura. Si no, este proyecto en términos de escalabilidad está relacionado en la aplicación en datasets más grandes o sistemas integrados.

La principal vía de extensión del proyecto considerada sería su incorporación a una interfaz de usuario donde los investigadores podrían subir sus datasets para un análisis usando el modelo entrenado. Una idea inspirada en la página web de BIO-HPC, específicamente el proyecto Bruselas (Luna, A. J. B., Sánchez, D. H. E. P., & Carrasco, D. J. P. C. (2019)).

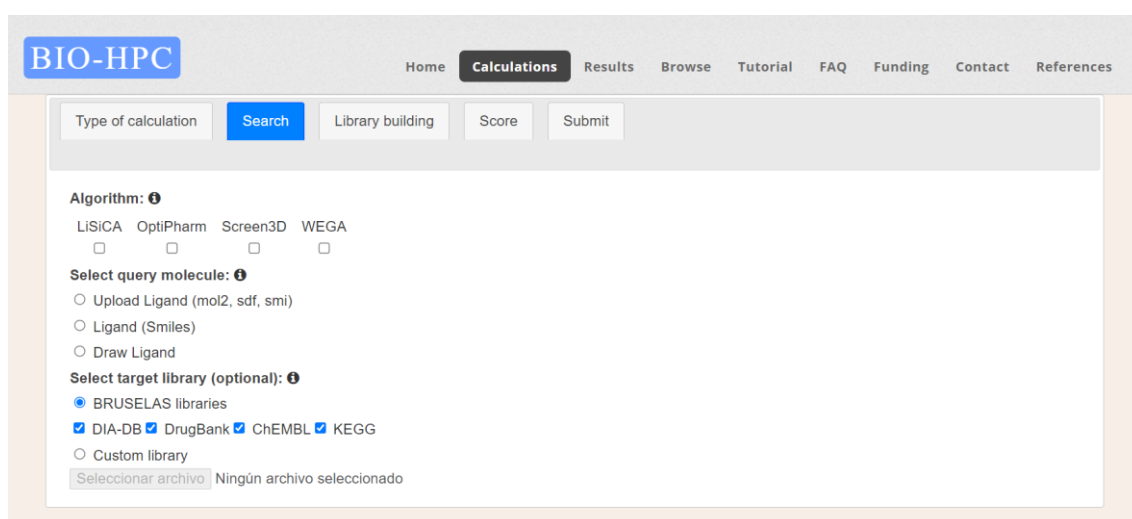


Ilustración 50. Captura página web del proyecto Bruselas.

De una manera muy similar se podría extender el actual proyecto, integrándolo en una interfaz, proporcionando accesibilidad a cualquier persona.

7.4.2. Mantenimiento y soporte

Respecto al mantenimiento del sistema, por ahora, está relacionada al control de versiones y su actualización. Sin embargo, si se considera continuar con la investigación, desarrollando más modelos o integrando futuros hallazgos de tecnología relacionada con algoritmos de DL y ML, será esencial establecer un

plan de mantenimiento que permita incorporar estas innovaciones de manera eficiente.

7.5. Plan de formación de usuarios

7.5.1. *Despliegue notebook*

La visualización es muy sencilla, solo se debe ingresar al siguiente link:

<https://colab.research.google.com/drive/1aqWwAyFFQyWthocZQnrV27VZdkWJchLB?usp=sharing>

Al ingresar se visualiza el notebook, en la izquierda se encuentra el índice para ir directamente a algún apartado específico. Además, se visualiza cómo están todas las celdas ejecutadas, se encuentra información sobre el dataset, cómo está compuesto, resultados de los modelos entrenados, ejecución de la interpretabilidad y demás.

7.5.2. *Interpretación de imágenes*

Para hacer comprensible el significado de las gráficas, interpretaremos dos en este apartado que se consideran un poco más confusas de entender. Igualmente, en el capítulo 6, se encuentran muchas gráficas con sus interpretaciones.

7.5.2.1. *Pairplot de Atributos Numéricos*

Este gráfico, nos ayuda a ver la relación entre tres características importantes de pacientes con cáncer de mama: la edad, el tamaño del tumor y la cantidad de nodos invadidos, que son indicadores de si el cáncer se ha extendido.

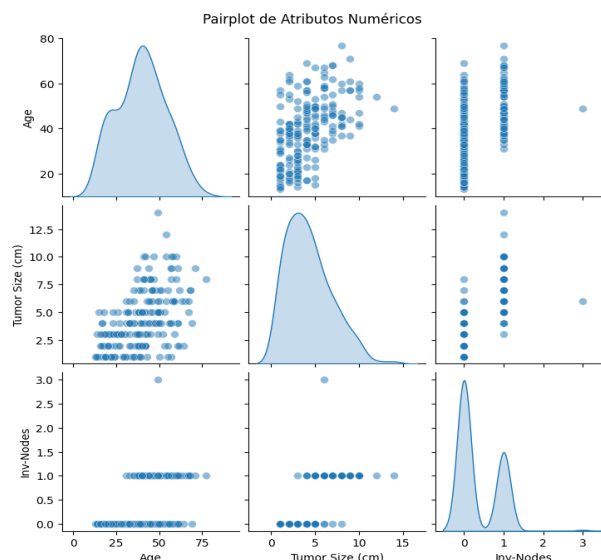


Ilustración 51. Pairplot atributos numéricos explicación. Código Google Colab.

Interpretación:

- Sobre las Personas:
 - Edad: La mayoría de los pacientes son alrededor de los 50 años. Esto significa que, en este grupo de pacientes, esa es la edad más común.
 - Tamaño del Tumor: Los tumores suelen ser pequeños, mayormente alrededor de 2.5 centímetros, siendo estos más fáciles de manejar.
 - Nodos Invadidos: La mayoría de los pacientes no tienen nodos invadidos, lo que es un indicativo de que el cáncer no se ha extendido mucho.
- Relaciones entre las Características:
 - Edad y Tamaño del Tumor: No parece haber una relación directa entre la edad de una persona y el tamaño de su tumor, es decir, el tamaño del tumor puede variar independientemente de la edad.
 - Edad y Nodos Invadidos: Al igual que con el tamaño del tumor, la edad no parece influir en la cantidad de nodos invadidos.
 - Tamaño del Tumor y Nodos Invadidos: No hay una relación clara que muestre que, a mayor tamaño del tumor, más nodos estén invadidos.

7.5.2.2. Matriz de Correlación

Este gráfico colorido muestra cómo diferentes variables, como la edad, el tamaño del tumor, y otros factores están relacionados entre sí. Los colores más cálidos (como el rojo) indican una correlación positiva fuerte, mientras que los colores más fríos (como el azul) indican una correlación negativa.

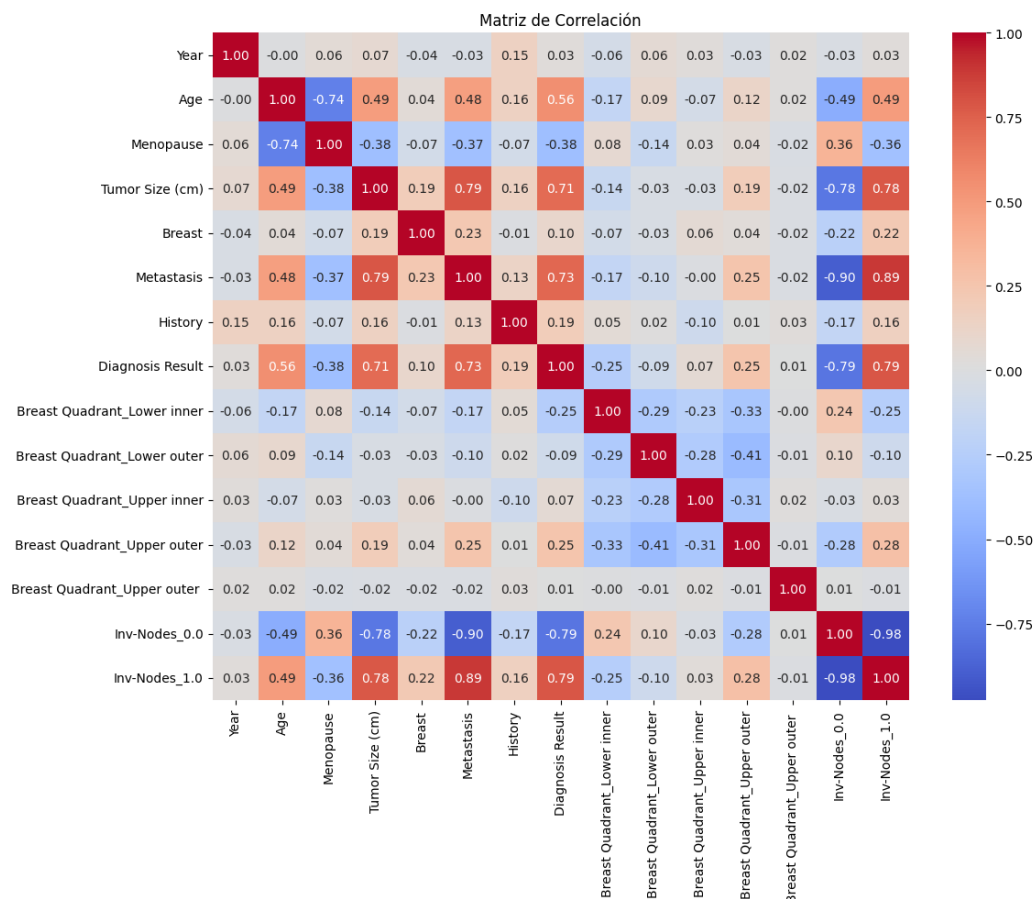


Ilustración 52. Explicación matriz de correlación. Código Google Colab.

Interpretación:

- **Relaciones Fuertes:** Variables como el tamaño del tumor y la historia clínica de los pacientes muestran algunas correlaciones fuertes. Por ejemplo, el tamaño del tumor está positivamente correlacionado con la presencia de metástasis, lo que podría indicar que tumores más grandes tienen mayor probabilidad de haberse diseminado.
- **Diferencias por Cuadrante:** La correlación también varía según el cuadrante del seno afectado, sugiriendo que la ubicación del tumor podría influir.

- Independencia de Otros Factores: La edad y el tamaño del tumor no están fuertemente correlacionados con muchos otros factores, lo que subraya la complejidad del cáncer de mama y la importancia de considerar múltiples variables al evaluar el riesgo y el tratamiento.

8. CONCLUSIONES

8.1. Conclusiones del trabajo y personales

Cuando tenía 11 años la madre de mi muy querida amiga en aquel entonces, Linda, sufría de cáncer de mama, recuerdo ver cómo usaba pañuelos en la cabeza para cubrir los efectos de la quimioterapia, era una increíble persona y una maravillosa madre, muy dulce, amigable y amable. Recuerdo estar saliendo del cine con todas mis amigas cuando nos contaron la noticia, mi amiga se había quedado sin madre ¿Qué es justo en la vida? ¿Cómo es posible que alguien tan joven pueda fallecer? No tenía ni 43 años.

Varios años después mi propia madre se llevó el gran susto, un tumor había aparecido, tenía 47 ¿Será cáncer? ¿Por qué tenemos la costumbre de que si aparece un tumor será maligno? Muchas preguntas te pasan por la cabeza y pocas respuestas obtienes.

La madre de mi pareja sufrió de cáncer de mama cuando tenía solo 33 años, un viaje muy doloroso, solo las valientes de mente pueden sobrevivir, porque decir “Yo sufrí de cáncer de mama y sobreviví” se ha convertido en un privilegio y un milagro.

¿Cómo las mujeres pueden saber qué evitar para no tener esta enfermedad, cuando ni los científicos tienen la respuesta? La motivación de este proyecto es muy clara en este punto, miedo.

Cuando eres mujer, lamentablemente, te expones mucho, incluso a morir por esta enfermedad y yo, al ser una, también. Una vez le pregunté a un sabio profesor ¿Por qué elegiste la bioinformática? Me dijo: “Ya que estamos aquí, habrá que hacer algo bueno” Así que, esta es la motivación de este proyecto, lleno de esperanza, amor e interés. Y aunque esto no cambie el mundo hoy, no significa que no lo pueda cambiar mañana.

8.2. Vías futuras

Este trabajo se considera no solo inspirado en numerosos trabajos relacionados con el campo, sino también basado en investigaciones actuales realizadas por diversos institutos científicos en diferentes partes del mundo. Se espera que, con el esfuerzo y el avance en el aprendizaje profundo y el aprendizaje automático, podamos alcanzar una predicción temprana del cáncer, no sólo identificándose una vez que ya ha comenzado, sino incluso antes de su aparición. Asimismo, se busca comprender las causas biológicas que subyacen a esta enfermedad.

Como se ha observado en el análisis de los gráficos, el riesgo de cáncer no se limita a las personas mayores de 50 años; incluso personas de 18 años pueden padecerlo. Por lo tanto, se espera llevar este proyecto a un nivel superior en el futuro, más allá de la fase de investigación. Sería un éxito personal convertirlo en una herramienta práctica y útil en la vida real, en lugar de dejarlo como un proyecto de fin de grado.

Se cree firmemente que este tipo de investigaciones tiene un gran potencial y un futuro prometedor. Con una mayor recolección de datos, no solo más recientes, sino también más amplios, se podrán obtener resultados más eficaces. La gran meta es implementar estas soluciones a nivel experto, para que los profesionales médicos puedan utilizarlo en su práctica diaria, beneficiando así no solo a la comunidad científica, sino también a la vida de las personas, teniendo un impacto positivo significativo en la sociedad.

9. BIBLIOGRAFÍA

Adesola, J. (2023, March 9). Model Evaluation Metrics for Binary Classification - Josiah Adesola - Medium. Medium. (<https://josiah-adesola.medium.com/model-evaluation-metrics-for-binary-classification-a-practical-and-well-explanatory-approach-608c89dce67f>)

American Cancer Society. (2021). Sentinel Lymph Node Biopsy. Recuperado de (<https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/tests-who-why-how-is-breast-cancer-diagnosed/sentinel-lymph-node-biopsy.html>)

Asana, T. (2024, 20 enero). Matriz de prioridades: cómo identificar lo que realmente importa y lograr mucho más [2024] • Asana. Asana. (<https://asana.com/es/resources/priority-matrix>)

Bermejo, M. (s. f.). El Kanban. Fundación Para la Universitat Oberta de Catalunya.

Biecek, P. (2018, 23 junio). DALEX: explainers for complex predictive models. arXiv.org. (<https://arxiv.org/abs/1806.08915>)

Bruselas. (s. f.). (<https://bio-hpc.ucam.edu/Bruselas/web/Calculations/Calculations.php>)

Cancer of the Breast (Female) - Cancer Stat Facts. (s. f.). SEER. (<https://seer.cancer.gov/statfacts/html/breast.html>)

Cruz-Roa, A., Gilmore, H., Basavanhally, A., Feldman, M., Ganesan, S., Shih, N., ... & Tomaszewski, J. (2018). Accurate and reproducible invasive breast cancer detection in whole-slide images: A Deep Learning approach for quantifying tumor extent. Scientific Reports, 7(1), 46450.

Explainable AI for Breast Cancer Detection: A LIME-Driven Approach. (2023, 18 diciembre). IEEE Conference Publication | IEEE Xplore. (<https://ieeexplore.ieee.org/document/10469341>)

Hastie, T., Tibshirani, R., & Friedman, J. (2017). The Elements of Statistical Learning (2.a ed.). Springer.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, 13, 8-17.

Layman, Williams, L., & Cunningham, L. (s. f.). Exploring Extreme Programming in Context: An Industrial Case Study. North Carolina State University, Department Of Computer Science.
(<https://collaboration.csc.ncsu.edu/laurie/Papers/ADC.pdf>)

Letelier, P., & Penadés, M. C. (2015, enero). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). ResearchGate.
(https://www.researchgate.net/profile/Patricio-Letelier/publication/28109707_Metodologias_agiles_para_el_desarrollo_de_software_eXtreme_Programming_XP/links/54ad00f10cf2479c2ee86820/Metodologias-agiles-para-el-desarrollo-de-software-eXtreme-Programming-XP.pdf)

Lundberg, S., & Lee, S. (2017, 22 mayo). A Unified Approach to Interpreting Model Predictions. arXiv.org. (<https://arxiv.org/abs/1705.07874>)

MULTIOMICA – Bygens. (s. f.). (<https://bygens.com/multiomica/>)

National Cancer Institute. (2021). Breast Cancer. Recuperado de (<https://www.cancer.gov/types/breast>)

Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013, diciembre). Revisión de metodologías ágiles para el desarrollo de software. Revisión de Metodologías Ágiles Para el Desarrollo de Software.
(<https://www.redalyc.org/pdf/4962/496250736004.pdf>)

Nhgri. (2019, 9 marzo). Transcriptoma. Genome.gov.
(<https://www.genome.gov/es/about-genomics/fact-sheets/Transcriptoma>)

Quiñónez-Ku, X., Casierra Cavada, J., Herrera-Izquierdo, L., & Mera Quiroz, J. (2019, junio). ANÁLISIS COMPARATIVO DE METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE: UNA REVISIÓN BIBLIOGRÁFICA. ResearchGate.
(https://www.researchgate.net/profile/Xavier-Quinonez-Ku-2/publication/334000094_ANALISIS_COMPARATIVO_DE_METODOLOGIAS_AGILES_DE_DESARROLLO_DE_SOFTWARE_UNA_REVISION_BIBLIOGRA)

[FICA/links/5d1248b2299bf1547c7ed1ce/ANALISIS-COMPARATIVO-DE-METODOLOGIAS-AGILES-DE-DESARROLLO-DE-SOFTWARE-UNA-REVISION-BIBLIOGRAFICA.pdf](https://www.fica.es/links/5d1248b2299bf1547c7ed1ce/ANALISIS-COMPARATIVO-DE-METODOLOGIAS-AGILES-DE-DESARROLLO-DE-SOFTWARE-UNA-REVISION-BIBLIOGRAFICA.pdf))

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). «Why Should I Trust You?»: Explaining the Predictions of Any Classifier. arXiv.org. (<https://arxiv.org/abs/1602.04938>)

Santos, P. S. M. D., Beltrão, A. C., De Souza, B. P., & Travassos, G. H. (2018). On the benefits and challenges of using kanban in software engineering: a structured synthesis study. Journal Of Software Engineering Research And Development, 6(1). (<https://doi.org/10.1186/s40411-018-0057-1>)

Sassa, A. C., Alves de Almeida, I., Fernandes Pereira, T. N., & Silva de Oliveira, M. (2023). Scrum: A Systematic Literature Review. (IJACSA) International Journal Of Advanced Computer Science And Applications, 14(4). (https://thesai.org/Downloads/Volume14No4/Paper_20-Scrum_A_Systematic_Literature_Review.pdf)

Schäfer, J. (2023, October 12). Die 20+ wichtigsten Scrum Statistiken für das Jahr 2023. Echometer. (<https://echometerapp.com/es/estadisticas-del-scrum/>)

Schwaber, K., & Sutherland, J. (2020, noviembre). La guía Scrum: La Guía Definitiva de Scrum: Las Reglas del Juego. 2020-Scrum-Guide-Spanish-European.pdf. (<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>)

Taghizadeh, E., Heydarheydari, S., Saberi, A., JafarpourNesheli, S., & Rezaeij, S. M. (2022). Breast cancer prediction with transcriptome profiling using feature selection and machine learning methods. BMC Bioinformatics, 23(1). (<https://doi.org/10.1186/s12859-022-04965-8>)

Tong, L., Mitchel, J., Chatlin, K., & Wang, M. D. (2020). Deep learning based feature-level integration of multi-omics data for breast cancer patients survival analysis. BMC Medical Informatics And Decision Making, 20(1). (<https://doi.org/10.1186/s12911-020-01225-8>)

World Health Organization: WHO. (2022, 3 febrero). Cáncer.
(<https://www.who.int/es/news-room/fact-sheets/detail/cancer>)

Xq. (2023, July 15). Exploring GANs to generate synthetic data - the Research Nest - medium. Medium. (<https://medium.com/the-research-nest/exploring-gans-to-generate-synthetic-data-ca48f8a4b518>)

