

TRABAJO FINAL DE MÁSTER

IA EN LA GESTIÓN DE CITAS MÉDICAS:
INNOVANDO EN CONFIRMACIÓN Y REDUCCIÓN
DE AUSENCIAS.

Descripción breve

La utilización de inteligencia artificial (IA) en la gestión de citas médicas tiene como objetivo mejorar la asignación de estas, disminuyendo los “non show” con el objetivo de mejorar la eficiencia de los servicios de salud ambulatorios

Grupo 3

Hernando Acevedo Aguilar
Michelle Alexandra Chicaiza Anrrango
Luis Marcelo Ortiz Carinao
Sergio Valdueza Lozano

Entrega Parcial 02 agosto 2024

Resumen ejecutivo del proyecto

Uno de los principales problemas de los servicios de salud tanto de España como de Latinoamérica son los tiempos de espera para consultas médicas de especialidades. En España para acceder a una atención especializada, los pacientes esperan una media de 79 días para ser atendidos en primera consulta con un rango entre 22 y 107 días (1). Mientras que el sistema de salud chileno enfrenta un desafío crítico donde al 31 de diciembre del 2022 se alcanzaron 61.191 prestaciones de especialidad retrasadas, donde cada prestación promediaba un retraso de 156.5 días (2).

Esto se explica por múltiples factores, tales como la brecha de especialistas (3), y la inasistencia a consultas médicas, que en algunos centros alcanza hasta el 20% del total de usuarios (4) (5).

Debido a esta problemática es que planteamos el desafío de integrar herramientas de inteligencia artificial con el objetivo de mejorar la asignación de citas médicas, mediante la predicción de una probabilidad de inasistencia ("non show") con herramientas de machine learning, y en base a esta probabilidad de inasistencia programar agendas médicas más compactas de forma que se usen los recursos de los centros médicos de forma más intensiva, reduciendo la cantidad de pacientes en listas de espera.

El proyecto "Aplicación de Inteligencia Artificial en la Gestión de Citas Médicas: Optimización y Reducción de Ausencias" aborda este reto crucial mediante el desarrollo de un sistema inteligente que optimiza la asignación de citas, reduce las ausencias y minimiza los tiempos de espera. El proyecto se estructura en cuatro fases interconectadas: Predicción de asistencia a citas médicas, Implementación de un sistema de overbooking inteligente, Creación de un asistente virtual para la gestión de citas e Integración mejora continua del sistema.

Los resultados obtenidos son notablemente prometedores. En la primera fase, se desarrolló un modelo predictivo de DP (NeuralNet y TabNet) con un AUC significativamente superior a los métodos tradicionales (modelos de ML), alcanzando un AUC de 0.74. La implementación del sistema de overbooking en la segunda fase demostró un potencial de ahorro económico considerable, estimando que un centro hospitalario con 20 consultas médicas podría ahorrar hasta 1.500.000€ anuales.

La tercera fase introdujo "Basilio", un asistente virtual de vanguardia altamente eficaz en la gestión de citas e interacción con pacientes, mejorando significativamente la accesibilidad y experiencia del usuario. Finalmente, el proceso de refinamiento continuo asegura la evolución y adaptación constante del sistema.

En conjunto, los resultados son positivos, logrando no solo los objetivos de optimización y eficiencia, sino también abriendo nuevas posibilidades para la aplicación de la inteligencia artificial en el sector sanitario. La reducción de tiempos de espera, la optimización de recursos y la mejora en la accesibilidad a servicios médicos son logros tangibles que demuestran el éxito del proyecto.

Sin embargo, el estudio también reconoce los desafíos inherentes a la implementación de tecnologías avanzadas en el sector sanitario, abordando con rigor aspectos como la precisión de las predicciones, la ética en el manejo de datos médicos y la necesidad de validación en entornos reales.

Contenido

Resumen ejecutivo del proyecto	1
Introducción.....	3
Núcleo del proyecto	6
Definición del proyecto y análisis de viabilidad.....	6
Descripción detallada de la empresa	6
Análisis interno y externo.....	7
Explicación detallada del proyecto	8
Fijación de los objetivos generales y específicos del proyecto	13
Planificación.....	14
Estimación de recursos económicos	14
Estimación de recursos materiales.....	14
Estimación de recursos humanos.....	14
Estimación de los recursos tiempo.....	14
Definición del alcance del proyecto	15
Elaboración del cronograma del proyecto.....	15
Desarrollo del proyecto	18
Fase 1. Predicción de Asistencia a Citas Médicas	18
Fase 2. Implementación de un sistema de overbooking	44
Fase 3. Creación de un Asistente Virtual basado en el Procesamiento de Lenguaje Natural (NLP)	58
Fase 4. Proceso de Integración del Sistema.....	64
Conclusiones	66
Bibliografía	70
Tabla de Figuras	72

Introducción

MediAgenda Solutions, S.L. (empresa ficticia para fines académicos), es una empresa líder en el sector de la salud, especializada en el uso intensivo de Inteligencia Artificial para la administración y gestión de agendas médicas para hospitales y centros médicos en España y Latinoamérica. Nuestra actividad se centra en gestionar las agendas de visita a los especialistas, con la misión de optimizar la asignación de horarios médicos, reducir los tiempos de espera y mejorar la calidad de la atención médica.

En España, para acceder a una atención especializada, los pacientes esperan una media de 79 días hasta ser atendidos en primera consulta, con un rango entre 22 y 107 días [\(1\)](#).

Mientras que en Latinoamérica se enfrenta un desafío de salud crítico, particularmente en Chile, donde pese a tener patologías priorizadas por ley con tiempos de espera máximo (garantías explícitas de salud GES), al 31 de diciembre del 2022 se alcanzaron 61.191 garantías retrasadas, con un promedio de 156.5 días de retraso por garantía [\(2\)](#). Estos retrasos se explican por múltiples factores, como la brecha de especialistas [\(3\)](#) y la inasistencia a consultas médicas, que en algunos centros alcanza hasta el 20% [\(4\)](#) [\(5\)](#).

La falta de asistencia a citas programadas, o "No Show", es un obstáculo recurrente que entorpece la eficiencia operativa, genera desequilibrios en la programación de los médicos y prolonga las listas de espera para ser atendido por un especialista.

Para abordar esta problemática, proponemos integrar herramientas de inteligencia artificial para mejorar la asignación de citas médicas mediante la predicción de la probabilidad de inasistencia ("No Show") usando técnicas de machine learning. Con base en esta probabilidad, se optimizarán las agendas médicas mediante un sistema de overbooking, similar al utilizado en otros sectores (aviación, hoteles, etc.), con el fin de mejorar la ocupación de los recursos, y reducir así los retrasos en las consultas de especialistas.

Además, desarrollaremos una interfaz de usuario que facilite la concertación y confirmación de citas de forma eficiente y amigable. Esta herramienta permitirá agendar citas, generar recordatorios y recopilar información útil para mejorar las predicciones de asistencia con nuestro modelo de IA, la integración de estos módulos se desarrollará en este proyecto solo a nivel teórico.

Para alcanzar los objetivos planteados, el proyecto se desarrollará en varias fases, con una primera fase que constituye el núcleo fundamental del mismo, la cual consta a su vez de varias subfases. Inicialmente, se desarrollará un modelo de IA para predecir la asistencia a citas médicas, comenzando con la recopilación de datos y un análisis exploratorio para identificar variables clave y crear características significativas. Se aplicará el particionamiento de los datos siguiendo distintos criterios para entrenar y evaluar los modelos, empleando algoritmos como regresión logística, árbol de decisión y redes neuronales, realizando una optimización de los modelos con búsqueda exhaustiva de los mejores parámetros. Los modelos se clasificarán según su rendimiento, seleccionando los más efectivos en cada particionamiento para su implementación en la gestión de overbooking. El proceso descrito anteriormente se respalda mediante el uso de Python como lenguaje de programación principal, aprovechando sus entornos de desarrollo como Google Colab, Visual Studio Code y Jupyter Notebook de Anaconda.

En la segunda fase, se utilizarán las predicciones de asistencia para optimizar las agendas médicas posibilitando la asignación de citas múltiples a un mismo slot, generando cupos de overbooking, para maximizar el uso de recursos sin generar tiempos de espera excesivos ni sobrecargar los servicios médicos. Para ello se definirán los costes de una agenda mal gestionada en función de los tiempos de inactividad en la clínica por falta de pacientes a atender, los tiempos de espera de los pacientes para

ser atendidos en la propia consulta médica, y los tiempos de trabajo extra realizados fuera del horario laboral oficial para atender a los pacientes rezagados. Dichos costes se evaluarán primeramente en las agendas médicas obtenidas con las reglas de asignación de citas tradicionales más comúnmente utilizadas, y a posteriori con agendas médicas obtenidas mediante una función que tiene en cuenta las probabilidades de asistencia de los pacientes, obtenidas con el modelo de predicción de la Fase 1. Finalmente se desarrolla un algoritmo que optimiza dicha función con un modelo de IA.

En la tercera fase, se desarrollará un Asistente Virtual basado en Procesamiento de Lenguaje Natural (NLP) para gestionar citas y recordatorios, mejorando la experiencia del usuario y optimizando el proceso de programación y seguimiento de citas médicas. Este asistente de tercera generación, además de agendar y gestionar citas, proporcionará información básica sobre especialidades médicas y medicamentos, e incluso podrá procesar imágenes. La implementación de este Asistente Virtual integrará tecnologías como Google Sheets para el manejo de datos y diversas APIs para funcionalidades específicas, resultando una herramienta que no solo mejorará la accesibilidad y satisfacción del paciente, sino que también optimizará los recursos del hospital, representando un paso significativo en la modernización de los servicios de salud.

En la cuarta fase se refinará el sistema a nivel teórico, integrando la base de datos que se utiliza en el Asistente Virtual con los modelos predictivos de asistencia y los de optimización para la asignación de citas, y cerrando el círculo con una retroalimentación periódica (por definir) a los modelos de IA que les ayude a mejorar la precisión en las predicciones y, por ende, en la optimización de agenda médica obtenida.

Resumen de los resultados finales obtenidos

Fase 1: Predicción de asistencia a citas médicas

Se desarrollaron y evaluaron varios modelos de machine learning, destacando TabNet como el más efectivo. Este modelo alcanzó un AUC (Área Bajo la Curva ROC) de 0.74, demostrando una capacidad discriminativa superior a los métodos tradicionales. TabNet también logró el mejor F1-score, llegando a 0.49, indicando un equilibrio óptimo entre precisión y recall. La implementación de técnicas de feature engineering y balanceo de datos como SMOTE-ENN y ADASYN contribuyó significativamente a estos resultados. Notablemente, se alcanzó un recall de hasta 83.7% utilizando una red neuronal en el conjunto de datos P1-ALL-ADASYN, lo cual es particularmente valioso para minimizar los falsos negativos en la predicción de no-shows.

Fase 2: Implementación de un sistema de overbooking

La implementación del sistema de overbooking basado en las predicciones de la Fase 1 demostró una mejora significativa en la eficiencia y reducción de costos. El escenario más efectivo (Escenario 3 ProbShow_0.8_0.15) logró una reducción del 26% en el costo total del sistema en comparación con el escenario tradicional sin overbooking. En términos económicos, se estimó un ahorro potencial de hasta 1,500,000€ anuales para un centro hospitalario con 20 consultas médicas. Además, el modelo demostró una mayor estabilidad, con una desviación estándar en los costos totales reducida casi a la mitad en comparación con otros escenarios, indicando una mayor robustez y previsibilidad de coste en la gestión de citas.

Fase 3: Creación de un asistente virtual basado en NLP

Se desarrolló con éxito "Basilio", un asistente virtual de tercera generación utilizando el modelo GPT-4o y técnicas de Generación Aumentada por Recuperación (RAG). Basilio demostró capacidades avanzadas en la gestión de citas médicas, incluyendo programación, modificación y cancelación de citas. El asistente logró procesar y responder a consultas de usuarios, manteniendo una buena precisión en la información proporcionada sobre especialidades médicas y medicamentos. La integración con Telegram permitió una mejor accesibilidad.

Fase 4: Integración del sistema

A nivel teórico, desarrollamos un proceso de refinamiento continuo que integra la base de datos de la interfaz creada para gestionar las citas y los recordatorios con los modelos predictivo y de optimización. Este proceso permite la actualización periódica de los modelos de IA, incluyendo incluso la posibilidad de introducir en algún momento nuevas características predictoras relevantes, mejorando la precisión de las predicciones de asistencia y, por ende, la optimización en la preparación de agendas médicas de menor coste. Además, la integración del sistema de gestión de citas con la interfaz de usuario facilita una experiencia más fluida y eficiente para los pacientes, otorgando funcionalidades paralelas que mejoran la experiencia de los usuarios y ayudan también a disminuir la carga de trabajo en las consultas médicas.

Resultados clave:

- **Reducción de Tiempos de Espera Global:** La implementación del sistema de overbooking, junto con las predicciones precisas del modelo de IA, permite una reducción en los tiempos de espera para la atención médica, mejorando la satisfacción de los pacientes.
- **Optimización de Recursos Médicos:** La utilización eficiente de los recursos médicos a través del sistema de overbooking y la mejora en la programación de citas contribuye a un uso más efectivo y racional de los mismos, generando ahorros económicos y operativos.
- **Mejora Atención al Cliente:** La puesta en marcha de un Asistente Virtual de 3^a generación permite interactuar con el Centro Hospitalario con un nuevo canal de comunicación, permitiendo una gestión ágil y práctica de las citas médicas, ofreciendo también otros servicios relacionados como recordatorios de citas médicas, consultas de prescripciones médicas, asesoramiento inicial para la concertación de visitas médicas, etc.

Núcleo del proyecto

Definición del proyecto y análisis de viabilidad

Descripción detallada de la empresa

Se plantea la creación de MediAgenda Solutions, S.L. como empresa líder en el sector de la salud, especializada en la administración inteligente de agendas médicas para hospitales y centros médicos en España y Latinoamérica.

Nos enfrentamos al desafío crítico de los largos tiempos de espera en la atención médica, una problemática que impacta tanto a los pacientes, quienes experimentan retrasos significativos en su atención, como a los profesionales de la salud, que deben hacer frente a agendas sobrecargadas y recursos limitados.

Un ejemplo ilustrativo de esta problemática se encuentra en el barómetro sanitario realizado en España por el Ministerio de Sanidad, Consumo y Bienestar Social (6) en colaboración con el Centro de Investigaciones Sociológicas. Según los datos recogidos en el informe correspondiente al año 2023, el 27.2% de los ciudadanos reportaron haber esperado "11 días o más" desde que solicitaron la cita hasta que fueron atendidos por el médico de familia, evidenciando así los prolongados tiempos de espera que afectan a la población.

SÓLO A QUIENES EN LOS ÚLTIMOS DOCE MESES HAN TENIDO ALGUNA CONSULTA CON UN/A MÉDICO/A DE CABECERA O DE FAMILIA DE LA SANIDAD PÚBLICA Y PASÓ MÁS DE UN DÍA HASTA QUE LES ATENDIERON PORQUE NO HABÍA CITA ANTES (1 o 3 en P6 y 3 en P6b) (N=1.388)	
--	--

Pregunta 6b01

¿Cuántos días?

1 día	0,1
2 días	11,4
3 días	12,7
4 días	6,8
5 días	7,3
6 días	1,8
7 días	18,5
Entre 8 y 10 días	12,4
11 y más días	27,2
No recuerda	1,7
N.C.	0,1
(N)	(1.388)
Media (días)	9,48
Desviación típica	9,47
(N)	(1.363)

Figura 1. Tiempo de espera en días para consulta a médico de familia.

Con el objetivo de abordar este problema, hemos centrado nuestros esfuerzos en el desarrollo de soluciones innovadoras que optimicen la gestión de citas médicas mediante la aplicación de tecnologías avanzadas, especialmente inteligencia artificial (IA). Nuestros avanzados algoritmos de machine learning analizan una amplia gama de datos, desde historiales médicos hasta patrones de comportamiento previos, para predecir con precisión la probabilidad de inasistencia de los pacientes a sus citas médicas. Esta información se utiliza para ajustar de manera inteligente las agendas médicas, maximizando así la eficiencia de los recursos médicos disponibles y reduciendo los tiempos de espera para los pacientes.

Además de los desafíos asociados con los largos tiempos de espera en la atención médica, otro problema significativo que enfrentan tanto pacientes como profesionales de la salud es el alto porcentaje de absentismo en las citas médicas. Este fenómeno no solo genera costos económicos para

los sistemas de salud, sino que también puede afectar negativamente la calidad y eficiencia de la atención médica.

Conscientes de esta problemática, MediAgenda Solutions, S.L. desarrollará una solución innovadora que aborde directamente el problema del absentismo. Nuestra plataforma utiliza algoritmos avanzados que permiten una mejor planificación de los recursos médicos con una oportunidad de atención más eficiente.

La eficacia de las innovaciones tecnológicas en el ámbito médico se ha demostrado en términos de ahorro de tiempo tanto para los pacientes como para los profesionales de la salud, lo que se traduce en una reducción de costos. Estas innovaciones no solo mejoran la organización de los horarios y una utilización óptima de los recursos hospitalarios, sino que también permiten una actualización inmediata de la información del paciente y una mayor flexibilidad en la programación de citas.

En resumen, la inversión en tecnología aplicada a la gestión de citas médicas es crucial para optimizar los procesos, aumentar la satisfacción de los usuarios y avanzar hacia sistemas de salud más accesibles y sostenibles.

La empresa tiene su sede principal en Barcelona, España, desde donde coordina sus operaciones y desarrollo tecnológico. Sin embargo, su alcance abarca a nivel nacional e internacional, con el objetivo de ofrecer soluciones a los problemas de tiempos de espera en los servicios de salud en países de habla hispana, especialmente en España y Latinoamérica.

Análisis interno y externo

Con el propósito de obtener una visión detallada y completa de la posición y perspectivas de MediAgenda Solutions, S.L. en el sector de la salud, se realizará un análisis tanto de los factores internos como externos que influyen en la empresa.

Análisis interno

Innovación Tecnológica: MediAgenda Solutions, S.L. se destaca por fomentar una cultura de innovación y colaboración, y se mantiene actualizada en las últimas tendencias y desarrollos en el campo de la inteligencia artificial y la tecnología de la salud.

Equipo Multidisciplinario: La empresa cuenta con un equipo altamente calificado y multidisciplinario, conformado por profesionales en áreas como inteligencia artificial, medicina, gestión de proyectos, diseño de experiencia de usuario (UX/UI) y desarrollo de software con el objetivo de desarrollar y ofrecer una solución innovadora y eficaz para abordar el problema de los largos tiempos de espera en las consultas médicas especializadas. Esta diversidad de talentos permite una visión integral en el desarrollo de sus productos y servicios.

Alianzas Estratégicas: MediAgenda Solutions, S.L. ha establecido alianzas estratégicas con instituciones médicas, centros de salud y organizaciones del sector para colaborar en la implementación y mejora continua de sus soluciones. Estas alianzas fortalecen su posición en el mercado y les permiten acceder a una base de clientes potenciales más amplia.

Análisis externo

Demandas en Crecimiento: La creciente demanda de soluciones para reducir los tiempos de espera en los servicios de salud representa una oportunidad clave para MediAgenda Solutions, S.L. La necesidad de optimizar la gestión de citas médicas es un problema extendido en el sector, tanto en España como en Latinoamérica, lo que brinda un mercado potencialmente amplio para sus servicios.

Competencia: Aunque MediAgenda Solutions, S.L. es líder en la integración de inteligencia artificial en la gestión de citas médicas, enfrenta competencia de otras empresas que ofrecen soluciones

similares o alternativas tradicionales. La capacidad de innovación y diferenciación será crucial para mantener su posición en el mercado.

Regulaciones y Normativas: La empresa opera en un sector altamente regulado, sujeto a normativas específicas en materia de protección de datos, seguridad y calidad de servicios de salud. Cumplir con estas regulaciones es fundamental para ganar la confianza de clientes y usuarios finales, así como para mantener la reputación y credibilidad de la empresa.

Explicación detallada del proyecto

Para la ejecución de este proyecto se plantean 4 fases o retos que nos ayuden a conseguir el objetivo planteado:

Fase 1: Realizar un modelo de IA capaz de predecir la Asistencia o No Asistencia de los pacientes a las citas médicas.

Esta fase es esencial ya que sienta las bases para el desarrollo de soluciones efectivas en las etapas posteriores del proyecto.

- **Elección, Recolección y Preparación de datos.** Se realiza una búsqueda de una base de datos que contenga información sobre las citas médicas programadas en un servicio de salud, así como datos relacionados con los usuarios que las solicitan. Se elige una base de datos que incluya una etiqueta que indique si el paciente asistió o no a su cita médica (show – no show), ya que esta información es esencial para el entrenamiento y la evaluación del modelo de inteligencia artificial.
- **Análisis Exploratorio de los Datos (EDA).** Se realiza un exhaustivo EDA para comprender la estructura, la distribución y las relaciones dentro del conjunto de datos. Durante este proceso, se identifican y preparan las variables predictoras relevantes para la predicción de inasistencia a las citas médicas. Esto implica la limpieza de datos para tratar inconsistencias o valores faltantes, así como explorar las variables presentes en la base de datos en busca de relaciones, tendencias o anomalías. Este paso es crucial para garantizar la calidad y la confiabilidad de los datos utilizados en el entrenamiento del modelo.
- **Feature Engineering.** El proceso de Feature Engineering desempeña un papel fundamental en la preparación de los datos para la construcción de modelos de inteligencia artificial. Durante esta etapa, transformamos los datos brutos del dataset en características significativas que permiten a los modelos aprender patrones y realizar predicciones precisas. Este proceso se divide en varias etapas clave que incluyen la extracción de características, la transformación de tipos de datos, la creación de nuevas características, la selección de características relevantes, y la normalización y manejo de los datos faltantes.

Durante la **extracción de características**, se identifican datos relevantes del dataset original, como la información demográfica de los pacientes y datos relacionados con el historial médico y las visitas anteriores, los cuales son importantes para predecir la asistencia o no asistencia a las citas médicas.

La **transformación de tipos** de datos garantiza que los datos estén en un formato adecuado para su procesamiento por parte de los modelos de inteligencia artificial. Además, se crean nuevas características que pueden incluir información climática, ubicación geográfica, y variables relacionadas con el historial del paciente, entre otras.

La **selección de características** se basa en la evaluación de su correlación con la variable objetivo (no show), asegurando que solo se utilicen aquellas que tengan un mayor impacto en la predicción.

Finalmente, se aplican **técnicas de normalización, reducción de dimensionalidad**, manejo de **variables categóricas** y **balanceo de clases** para garantizar la calidad y eficacia del modelo de inteligencia artificial.

Este proceso de Feature Engineering sienta las bases para la construcción de modelos predictivos precisos en la gestión de citas médicas, permitiendo una mejor comprensión de los factores que influyen en la asistencia o no asistencia a las citas médicas.

- **Entrenar diferentes modelos de machine learning.** Se entrenarán diferentes modelos de machine learning, aplicando diversas técnicas y estrategias para asegurar una predicción precisa de la asistencia de los pacientes a sus citas médicas. Para el entrenamiento de los modelos de machine learning, se seguirá un enfoque metodológico detallado:
 - **Selección del conjunto de prueba.** Dado que no hay un conjunto de datos de prueba separado, se reservará una parte del conjunto original. Este subconjunto mantendrá la distribución de clases de original. La elección de este conjunto no se hará de forma aleatoria, se ordenará el dataset por fecha de cita programada, en orden descendente, seleccionando sólo la última cita de cada paciente, de forma que no se repita paciente alguno.
 - **Hipótesis y segmentación de datos.** Durante la preparación de los datos, se considerarán varias hipótesis para crear conjuntos de datos específicos que mejoren la capacidad predictiva de los modelos:
 - Conjunto de datos completo.
 - Pacientes sin condiciones médicas.
 - Pacientes de edad entre los 5 y 30 años.
 - Pacientes con citas programadas para otro día.
 - Pacientes de barrios con centro médico.
 - Pacientes segmentados por grupos de edad:
 - Niños: Menores de 12 años
 - Adolescentes: 13 - 18 años
 - Jóvenes adultos: 19 - 35 años
 - Adultos: 36 - 64 años
 - Adultos mayores: 65 años en adelante
 - **Entrenamiento de modelos de machine learning.** Para cada uno de los conjuntos de datos generados, se entrenarán varios modelos de machine learning. Entre los modelos seleccionados se incluyen regresión logística, árboles de decisión y redes neuronales. Durante el entrenamiento, se realizarán búsquedas de hiperparámetros y validaciones cruzadas para optimizar el rendimiento de cada modelo.
 - **Evaluación y optimización.** Con el objetivo de mejorar la precisión y la eficacia de los modelos, se realizará una evaluación exhaustiva de su desempeño. Esto incluirá la *optimización de la precisión* para minimizar el overbooking, es decir, para reducir al mínimo la situación en la que se asignan más citas de las que puede manejar el médico, evitando así la sobrecarga de pacientes en determinados horarios o días. Asimismo, se buscará *optimizar el recall* para minimizar los casos en los que los pacientes que se predijo que asistirían finalmente no lo hacen, lo que ayudará a reducir los huecos libres en las agendas médicas y a maximizar la eficiencia en la utilización de los recursos disponibles.

Fase 2: Implementar un Sistema de Overbooking en la Programación de Citas

El objetivo de esta 2^a fase es maximizar el uso de los recursos de los centros hospitalarios, sin excederse en los costos provocados por la propia implementación del sistema de overbooking, es decir, aquellos costos relacionados con los tiempos de espera de los pacientes, o de las horas extra realizadas por el personal sanitario.

- **Estudio de Bibliografía relacionada.** Existe una extensa bibliografía desarrollada en torno a los costes provocados por la infrautilización o sobreutilización de recursos médicos, así como los provocados por una reducción en la calidad del servicio debida a los tiempos de espera de los pacientes. Hay muchas formas de medir los costes, son muchas las variables que impactan en dichos costes, y decenas de variantes a la hora de definir los métodos de asignación de citas para tratar de reducirlos.
- **Elección de las hipótesis para la resolución del problema y fórmula para el cálculo del Coste del Sistema a minimizar.** Una vez estudiada la bibliografía existente, se procede a determinar las hipótesis que acoten los posibles escenarios a desarrollar, y que se ajusten al dataset elegido en la Fase 1 para el Modelo Predictivo de Asistencia, pues se necesitan dichos datos y sus respectivas predicciones para generar el modelo optimizador de overbooking. Dichas hipótesis servirán para definir y delimitar diferentes casos de estudio, así como para generar restricciones que se tendrán que aplicar al modelo para restringir sus grados de libertad a la hora de optimizar los slots de overbooking.

De forma paralela, se elige la fórmula para el cálculo del Coste del Sistema, función que se desea minimizar para optimizar la asignación de citas médicas, teniendo en cuenta toda la bibliografía encontrada al respecto, y también buscando compatibilidad con las características presentes en el dataset escogido en la Fase 1.

- **Cálculo del Coste del Sistema con método tradicional basado en la aplicación de diferentes reglas de asignación de citas médicas.** Una vez escogidas hipótesis y fórmula, se procede a realizar el cálculo aritmético puro de los Costes del Sistema usando diferentes reglas tradicionales de asignación de citas, sin sistema de optimización alguno, sin y con diferentes grados de overbooking.

Cada uno de estos escenarios se ejecutará al mismo conjunto de pruebas de la Fase 1, esto para garantizar robustez y comparabilidad entre los diferentes escenarios en el cálculo medio del Coste del Sistema de Asignación de Citas. Concretamente se usará una partición del 30% de los datos del set de pruebas de la Fase 1, llamado set de validación, porque el 70% restante se utilizará para entrenar el modelo ML de optimización.

- **Cálculo del Coste del Sistema con función basada en las predicciones de asistencia (Fase 1) para la asignación de citas médicas.** Se procede al mismo cálculo aritmético del punto anterior, pero esta vez realizando la asignación de citas mediante una función que tenga en cuenta la probabilidad de asistencia de los pacientes ya programados para una fecha determinada.

Se ejecutarán varios escenarios, con diferentes parámetros en la función que define la asignación de citas, aplicados al mismo set de validación usado en el punto anterior, con tal de garantizar la comparabilidad de resultados.

- **Desarrollo de modelo de ML para el cálculo óptimo de overbooking.** Pura aplicación de IA para desarrollar un modelo ML que sea capaz de optimizar los parámetros de la función de asignación de cita, teniendo en cuenta las probabilidades de asistencia de los pacientes ya programados. Se trata de una mejora sustancial al punto anterior, donde la búsqueda de los parámetros ya no se hace de forma manual, sino que es un algoritmo de IA quien los define, minimizando al

máximo el Coste del Sistema, calculado según la fórmula previamente escogida, y conforme a las restricciones impuestas en las hipótesis previamente establecidas.

El modelo se entrenará con la partición del 70% de los datos del set de pruebas de la Fase 1, usando el 30% restante como set de validación para comparar los resultados con los obtenidos en los dos puntos anteriores.

- **Comparativa de Costes y Conclusiones en la optimización del overbooking.** Se compararán los Costes obtenidos en todos los escenarios comentados, detallando las conclusiones.

Fase 3: Creación de un Asistente Virtual basado en el Procesamiento de Lenguaje Natural (NLP).

En esta fase del proyecto, se desarrolla **Basilio**, un asistente virtual avanzado diseñado para optimizar la gestión de citas médicas en centros de salud. Basilio emplea técnicas de Procesamiento de Lenguaje Natural (NLP) y modelos de **Large Language Models (LLM)**, como **GPT-4o**, para ofrecer una experiencia interactiva eficiente y amigable a través de plataformas de mensajería como Telegram.

- **Objetivos generales**
 - **Gestión de citas médicas:** Facilitar la solicitud, modificación y cancelación de citas médicas de manera rápida y eficiente, reduciendo la carga administrativa y mejorando la disponibilidad de recursos médicos.
 - **Proporcionar información relevante:** Ofrecer detalles sobre disponibilidad de citas, servicios médicos, especialidades y resolver preguntas frecuentes relacionadas con el proceso de citas médicas.
- **Enfoque del Asistente Virtual**
 - Basilio se integra con técnicas avanzadas de **NLP** y **RAG** (Retrieval-Augmented Generation) para proporcionar respuestas contextualmente relevantes y precisas. Este enfoque permite manejar interacciones complejas y ofrece un servicio accesible y eficiente a los pacientes.
- **Beneficios clave**
 - **Reducción de errores humanos:** Automatización del proceso de asignación de citas, minimizando errores manuales.
 - **Disponibilidad 24/7:** Los pacientes pueden gestionar sus citas en cualquier momento, sin restricciones de horario.
 - **Optimización de recursos:** Mejor utilización de los recursos médicos.
 - **Mejora de la experiencia del paciente:** Interacción más fluida y amigable, aumentando la satisfacción del paciente con el servicio.
- **Descripción general del funcionamiento:** Basilio permite a los pacientes:
 - **Agendar citas:** Solicitar nuevas citas especificando especialidad, fecha y hora deseadas.
 - **Consultar citas:** Verificar las citas programadas proporcionando información de identificación.
 - **Modificar o cancelar citas:** Cambiar la fecha y hora de citas existentes o cancelarlas.
 - **Orientación médica:** Ofrecer información sobre qué especialidad consultar según los síntomas descritos.
 - **Información sobre medicamentos:** Proporcionar detalles sobre el uso de medicamentos y la necesidad de recetas médicas.

- **Integraciones tecnológicas**

- **API de OpenAI:** Tras revisar diversas plataformas de chatbots como Dialogflow, Amazon Lex, Microsoft Bot Framework y developers OpenAI, se decide utilizar la API de OpenAI debido a su avanzada capacidad para generar respuestas naturales y precisas, y su flexibilidad para la integración. Se implementa el modelo GPT-4o para facilitar interacciones avanzadas, generando respuestas contextuales a través de instrucciones de sistema que establecen el contexto y comportamiento general del asistente. También se aplicarán técnicas de file search para acceder a la información relevante en la base de datos de citas médicas.
- **Google Sheets:** Utilizado para almacenar y gestionar datos relacionados con citas médicas y disponibilidad de médicos. La integración con esta base de datos permite a Basilio acceder a información actualizada sobre disponibilidad de citas, horarios de médicos y especialidades, asegurando respuestas precisas.
- **Telegram:** Plataforma de mensajería utilizada para la interacción con los pacientes, proporcionando un acceso conveniente y en tiempo real.
- **Procesamiento de imágenes:** Integración de librerías para identificar medicamentos a partir de imágenes enviadas por los usuarios.

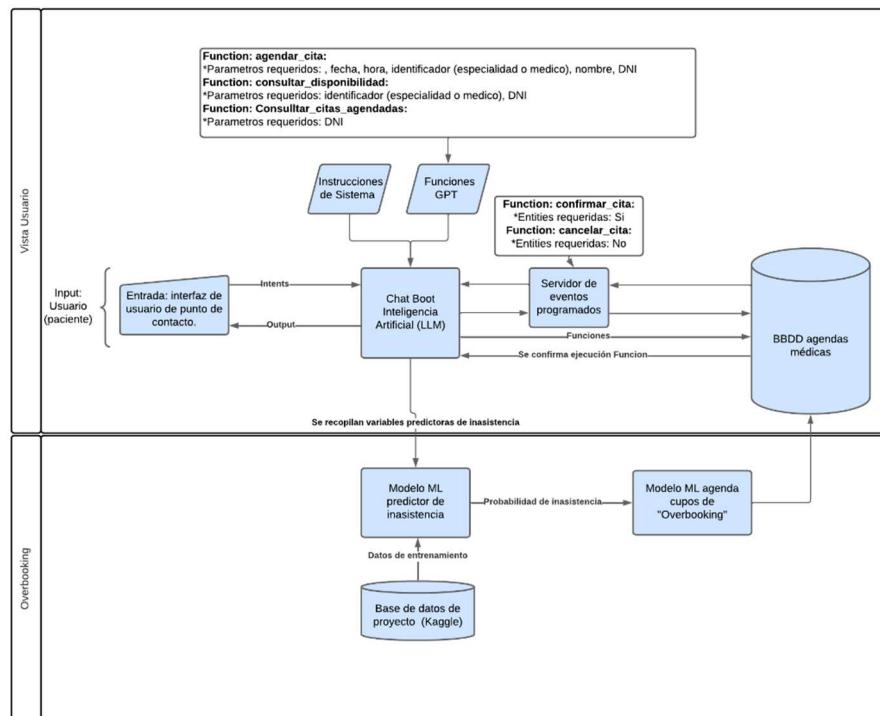


Figura 2. Diagrama de flujo de la propuesta de asistente virtual

En resumen, Basilio, como asistente virtual de tercera generación, representa un avance significativo en la gestión de citas médicas. Su capacidad para interactuar de manera eficiente y amigable a través de una interfaz "text to text", junto con sus técnicas avanzadas de NLP y RAG, optimiza tanto la experiencia del paciente como la eficiencia operativa del hospital.

Fase 4: Desarrollar integración con interfaz para recordatorios de las citas a los pacientes

Desarrollar, a nivel teórico, un proceso de refinamiento del sistema que integre la base de datos de la interfaz creada para gestionar las citas y los recordatorios a los pacientes (Fase 3) con los modelos predictivo y de optimización generados en las Fases 1 y 2.

- Selección de posibles nuevas características relevantes provenientes de la aplicación.
- Adhesión de estas nuevas características predictivas al dataset definitivo elaborado en la Fase 1.
- Desarrollo fine-tuning del modelo de IA seleccionado en la Fase 1, con el objetivo de mejorar en precisión las predicciones de asistencia.
- Integrar los nuevos resultados en el modelo de IA creado en la Fase 2, analizando la optimización / disminución de coste obtenida.
- Fusionar este análisis de probabilidad de inasistencia y “overbooking” con una base de datos integrada a una interfaz de usuario.

Fijación de los objetivos generales y específicos del proyecto

El proyecto tiene como **objetivo general mejorar la eficiencia y reducir los costos asociados con la gestión de citas médicas** mediante el uso de técnicas de Inteligencia Artificial (IA).

Para alcanzar el objetivo general mencionado anteriormente, se plantean los siguientes **objetivos específicos**:

- **Desarrollo de modelos predictivos de ausencias.** Investigar y desarrollar modelos de IA capaces de predecir la no asistencia a las citas médicas con alta precisión. Estos modelos se basarán en datos específicos de pacientes y variables relevantes, buscando minimizar los falsos positivos y negativos.
- **Comparación de algoritmos de aprendizaje automático.** Evaluar y comparar la eficacia de diferentes algoritmos de aprendizaje automático en la predicción del absentismo. Se optimizará el rendimiento de los algoritmos para adaptarse a las características específicas de los datos médicos, con el objetivo de seleccionar el más adecuado para el propósito del proyecto.
- **Optimización de la curva AUC (Area Under the Curve).** Obtener la mejor AUC para el modelo de predicción, buscando el mejor balance entre precisión y recall. Esto implica minimizar los falsos positivos (predicciones incorrectas de que un paciente no asistirá a una cita cuando sí lo hace) y los falsos negativos (predicciones incorrectas de que un paciente asistirá a una cita cuando no lo hace). Reducir estos errores mejora la eficiencia del modelo en la gestión de citas médicas y disminuye los costos asociados.
- **Obtención de la fórmula de costes asociados a una atención médica ineficiente.** Estudiar los costes típicos asociados a una agenda médica ineficazmente programada. Identificar las variables que afectan a dichos costes y definir las hipótesis que mejor se ajustan al problema planteado. Definir una fórmula simple pero precisa es fundamental para que el modelo de IA que optimice la asignación de citas médicas funcione correctamente.
- **Desarrollo de un modelo de IA para la asignación de citas.** Desarrollar un modelo de ML que optimice la asignación de citas buscando el mínimo coste medio de las consultas médicas, el cual se retroalimenta, en cada iteración, de una serie temporal de citas ya asignadas previamente.
- **Creación de un Asistente Virtual para gestión de citas.** Generar un Asistente Virtual con comunicación efectiva. Este asistente gestionará citas con precisión y facilitará la gestión a través de Telegram. Se configurará al Asistente Virtual para ofrecer información médica general, responder preguntas frecuentes sobre el proceso de citas médicas, así como proporcionar orientación básica sobre síntomas y medidas preventivas. Estas funciones constituirán su tarea principal y mejorarán la experiencia del usuario.
- **Desarrollo de un proceso de mejora continua.** Establecer un proceso teórico de mejora continua, donde el modelo predictivo y la información extraída por el Asistente Virtual se retroalimenten

periódicamente. El objetivo es mejorar los resultados predictivos del modelo y reducir los costos asociados a inasistencias en la gestión de citas médicas.

Planificación

Estimación de recursos económicos

Para poder realizar esta estimación se espera a las decisiones tomadas por las directivas, posterior al estudio de la prefactibilidad.

Estimación de recursos materiales

Para estimar los recursos materiales del asistente virtual se deben tener en cuenta los siguientes recursos:

- Un servidor o infraestructura en la nube, para alojar y ejecutar el programa del asistente virtual que se decida usar; debe tener capacidad suficiente para procesar y almacenar la data. Y una amplia capacidad de procesar un alto volumen de citas.
- Dispositivos de acceso como computadores - terminales, tablets o teléfonos inteligentes.
- Línea de acceso a Internet, ¿canal dedicado?
- Software licenciado para el programa - Incluye sistema operativo, Python, Visual Studio, u otras soluciones más robustas, que se decida adoptar; librerías y herramientas para los procesamientos.
- Otro servidor o infraestructura en la nube, para alojar y ejecutar los programas de prueba o nuevos desarrollos.
- Un sistema de gestión para almacenamiento de las bases de datos, y de copias de respaldo.
- El espacio físico para alojar los servidores y las personas responsables del proyecto.

Estimación de recursos humanos

Las personas que harán parte del proyecto se pueden dividir en dos momentos, planeación, desarrollo e implementación del proyecto y personal de funcionamiento y mantenimiento del asistente virtual.

- La primera parte debe contar con al menos un desarrollador de software.

Para todo el proyecto se necesitan personas de coordinación y soporte, como son las que siguen:

- Gerente del proyecto.
- Especialistas de seguridad de datos.
- Personal de soporte para hardware y software.
- Al menos un experto en IA, inicialmente puede ser medio tiempo.

De otro lado se debe tener en cuenta los implicados directamente con el proyecto que deben tener representación en todo el desarrollo del proyecto, al menos un médico, personal administrativo, personal de atención al cliente.

Estimación de los recursos tiempo

Se ha estimado que para el estudio de prefactibilidad un mes. En cuanto hace referencia a la planeación, desarrollo e implementación del proyecto: un mes y medio. Para la parte de pruebas y ajustes un mes.

Lo anterior está condicionado a la cantidad de personas que puedan contratarse para este proyecto.

Definición del alcance del proyecto

Una vez determinada la necesidad de realizar el presente proyecto por la institución se ha procedido a realizar las actividades descritas. Y continuamos describiendo el alcance del asistente virtual.

Recolección de datos, se obtuvo una base de datos sobre la que se realizará el desarrollo del presente asistente virtual. En el que se están desarrollando actividades de limpieza de datos, errores y manejo de datos atípico.

Integración con el sistema de gestión de citas, que se va a ir realizando a medida que avanza el proyecto, que incluye la implementación de una interfaz de programación de aplicaciones (API - Application Programming Interfaces).

Desarrollo del modelo del asistente virtual, núcleo del presente trabajo que se ha venido diseñando y adaptando desde el inicio de este proyecto.

Validación y pruebas, a medida que se va avanzando con el proyecto, se evaluará el rendimiento del asistente virtual, usando datos de prueba y validación; pruebas de carga para verificar la escalabilidad y rendimiento, bajo diferentes condiciones de carga.

Despliegue e implementación, se realizará capacitación al personal involucrado en la asignación de citas, asistentes de atención al cliente, personas del área administrativa, médicos y personal de enfermería. Se realizará supervisión y soporte directo por un periodo de tiempo por definir.

Entregables del proyecto, se plantea realizar un documento de los requerimientos del sistema de asistencia virtual, el conjunto de los datos preprocesados, el modelo del asistente virtual entrenado y desplegado. Informe de las pruebas y validación, sistema de asistente virtual implementado y en funcionamiento, así como la documentación de soporte y capacitación.

Elaboración del cronograma del proyecto

El siguiente cronograma detalla las fechas de inicio y término de cada entrega, junto con el contenido relevante para cada una. Además, se describe la dinámica de trabajo y las reuniones semanales para asegurar el seguimiento y la coordinación del proyecto.

Reuniones semanales

- **Frecuencia:** Todos los viernes
- **Duración:** 2 horas
- **Objetivo:** Revisar el avance semanal, resolver dudas y organizar la distribución del trabajo.
- **Dinámica:** Cada miembro del equipo presenta su progreso y se discuten aspectos relevantes del proyecto. Se asignan tareas para la semana siguiente.

Proceso de trabajo

- **Desarrollo del proyecto:** Todas las personas del equipo (4 miembros) participan en el desarrollo del proyecto.
- **Distribución del contenido del documento:** Para la elaboración del documento final, los apartados se distribuyen de forma individual. Cada miembro se encarga de redactar su sección correspondiente.
- **Revisión conjunta:** Posteriormente, se realiza una revisión conjunta para integrar y unificar el contenido del documento.

Cronograma y entregas

Entrega	Fecha de inicio	Fecha de término	Contenido relevante
1 Entrega	23 Feb 2024	9 Abr 2024	Planteamiento del problema, posibles soluciones, análisis preliminar de bases de datos.
2 Entrega	19 May 2024	4 Jun 2024	Modelos de predicción de no show, algoritmo para optimización de agendas médicas, planteamiento de modelo de chatbot.
3 Entrega	5 Jun 2024	2 Ago 2024	Desarrollo final de modelos de predicción, optimización de algoritmos, desarrollo preliminar del Asistente Virtual.
Entrega final preliminar	19 Ago 2024	16 Sep 2024	Documento final del TFM, unificación de trabajo, conclusiones finales, Resumen Ejecutivo.
Entrega final	1 Oct 2024		Entrega final del proyecto.

A continuación, se detalla cada entrega para ofrecer información más específica sobre el trabajo realizado en cada una.

Entrega 1

Fecha de inicio: 23 Feb 2024

Fecha de término: 9 Abr 2024

Contenido:

1. Búsqueda de base de datos de no show de pacientes
 - a. Planteamiento del problema y posibles soluciones:
 - i. Identificar y analizar las causas de no asistencia a citas médicas.
 - ii. Investigar posibles soluciones para mejorar la asistencia.
 - b. Análisis preliminar de las bases de datos:
 - i. Recopilar y analizar datos sobre no asistencia a citas médicas.
 - ii. Identificar tendencias y patrones en los datos.

Entrega 2

Fecha de inicio: 19 May 2024

Fecha de término: 4 Jun 2024

Contenido:

1. Desarrollo de modelos de predicción de no show (Fase 1):
 - a. Finalización del análisis exploratorio de datos (EDA): Analizar y visualizar los datos para comprender mejor su estructura y distribución.
 - b. Planteamiento de hipótesis con particionamiento del dataset: Dividir el conjunto de datos en varios subconjuntos, cada uno representando una hipótesis subyacente sobre la predicción de asistencias a citas médicas.
 - c. Optimización de modelos de machine learning para predicción de inasistencia de pacientes.

- i. Búsqueda de mejores resultados de particiones: Evaluar y comparar diferentes particiones del dataset.
 - ii. Búsqueda de mejores modelos de machine learning y mejor red neuronal: Evaluar y comparar diferentes modelos de machine learning y redes neuronales.
 - iii. Optimización de hiperparámetros de modelos de machine learning: Ajustar los parámetros de los modelos para mejorar su rendimiento.
 - iv. Conclusiones iniciales comparando nuestros resultados con las bibliografías revisadas: Comparar los resultados obtenidos con los reportados en la literatura.
2. Planteamiento de algoritmo para optimización de agendas médicas según resultados de fase 1 (Fase 2):
- a. Finalizar búsqueda bibliográfica sobre el tema: Investigar y recopilar información sobre algoritmos de optimización de agendas médicas.
 - b. Selección de hipótesis y función de coste. Determinar las principales hipótesis que delimiten el problema a resolver, así como la función de coste a calcular.
3. Planteamiento de modelo de asistente virtual de interacción con pacientes (Fase 3):
- a. Iniciar búsqueda bibliográfica sobre el tema: Investigar y recopilar información sobre asistente virtual médicos.
 - b. Definir el framework con el que elaborar el asistente virtual: Seleccionar un framework para desarrollar el asistente virtual.
 - c. Definir modalidad de interacción del asistente virtual.

Entrega 3

Fecha de inicio: 4 Jun 2024

Fecha de término: 2 Ago 2024

Contenido:

1. Desarrollo final de modelos de predicción de no show (Fase 1):
 - a. Optimización de modelos de Machine y Deep Learning para cada grupo de usuarios: Ajustar los modelos para cada grupo de usuarios.
 - b. Conclusiones de nuestros resultados con las bibliografías revisadas: Comparar los resultados obtenidos con los reportados en la literatura.
2. Algoritmos para optimización de agendas médicas según resultados de fase 1 (Fase 2):
 - a. Terminar de definir la función de coste a utilizar: Implementar un coste exponencial a los tiempos de espera del paciente.
 - b. Analizar Reglas de Asignación de Cita tradicionales: Estudiar, comparar y calcular el coste de los algoritmos utilizando estas reglas que tratan de optimizar agendas médicas, con especial interés en aquellas que generan overbooking.
 - c. Definir y evaluar una función de asignación de cita que utilice las probabilidades de asistencia a la cita médica obtenidas en la Fase 1.
 - d. Desarrollar un algoritmo de optimización: necesario para optimizar la función de asignación de cita mencionada en el punto anterior, minimizando los costos de no asistencia y overbooking.
 - e. Comparar costes de algoritmos: Evaluar y comparar los resultados de los algoritmos.
 - f. Generar conclusiones: Presentar los resultados y conclusiones sobre la optimización de agendas médicas.
3. Desarrollo preliminar de Asistente Virtual de interacción con pacientes (Fase 3):
 - a. Crear/inventar bases de datos: Crear bases de datos de médicos por especialidad y ubicación, y agendas médicas con información de médico, especialidad, ubicación y fecha (libre o Id. Paciente).

- b. Configuración del Asistente Virtual de 3^a generación: Redactar Instrucciones y definir archivos para usar en el file search, así como los parámetros de Temperatura y Top-p (nucleus sampling) para controlar los flujos de conversación entre el Asistente y los Usuarios, permitiendo proporcionar información básica sobre síntomas y medicamentos.
 - c. Crear funciones de búsqueda, lectura y escritura a dichas bases de datos: Implementar funciones para interactuar con las bases de datos
 - d. Probar y refinar el Asistente Virtual.
 - e. Generar conclusiones: Presentar los resultados y conclusiones sobre el Asistente Virtual.
4. Se realiza un planteamiento teórico de una integración de los 3 puntos anteriores (Fase 4).

Entrega final preliminar

Fecha de inicio: 19 Ago 2024

Fecha de término: 16 Sep 2024

Contenido:

1. Elaboración y compilación del documento final del TFM:
 - a. Unificación de trabajo de las 3 fases: Integrar los resultados de las tres fases.
 - b. Conclusiones finales.
 - c. Revisión final y cita de bibliografías.

Entrega final

Fecha inicio: 17 Sep 2024

Fecha de término: 1 Oct 2024

Contenido:

1. Desarrollo de presentación:
 - a. Crear una presentación en PowerPoint que resuma los hallazgos y conclusiones del proyecto.
 - b. Incluir gráficos, visualizaciones y otros elementos para comunicar de manera efectiva los resultados.
 - c. Revisar la presentación para su entrega final.

Desarrollo del proyecto

Fase 1. Predicción de Asistencia a Citas Médicas

Elección, recolección y preparación de datos

Un buen modelo predictivo de asistencia a citas médicas requiere un set de datos con las siguientes características:

- Presencia de Variable Target indicando la Asistencia o No Asistencia a la cita médica.
- Presencia de Variables Predictoras diversas y heterogéneas con las que poder predecir si ocurrirá asistencia o no, contra más variables predictoras mejor.
- Gran número de muestras, necesarias para poder generar un modelo predictivo robusto y confiable.

Se hizo una búsqueda por internet y se escogió el siguiente dataset de Kaggle: “Medical Appointment No Shows” ([7](#)), publicado por Jonihoppen.

Dicho Dataset contiene 110.527 muestras (citas médicas) de 62.299 pacientes distintos, de los que se recogen hasta 13 características de las citas y pacientes mencionados, así como la imprescindible característica objetivo de asistencia o no a la propia cita solicitada.

Se encontraron otros conjuntos de datos que fueron descartados por contener menor número de muestras, o peores características predictoras, o simplemente por no disponer de la variable objetivo de Show – No Show.

Escogido el dataset, y antes incluso de realizar el pertinente Análisis Exploratorio de los Datos (EDA), revisamos la nomenclatura de las columnas para corregir errores topográficos en las mismas: Hypertension por Hipertension, Handicap por Handcap, y NoShow por No-show.

Análisis Exploratorio de Datos (EDA)

La primera actividad de peso en el desarrollo de cualquier tarea de ML es realizar un buen Análisis Exploratorio de Datos para extraer toda la información posible de los mismos, y así diseñar la mejor estrategia de datos que le sirva posteriormente al modelo predictivo.

Primero se analiza la información general del dataset: tipos de datos (los cuales se modifican para poder trabajar la información que contiene cada variable), valores únicos, valores nulos, etc.

A continuación, se empieza analizando la variable target u objetivo, y posteriormente se seguirá con el resto de las 13 variables predictoras.

Variable target “NoShow”

Sólo tiene 2 valores posibles: 0 (False) = Show o 1 (True) = No Show. La distribución de valores indica que, como era de esperar, el dataset está fuertemente desbalanceado, revelando un 20.19% de citas sin asistencia.

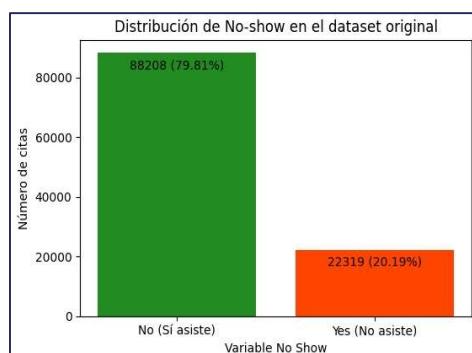


Figura 3. Distribución de citas (asistencias e inasistencias) en el dataset original

Variable predictora: “PatientId”

El dataset consta con información de 62.299 pacientes distintos, identificados con números enteros que van desde el 39217 hasta el 999981631772427.

Esto significa que el dataset contiene datos históricos de pacientes que realizan más de una solicitud de cita médica. Esta información puede ser muy relevante a la hora de determinar la probabilidad de asistencia de un paciente, pues en muchos casos ya se tendrá información relevante de su comportamiento en citas previas.

Aun así, tal y como se observa en la siguiente tabla, para el 60.87% de los pacientes sólo se tiene información de una única cita.

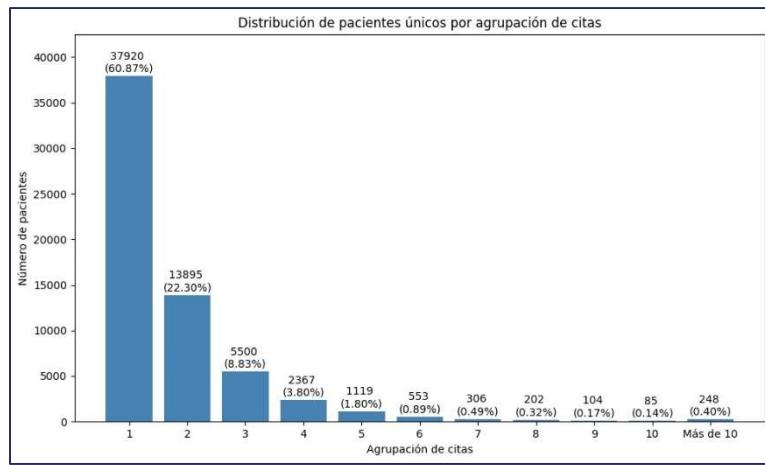


Figura 4. Distribución de pacientes únicos por agrupación de citas médicas

A simple vista no se observa ninguna relación o información de relevancia en el número de identificación del paciente, dato que nos confirmará posteriormente el cálculo de la matriz de correlación entre variables.

Variable predictora: "AppointmentID"

Los valores únicos de esta variable son el total de muestras del dataset: 110.527, identificados con números enteros desde el 5030230 al 5790484.

Tampoco parece haber mayor correlación numérica salvo con la asignación temporal, es decir, que es una identificación numérica correlativa en función de la fecha en la que se solicita la cita. La posible información de utilidad contenida en esa correlación temporal ya está contenida en otras variables más directas, como veremos a continuación.

Variable predictora: "Gender"

Sólo se toman en cuenta dos identificaciones de género: Female and Male, distribuidos en un 65% y 35% de los datos, respectivamente, por lo que esta variable está ligeramente desbalanceada, pero no se considera significativo.

Una de las comprobaciones importantes de esta variable, para garantizar coherencia en los datos, es que cada paciente tenga una identificación única e inequívoca del género asignado en cada una de sus correspondientes citas, y así es.

Respecto a la distribución de asistencia según género, las mujeres tienen una proporción mayor de No Shows (20.31%) respecto a los hombres (19.97%), pero la diferencia es mínima.

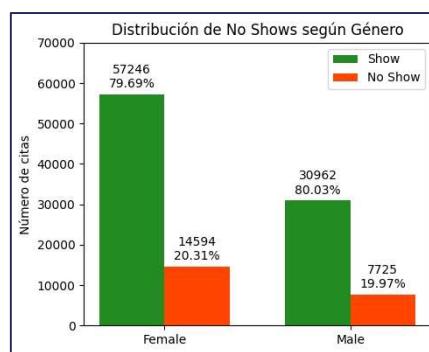


Figura 5. Distribución de citas (asistencias e inasistencias) según el género

Variable predictora: "ScheduledDay"

Esta variable muestra la fecha (día y hora) en la que se solicitó (y programó) la cita médica. Existen 103.549 fechas con día y hora distintos, por lo que hay días en los que se han solicitado citas exactamente en el mismo segundo (pocas, pero las hay).

El primer día que se solicitó una cita médica dentro de este conjunto de datos es el 10 de noviembre de 2015, a las 07:13:56, y la última el 8 de junio de 2016 a las 20:07:23.

El siguiente gráfico muestra como el grueso de fechas de solicitud de citas se concentra en los meses de abril, mayo y junio.

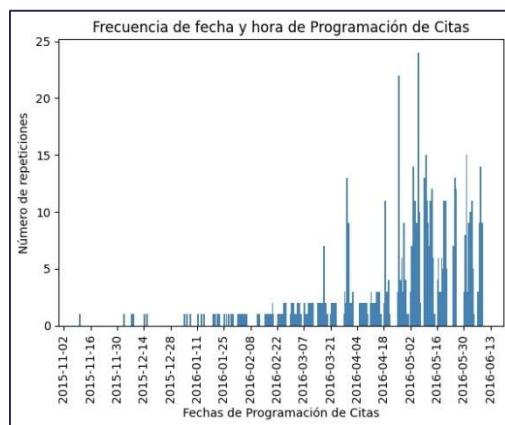


Figura 6. Distribución de citas por fecha de programación

Variable predictora: "AppointmentDay"

A diferencia de "ScheduledDay", esta variable sólo guarda información del día en que se tiene la cita médica, sin la hora a la que está convocado cada paciente. Dentro de este dataset, existen 27 días distintos en los que se atienden consultas médicas, desde el 29 de abril al 8 de junio de 2016.

El hecho de contar con 2 variables Datetime relacionadas con la fecha en la que se solicita la consulta y la fecha en la que se atiende, nos permite estudiar posibles dependencias entre el lapso existente entre ambas fechas y la propia asistencia, característica que se estudiará durante el Feature Engineering.

También se prevé estudiar la posible correlación entre las asistencias y el día semanal en el que se atienden, pero tal y como se muestra en el siguiente gráfico de barras, las variaciones de asistencia según el día de la semana tampoco son exageradas.

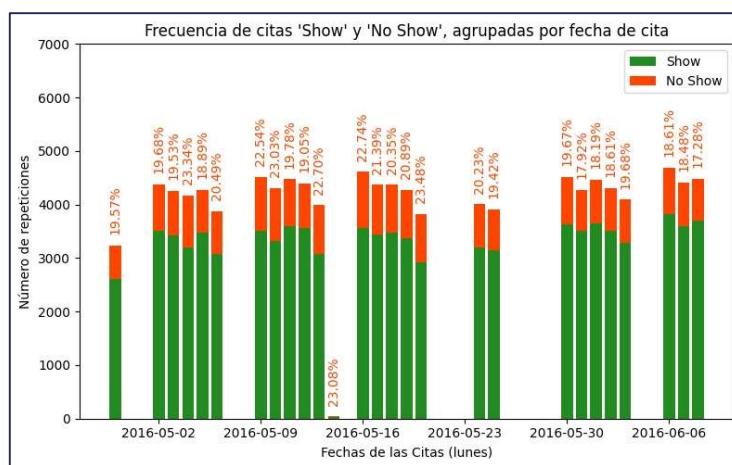


Figura 7. Distribución de citas (asistencias e inasistencias) por fecha de cita

Las fechas indicadas en el gráfico corresponden a todos los lunes de la variable "AppointmentDay", y cada barra representa un día de la semana.

También se observa que el balanceo de los datos para cada día de la semana está bastante equilibrado, salvo para el único sábado del que se tiene constancia (14 mayo), cuyas citas se eliminaran del dataset final para evitar outliers que generen ruido al modelo.

Variable predictora: "Age"

Respecto a la edad, el dataset comprende 104 edades distintas entre todos los pacientes, desde -1 a 115, según la siguiente distribución agrupadas las edades de 5 en 5:

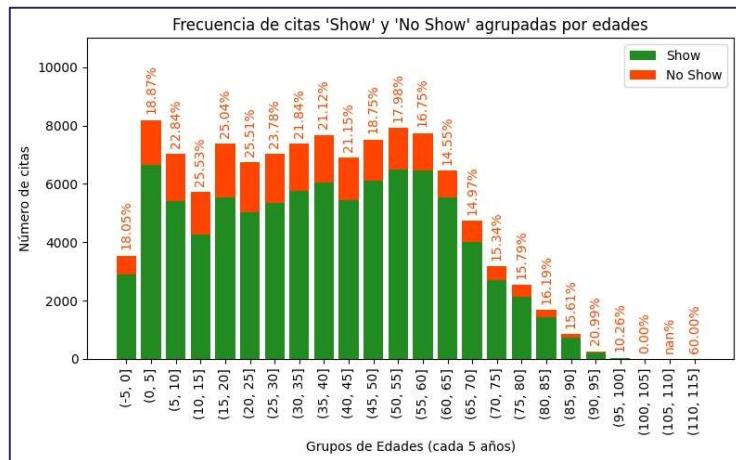


Figura 8. Distribución de citas (asistencias e inasistencias) por rangos de edad

Los pacientes que más faltan a las citas médicas son los comprendidos entre los 5 y los 30 años, así como los mayores de 110, pero de este último grupo hay un único paciente, por lo que no es representativo.

En esta variable se observa un severo desbalanceo en los pacientes de mayor de edad, como es natural, pues el número de pacientes es mucho menor.

En esta variable es importante hacer un estudio más detallado de posibles outliers. Las conclusiones referentes a las edades reflejadas en ambos extremos de los rangos de edad son las siguientes:

- Edades inferiores:
 - -1: Sólo existe una cita para esta edad. Entendemos que es la madre atendiendo una consulta de ginecología.
 - 0, 1, 2, ...: Existen 3539, 2273, 1618, ... citas para estas edades, las cuales no son deseñables. Obviamente son citas en las que los pacientes (bebés) van acompañados por alguno de sus padres.
- Edades superiores:
 - 99: 1 sola cita
 - 100: 4 citas correspondientes a 3 pacientes distintos.
 - 102: 2 citas correspondientes a 2 pacientes distintos.
 - 115: 5 citas correspondientes a un único paciente.

La edad es una información importante para pronosticar la asistencia o no a las citas médicas, por lo que se deja toda la información intacta.

Sin embargo, se hacen un par de comprobaciones más para determinar la coherencia y robustez de los datos:

- Se comprueba que ningún bebé recién nacido presenta antecedentes de hipertensión, diabetes o alcoholismo (otras variables predictoras presentes en el dataset): OK.
- Se comprueba que los pacientes sólo tengan una edad asignada: Se detectan 1.168 pacientes con más de una edad asignada, pero cuyas diferencias nunca son mayores de 1 año, por lo que se asume que son pacientes que han cumplido años entre una cita y la siguiente.

Variable predictora: "Neighbourhood"

El dataset contiene información de 81 barrios distintos, todos correspondientes a la ciudad de Vitória, en Brasil.

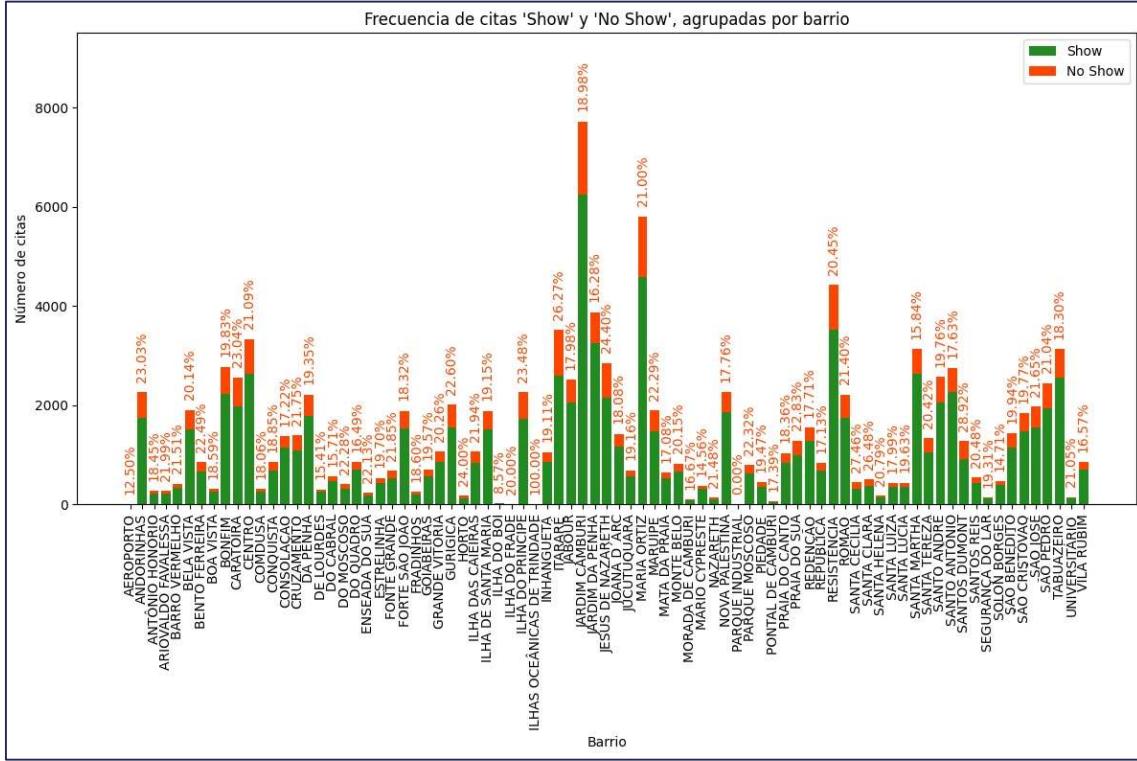


Figura 9. Distribución de citas (asistencias e inasistencias) por barrios

El gráfico muestra un gran desbalanceo de datos en función del barrio considerado, y unas frecuencias de asistencia bastante dispares.

Con el objetivo de mejorar este desbalanceo y tener un modelo que haga unas predicciones más robustas, agruparemos los barrios por clúster de proximidad en el Feature Engineering.

Variable predictora: "Scholarship"

Esta variable contiene información sobre unas becas de ayuda económica que se otorgan en Brasil, siendo una variable binaria indicando si el paciente que solicita la cita dispone de dicha beca o no.

Esta variable está fuertemente desbalanceada, pues sólo un 9.83% de las citas pertenecen a pacientes que disponen de la "Scholarship", los cuales son los que exhiben un mayor porcentaje de No Show.

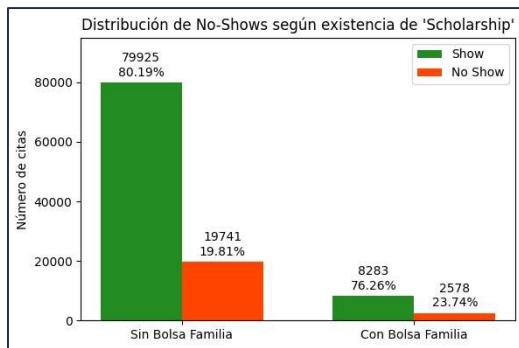


Figura 10. Distribución de citas (asistencias e inasistencias) según ayuda económica

Variable predictora: "Hypertension"

Esta variable también es una variable binaria para indicar si el paciente que acude a la cita tiene hipertensión o no.

Es una variable muy desbalanceada, con un 19.72% de las citas pertenecientes a pacientes que sufren de hipertensión, aunque son los que mejor índice de asistencia tienen.

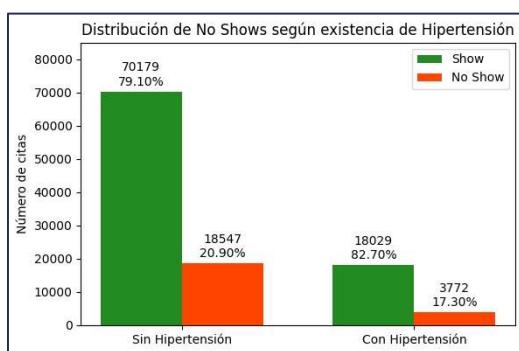


Figura 11. Distribución de citas (asistencias e inasistencias) por hipertensión

Variable predictora: "Diabetes"

Al igual que las dos variables anteriores, ésta también es binaria, indicando si el paciente sufre de diabetes o no.

Esta variable predictora también está fuertemente desbalanceada, con tan sólo un 7.19% de las citas indicando que el paciente solicitante sufre de diabetes, y, al igual que pasaba con los pacientes que sufrían de hipertensión, son los que menos faltan a las citas programadas.

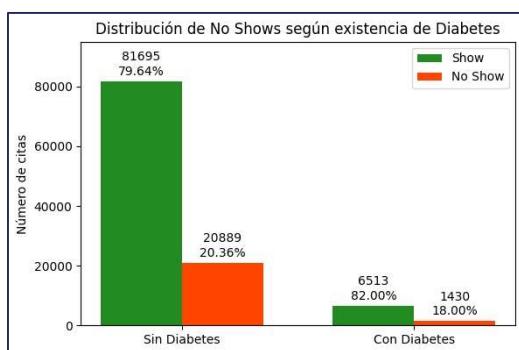


Figura 12. Distribución de citas (asistencias e inasistencias) por diabetes

Variable predictora: "Alcoholism"

Variable predictora binaria que indica si el paciente solicitante de la cita médica es alcohólico o no.

Esta variable está fuertemente desbalanceada, pues tan sólo un 3.04% de las citas del dataset corresponden a pacientes con alcoholismo. Aun así, tampoco es que existan diferencias relevantes en la frecuencia de asistencia a las citas médicas entre un grupo y otro.



Figura 13. Distribución de citas (asistencias e inasistencias) por alcoholismo

Variable predictora: "Handicap"

Esta variable indica si el paciente que solicita la cita tiene algún grado de discapacidad. Se miden hasta 4 grados de discapacidad, por lo que su valor varía entre 0 (ninguna discapacidad), 1 (grado de discapacidad 1), 2 (grado de discapacidad 2), 3 (grado de discapacidad 3) y 4 (grado de discapacidad 4).

Esta es la variable predictora más desbalanceada de todas: tan sólo el 1.8475% (2042 citas) corresponde a citas de pacientes con un grado de discapacidad 1, un 0.1656% (183 citas) corresponde citas de pacientes con un grado de discapacidad 2, un 0.0118% (13 citas) corresponden a un grado de discapacidad 3, y un ínfimo 0.0027% corresponden a 3 citas de 3 pacientes distintos con un grado de discapacidad 4.

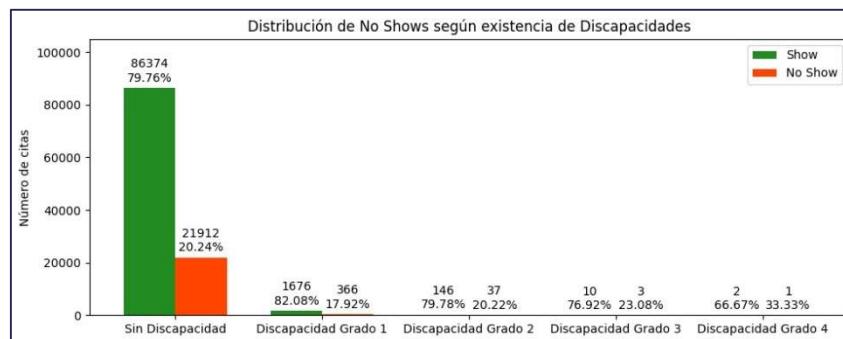


Figura 14. Distribución de citas (asistencias e inasistencias) por grado de discapacidad

Es por eso por lo que todas las citas que indican algún grado de discapacidad en el paciente se consideran outliers. Con tal de mejorar la robustez del modelo se procede a agrupar todos los pacientes con discapacidad a un solo grupo, independientemente del grado de discapacidad médica.

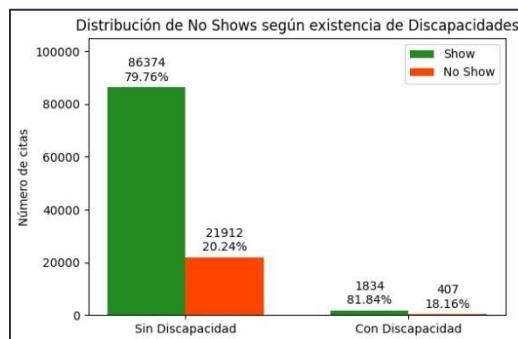


Figura 15. Distribución de citas (asistencias e inasistencias) por discapacidad

Aun así, tal y como se observa en la gráfica anterior, no se consigue arreglar el desbalanceo, pues la suma de todas las citas con indicación de algún grado de discapacidad tan sólo representa el 2.0276%.

Es por ello por lo que, vistos también los desbalances existentes en todas las variables predictoras que describen la historia clínica de los pacientes, se trabajarán nuevas características predictoras en la fase de Featuring Engineering que aglutinen los pacientes con dolencias médicas, independientemente de cuál sea.

Variable predictora: "SMS_received"

Esta es la última variable predictora original del dataset, y se trata de una variable binaria que indica si el paciente que asistía a la cita médica recibió [0] o no [1] algún mensaje de texto recordatorio para dicha asistencia.

La variable está desbalanceada (*aunque no tanto como las que describían el historial médico de los pacientes*), pues tan sólo se enviaron mensajes de texto en un 32.10% de las citas contenidas en este dataset.

Curiosamente, y pareciendo “a priori” fuera de toda lógica, los pacientes que recibieron un mensaje de texto recordatorio son los que muestran un porcentaje de No Show mayor.

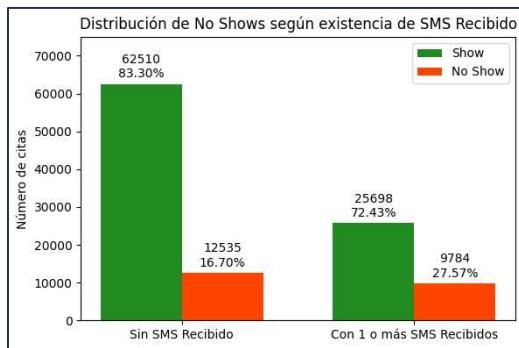


Figura 16. Distribución de citas (asistencias e inasistencias) según SMS recibido

Sin embargo, este comportamiento “fuera de lógica” guarda una correlación encubierta con el lapsus entre la fecha de programación de cita y la fecha de atención a la cita. Los mensajes de texto sólo se envían cuando dicho lapsus supera cierto margen (no se envían mensajes de textos si la cita se programa para los próximos días).

Feature Engineering

La fase de Feature Engineering en este proyecto se enfoca en transformar y seleccionar las características más relevantes del conjunto de datos para mejorar el rendimiento de los modelos predictivos de IA, específicamente diseñados para predecir la probabilidad de las ausencias (no shows). A continuación, se describen las acciones y transformaciones específicas que se han realizado:

Creación de nuevas características

Para mejorar la capacidad predictiva sobre la asistencia a citas médicas, se han creado nuevas variables predictoras basadas en diferentes aspectos, desde patrones temporales hasta factores geográficos y meteorológicos. Estas variables se han creado para capturar aspectos clave que podrían influir en el comportamiento de los pacientes. A continuación, se describen en detalle estas nuevas características.

Variables temporales

Se han creado nuevas variables predictoras basadas en las fechas de solicitud (ScheduledDay) y asistencia a la cita (AppointmentDay). Se parte de la hipótesis de que el día de la semana en que se

programa una cita médica puede afectar la probabilidad de asistencia, dado que ciertos días pueden presentar una mayor disponibilidad o disposición por parte de los pacientes para asistir. Así mismo, se plantea que el tiempo transcurrido entre la programación de la cita hasta la fecha real de la misma podría influir en la probabilidad de asistencia.

Nueva variable	Descripción	Variable original
App_DayOfWeek	Día de la semana en que se atiende la cita médica. Valores posibles: 'Friday' 'Wednesday' 'Monday', etc.	AppointmentDay
Time_SchDay_to_AppDay	Tiempo (en segundos) transcurrido entre la fecha de solicitud y la fecha de la cita.	AppointmentDay,
Days_since_last_App	Tiempo (en días) transcurrido entre las citas consecutivas por paciente.	AppointmentDay

Variables geográficas

Se han introducido variables predictoras adicionales basadas en la información geográfica de los barrios, incluyendo su ubicación (8) y la presencia de centros de salud en ellos (9). La hipótesis subyacente sugiere que la proximidad geográfica a los centros médicos y la disponibilidad de atención médica en el barrio pueden influir en la probabilidad de asistencia a las citas médicas.

Nueva variable	Descripción	Variable original
Neigh_Cluster	Etiquetas de clúster asignadas a cada barrio mediante el modelo de K-Means. Esto permite agrupar los barrios según su proximidad geográfica.	LatitudeNeigh, LongitudeNeigh
Health_Centre	Identificación binaria que especifica la presencia (1) o ausencia (0) de Centro Médico en cada barrio.	

Se ha utilizado el algoritmo **K-Means**, una técnica de aprendizaje no supervisado, para entender cómo la proximidad geográfica y la disponibilidad de servicios de salud afectan la tasa de ausencia a las citas. El proceso se llevó a cabo de la siguiente manera:

- Selección de características.** Para agrupar los barrios, se seleccionaron las coordenadas geográficas (latitud y longitud) de cada barrio. Estas características permiten agrupar los barrios en función de su proximidad geográfica.
- Determinación del número de clústers.** Se decidió agrupar los barrios en 12 clústers. Esta decisión se basó en la distribución geográfica y la cantidad total de barrios, buscando un equilibrio entre la granularidad de los grupos y la manejabilidad de los datos.
- Aplicación de K-Means.** Utilizando el algoritmo K-Means, se asignó a cada barrio una etiqueta de clúster que indica a qué grupo pertenece.
- Ajuste manual de la clusterización.** Tras la clusterización inicial, se realizó un ajuste manual para balancear mejor el número de citas médicas entre los clústers. Este ajuste consistió en mover ciertos barrios entre clústers para asegurar una distribución más equitativa y efectiva de los recursos médicos.

Para facilitar la comprensión y comunicación de los resultados, se creó un mapa interactivo utilizando Folium. Este mapa muestra los barrios agrupados por clústeres, con diferentes colores para cada uno, y señala la ubicación de los centros de salud en rojo. Además, el mapa incluye información sobre el número de citas médicas por barrio, lo que permite identificar de manera clara y accesible la distribución geográfica de los barrios, los recursos de salud disponibles y la carga de trabajo en cada área.

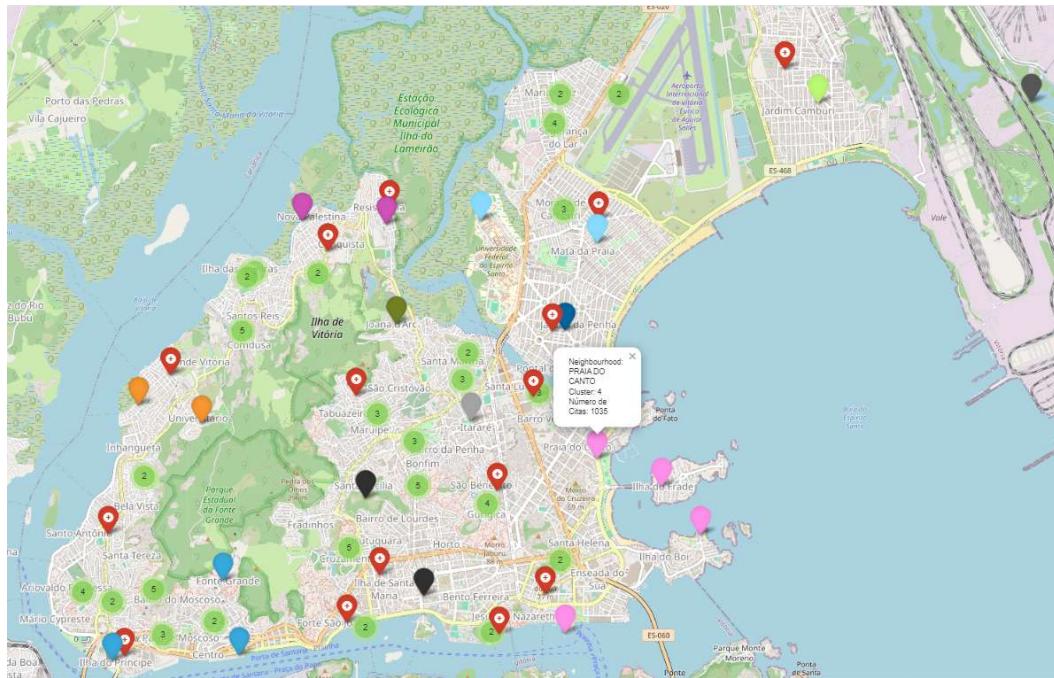


Figura 17. Mapa de distribución de citas médicas por barrio y centros médicos (KMeans)

Variables meteorológicas

Otra hipótesis plantea que la asistencia a la cita médica esté fuertemente influenciada por las condiciones meteorológicas del día en cuestión, como la temperatura, la velocidad del viento y la lluvia. Para investigar esta suposición, se procedió a extraer la información meteorológica de la ciudad de Vitória para el mes de mayo de 2016, obtenida de Weather and Climate (10). Posteriormente, se integraron al conjunto de datos las nuevas variables predictoras: ‘Temperature’, ‘WindSpeed’, ‘Precipitation’.

Variables del historial médico

Bajo la premisa de que la combinación de las condiciones médicas de un paciente puede tener relevancia en la predicción de la asistencia a las citas médicas, se han introducido variables adicionales basadas en el historial médico de los pacientes.

Nueva variable	Descripción	Variable original
Number_Health_Conds	Número total de condiciones médicas que tiene cada paciente. Valores posibles: 0, 1, 2, 3, 4.	Hypertension, Diabetes, Alcoholism, Handicap
Presence_Health_Conds	Identificación binaria que especifica la presencia (1) o ausencia (0) de condiciones médicas por paciente.	Number_Health_Conds

Además, para determinar la probabilidad de asistencia del paciente, se han considerado las siguientes estadísticas:

Nueva variable	Descripción
Prior_Apps_byPatient	Número de citas previas de cada paciente.
Prior_NoShows_byPatient	Indica la cantidad de veces que un paciente no asistió a citas previas.
Prob_NoShow_byPatient	Porcentaje de no asistencia del paciente, basado en su historial de citas.

Es importante señalar que el porcentaje de no asistencia se inicializa para cada paciente con el valor promedio de no asistencia en todo el conjunto de datos. Posteriormente, este valor se actualiza según el historial específico de cada paciente a medida que tiene citas médicas.

IMPORTANTE: En el fichero jupyter habría que volver a calcular los valores de NUMBER_NO_SHOWS / NUMBER_SAMPLES porque se han eliminado registros.

Transformación de tipos de datos

Durante el proceso de preparación de los datos, se realizaron varias transformaciones para garantizar la coherencia y la adecuación de los tipos de datos. A continuación, se detallan las principales transformaciones llevadas a cabo:

Conversiones de tipo de datos numéricos. Se ajustaron los tipos de datos numéricos, como enteros o flotantes, para garantizar la consistencia en las operaciones matemáticas y el análisis estadístico. Esto incluyó la conversión de variables como el identificador del paciente (PatientId) y el género (Gender).

Manipulación de fechas y horas. Las variables relacionadas con fechas y horas, como las fechas de programación (ScheduledDay) y las fechas de las citas médicas (AppointmentDay), se convirtieron al formato de fecha y hora adecuado para facilitar su manipulación y análisis.

Codificación de variables categóricas. Se utilizó LabelEncoder para convertir variables categóricas, como el género (Gender), día de la semana de la cita (App_DayOfWeek) y el barrio (Neighbourhood), en valores numéricos. Esto se realizó para permitir el procesamiento de algoritmos de aprendizaje automático que requieren datos numéricos como entrada.

Además, la **variable target “NoShow” fue transformada a tipo booleano** para facilitar su manipulación y análisis durante el modelado predictivo: 0 (sí asiste), 1 (no asiste).

Tratamiento de valores atípicos y missing values

Durante el análisis de datos, se identificaron valores atípicos y algunos registros que requirieron un tratamiento especial. A continuación, se detallan las acciones tomadas:

Condición de discapacidad. La variable “Handicap” originalmente incluía valores de 0 a 4, indicando niveles de discapacidad. Sin embargo, el análisis exploratorio de datos (EDA) reveló que los niveles superiores a 0 no eran relevantes. Por lo tanto, se simplificó la variable para que solo indique si un paciente tiene discapacidad (1) o no (0). Ver [Figura 15](#).

Eliminación de citas programadas para los sábados. La variable “App_DayOfWeek” mostró que solo había 39 citas programadas para los sábados. Para evitar posibles sesgos debido al pequeño tamaño de esta muestra y reducir el ruido en el modelo, se decidió eliminar estas citas del conjunto de datos.

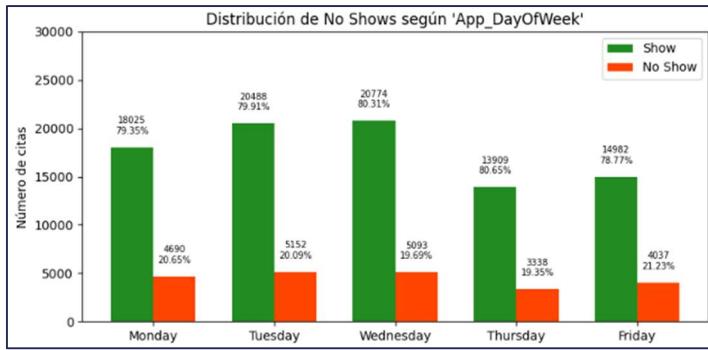


Figura 18. Distribución de citas (asistencias e inasistencias) según el día programado

Gestión de valores negativos en el tiempo entre programación y cita. Se identificaron 5 registros en la variable “Time_SchDay_to_AppDay” con valores negativos, indicando que la fecha de programación era posterior a la fecha de la cita. Estos casos se corrigieron asignando un valor de 0, asumiendo que el paciente programó la cita el mismo día de la consulta.

PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	App_DayOfWeek	Time_SchDay_to_AppDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handicap	SMS_received	NoShow	
72362	3787481966821	M	5655637	2016-05-04 06:50:57	2016-05-03	Tuesday	-111057	7	TABUAZEIRO	0	0	0	0	0	True	
27033	7839272661752	M	5679978	2016-05-10 10:51:53	2016-05-09	Monday	-125513	38	RESISTÊNCIA	0	0	0	0	1	0	True
55226	7896293967868	F	5715660	2016-05-18 14:50:41	2016-05-17	Tuesday	-139841	19	SANTO ANTÔNIO	0	0	0	0	1	0	True
64175	24252258389979	F	5664962	2016-05-05 13:43:58	2016-05-04	Wednesday	-135838	22	CONSOLAÇÃO	0	0	0	0	0	0	True
71533	998231581612122	F	5686628	2016-05-11 13:49:20	2016-05-05	Thursday	-568160	81	SANTO ANTÔNIO	0	0	0	0	0	0	True

Figura 19. Paciente con fecha de programación posterior a la fecha de la cita

```
[ ] # Corrección a 0's de todos los 'Time_SchDay_to_AppDay' negativos:
med_app_FE['Time_SchDay_to_AppDay'][med_app_FE['Time_SchDay_to_AppDay'] <= 0] = 0
```

Figura 20. Modificación de los valores negativos de “Time_SchDay_to_AppDay” a 0

Asignación de valor -1 a pacientes sin citas previas. En la variable “Days_since_last_App”, los pacientes sin citas previas recibieron un valor de -1, indicando la falta de datos previos para calcular el tiempo transcurrido.

```
[ ] # Cálculo e inserción de la variable del tiempo entre citas de un mismo paciente
med_app_FE.insert(loc = 7,
                   column = 'Days_since_last_App',
                   value = med_app_FE.groupby('PatientId')['AppointmentDay'].diff().dt.days)

# Asignación -1 a todos las citas de pacientes que no han tenido citas previas
med_app_FE['Days_since_last_App'].fillna(-1, inplace=True)
```

Figura 21. Asignación del valor -1 a pacientes sin citas previas en “Days_since_last_App”

Ajuste de clústeres en “Neigh_Cluster”. Se detectaron clústeres con solo 1 o 2 muestras (específicamente los clústeres 1 y 5). Estos clústeres se eliminaron y se reorganizaron los restantes, reduciendo el número de clústeres de 12 a 10 para eliminar los menos representativos.

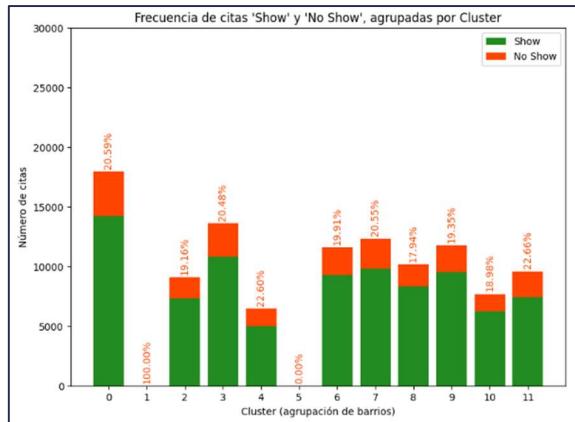


Figura 22. Distribución de citas (asistencias e inasistencias) agrupadas por cluster

```
[ ] # Eliminación de los 2 clusters con una o dos cita:
indexes = med_app_FE.loc[(med_app_FE['Neigh_Cluster'] == 1) | \
                           (med_app_FE['Neigh_Cluster'] == 5)]
                           ].index.tolist()

med_app_FE.drop(indexes, inplace = True)

# Corremos todos los valores de los cluster del 0 al 9:
for i in range(2, 12):
    if i < 6:
        med_app_FE.loc[med_app_FE['Neigh_Cluster'] == i, 'Neigh_Cluster'] = i-1
    else:
        med_app_FE.loc[med_app_FE['Neigh_Cluster'] == i, 'Neigh_Cluster'] = i-2
```

Figura 23. Eliminación de cluster con bajo número de citas médicas

Es importante señalar que en el dataset original no se encontraron valores faltantes. Estas medidas aseguran la integridad y coherencia de los datos utilizados para el análisis posterior.

Validación de características

El análisis de correlación se utilizó para identificar relaciones lineales entre las características y la variable objetivo, así como entre las propias características. Esto incluye:

Matriz de correlación. Se calculó la matriz de correlación de Pearson para todas las características numéricas, lo que permitió identificar características redundantes que podrían ser eliminadas o combinadas.

Correlación con la variable objetivo. Se analizaron las correlaciones entre cada característica y la variable objetivo ('NoShow') para identificar las características con mayor poder predictivo.

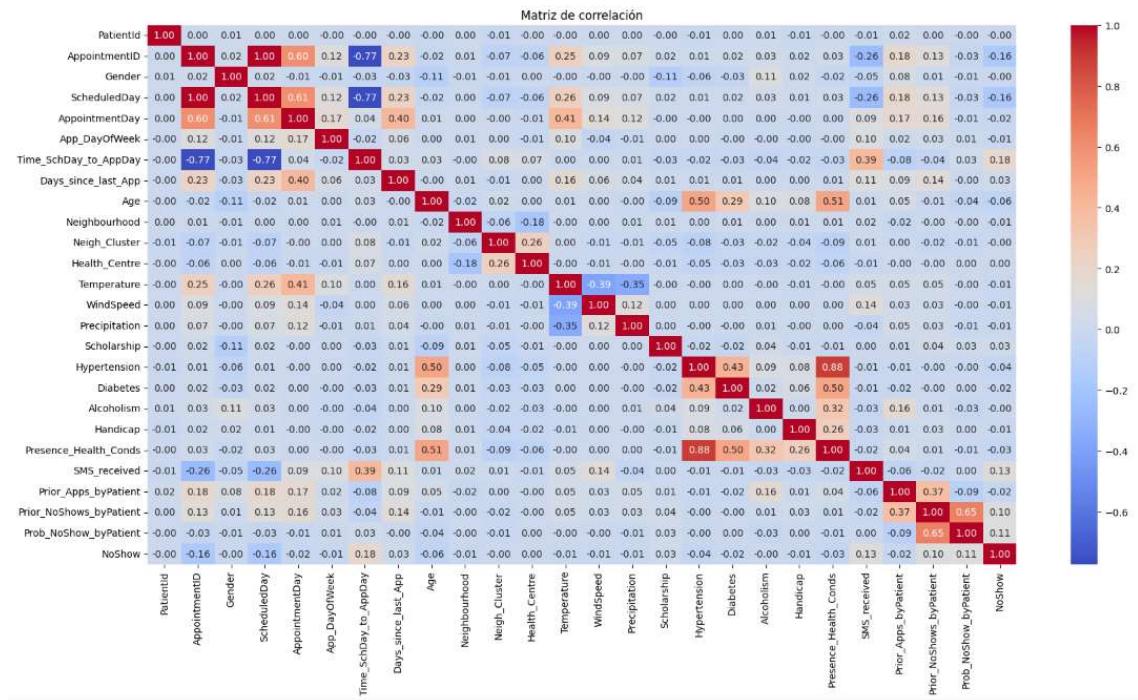


Figura 24. Matriz de correlación sin reducción de dimensionalidad

Selección de características

En esta sección, se presenta una comparación detallada entre la estructura del dataset original y el dataset resultante después de aplicar técnicas de ingeniería de características (Feature Engineering).

Dataset Original

El dataset inicial contiene una serie de variables básicas y diversas para la gestión de citas médicas. Estas variables proporcionan una base inicial que, aunque útil, requiere refinamiento para mejorar su capacidad predictiva.

Dataset Tratado

Tras aplicar técnicas de ingeniería de características, el dataset se modificó significativamente. Se añadieron nuevas variables predictoras y se ajustaron los tipos de datos de varias de las variables originales. En la imagen incluida, se destacan en color rojo las modificaciones y las nuevas variables añadidas. Esta transformación es fundamental para aumentar la capacidad del modelo de IA para identificar patrones y tendencias relacionadas con la asistencia a citas médicas.

Dataset Final

Finalmente, se realizó una selección exhaustiva de características para crear el dataset final. Este conjunto de datos, que se utilizará para la normalización, escalado, reducción de dimensionalidad y particionamiento, incluye solo las variables más relevantes y útiles para la predicción de asistencia a citas médicas. En la imagen, las variables eliminadas se muestran en gris, mientras que las variables seleccionadas se destacan en negrita.

The diagram illustrates the iterative process of dataset transformation. It consists of three tables arranged horizontally, connected by three large grey arrows pointing from left to right, indicating the progression of the feature engineering process.

DATASET ORIGINAL	
PatientId	float64
AppointmentID	int64
Gender	object
ScheduledDay	object
AppointmentDay	object
Age	int64
Scholarship	int64
Hypertension	int64
Diabetes	int64
Alcoholism	int64
Handicap	int64
SMS_received	int64
NoShow	object

DATASET TRATADO	
PatientId	int64
AppointmentID	int64
Gender	int32
ScheduledDay	datetime64[ns]
AppointmentDay	datetime64[ns]
App_DayOfWeek	int32
Time_SchDay_to_AppDay	int64
Days_since_last_App	int64
Age	int64
Neighbourhood	int32
Neigh_Cluster	int64
Health_Centre	int64
Temperature	float64
WindSpeed	float64
Precipitation	float64
Scholarship	int64
Hypertension	int64
Diabetes	int64
Alcoholism	int64
Handicap	int64
Presence_Health_Conds	int64
SMS_received	int64
Prior_Apps_byPatient	int64
Prior_NoShows_byPatient	int64
Prob_NoShow_byPatient	float64
NoShow	bool

DATASET FINAL	
PatientId	int64
AppointmentID	int64
Gender	int32
ScheduledDay	datetime64[ns]
AppointmentDay	datetime64[ns]
App_DayOfWeek	int32
Time_SchDay_to_AppDay	int64
Days_since_last_App	int64
Age	int64
Neighbourhood	int32
Neigh_Cluster	int64
Health_Centre	int64
Temperature	float64
WindSpeed	float64
Precipitation	float64
Scholarship	int64
Hypertension	int64
Diabetes	int64
Alcoholism	int64
Handicap	int64
Presence_Health_Conds	int64
SMS_received	int64
Prior_Apps_byPatient	int64
Prior_NoShows_byPatient	int64
Prob_NoShow_byPatient	float64
NoShow	bool

Figura 25. Transformaciones del dataset

Iteraciones y refinamiento

Durante el proceso de Feature Engineering para el proyecto de predicción de asistencia a citas médicas, se llevaron a cabo varias iteraciones y refinamientos para optimizar las características utilizadas en el modelo predictivo. Estas iteraciones se basaron en los hallazgos del análisis exploratorio de datos, así como en la retroalimentación de la validación del modelo y los cambios en los requisitos del proyecto.

Iteraciones

- Se realizaron iteraciones en la selección y creación de características en respuesta a los patrones observados durante el análisis exploratorio de datos. Por ejemplo, se exploraron diferentes formas de representar el historial médico de los pacientes para capturar de manera efectiva su impacto en la asistencia a las citas.
- Se llevaron a cabo ajustes en las características temporales, como el tiempo transcurrido entre la programación de la cita y la fecha real de la misma, para capturar con mayor precisión la influencia de los factores temporales en la probabilidad de asistencia.
- Se exploraron técnicas avanzadas de transformación de variables geográficas, como la segmentación espacial, para capturar la influencia de la ubicación geográfica en la asistencia a las citas médicas.

Refinamiento

- Se refinaron las características seleccionadas en función de su capacidad predictiva y su interpretabilidad. Por ejemplo, se realizaron ajustes en las características basadas en el historial médico de los pacientes para equilibrar la representación de diferentes condiciones médicas.
- Se evaluaron y refinaron las técnicas de imputación de valores faltantes y manejo de valores atípicos para mejorar la calidad de los datos y reducir el impacto de datos erróneos en el modelo.

Normalización y escalado de características

La estandarización y el escalado de características son procesos fundamentales en el preprocesamiento de datos para muchos algoritmos de aprendizaje automático. En nuestro proyecto de gestión de citas médicas, estos pasos son esenciales para asegurar que las características de entrada estén en una escala comparable, facilitando así la convergencia de los modelos y mejorando su capacidad predictiva.

Utilizamos la clase **StandardScaler** de la biblioteca scikit-learn para estandarizar nuestras bases de datos. Este proceso implica ajustar el escalador usando solo los datos de entrenamiento, permitiendo que el escalador "aprenda" de estos datos. Luego, aplicamos la misma transformación a los datos de prueba utilizando el escalador ajustado, garantizando que la distribución de cada característica tenga una media de 0 y una desviación estándar de 1.

```
# Se crea una instancia de StandardScaler
scaler = StandardScaler()

# Se ajusta el escalador a los datos de entrenamiento y se transforman
X_train_scaled = scaler.fit_transform(X_train_set)
# Se transforman los datos de test utilizando el mismo escalador que se ajustó a los datos de entrenamiento
X_test_scaled = scaler.transform(X_test_set)
```

Figura 26. Estandarización con StandardScaler

En la gestión de citas médicas, la estandarización es crucial debido a la diversidad de las características consideradas, como la edad del paciente y el tiempo entre la solicitud y la cita. Para la edad de los pacientes, originalmente de -1 a 115 años, la estandarización ha ajustado el rango a valores entre -1 y 3. Esto asegura que las variaciones de las variables no afecten negativamente el rendimiento de los modelos, manteniendo la integridad de los datos originales.

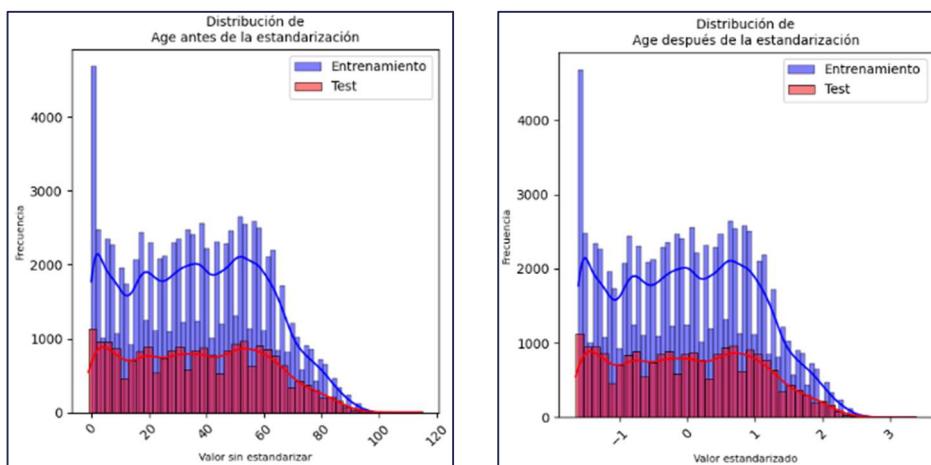


Figura 27. Distribución de citas por edad sin estandarizar (izquierda) y estandarizado (derecha)

Para abordar el desbalanceo de clases en nuestro conjunto de datos de entrenamiento, hemos aplicado dos técnicas de **Data Augmentation** de manera independiente: SMOTE-ENN y ADASYN.

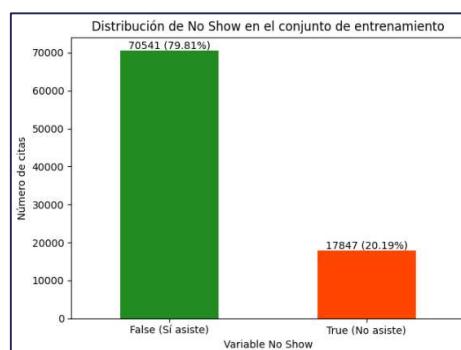


Figura 28. Dataset de entrenamiento - Distribución de citas antes de normalizar

SMOTE-ENN

Implementamos la técnica SMOTE-ENN mediante el uso de Pipeline de scikit-learn. SMOTE (*Synthetic Minority Over-sampling Technique*) genera nuevas muestras sintéticas de la clase minoritaria,

mientras que ENN (*Edited Nearest Neighbours*) elimina instancias ruidosas, logrando así un equilibrio de clases en este conjunto de datos como se visualiza en la imagen.

```
[ ] # Se crea una instancia de SMOTE y ENN
smote = SMOTE(sampling_strategy = 'minority',
               random_state = 42)
enn = EditedNearestNeighbours(sampling_strategy = 'not minority',
                               kind_sel = 'all',
                               n_neighbors = 2)

# Se crea una instancia de la clase Pipeline
pipeline = Pipeline([('smote', smote), ('enn', enn)])

# Se aplica el pipeline a los datos
X_train_resampled, y_train_resampled = pipeline.fit_resample(X_train_scaled, y_train_set)
```

Figura 29. Aplicación de SMOTE-ENN

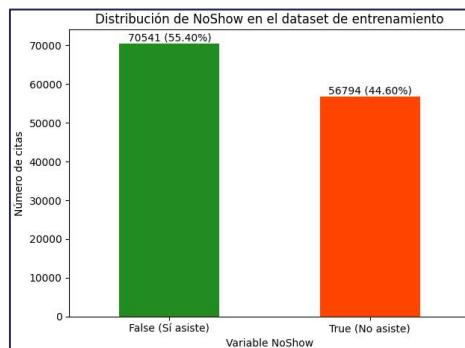


Figura 30. Dataset de entrenamiento - Distribución de citas tras SMOTE-ENN

ADASYN

En otra copia independiente de los datos de entrenamiento, aplicamos ADASYN (Adaptive Synthetic Sampling Approach). ADASYN es una técnica que no sólo genera nuevas muestras sintéticas para la clase minoritaria, sino que lo hace de manera adaptativa. Esto significa que ADASYN prioriza la generación de nuevas muestras en las áreas del espacio de características donde los datos son más escasos y difíciles de clasificar. Al enfocarse en las casuísticas con menor densidad de datos, se espera que ADASYN mejore la capacidad del modelo para manejar casos difíciles y, por lo tanto, aumente la precisión del modelo en la clasificación de la clase minoritaria.

```
[ ] # Crear una instancia de ADASYN
adasyn = ADASYN(sampling_strategy='minority', random_state=42)

# Aplicar ADASYN a los datos de entrenamiento
X_train_resampled, y_train_resampled = adasyn.fit_resample(X_train_scaled, y_train_set)
```

Figura 31. Aplicación de ADASYN.

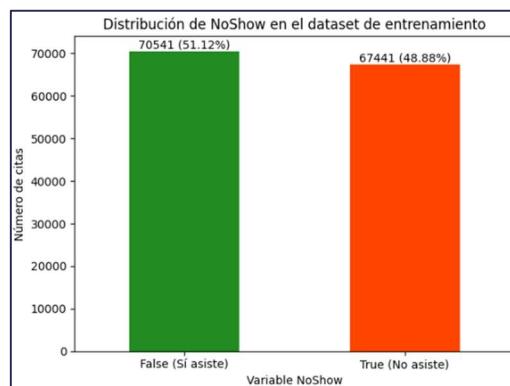


Figura 32. Dataset de entrenamiento - Distribución de citas tras ADASYN

Reducción de dimensionalidad

La reducción de dimensionalidad es una técnica fundamental para simplificar el conjunto de datos y mejorar la eficiencia de los modelos de inteligencia artificial sin perder información importante. Para este propósito, se ha implementado el **Análisis de Componentes Principales (PCA)**, una técnica ampliamente utilizada en aprendizaje automático.

Con el PCA, nuestro objetivo era conservar al menos el 95% de la varianza total mientras reducíamos la dimensionalidad de nuestro conjunto de datos.

```
[ ] # Aplicar PCA para reducir la dimensionalidad mientras se conserva la mayor cantidad de varianza posible
pca = PCA(n_components = 0.95,
           svd_solver = 'full')

# Ajustar y transformar los datos de entrenamiento
X_train_pca = pca.fit_transform(X_train_resampled)
X_test_pca = pca.transform(X_test_scaled)
```

Figura 33. Aplicación de PCA

Sin embargo, después de aplicar el PCA, observamos que, de las 19 variables predictoras originales, solo se seleccionaron 16 componentes principales. A pesar de ser un número menor de variables, estos componentes lograron explicar el 96.66% de la varianza total de los datos, proporcionando una representación efectiva de la información contenida en el conjunto de datos original.

```
Número de componentes: 16
Varianza explicada por cada componente: [0.13707614 0.12212461 0.09235221 0.0689055  0.06253879 0.05901496
 0.05718571 0.05134638 0.05082864 0.05010441 0.04323952 0.03925022
 0.03778197 0.03715579 0.03256729 0.02512141]
Varianza total explicada: 0.9665935231029232
```

Figura 34. Resultado de aplicación de PCA

Dada la posibilidad de que la reducción en el número de variables conlleve a una pérdida de información, consideramos que era más prudente mantener todas las variables originales en nuestros modelos. Esta decisión se basó en la observación de que el PCA eliminó solo un pequeño número de variables, y que esta pérdida no justificaba la posible reducción en la calidad de nuestras predicciones, especialmente considerando que ya estábamos conservando una varianza explicada del 95%.

Entrenamiento de modelos

Particionamiento de datos

Dado que no se dispone de un conjunto de datos de pruebas, se procede a generar uno mediante el particionamiento de los datos disponibles. Este proceso se realiza con el propósito de evaluar el rendimiento de los modelos predictivos en datos no vistos durante el entrenamiento. Previamente, se decide crear conjuntos de datos de acuerdo a hipótesis sobre los que posteriormente se aplicará el particionamiento.

Hipótesis para la creación de conjuntos de datos:

- Mantenimiento de la totalidad del conjunto de datos.** Bajo la premisa de que la retención de la totalidad de los datos maximiza la información disponible y potencialmente mejora la capacidad predictiva, se mantiene el conjunto de datos original sin ningún tipo de filtrado.
- Pacientes sin condiciones médicas.** Se crea un conjunto de datos que incluye solo a los pacientes del conjunto original que no tienen condiciones médicas registradas. Esto se basa en la hipótesis subyacente de que los pacientes sin condiciones médicas pueden tener una mayor probabilidad de inasistencia. Al generar este dataset se eliminan las variables 'Hypertension', 'Diabetes', 'Alcoholism', 'Handicap', 'Presence_Health_Conds'.

- C. **Pacientes de edad entre 5 y 30 años.** Se genera un conjunto de datos que contiene únicamente a los pacientes del conjunto original cuya edad se encuentra entre 5 y 30 años. Esto se basa en la hipótesis subyacente de que este grupo demográfico tiende a faltar más a las citas médicas.
- D. **Pacientes con citas programadas para otro día.** Se forma un conjunto de datos que incluye exclusivamente a los pacientes del conjunto original cuyas citas están programadas para un día distinto al de su solicitud. Esto se sustenta en la hipótesis subyacente de que existe una correlación significativa entre este grupo de pacientes y la inasistencia.
- E. **Pacientes de barrios con centro médico.** Se establece un conjunto de datos que contiene únicamente a los pacientes del conjunto original de barrios donde existe un centro médico. Esto se basa en la hipótesis subyacente de que los pacientes que están en barrios con centros médicos pueden tener una menor sensación de urgencia o responsabilidad al tener acceso fácil a atención médica.
- F. **Pacientes segmentados por grupos de edad.** Se crean conjuntos de datos específicos para diferentes grupos de edad, basados en la hipótesis de que la edad puede influir en los patrones de asistencia a citas médicas:
 - Niños: Menores de 12 años
 - Adolescentes: 13 - 18 años
 - Jóvenes adultos: 19 - 35 años
 - Adultos: 36 - 64 años
 - Adultos mayores: 65 años en adelante

Procedimiento de particionamiento de datos:

Una vez formuladas las hipótesis y generados los conjuntos de datos correspondientes, se procede al particionamiento de los datos para entrenamiento y prueba. Este proceso se lleva a cabo con el objetivo de evaluar y validar la eficacia de los modelos de aprendizaje automático en datos no vistos durante el entrenamiento. El procedimiento se detalla a continuación:

1. **Cálculo de variables relevantes.** Para cada hipótesis, se calcula el número de pacientes en el conjunto de datos (NUMBER_PATIENTS), el número de citas correspondientes a no asistencia (NUMBER_NO_SHOWS), y el número total de muestras en el conjunto de datos (NUMBER_SAMPLES).
2. **Determinación del tamaño del conjunto de prueba.** Se determina el tamaño del conjunto de prueba multiplicando el número total de muestras por un porcentaje predefinido, usualmente el 20%. Esto garantiza una división adecuada entre los conjuntos de entrenamiento y prueba.

```
[ ] NUMBER_Apps_inTest = int(NUMBER_SAMPLES * 0.20)
print(f"Para alcanzar el 20% del muestreo en nuestro set de pruebas\n \
necesitamos {NUMBER_Apps_inTest} citas de pacientes distintos, es decir,\n \
muestras de un {NUMBER_Apps_inTest / NUMBER_PATIENTS:.2%} de los pacientes.")

➡ Para alcanzar el 20% del muestreo en nuestro set de pruebas\n \
necesitamos 22097 citas de pacientes distintos, es decir,\n \
muestras de un 35.48% de los pacientes.
```

Figura 35. Determinación del tamaño del conjunto de prueba

3. **Creación de subconjuntos de datos.** Se crea un subconjunto de datos que contiene únicamente la última cita de cada paciente, ordenado según la fecha de la cita, para cada hipótesis generada. Esto asegura que cada paciente esté representado por una única muestra en el conjunto de prueba.
4. **Determinación de la proporción de no asistencia.** Se calcula la cantidad de pacientes que no asistieron y que asistieron, necesarios para mantener la misma proporción en el conjunto de prueba como en el conjunto de datos completo, para cada hipótesis. Esto asegura que la distribución de asistencia y no asistencia se mantenga en el conjunto de prueba.

```

# Cálculo del número de citas NoShow = True que se requieren para mantener la estratificación de Clases
NUMBER_NoShows_inTest = int(NUMBER_Apps_inTest * (NUMBER_NO_SHOWS / NUMBER_SAMPLES))
NUMBER_Shows_inTest = int(NUMBER_Apps_inTest - NUMBER_NoShows_inTest)
print(f"Se requieren {NUMBER_NoShows_inTest} pacientes con 'NoShow' = True en el Set de Prueba, y\n \
{NUMBER_Shows_inTest} pacientes con 'NoShow' = False.\n")

```

Figura 36. Determinación de la proporción de no asistencia

5. **Selección de citas para el conjunto de prueba.** Se seleccionan las últimas citas de pacientes que no asistieron y asistieron según el número calculado anteriormente, para cada hipótesis. Estas citas se utilizarán para formar el conjunto de datos de prueba final.
6. **Filtrado del conjunto de datos original.** Después de haber seleccionado las citas necesarias para el conjunto de prueba, se utilizan esas citas para filtrar el conjunto de datos original y crear tanto el conjunto de prueba como el conjunto de entrenamiento definitivos, para cada hipótesis. Esto garantiza que cada conjunto de prueba final contenga exactamente las citas necesarias, mientras que cada conjunto de entrenamiento final contenga todas las demás citas que no están en el conjunto de prueba.

Entrenamiento

Se procede a entrenar para cada una de las distintas bases de datos 4 modelos de machine learning: regresión logística, árbol de decisión, en ambos casos utilizando la librería Sklearn de Python y 2 redes neuronales: una red neuronal densa (NeuralNet) y un modelo de atención secuencial (TabNet) utilizando la librería Pytorch entrenadas utilizando GPU. Se desestima de momento la utilización de otros modelos de machine learning (SVM y modelos ensamblados como el Random Forest, AdaBoost y Stacking) debido a los tiempos de entrenamiento en el hardware que disponemos. A continuación, se detalla el entrenamiento de cada modelo.

Para cada modelo de Sklearn, se entrenaron tres instancias de GridSearchCV, cada una optimizada para una métrica de rendimiento (scoring) diferente: Accuracy (Exactitud), Precision (Precisión), Recall (Sensibilidad).

Regresión logística

Se procede a realizar una búsqueda de mejores hiperparámetros utilizando búsqueda en Cuadrícula (Grid Search): Se utiliza GridSearchCV para explorar un espacio de hiperparámetros:

```

# Se define el rango de hiperparámetros para la búsqueda en cuadrícula
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
    'max_iter': [100, 200, 300, 500, 1000]
}

```

Figura 37. Hiperparámetros en regresión logística

Posteriormente se realiza validación Cruzada (K-Fold Cross-Validation): Se utiliza un KFold con 5 particiones (`n_splits=5`) y se barajan los datos (`shuffle=True`) para asegurar una evaluación robusta del modelo. Durante cada iteración del K-Fold, se ajusta el modelo y se evalúa el rendimiento en un conjunto de validación. El modelo con la mejor puntuación de validación se selecciona como el mejor, se guarda y se utiliza en el conjunto de prueba.

Árbol de decisión

Al igual que en la regresión logística se procede a realizar una búsqueda de mejores hiperparámetros utilizando búsqueda en Cuadrícula (Grid Search): Se utiliza GridSearchCV para explorar un espacio de hiperparámetros:

```

# Se define el rango de hiperparámetros para la búsqueda en cuadros
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [10, 15, 20],
    'min_samples_split': [2, 3, 5],
    'min_samples_leaf': [2, 3, 4]
}

```

Figura 38. Hiperparámetros en árbol de decisión

Se repite el mismo proceso de validación Cruzada (K-Fold Cross-Validation): Se utiliza un KFold con 5 particiones (`n_splits=5`), se barajan los datos (`shuffle=True`) y se evalúa el rendimiento en un conjunto de validación. El modelo con la mejor puntuación se selecciona, se guarda y se utiliza en el conjunto de prueba.

Redes neuronales: NeuralNet

NeuralNet es una red neuronal densa, lo que significa que cada capa está conectada directamente a todas las capas posteriores. Se define una clase NeuralNet que hereda de `nn.Module`. La red consta de:

- Capas lineales (`nn.Linear`): Estas capas son responsables de las transformaciones lineales de los datos de entrada, aprendiendo pesos y sesgos que se ajustan durante el entrenamiento.
- Normalización por lotes (`nn.BatchNorm1d`): Reduce el desplazamiento interno de covariables, haciendo el entrenamiento más eficiente.
- Funciones de activación ReLU (`nn.ReLU`): Introducen no linealidades en el modelo, permitiendo que la red neuronal aprenda relaciones complejas en los datos.
- Función de activación Sigmoid (`nn.Sigmoid`): Se utiliza en la capa de salida para transformar las salidas en probabilidades, es útil ya que es un problema de clasificación binaria, y fundamental para la Fase 2 del proyecto.
- Capa de Dropout (`nn.Dropout`): desactiva aleatoriamente algunas neuronas durante el entrenamiento, ayuda a prevenir el sobreajuste.

Validación: Se utiliza StratifiedKFold con 5 particiones (`n_splits=5`) y se barajan los datos (`shuffle=True`) para asegurar una evaluación robusta del modelo y se divide el conjunto de datos en pliegues estratificados para mantener la proporción de clases en cada pliegue. Para cada pliegue, se crean tensores de entrenamiento y validación. Se utilizan DataLoader para cargar los datos en mini-lotes (`batch_size=64`), lo que facilita el entrenamiento por lotes. En cada época, se entrena el modelo, se calcula la pérdida de entrenamiento y validación, y se ajusta el optimizador.

Entrenamiento: Se establecen los parámetros de entrenamiento como:

- Número de épocas (`num_epochs=100`)
- Criterio de pérdida (`nn.BCELoss`), optimizador (`optim.Adam`) con `lr=1e-4, weight_decay=1e-5` útil en problemas de clasificación binaria.
- Scheduler para ajustar la tasa de aprendizaje `ReduceLROnPlateau` con `factor=0.5, patience=3`. Lo que permite ajustar dinámicamente la tasa de aprendizaje, mejorando la convergencia del modelo.
- Se implementa el Early Stopping con `patience=3` para detener el entrenamiento si no hay mejora en la pérdida de validación, evitando así el sobreajuste.

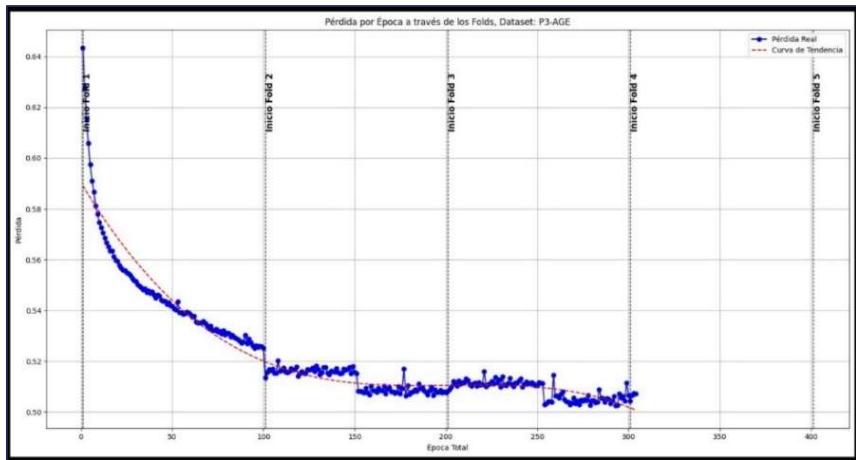


Figura 39. Pérdida por época en red neural

Se calculan las pérdidas de entrenamiento y validación en cada época, con los parámetros mencionados, como se puede observar en la imagen existe una curva descendente consistente en las perdidas hasta el 4to fold, donde Early Stopping detiene el entrenamiento para así evitar el sobreajuste.

Una vez completado el entrenamiento en todos los pliegues, se guarda el mejor modelo encontrado para ser utilizado con los datos de prueba.

Redes neuronales: TabNet

TabNet es un modelo transformer diseñado específicamente para datos tabulares, introduce un mecanismo de atención secuencial que permite a la red enfocarse en las características más relevantes en cada paso, en lugar de procesar todas las características al mismo tiempo, a diferencia de NeuralNet.

Esta capacidad de atención secuencial permite a TabNet modelar de manera más efectiva las relaciones complejas entre las variables y capturar el orden significativo en las columnas de datos tabulares. En otras palabras, TabNet puede aprender a "leer" los datos tabulares de manera similar a como un humano leería una tabla, prestando atención a las partes más importantes y relevantes en cada momento.

Ventajas de TabNet:

- **Interpretabilidad:** TabNet proporciona información sobre la importancia de las características, lo que ayuda a comprender cómo el modelo toma decisiones.
- **Rendimiento:** En muchos casos, TabNet supera a otras arquitecturas de redes neuronales en tareas de clasificación y regresión con datos tabulares.
- **Eficiencia:** A pesar de su complejidad, TabNet puede ser entrenado de manera eficiente en conjuntos de datos grandes.

Para implementar TabNet se adaptó la arquitectura base de NeuralNet descrita anteriormente, para incorporar el mecanismo de atención secuencial de TabNet. Se mantuvieron constantes otros aspectos del entrenamiento y la configuración para hacer homologables los resultados de ambos modelos.

Evaluación de modelos

Para evaluar el desempeño de los modelos (regresión logística, árbol de decisión, NeuralNet y TabNet), se entrenaron y validaron utilizando tanto hardware local con GPU y CPU. Este enfoque híbrido permitió aprovechar la aceleración de la GPU para tareas computacionalmente intensivas, como el entrenamiento de las redes neuronales, y la flexibilidad de la CPU para otras etapas del proceso, como el entrenamiento de los modelos de Scikit-learn.

Una vez finalizado el entrenamiento, se procedió a una evaluación de los datos de entrenamiento y prueba. Se calcularon diversas métricas de rendimiento para cada modelo y dataset, incluyendo:

- Accuracy (Exactitud): Proporción de predicciones correctas sobre el total de predicciones.
- Precision (Precisión): Proporción de verdaderos positivos sobre el total de positivos predichos.
- Recall (Sensibilidad): Proporción de verdaderos positivos sobre el total de positivos reales.
- F1-score: Media armónica entre precisión y recall, útil para equilibrar ambas métricas.
- Curva ROC: Representación gráfica de la capacidad de discriminación del modelo a diferentes umbrales de clasificación.

Dataset	Model	Train Accuracy	Train Precision	Train Recall	Train F1 Score	Train AUC
P1-ALL-ADASYN	Logistic Regression (Precision)	60.88%	62.86%	48.76%	0.55	0.65
	Decision Tree (Precision)	80.51%	76.66%	86.44%	0.81	0.90
	Logistic Regression (Recall)	60.88%	62.86%	48.76%	0.55	0.65
	Decision Tree (Recall)	73.29%	67.21%	88.56%	0.76	0.81
	Logistic Regression (Accuracy)	60.88%	62.86%	48.76%	0.55	0.65
	Decision Tree (Accuracy)	80.52%	76.65%	86.51%	0.81	0.90
	Neural Network (NeuralNet)	68.30%	63.17%	84.26%	0.72	0.74
P1-ALL	Neural Network (TabNet)	69.13%	63.26%	87.90%	0.74	0.75
	Logistic Regression (Precision)	63.90%	63.69%	44.35%	0.52	0.70
	Decision Tree (Precision)	84.89%	82.69%	83.62%	0.83	0.93
	Logistic Regression (Recall)	63.90%	63.69%	44.35%	0.52	0.70
	Decision Tree (Recall)	82.65%	78.68%	83.81%	0.81	0.92
	Logistic Regression (Accuracy)	63.90%	63.69%	44.35%	0.52	0.70
	Decision Tree (Accuracy)	84.87%	82.66%	83.61%	0.83	0.93
P2-NOCONDITIONS	Neural Network (NeuralNet)	71.12%	64.90%	76.78%	0.70	0.79
	Neural Network (TabNet)	68.96%	61.39%	81.96%	0.70	0.77
	Logistic Regression (Precision)	63.89%	63.17%	43.84%	0.52	0.70
	Decision Tree (Precision)	84.01%	81.65%	82.29%	0.82	0.93
	Logistic Regression (Recall)	63.92%	63.18%	43.96%	0.52	0.70
	Decision Tree (Recall)	78.46%	76.16%	74.60%	0.75	0.88
	Logistic Regression (Accuracy)	63.91%	63.18%	43.92%	0.52	0.70
P3-AGE	Decision Tree (Accuracy)	83.65%	81.95%	80.78%	0.81	0.93
	Neural Network (NeuralNet)	71.98%	65.93%	75.68%	0.70	0.80
	Neural Network (TabNet)	70.92%	63.90%	78.59%	0.70	0.79
	Logistic Regression (Precision)	64.26%	63.13%	43.84%	0.52	0.72
	Decision Tree (Precision)	83.50%	77.11%	88.54%	0.82	0.92
	Logistic Regression (Recall)	64.31%	63.13%	44.14%	0.52	0.72
	Decision Tree (Recall)	78.82%	71.20%	86.58%	0.78	0.88
P4-TIME	Logistic Regression (Accuracy)	64.31%	63.13%	44.14%	0.52	0.72
	Decision Tree (Accuracy)	83.42%	77.07%	88.35%	0.82	0.92
	Neural Network (NeuralNet)	71.96%	64.66%	79.11%	0.71	0.80
	Neural Network (TabNet)	68.59%	60.20%	83.04%	0.70	0.76
	Logistic Regression (Precision)	62.78%	59.03%	26.24%	0.36	0.64
	Decision Tree (Precision)	73.11%	65.51%	70.84%	0.68	0.82
	Logistic Regression (Recall)	62.78%	58.95%	26.44%	0.37	0.64
P5-HEALTHCENTRE	Decision Tree (Recall)	77.59%	71.29%	74.69%	0.73	0.87
	Logistic Regression (Accuracy)	62.78%	59.03%	26.24%	0.36	0.64
	Decision Tree (Accuracy)	81.14%	88.37%	61.49%	0.73	0.90
	Neural Network (NeuralNet)	66.70%	64.65%	38.92%	0.49	0.71
	Neural Network (TabNet)	64.39%	60.55%	34.45%	0.44	0.67
	Logistic Regression (Precision)	64.66%	64.45%	45.71%	0.53	0.70
	Decision Tree (Precision)	84.89%	81.35%	85.63%	0.83	0.93
P6A-KIDS	Logistic Regression (Recall)	64.66%	64.45%	45.71%	0.53	0.70
	Decision Tree (Recall)	84.26%	80.41%	85.40%	0.83	0.93
	Logistic Regression (Accuracy)	64.66%	64.45%	45.71%	0.53	0.70
	Decision Tree (Accuracy)	83.56%	80.13%	83.81%	0.82	0.92
	Neural Network (NeuralNet)	72.32%	65.87%	78.28%	0.72	0.80
	Neural Network (TabNet)	71.54%	65.44%	76.23%	0.70	0.80
	Logistic Regression (Precision)	66.36%	66.94%	51.27%	0.58	0.74
P6B-ADOLESCENTS	Decision Tree (Precision)	89.36%	89.33%	86.96%	0.88	0.96
	Logistic Regression (Recall)	66.40%	66.87%	51.62%	0.58	0.74
	Decision Tree (Recall)	77.65%	70.42%	87.62%	0.78	0.87
	Logistic Regression (Accuracy)	66.40%	66.87%	51.62%	0.58	0.74
	Decision Tree (Accuracy)	89.61%	89.65%	87.20%	0.88	0.96
	Neural Network (NeuralNet)	72.75%	66.39%	81.06%	0.73	0.80
	Neural Network (TabNet)	71.87%	64.93%	82.80%	0.73	0.80
P6C-YOUNGADULTS	Logistic Regression (Precision)	65.98%	65.03%	48.17%	0.55	0.73
	Decision Tree (Precision)	91.76%	93.26%	87.50%	0.90	0.96
	Logistic Regression (Recall)	66.01%	64.70%	49.17%	0.56	0.73
	Decision Tree (Recall)	91.43%	91.40%	88.78%	0.90	0.96
	Logistic Regression (Accuracy)	65.92%	64.80%	48.45%	0.55	0.73
	Decision Tree (Accuracy)	91.46%	91.67%	88.54%	0.90	0.96
	Neural Network (NeuralNet)	72.15%	67.21%	70.99%	0.69	0.80
P6D-ADULTS	Neural Network (TabNet)	75.56%	67.63%	84.70%	0.75	0.84
	Logistic Regression (Precision)	64.12%	63.15%	43.90%	0.52	0.72
	Decision Tree (Precision)	87.36%	88.58%	81.74%	0.85	0.95
	Logistic Regression (Recall)	63.94%	62.75%	43.95%	0.52	0.72
	Decision Tree (Recall)	77.83%	71.96%	81.11%	0.76	0.87
	Logistic Regression (Accuracy)	64.12%	63.15%	43.90%	0.52	0.72
	Decision Tree (Accuracy)	88.13%	89.06%	83.17%	0.86	0.95
P6E-OLDERADULTS	Neural Network (NeuralNet)	73.36%	65.93%	81.40%	0.73	0.81
	Neural Network (TabNet)	71.43%	63.66%	81.36%	0.71	0.79
	Logistic Regression (Precision)	63.56%	63.95%	45.33%	0.53	0.69
	Decision Tree (Precision)	85.60%	84.43%	83.75%	0.84	0.94
	Logistic Regression (Recall)	63.56%	63.95%	45.33%	0.53	0.69
	Decision Tree (Recall)	75.12%	69.05%	81.97%	0.75	0.85
	Logistic Regression (Accuracy)	63.56%	63.95%	45.33%	0.53	0.69
P6F-ADULTS	Decision Tree (Accuracy)	85.98%	83.60%	85.99%	0.85	0.94
	Neural Network (NeuralNet)	70.74%	65.07%	76.83%	0.70	0.79
	Neural Network (TabNet)	69.81%	61.93%	87.12%	0.72	0.79
	Logistic Regression (Precision)	58.95%	60.64%	37.40%	0.46	0.63
	Decision Tree (Precision)	83.97%	77.64%	92.83%	0.85	0.92
	Logistic Regression (Recall)	58.95%	60.64%	37.40%	0.46	0.63
	Decision Tree (Recall)	79.14%	71.98%	91.48%	0.81	0.88
P6G-ADULTS	Logistic Regression (Accuracy)	58.95%	60.64%	37.40%	0.46	0.63
	Decision Tree (Accuracy)	88.00%	86.40%	88.55%	0.87	0.95
	Neural Network (NeuralNet)	73.26%	69.53%	77.29%	0.73	0.81
	Neural Network (TabNet)	70.01%	67.05%	71.85%	0.69	0.77

Figura 40. Resultados datos de entrenamiento por modelo para cada dataset

Dataset	Model	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test AUC
P1-ALL-ADASYN	Logistic Regression (Precision)	64.57%	30.70%	60.03%	0.41	0.66
	Decision Tree (Precision)	65.18%	28.89%	49.61%	0.37	0.64
	Logistic Regression (Recall)	64.57%	30.70%	60.03%	0.41	0.66
	Decision Tree (Recall)	61.40%	30.50%	71.33%	0.43	0.72
	Logistic Regression (Accuracy)	64.57%	30.70%	60.03%	0.41	0.66
	Decision Tree (Accuracy)	65.04%	28.69%	49.25%	0.36	0.64
	Neural Network (NeuralNet)	53.57%	28.15%	83.70%	0.42	0.69
P1-ALL	Neural Network (TabNet)	58.90%	30.19%	78.95%	0.44	0.73
	Logistic Regression (Precision)	67.73%	32.05%	53.44%	0.40	0.66
	Decision Tree (Precision)	70.69%	30.53%	35.42%	0.33	0.63
	Logistic Regression (Recall)	67.70%	32.03%	53.46%	0.40	0.66
	Decision Tree (Recall)	69.99%	32.20%	44.00%	0.37	0.65
	Logistic Regression (Accuracy)	67.72%	32.05%	53.44%	0.40	0.66
	Decision Tree	71.05%	31.14%	35.84%	0.33	0.63
P2-NOCONDITIONS	Neural Network (NeuralNet)	59.21%	28.44%	67.34%	0.40	0.67
	Neural Network (TabNet)	57.96%	29.80%	79.85%	0.43	0.72
	Logistic Regression (Precision)	67.48%	32.76%	52.66%	0.40	0.66
	Decision Tree (Precision)	71.53%	32.92%	34.78%	0.34	0.64
	Logistic Regression (Recall)	67.42%	32.72%	52.71%	0.40	0.66
	Decision Tree (Recall)	70.76%	34.63%	44.79%	0.39	0.68
	Logistic Regression (Accuracy)	67.45%	32.74%	52.71%	0.40	0.66
P3-AGE	Decision Tree (Accuracy)	72.47%	34.09%	33.84%	0.34	0.63
	Neural Network (NeuralNet)	61.85%	31.08%	67.60%	0.43	0.69
	Neural Network (TabNet)	64.14%	32.70%	67.49%	0.44	0.72
	Logistic Regression (Precision)	69.01%	39.82%	51.93%	0.45	0.69
	Decision Tree (Precision)	68.96%	37.92%	41.99%	0.40	0.62
	Logistic Regression (Recall)	68.93%	39.76%	52.17%	0.45	0.69
	Decision Tree (Recall)	64.91%	35.92%	55.24%	0.44	0.65
P4-TIME	Logistic Regression (Accuracy)	68.93%	39.76%	52.17%	0.45	0.69
	Decision Tree (Accuracy)	69.15%	38.10%	41.57%	0.40	0.62
	Neural Network (NeuralNet)	62.19%	35.79%	68.49%	0.47	0.69
	Neural Network (TabNet)	63.18%	36.68%	69.34%	0.48	0.71
	Logistic Regression (Precision)	67.37%	38.86%	25.16%	0.31	0.59
	Decision Tree (Precision)	64.00%	34.38%	28.89%	0.31	0.53
	Logistic Regression (Recall)	67.37%	38.93%	25.38%	0.31	0.59
P5-HEALTHCENTRE	Decision Tree (Recall)	64.44%	34.29%	26.99%	0.30	0.52
	Logistic Regression (Accuracy)	67.37%	38.86%	25.16%	0.31	0.59
	Decision Tree (Accuracy)	65.12%	31.77%	19.45%	0.24	0.52
	Neural Network (NeuralNet)	64.18%	35.50%	31.35%	0.33	0.57
	Neural Network (TabNet)	69.68%	39.67%	12.16%	0.19	0.54
	Logistic Regression (Precision)	69.19%	33.12%	50.64%	0.40	0.67
	Decision Tree (Precision)	70.80%	32.12%	39.29%	0.35	0.63
P6A-KIDS	Logistic Regression (Recall)	69.19%	33.12%	50.64%	0.40	0.67
	Decision Tree (Recall)	70.15%	31.53%	40.05%	0.36	0.63
	Logistic Regression (Accuracy)	69.19%	33.12%	50.64%	0.40	0.67
	Decision Tree (Accuracy)	71.69%	33.73%	40.81%	0.37	0.64
	Neural Network (NeuralNet)	61.13%	30.52%	71.51%	0.43	0.70
	Neural Network (TabNet)	68.41%	34.15%	59.74%	0.43	0.73
	Logistic Regression (Precision)	69.05%	34.55%	57.26%	0.43	0.72
P6B-ADOLESCENTS	Decision Tree (Precision)	71.79%	29.93%	28.22%	0.29	0.58
	Logistic Regression (Recall)	68.96%	34.56%	57.84%	0.43	0.72
	Decision Tree (Recall)	67.79%	31.12%	47.27%	0.38	0.67
	Logistic Regression (Accuracy)	68.96%	34.56%	57.84%	0.43	0.72
	Decision Tree (Accuracy)	72.71%	31.04%	27.29%	0.29	0.58
	Neural Network (NeuralNet)	61.49%	31.18%	73.05%	0.44	0.68
	Neural Network (TabNet)	69.69%	35.09%	56.56%	0.43	0.74
P6C-YOUNGADULTS	Logistic Regression (Precision)	67.51%	41.38%	53.41%	0.47	0.69
	Decision Tree (Precision)	69.01%	36.92%	23.44%	0.29	0.56
	Logistic Regression (Recall)	66.80%	40.63%	54.01%	0.46	0.69
	Decision Tree (Recall)	69.16%	39.53%	30.27%	0.34	0.59
	Logistic Regression (Accuracy)	67.82%	41.95%	54.90%	0.48	0.68
	Decision Tree (Accuracy)	70.03%	41.57%	31.45%	0.36	0.59
	Neural Network (NeuralNet)	62.30%	36.77%	58.16%	0.45	0.65
P6D-ADULTS	Neural Network (TabNet)	62.70%	36.40%	54.01%	0.43	0.65
	Logistic Regression (Precision)	69.59%	41.39%	50.80%	0.46	0.70
	Decision Tree (Precision)	69.95%	37.18%	28.60%	0.32	0.63
	Logistic Regression (Recall)	69.51%	41.34%	51.33%	0.46	0.70
	Decision Tree (Recall)	63.22%	33.46%	47.07%	0.39	0.65
	Logistic Regression (Accuracy)	69.59%	41.39%	50.80%	0.46	0.70
	Decision Tree (Accuracy)	68.44%	34.61%	28.95%	0.32	0.61
P6E-OLDERADULTS	Neural Network (NeuralNet)	63.80%	38.44%	73.54%	0.50	0.70
	Neural Network (TabNet)	63.98%	37.88%	68.03%	0.49	0.71
	Logistic Regression (Precision)	68.96%	31.07%	48.07%	0.38	0.66
	Decision Tree (Precision)	71.56%	27.97%	28.74%	0.28	0.61
	Logistic Regression (Recall)	68.95%	31.07%	48.07%	0.38	0.66
	Decision Tree (Recall)	64.76%	29.17%	56.02%	0.38	0.68
	Logistic Regression (Accuracy)	68.96%	31.07%	48.07%	0.38	0.66
P6F-OLDERADULTS	Decision Tree (Accuracy)	70.34%	27.95%	32.65%	0.30	0.61
	Neural Network (NeuralNet)	58.97%	28.16%	70.64%	0.40	0.69
	Neural Network (TabNet)	59.26%	28.88%	73.92%	0.42	0.70
	Logistic Regression (Precision)	67.37%	21.30%	41.03%	0.28	0.60
	Decision Tree (Precision)	65.57%	19.76%	39.91%	0.26	0.57
	Logistic Regression (Recall)	67.37%	21.30%	41.03%	0.28	0.60
	Decision Tree (Recall)	62.40%	21.25%	52.69%	0.30	0.61
P6G-ADULTS	Logistic Regression (Accuracy)	67.37%	21.30%	41.03%	0.28	0.60
	Decision Tree (Accuracy)	72.45%	19.51%	24.89%	0.22	0.54
	Neural Network (NeuralNet)	53.86%	18.27%	56.95%	0.28	0.58
	Neural Network (TabNet)	67.82%	22.09%	42.60%	0.29	0.63

Figura 41. Resultados datos de prueba por modelo para cada dataset

Se adjuntan como anexo un archivo Júpiter con el código, los resultados, matrices de confusión y curvas ROC de cada uno de los modelos entrenados:

- https://github.com/SValduezaL/TFM-Master-IA-INESDI/blob/main/Fase1_ML_Prediction_NoShows/Machine_learning_analisis_v6.ipynb

Análisis de resultados

El análisis de los resultados revela hallazgos significativos que arrojan luz sobre el rendimiento de los modelos predictivos implementados. En primer lugar, es destacable la consistencia observada entre los resultados de entrenamiento y prueba, lo cual sugiere que las estrategias empleadas durante el proceso de entrenamiento fueron efectivas en prevenir el overfitting, permitiendo a los modelos capturar patrones relevantes en los datos, con un nivel de complejidad adecuado, sin caer en el underfitting, evidenciando así un equilibrio adecuado en su capacidad de generalización.

Un aspecto fundamental que respalda la validez de todos los modelos evaluados es su rendimiento superior al azar, reflejado en valores de AUC (Área Bajo la Curva ROC) consistentemente por encima de 0.5. Este resultado confirma su capacidad para discriminar entre las clases de interés de manera efectiva.

Sin embargo, el análisis del mapa de calor revela un desafío persistente en términos de precisión. Esta métrica se muestra como la más difícil de optimizar para todos los modelos, independientemente del conjunto de datos utilizado. Incluso los esfuerzos específicos de optimización mediante búsqueda en cuadrícula (GridSearchCV) en los modelos de Scikit-Learn no lograron elevar la precisión por encima del 50%. Esta observación tiene implicaciones prácticas significativas, ya que indica que en más de la mitad de los casos en que se predice la no asistencia de un paciente, éste termina asistiendo. Tal situación plantea retos importantes para la gestión eficiente de citas y recursos médicos.

En contraste con los desafíos en la precisión, el Recall emerge como la métrica más destacada en el rendimiento de los modelos. Se alcanzaron valores notables, llegando hasta un 83.7% en la base de datos P1-ALL-ADASYN utilizando el modelo de NeuralNet. Este alto recall es particularmente valioso en el contexto del proyecto, ya que minimiza la predicción errónea de asistencias que no se materializan, contribuyendo así a reducir los espacios vacíos en las agendas médicas y optimizar la utilización de recursos en la fase operativa del proyecto.

Entre los modelos evaluados, TabNet se distingue por su rendimiento superior en métricas clave. Alcanza el mejor AUC de 0.74 en el conjunto de datos P6A-KIDS, demostrando una excelente capacidad discriminativa. Además, TabNet logra los mejores F1 score, llegando hasta 0.49 en el conjunto de datos P6C-YOUNGADULTS, lo que indica un mejor equilibrio entre Precisión y Recall que los otros modelos analizados. Esta consistencia en diferentes métricas posiciona a TabNet como el modelo más prometedor para el proyecto.

A modo de discusión, queda pendiente por evaluar el rendimiento de modelos adicionales como Support Vector Machine (SVM) y modelos ensamblados. Estos enfoques no han sido valorados hasta ahora debido a limitaciones de hardware, como se planteó anteriormente. Sin embargo, explorar estos modelos podría ofrecer nuevas perspectivas y mejoras adicionales en el rendimiento predictivo.

Fase 2. Implementación de un sistema de overbooking

Estudio de bibliografía relacionada

La extensa bibliografía desarrollada en relación con los Sistemas “System Appointment Scheduling” (SAS) es un buen indicador de la dificultad para resolver el problema, como también de la importancia que tiene mejorar el servicio en atención médica y disminuir costos asociados.

En “A Review of Optimization Studies for System Appointment Scheduling” (11) se definen las características principales de un SAS, destacando que, en los servicios ambulatorios, una cuestión central para la programación de las citas es cómo asignar los espacios de tiempo disponibles para los pacientes, reduciendo las demoras de los pacientes y la disponibilidad de los médicos o el tiempo adicional.

Son muchos los factores que complican la resolución real del problema, empezando por la selección del marco en el que se toman las decisiones para optimizar las citas, según el cual se identifican 3 tipos de decisiones:

- **Estratégicas:** decisiones a largo plazo que determinan la estructura principal del Sistema. Las políticas tradicionales implican que toda la capacidad de la clínica es cubierta con pacientes preasignados mediante cita previa. Existen otro tipo de centros médicos donde el acceso es abierto, y las consultas se van atendiendo a medida que los pacientes acceden al centro (walk-ins). También se pueden implementar estrategias híbridas, donde convivan pacientes con cita previa y pacientes que se presentan a consulta el mismo día (walk-ins), con más o menos restricciones al respecto. Nuestro proyecto de estudio se comporta como un sistema híbrido, ya que en la base de datos hay muchos pacientes que tienen el “AppointmentDay” el mismo día del “ScheduledDay”. Aceptar pacientes sin cita previa disminuye los costes de inactividad asociados al absentismo, pero aumenta la complejidad del modelo. Otra decisión estratégica dentro las políticas tradicionales es realizar una programación de citas “offline” o “online”. El método “offline” implica esperar a asignar las citas una vez se han recibido todas las solicitudes (sólo apto para usos muy particulares), mientras que en el método “online” todas las citas se asignan inmediatamente una vez se recibe la solicitud (nuestro caso).
- **Tácticas:** decisiones más a medio plazo para determinar cómo se utilizan los servicios, si se permite la elección de médico, si se otorga preferencia a determinados grupos (si se va a priorizar el acceso a consulta a pacientes con múltiples dolencias médicas, o a aquellos que tienen becas de acceso al centro médico, o a los pertenecientes a unas determinadas mutuas, etc.). Otra decisión táctica muy importante es definir si se permite o no overbooking, y en qué grado. En este caso particular, poner muchas restricciones al overbooking puede mejorar la calidad del servicio, pero seguramente a expensas de aumentar los costes de inactividad. La definición de la duración de los slots en los que se asigna y atiende a los pacientes, o de la diferenciación o no de los slots en bloques, también afecta considerablemente a la capacidad de la clínica.
- **Operativas:** son las decisiones a corto plazo que hay que tomar a la hora de asignar las citas. Si éstas se toman con enfoques basados en reglas bien definidas (RBA – Rule Based Approaches), o con enfoques más basados en la optimización de algún parámetro (OBA – Optimization Based Approaches). Las RBA son más sencillas de aplicar, pero no garantizan un rendimiento óptimo, mientras que las OBA están específicamente estudiadas para alcanzar el mejor comportamiento, aunque son mucho más difíciles de diseñar. Nuestro objetivo en este proyecto es estudiar y comparar diferentes decisiones operativas, incluyendo RBA tradicionales, otras RBA no tan tradicionales, y una OBA basada en la optimización de una función de asignación de citas para minimizar los costes del SAS.

La siguiente tabla, extraída de “A Review of Optimization Studies for System Appointment Scheduling” (11), resume las principales características relacionadas con el marco de decisiones de un SAS.

Decision Types	Features
Strategic decisions	(a) Long-term decisions; (b) Determine the main system structure; (c) Include access policy; number of resources; policy on acceptance of walk-ins; types of scheduling.
Tactical decisions	(a) Medium-term decisions; (b) Explaining how patients are placed as a whole; (c) Maximizing resource utilization and the accessibility of nursing services.
Operational decisions	(a) Short-term decisions; (b) Focused on the efficient scheduling of individual patients; (c) Incorporating both the rule-based approach (RBA) and the optimization-based approach (OBA).

Figura 42. Marco de decisiones de un Sistema SAS

Así pues, la dificultad en la resolución del problema radica en la multitud de decisiones que se deben realizar para definir el problema de optimización estocástica, así como en el grado de incertidumbre que tienen los parámetros y variables que finalmente se decide tener en cuenta.

Parámetros que aportan mayor incertidumbre y variabilidad a los Sistemas SAS:

- **Incertidumbre en las predicciones de asistencia.** Los modelos de predicción están lejos de ser perfectos. Aunque los resultados son mejores que una mera asignación de probabilidad media de asistencia o una predicción al azar, las métricas de precisión y sensibilidad siguen proporcionando un grado de incertidumbre bastante elevado, y que será automáticamente trasladado al modelo de optimización.
- **Variedad e incertidumbre en los factores ambientales.** El número de factores ambientales que se pueden aplicar al sistema son muy numerosos, y seguramente diferentes para cada centro hospitalario o de salud, pues dependen de las decisiones estratégicas y tácticas mencionadas más arriba. Adicional, muchos de ellos añaden unos valores de incertidumbre que ni siquiera están estudiados por modelos predictivos. Así pues, podemos distinguir entre factores más certeros (número de servicios integrados, número de doctores, número de citas por sesión, prioridad en la atención) y más inciertos (puntualidades, tiempos de atención, nivel de interrupción de los doctores, llegadas espontáneas).
- **Variedad en las reglas de cita y en reglas de secuencia.** Las reglas de cita definen las restricciones que imponemos al modelo para limitar las asignaciones (número de pacientes que se puede asignar a un único slot, duración de los slots, etc.), y las reglas de secuencia definen el orden con los que los pacientes son asignados a los slots en función de una determinada clasificación (prioritarios, primerizos, recurrentes, etc.). Forman parte de las decisiones operativas mencionadas por Tiantian et al (11). Sólo Cayirli, Veral y Rosen (12) ya realizaron un estudio probando 6 diferentes reglas de secuencia con 7 reglas de cita, lo que hace un total de 42 Sistemas de Asignación (AS, del inglés) diferentes. Para hacernos otra idea de la magnitud y complejidad del problema, Ho y Lau (13) comparaban 9 diferentes sistemas de citas en 27 diferentes escenarios clínicos caracterizados por 3 factores ambientales (probabilidad de ausencias, variación de tiempos de atención y el número de pacientes por sesión).
- **Definición del Coste.** Para minimizar la Función de Coste que permite realizar la mejor programación de citas médicas se tiene que definir primero cuál es dicho coste y cómo calcularlo, lo cual no es banal. Y como se indica a continuación, las posibilidades son prácticamente infinitas.

Para empezar, según Tiantian et al. (11) la optimización varía en función del objetivo que se pretenda conseguir: beneficio social, rentabilidad económica de la clínica, máxima amortización de recursos, etc. Así pues, Chew (14) y Kaandorp and Koole (15) dividen la función en 3 tipologías

de coste distintas: el tiempo de espera del paciente en consulta, el tiempo de inactividad de toda la infraestructura médica por falta de paciente, y el tiempo extra dedicado por la infraestructura médica a atender los pacientes después de una jornada laboral normal. Ho y Lau (13) realizan de forma similar, pero sin tener en cuenta el tiempo extra de los médicos. Y Harris and Samorani (16) utilizan los mismos conceptos, pero sin tener en cuenta el tiempo de inactividad de la infraestructura médica. Con otra perspectiva, Almaktoom (17) calcula el coste respecto a los costos de realizar una consulta, los de operación de la clínica y los de overbooking, referenciándolos al número de No-Shows y de Overbookings. Sin embargo, en Valenzuela-Núñez et al. (18) no consideran minimizar un coste, sino maximizar la utilidad que genera el centro sanitario por cada cita atendida, menos una penalidad por sobrecarga de overbooking, directamente proporcional a los casos de overbooking registrados. Y Lawley and Muthuraman (19) usaron un modelo estocástico para minimizar el número de pacientes desbordados de una cita a la siguiente.

Elección de las hipótesis para la resolución del problema y fórmula para el cálculo del Coste del Sistema a minimizar

Para definir el Problema de Optimización Estocástico se tiene que decidir sobre:

- **Función objetivo:** lo que se quiere optimizar, relacionado con los marcos de optimización mencionados por Tiantian et al. (11).
- **Variables de decisión:** aquello que se pretende ajustar en el modelo. En nuestro caso, la distribución de pacientes en los slots que dura una consulta médica.
- **Restricciones:** limitaciones con las que se restringen las posibles soluciones. Estas restricciones pueden ser el número de pacientes esperando, la capacidad máxima de pacientes que se puede atender en un día, o el horario máximo de atención, por ejemplo. Y formarán parte de las hipótesis a fijar en este apartado.
- **Variables aleatorias:** los elementos que están sujetos a variabilidad y que se modulan mediante distribuciones de probabilidad.

En nuestro caso, en aras de simplificar la resolución del problema, minimizamos las variables del modelo como sigue:

Factores ambientales

- Predicciones de Asistencia. Extraídas de la Fase 1.
- Tiempo de Atención al Paciente = 20 min. Igual para todos los pacientes sin excepción.
- Horario de Atención normal. De 9:00 a 13:00 y de 15:00 a 19:00 (8 horas, separadas en 2 bloques de 4 horas).
- Número de slots por sesión = 24 (consideradas las consultas de veinte minutos en una jornada normal de 8 horas).
- Número de slots por bloque = 12 (dos bloques de 4 horas por sesión, consideradas las consultas de veinte minutos).
- El resto de los factores los anulamos directamente:
 - No hay impuntualidad.
 - Tiempo de consulta fijo e invariable para todo paciente. No hay demoras en la atención, ni las consultas terminan antes de tiempo.
 - Un único servicio integrado.
 - etc.

Reglas de Asignación de Citas

Definimos 4 escenarios donde aplicamos distintas reglas de asignación de citas, para así poder a posteriori calcular los Costes del Sistema resultantes y compararlos:

Escenario 1 - Tradicional sin Overbooking (caso R11_24)

Sólo se permite un paciente por slot, sin posibilidad de espera por parte de los pacientes ni tiempos de atención extra, pero sí mucho coste por inactividad de los servicios médicos. Esto equivale a una asignación de 24 pacientes (24 slots) por servicio médico y día.

Escenario 2 - Tradicional con Overbooking (casos R1231_28, R1241_28, R1221_30, R1231_30, R133112_32 y R133112211241_32)

Sabiendo que hay pacientes que no asisten a su cita médica, se contempla la posibilidad de generar overbooking en algunos slots. Como la probabilidad promedio de No Show en nuestro dataset es del 20,19%, se decide programar más pacientes a la consulta médica a razón de 1,2019 pacientes extra, lo que equivale a una asignación de 30 pacientes por servicio médico y día, lo que, al estar distribuidos entre 24 slots, genera el overbooking. La regla de asignación mencionada es la de 1 slot con 2 pacientes, 3 slots con un paciente, y así sucesivamente hasta completar la asignación de los 30 pacientes. La siguiente figura muestra las diferentes Reglas de Asignación de Citas contempladas en este estudio.

Regla Asignación Cita	Número Slot - Bloque Mañana (9:00 a 13:00)												Descanso												Número Slot - Bloque Tarde (15:00 a 19:00)															
	Comida												Comida												Comida															
R1231_28	2	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	= 28 pacientes
R1241_28	2	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	= 28 pacientes
R1221_30	2	1	1	1	2	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	= 30 pacientes
R1231_30	2	1	1	1	2	1	1	1	2	1	1	1	1	2	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	= 30 pacientes
R133112_32	3	1	1	1	1	2	1	1	1	2	1	1	1	3	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	= 32 pacientes
R133112211241_32	3	1	1	1	1	2	1	1	1	2	1	1	1	3	1	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	= 32 pacientes

Figura 43. Reglas Asignación de Citas para el Escenario 2.

Escenario 3 - Usando Probabilidades de Asistencia con Overbooking (caso ProbShow)

En este caso no se hace una asignación siguiendo una regla de orden natural, sino que incluimos la variable de probabilidad de asistencia a la cita, determinada mediante el Modelo de ML de la Fase 1, para determinar si un paciente tiene cabida o no en un determinado slot. A la hora de asignar un paciente a una agenda médica, se suma, en los slots ya abiertos, las Probabilidades de Show de los pacientes ya asignados al slot en estudio y la del propio paciente que se está asignando, incluyéndolo en dicho slot sólo cuando dicho valor no supera un cierto valor del “Listón ProbShow”.

Este “Listón ProbShow” no es igual para cada slot, pues para minimizar el coste interesa asignar más pacientes en los primeros slots (para minimizar costes de inactividad) y menos en los últimos (para minimizar costes de tiempo extra), por lo que lo definimos como una función (lineal o cuadrática, según el caso estudiado) que va de más a menos, diferenciándola para los 2 bloques de slots definidos (del 1 al 12, y del 13 al 24). La siguiente figura muestra un par de ejemplos de los Listones ProbShow estudiados.

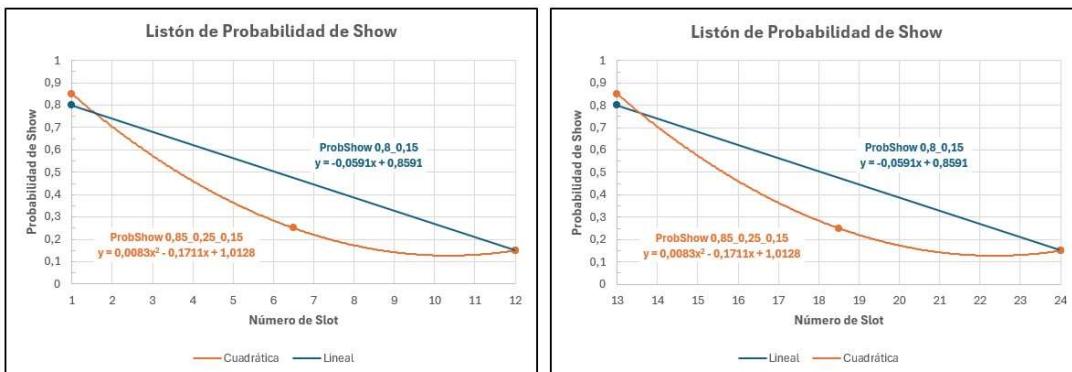


Figura 44. Un par de ejemplos de Listones de Probabilidad de Show.

Si después de repasar todos los slots abiertos de una consulta, se determina que el paciente no cabe en ninguno de ellos, se le asigna al siguiente slot disponible (es decir, siempre y cuando no se hayan abierto ya los 24 slots de una consulta).

Adicional, para evitar controlar mejor los costes de espera de los pacientes, (y a su vez evitar situaciones incómodas con los mismos), se incluye una Restricción de Contorno extra en la que no se permite agendar más de 3 pacientes para un mismo slot.

Escenario 4 - Modelo ML para Optimización de Overbooking

Este caso es el mismo caso del Escenario 3, pero en lugar de definir previamente la forma y valor del Listón de ProbShow, entrenaremos un modelo de ML para que encuentre el Listón de ProbShow óptimo a usar en cada Slot, convirtiéndose dichos Listones en los parámetros aprendidos que minimicen la función de coste.

Reglas de secuencia

Utilizaremos una secuencia FCFA (First Call First Appointment), también conocida directamente como Regla de No Secuencia, pues simplemente se asignará al paciente en el mejor slot disponible calculado mediante el modelo entrenado de overbooking (escenarios 3 y 4), o según la Regla de Asignación de Cita definida en el punto anterior para los escenarios 1 y 2.

Función de coste

Utilizaremos una fórmula parecida a la usada por Chew ([14](#)), dividiendo y calculando la función de coste en 3 términos independientes:

- **T_w = Tiempo de espera del paciente (Wait Cost).** Calculado como unidad temporal que debe esperar un paciente a ser atendido por encontrarse el slot sobrecargado. Se entiende que la paciencia de los pacientes no es indefinida, y, por lo tanto, el valor que tiene su tiempo no es igual si tiene que esperar una sola vez, a que tenga que esperar múltiples slots para ser atendido. Es por ello por lo que se hace una elevación cuadrática a la unidad temporal que tiene que esperar, siendo igual a 1 si sólo tiene que esperar 1 slot, 3 si tiene que esperar 2 slots, 6 si tiene que esperar 3 slots, etc. Dicha función queda representada por la siguiente fórmula:

$$T_{w_i}(y) = \frac{y^2 + y}{2}$$

y = número de slots a esperar por un paciente i

- **T_o = Tiempo extra a realizar por el doctor (Overtime Cost).** Calculado como unidad temporal que tiene que trabajar fuera de la sesión o jornada laboral normal de 8 horas. Si no tiene que trabajar ningún slot será igual a cero, si tiene que trabajar un slot será igual a 1, y así sucesivamente. (unidad temporal = 20 min = 1 slot).
- **T_i = Tiempo de inactividad en la clínica (Idle Cost).** Calculado como unidad temporal sin estar atendiendo a ningún paciente. Si el doctor está siempre ocupado será igual a cero, si queda un slot vacío será igual a 1, y así sucesivamente. (unidad temporal = 20 min = 1 slot).

A cada término independiente le damos unos costes monetarios distintos, pues queda claro que no tienen el mismo valor. Se entiende de que hay valores muy subjetivos en el cálculo de dichos importes, como pueden ser el coste que tiene para un paciente la hora de espera, o para un doctor la hora extra de trabajo extra, pero una buena aproximación es la siguiente

- **I_c / minuto = 5,00 €/min** (considerando el costo de una consulta de 20 min de un médico por 80 €, más 1 €/min extra de otros costes operativos de la clínica).

- **Wc / minuto = 0,50 €/min** (ya considerando el componente cuadrático de este coste en el cálculo de Tw).
- **Oc / minuto = 7,00 €/min** (considerando el mismo costo de un doctor para una consulta de 20 min, multiplicado x 1,5 por estar en horas extra, más el mismo 1 €/min extra de otros costes operativos de la clínica).

$$\begin{aligned} \text{Coste Total} &= T_I(x) * 5 \text{ €/min} * 20\text{min} + \\ &+ \sum_{i=1}^N T_{w_i}(y) * 0,5 \text{ €/min} * 20\text{min} + \\ &+ T_O(z) * 7 \text{ €/min} * 20\text{min} \end{aligned}$$

$T_I(x) = x$ = número de slots de inactividad.

$y =$ número de slots a esperar por un paciente i .

$N =$ número total de pacientes en una consulta.

$T_O(z) = z$ = número de slots a realizar tiempo extra.

Cálculo del Coste del Sistema en métodos tradicionales de asignación de citas médicas

Los métodos tradicionales de asignación de citas médicas corresponden a los Escenarios 1 y 2 citados anteriormente. Para calcular el coste se necesita realizar una serie de tareas previas, explicadas a continuación.

Creación de los Sets de Entrenamiento para Fase 2 y Validación

Eventualmente podríamos realizar el cálculo con todos los datos del Set de Pruebas, pero como se quiere entrenar un modelo de IA, y luego comparar resultados, es mejor realizar el cálculo sólo sobre un set de validación, idéntico para todos los escenarios.

Primero cargamos el Set de Pruebas heredado de la “Fase 1. Predicción de Asistencias de Citas Médicas”, seleccionando sólo las variables que necesitamos para calcular los Costes del Sistema SAS (y entrenar el modelo de IA del Escenario 4): 'PatientId', 'ScheduledDay', 'AppointmentDay', 'NoShow'.

A continuación, agrupamos los datos por “AppointmentDay”, y revisamos el número de citas y los porcentajes de “NoShow” según fecha de cita. De esta forma se observa que para los 2 primeros días de cita hay muy pocas citas en la base de datos, y prácticamente todas son No Show. Esto se debe a la forma en que se generaron los Sets de Entrenamiento y Prueba en la Fase 1. Cabe recordar que la partición se hizo para meter en el Set de Prueba las últimas citas de pacientes únicos, en estricto orden descendente de la fecha de cita, y, al forzar para que tuviese exactamente el mismo porcentaje de “NoShow”s que el dataset original, se fue completando con pacientes “NoShow” de fechas más antiguas. Debido a esta situación se decide descartar del Set de Pruebas las citas correspondientes a dichos días (ver la siguiente Figura).

Sets de Prueba completos:
El set para el 2016-05-30 tiene 60 citas, con un porcentaje de NoShows del 100.00%.
Descartamos los datos del 2016-05-30 por insuficiencia de datos y/o por haber excesivos No Show.
El set para el 2016-05-31 tiene 765 citas, con un porcentaje de NoShows del 66.80%.
Descartamos los datos del 2016-05-31 por insuficiencia de datos y/o por haber excesivos No Show.
El set para el 2016-06-01 tiene 3061 citas, con un porcentaje de NoShows del 18.75%.
El set para el 2016-06-02 tiene 3248 citas, con un porcentaje de NoShows del 18.29%.
El set para el 2016-06-03 tiene 3163 citas, con un porcentaje de NoShows del 19.60%.
El set para el 2016-06-06 tiene 3802 citas, con un porcentaje de NoShows del 18.73%.
El set para el 2016-06-07 tiene 3884 citas, con un porcentaje de NoShows del 17.64%.
El set para el 2016-06-08 tiene 4114 citas, con un porcentaje de NoShows del 17.14%.

Figura 45. Fechas descartadas del Set de Pruebas

Esta actuación rebaja el porcentaje de “NoShows” del 20,18% en el Set de Pruebas original al 18,29% en el nuevo conjunto, pero esto carece de relevancia para el cálculo y comparativa de costes en los diferentes escenarios.

La partición la hacemos del 70% para datos de entrenamiento de la Fase 2, y del 30% para datos de validación, manteniendo la misma proporción de “NoShows” (18,29%) en ambos conjuntos.

Formación de Agendas Médicas en función de las Reglas de Asignación

La asignación de pacientes a los diferentes slots de una consulta médica se realiza en función de las Reglas de Asignación de Citas definidas para cada Escenario, aplicando la Regla de Secuencia FCFA (First Call First Appointment).

Para aplicar la Regla de Secuencia FCFA lo primero que tenemos que hacer es ordenar el Set de Validación por fecha de solicitud de cita (“ScheduledDay”), y así poder iterar paciente por paciente en estricto orden de llamada.

Para el Escenario 1 – Tradicional sin Overbooking, sólo existe una regla de asignación (R11_24), ya que simplemente se trata de ir asignando a cada paciente el primer slot vacío de una consulta. En caso de no quedar slots disponibles, se considera como hipótesis que tenemos más doctores disponibles y, por lo tanto, se abre otra consulta. Así indefinidamente hasta que se hayan asignado todos los pacientes a una consulta para la fecha solicitada (“AppointmentDay”).

En el Escenario 2 – Tradicional con Overbooking, realizamos exactamente la misma operación, sólo que, para algunos slots (en función de la Regla de Asignación de Citas), en lugar de asignar un solo paciente, se asigna una lista de 2 o 3 pacientes, según el caso.

Las Agendas Médicas generadas son diccionarios con la siguiente estructura:

```
Agenda_medica_R1231_30 = {
    "2016-06-01": {
        "consulta_001": {
            "slot_01": [id_paciente, id_paciente],
            "slot_02": [id_paciente],
            ...
            "slot_24": [id_paciente]
        },
        ...
        "consulta_032": {
            "slot_01": [id_paciente, id_paciente],
            ...
        }
    },
    ...
    "2016-06-08": {
        "consulta_001": {
            "slot_01": [id_paciente],
            ...
        }
    }
}
```

Aunque posteriormente estos diccionarios se aplana para poder generar bases de datos tabulares y guardar las agendas en archivos tipo Excel.

Cálculo de Costes

Finalmente aplicamos la función de coste a cada una de las agendas generadas para los Escenarios 1 y 2. En el anexo correspondiente al cuaderno Jupyter Notebook Schedule_Optimization.ipynb se puede consultar el código aplicado para calcular dicho coste.

El siguiente resumen sintetiza el proceso:

- Se itera para cada consulta de cada fecha slot por slot, y paciente por paciente asignado a cada slot.
- Se comprueba si el slot está ocupado en consulta médica y si el paciente asistió a la cita.
- Si el slot está libre y el paciente asiste, se disminuye el **Coste de Inactividad**.
- Si el slot está ocupado y el paciente asiste, se traslada este paciente a la cabeza de fila del siguiente slot, y se empieza a calcular el **Coste de Espera**, aplicando la fórmula exponencial en función del número de slots que le va a tocar esperar.
- Si el paciente no asiste, se pasa al siguiente paciente del slot.
- Cuando se acaban los pacientes de un slot, se pasa al siguiente slot, teniendo en cuenta los cambios habidos por transferencia de pacientes.
- Para los slots 12 y 24 se calcula también el **Coste de Tiempo Extra**, identificando el número de pacientes que quedan por ser atendidos en dichos slots.
- En el Slot 12 añadimos una restricción sobre la cual no se pueden atender a más de 3 pacientes después de las 13:00 horas. El resto de los pacientes en espera, de existir, se mueven directamente al Slot 13, para ser atendidos a partir de las 15:00.

Cabe destacar que sólo se calculan los costes para las primeras 20 consultas médicas generadas de cada día (en total 120 consultas médicas para 6 días contemplados). El motivo de esta decisión se explica en el siguiente apartado, cuando se generan las agendas médicas usando probabilidades de asistencia.

Los resultados para el Escenario 1 y del Escenario 2 se muestran en las siguientes figuras.

COSTES EN CITAS MÉDICAS POR NO ASISTENCIA y OVERBOOKING								
REGLA DE ASIGNACIÓN	C_i : Coste Inactividad (consulta médica)		C_w : Coste de Espera (pacientes)		C_o : Coste Tiempo Extra (consulta médica)		COSTE TOTAL	
	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.
Regla R11_24 pacientes	690,83 €	49,03 €	0,00 €	0,00 €	0,00 €	0,00 €	690,83 €	49,03 €

Coste unitario del Idle Cost (C_i) / minuto = 5,00 €	Coste una consulta = $4 \text{ €} \times 20 \text{ min} = 80 \text{ €} + \text{otros Costes Operativos (1 €/min)}$
Coste unitario del Waiting Cost (C_w) / minuto = 0,50 €	Coste con incremento exponencial: $(x2 + x) / 2$, siendo x el número de slots de espera.
Coste unitario del Overtime Cost (C_o) / minuto = 7,00 €	Coste una consulta = $(4 \text{ €} \times 1,5) \times 20 \text{ min} = 120 \text{ €} + \text{otros Costes Operativos (1 €/min)}$

Figura 46. Coste Escenario 1 - Tradicional sin Overbooking.

COSTES EN CITAS MÉDICAS POR NO ASISTENCIA y OVERBOOKING								
REGLA DE ASIGNACIÓN	C_i : Coste Inactividad (consulta médica)		C_w : Coste de Espera (pacientes)		C_o : Coste Tiempo Extra (consulta médica)		COSTE TOTAL	
	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.
Regla R1231_28 pacientes	407,50 €	58,29 €	82,25 €	21,43 €	25,67 €	8,48 €	515,42 €	52,89 €
Regla R1241_28 pacientes	409,17 €	59,11 €	77,42 €	16,06 €	28,00 €	8,85 €	514,58 €	54,78 €
Regla R1221_30 pacientes	295,83 €	54,17 €	161,17 €	24,77 €	87,50 €	27,20 €	544,50 €	69,98 €
Regla R1231_30 pacientes	310,00 €	47,43 €	133,42 €	20,11 €	107,33 €	29,59 €	550,75 €	52,38 €
Regla R133112_32 pacientes	197,50 €	58,20 €	323,58 €	37,06 €	196,00 €	25,81 €	717,08 €	49,10 €
Regla R133112211241_32 pacientes	193,33 €	55,11 €	346,58 €	41,18 €	190,17 €	30,49 €	730,08 €	56,57 €

Coste unitario del Idle Cost (C_i) / minuto = 5,00 €	Coste una consulta = $4 \text{ €} \times 20 \text{ min} = 80 \text{ €} + \text{otros Costes Operativos (1 €/min)}$
Coste unitario del Waiting Cost (C_w) / minuto = 0,50 €	Coste con incremento exponencial: $(x2 + x) / 2$, siendo x el número de slots de espera.
Coste unitario del Overtime Cost (C_o) / minuto = 7,00 €	Coste una consulta = $(4 \text{ €} \times 1,5) \times 20 \text{ min} = 120 \text{ €} + \text{otros Costes Operativos (1 €/min)}$

Figura 47. Costes Escenario 2 - Tradicional con Overbooking.

Cálculo del Coste del Sistema con función basada en las predicciones de asistencia (Fase 1) para la asignación de citas médicas

Este apartado se corresponde al Escenario 3 – Usando Probabilidades de Asistencia con Overbooking (caso ProbShow), donde se aprovechan los cálculos de predicciones de asistencia obtenidas en la Fase 1 del Proyecto para optimizar las asignaciones de cita de los pacientes.

Creación de los Sets de Entrenamiento para Fase 2 y Validación

Para poder comprobar los resultados a posteriori, se tiene que usar en el Escenario 3 el mismo Set de Pruebas al utilizado en los Escenarios 1 y 2, pero se ha de completar con la columna correspondiente a la probabilidad de NoShow calculada con el mejor modelo de IA obtenido en la Fase 1, pues es la base del nuevo método para la asignación de pacientes en las agendas médicas.

Una vez completado el Set de Pruebas con esta nueva variable “Prob_NoShow”, se sigue exactamente el mismo procedimiento para dividirlo en el Set de Entrenamiento para Fase 2 y el Set de Validación. Así pues, los Sets de Validación de los diferentes escenarios son iguales, aunque en este escenario se haya añadido la variable “Prob_NoShow”.

Formación de Agendas Médicas en función de las Probabilidades de Asistencia de los Pacientes

Al igual que para los métodos tradicionales con Reglas de Asignación de Citas, lo primero que hay que hacer es ordenar el Set de Validación por fecha de solicitud de cita (“ScheduledDay”), y así poder iterar paciente por paciente en estricto orden de llamada.

Adicional, se calcula la Probabilidad de Show como [ProbShow = 1 – Prob_NoShow], que es la que se utiliza para medir cuando un grupo de pacientes excede o no el “Listón ProbShow”.

Por otro lado, se define la función del “Listón ProbShow” (ver [Figura 44](#)) como una Interpolación Polinómica en la forma de Lagrange (ver [Figura 48](#)), según la cual, para “n” puntos dados, se obtiene una interpolación polinómica de grado “n-1” que pasa por todos dichos puntos. Así pues, si se definen el listón de probabilidad de asistencia máximo para los slots 1 y 12, por ejemplo, obtenemos la línea recta (función polinómica de grado 1) que define el listón de probabilidad de asistencia máximo para todos los slots intermedios. Pero si al mismo tiempo fijamos el listón de probabilidad de asistencia máximo en un tercer punto intermedio, se obtiene en su lugar una función cuadrática (función polinómica de grado 2) para el “Listón ProbShow”.

```
def calculo_liston_ProbShow(x_values: list, y_values: list, x: int) -> float:
    ...
    Calcula la coordenada "y" de una función polinómica para una coordenada "x" dada,
    usando cualquier número de puntos (Interpolación polinómica de Lagrange)
    Parámetros:
        x_values: Lista de coordenadas "x" de los puntos conocidos por donde debe pasar
                  la función polinómica.
        y_values: Lista de coordenadas "y" de los 3 puntos conocidos por donde debe pasar
                  la función polinómica.
        x: La coordenada para la cual se desea calcular la coordenada "y" de la
            función polinómica de Lagrange.
    Retorna:
        float: La coordenada "y" correspondiente en la función polinómica definida
               por un número indeterminado de puntos.
    ...
    grado_funcion = len(x_values)
    y = 0.0

    for i in range(grado_funcion):
        # Calcula el polinomio L_i(x)
        L_i = 1
        for j in range(grado_funcion):
            if i != j:
                L_i *= (x - x_values[j]) / (x_values[i] - x_values[j])

        # Añade el término correspondiente a y_i
        y += y_values[i] * L_i

    return y
```

[Figura 48. Interpolación Polinómica en la forma de Lagrange](#)

En el Escenario 3 se prueba con diferentes listones, tanto lineales como cuadráticos, para buscar y encontrar el mínimo Coste del sistema SAS.

El siguiente paso para seguir minimizando el coste sería seguir iterando con funciones polinómicas de grado superior, pero el número de variantes a desarrollar es demasiado alto como para ser capaces de, mediante iteraciones manuales, encontrar el mínimo global de la función. La función polinómica ideal sería una de grado 11, que es la que nos proporcionaría el grado de libertad máximo para cada uno de los 12 slots que definen un bloque, y diferenciar la función del bloque 1 de la del bloque 2 con valores distintos, ya que en el Escenario 3 se usa la misma función para ambos bloques (ver [Figura 44](#)). En el Escenario 4 es donde se consigue esta optimización, usando un modelo de IA que, entrenado con los datos del Set de Entrenamiento preparado en esta Fase 2, devuelva el “Listón Prob_Show” óptimo (aquel que minimiza el coste) para cada uno de los 24 Slots de una consulta médica.

Las Agendas Médicas generadas en este Escenario 3 son diccionarios que difieren un poco a los de los Escenarios 1 y 2, ya que para cada slot no sólo se han de guardar los id's de los pacientes, sino también la probabilidad de asistencia a la consulta de dicho paciente, pues es un dato crucial para poder seguir asignando pacientes. Concretamente presentan la siguiente estructura:

```
Agenda_medica_ProbShow_0.85_0.25_0.15 = {
    "2016-06-01": {
        "consulta_001": {
            "slot_01": [(id_paciente, ProbShow), (id_paciente, ProbShow)],
            "slot_02": [(id_paciente, ProbShow), (id_paciente, ProbShow)],
            "slot_03": [(id_paciente, ProbShow)],
            ...
            "slot_24": [(id_paciente, ProbShow)]
        },
        "consulta_002": {
            "slot_01": [(id_paciente, ProbShow), (id_paciente, ProbShow), (id_paciente, ProbShow)],
            ...
        },
        ...
        "consulta_034": {
            "slot_01": [(id_paciente, ProbShow)],
            ...
        }
    },
    ...
    "2016-06-08": {
        "consulta_001": {
            "slot_01": [id_paciente, ProbShow],
            ...
        }
    }
}
```

Al igual que en los Escenarios 1 y 2, estos diccionarios se aplanan posteriormente para poder generar bases de datos tabulares y guardar las agendas en archivos tipo Excel.

Analizando las Agendas Médicas generadas para cada sub-escenario de ProbShow, se observa que las primeras consultas generadas son las que tienen más pacientes asignados. Esto se debe a que cuando un paciente solicita una consulta para un día determinado, lo primero que se hace es ir repasando los slots ya generados, uno a uno, desde la primera consulta, para ver si encaja en alguno de ellos. Por lo tanto, es natural que las primeras consultas sean las más saturadas, dejando siempre a los pacientes con mayores probabilidades de asistencia al final del recorrido (pues ya no caben en los slots abiertos). En la realidad esto no sucede así, ya que no se pueden abrir consultas médicas de forma indefinida (existe un número limitado de doctores, siendo incluso sólo uno cuando se busca una atención especializada), y lo que se hace realmente es buscar otro día de consulta. La realidad se parece

más a las primeras consultas de nuestra serie, pues siempre se intenta copar la agenda médica de un día determinado.

Por lo tanto, para evitar esta distorsión de la realidad, limitamos el cálculo de los Costes del Sistema SAS a las primeras **20 consultas** para cada fecha de consulta, pues es donde, de media, se conserva una distribución de número de pacientes más uniforme y parecida a la inicial, tal y como se aprecia en las siguientes figuras.

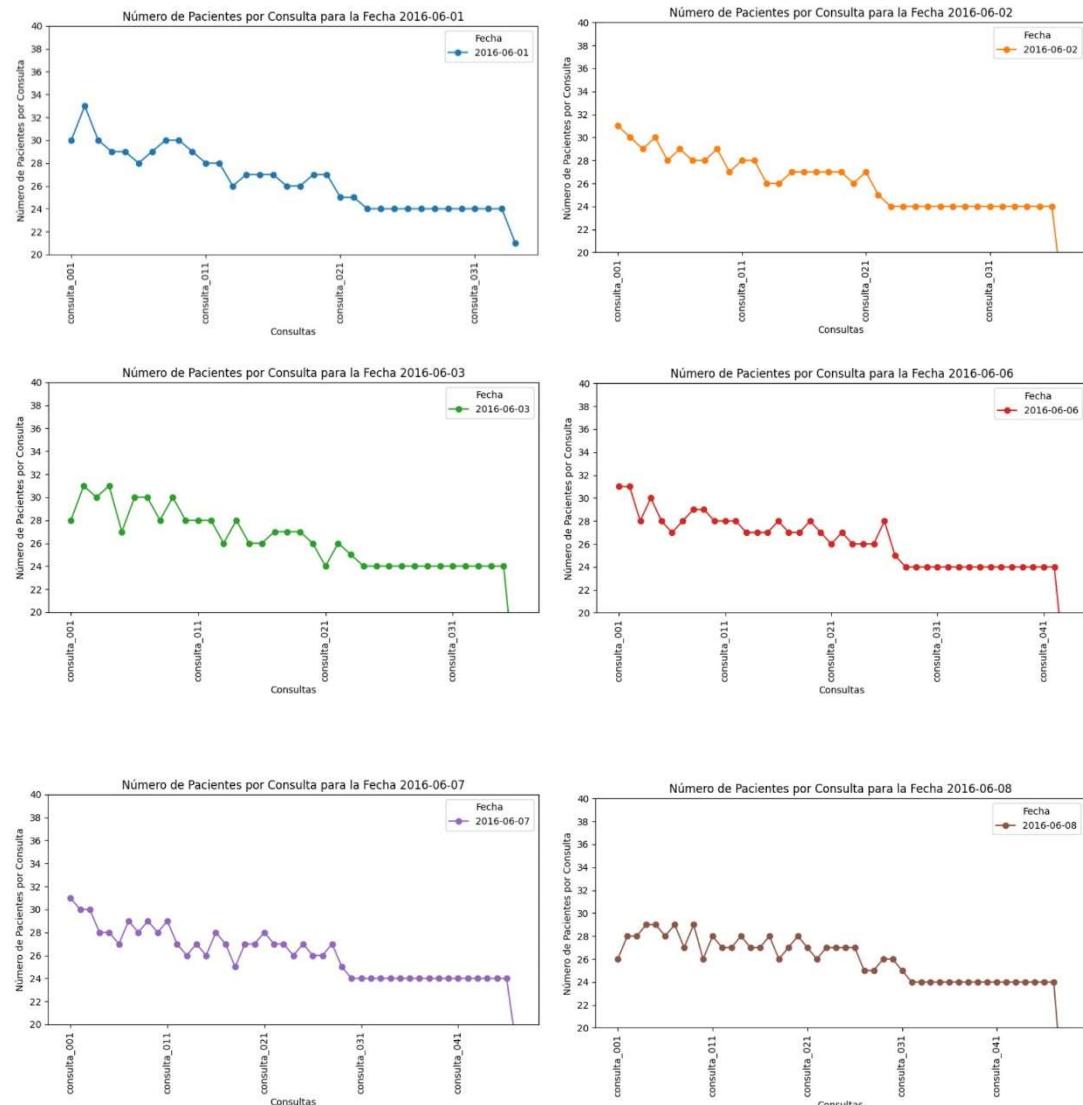


Figura 49. Número Pacientes por Consulta en Escenario 3 ProbShow_0.85_0.25_0.15

Cálculo de Costes

El procedimiento para el cálculo de los costes es igual, utilizando exactamente las mismas funciones, al utilizado en los Escenarios 1 y 2.

Los resultados para el Escenario 3, tanto para las funciones del “Listón ProbShow” lineal como cuadrática, se muestran en las siguientes figuras.

REGLA DE ASIGNACIÓN	C_I : Coste Inactividad (consulta médica)		C_W : Coste de Espera (pacientes)		C_O : Coste Tiempo Extra (consulta médica)		COSTE TOTAL	
	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.
Regla usando Probabilidad Show 0,65_0,15	521,67 €	33,12 €	43,50 €	18,82 €	5,83 €	2,86 €	571,00 €	42,50 €
Regla usando Probabilidad Show 0,65_0,2	519,17 €	30,56 €	43,58 €	17,42 €	8,17 €	6,88 €	570,92 €	41,93 €
Regla usando Probabilidad Show 0,65_0,25	513,33 €	29,44 €	48,08 €	19,26 €	11,67 €	7,23 €	573,08 €	46,62 €
Regla usando Probabilidad Show 0,7_0,15	470,83 €	22,23 €	70,67 €	34,36 €	17,50 €	16,42 €	559,00 €	63,59 €
Regla usando Probabilidad Show 0,7_0,2	460,83 €	27,82 €	69,75 €	20,10 €	16,33 €	8,48 €	546,92 €	46,50 €
Regla usando Probabilidad Show 0,7_0,25	461,67 €	27,87 €	78,42 €	24,64 €	19,83 €	6,88 €	559,92 €	51,13 €
Regla usando Probabilidad Show 0,7_0,3	452,50 €	27,70 €	77,92 €	30,65 €	28,00 €	10,84 €	558,42 €	67,45 €
Regla usando Probabilidad Show 0,75_0,1	412,50 €	26,22 €	104,33 €	17,71 €	24,50 €	13,10 €	541,33 €	44,03 €
Regla usando Probabilidad Show 0,75_0,15	408,33 €	31,09 €	118,83 €	17,73 €	33,83 €	19,00 €	561,00 €	59,08 €
Regla usando Probabilidad Show 0,75_0,2	399,17 €	21,31 €	129,92 €	15,36 €	35,00 €	7,67 €	564,08 €	29,05 €
Regla usando Probabilidad Show 0,75_0,25	380,00 €	37,82 €	130,25 €	14,93 €	40,83 €	10,30 €	551,08 €	53,55 €
Regla usando Probabilidad Show 0,75_0,3	380,00 €	20,00 €	143,75 €	27,59 €	43,17 €	17,38 €	566,92 €	55,37 €
Regla usando Probabilidad Show 0,8_0,1	345,00 €	16,43 €	162,50 €	23,27 €	49,00 €	15,96 €	556,50 €	51,08 €
Regla usando Probabilidad Show 0,8_0,15	333,33 €	15,06 €	142,00 €	15,14 €	37,33 €	16,37 €	512,67 €	37,22 €
Regla usando Probabilidad Show 0,8_0,2	324,17 €	21,08 €	178,42 €	31,88 €	57,17 €	14,96 €	559,75 €	51,38 €
Regla usando Probabilidad Show 0,8_0,25	323,33 €	20,66 €	184,25 €	21,97 €	65,33 €	9,56 €	572,92 €	41,54 €
Regla usando Probabilidad Show 0,8_0,3	304,17 €	22,68 €	182,92 €	31,62 €	65,33 €	26,81 €	552,42 €	76,28 €
Regla usando Probabilidad Show 0,85_0,1	287,50 €	20,92 €	224,33 €	40,39 €	77,00 €	21,23 €	588,83 €	76,40 €
Regla usando Probabilidad Show 0,85_0,15	265,83 €	19,34 €	274,08 €	39,61 €	92,17 €	28,49 €	632,08 €	80,60 €
Regla usando Probabilidad Show 0,85_0,2	256,67 €	22,29 €	273,58 €	43,34 €	101,50 €	23,32 €	631,75 €	75,31 €

Coste unitario del Idle Cost (C_I) / minuto =	5,00 €
Coste unitario del Waiting Cost (C_W) / minuto =	0,50 €
Coste unitario del Overtime Cost (C_O) / minuto =	7,00 €

Coste una consulta = 4 € x 20 min = 80 € + otros Costes Operativos (1 €/min)

Coste con incremento exponencial: $(x2+x) / 2$, siendo x el número de slots de espera.

Coste una consulta = (4 € x 1,5) x 20 min = 120 € + otros Costes Operativos (1 €/min)

Figura 50. Costes Escenario 3 - Función lineal del "Listón ProbShow"

REGLA DE ASIGNACIÓN	C_I : Coste Inactividad (consulta médica)		C_W : Coste de Espera (pacientes)		C_O : Coste Tiempo Extra (consulta médica)		COSTE TOTAL	
	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.
Regla usando Probabilidad Show 0,7_0,3_0,2	505,83 €	31,53 €	53,58 €	25,24 €	12,83 €	10,30 €	572,25 €	49,10 €
Regla usando Probabilidad Show 0,7_0,35_0,2	489,17 €	24,78 €	55,17 €	23,89 €	9,33 €	5,72 €	553,67 €	42,45 €
Regla usando Probabilidad Show 0,7_0,4_0,2	479,17 €	22,23 €	61,00 €	21,89 €	14,00 €	9,90 €	554,17 €	43,66 €
Regla usando Probabilidad Show 0,7_0,45_0,2	460,83 €	27,82 €	69,75 €	20,10 €	16,33 €	8,48 €	546,92 €	46,50 €
Regla usando Probabilidad Show 0,75_0,2_0,1	470,00 €	22,14 €	51,08 €	17,70 €	2,33 €	3,61 €	523,42 €	35,66 €
Regla usando Probabilidad Show 0,75_0,25_0,1	462,50 €	23,18 €	62,50 €	9,74 €	8,17 €	5,27 €	533,17 €	24,36 €
Regla usando Probabilidad Show 0,75_0,3_0,1	450,83 €	23,96 €	68,92 €	12,24 €	10,50 €	7,34 €	530,25 €	28,18 €
Regla usando Probabilidad Show 0,75_0,35_0,1	446,67 €	23,59 €	68,42 €	23,68 €	15,17 €	14,29 €	530,25 €	43,54 €
Regla usando Probabilidad Show 0,8_0,25_0,15	435,83 €	25,58 €	84,83 €	13,60 €	15,17 €	12,06 €	535,83 €	31,68 €
Regla usando Probabilidad Show 0,8_0,3_0,15	416,67 €	20,41 €	98,67 €	16,02 €	18,67 €	9,56 €	534,00 €	34,75 €
Regla usando Probabilidad Show 0,8_0,35_0,15	402,50 €	17,54 €	123,58 €	23,59 €	28,00 €	15,96 €	554,08 €	40,73 €
Regla usando Probabilidad Show 0,8_0,4_0,15	385,83 €	16,25 €	134,92 €	34,81 €	38,50 €	7,34 €	559,25 €	32,35 €
Regla usando Probabilidad Show 0,85_0,25_0,15	393,33 €	32,66 €	93,83 €	26,24 €	18,67 €	15,76 €	505,83 €	69,23 €
Regla usando Probabilidad Show 0,85_0,3_0,15	373,33 €	23,38 €	125,42 €	33,99 €	28,00 €	17,15 €	526,75 €	42,82 €
Regla usando Probabilidad Show 0,85_0,35_0,15	361,67 €	8,76 €	140,83 €	25,63 €	47,83 €	18,48 €	550,33 €	44,34 €
Regla usando Probabilidad Show 0,85_0,4_0,15	340,83 €	15,94 €	149,00 €	32,86 €	47,83 €	18,48 €	537,67 €	55,53 €

Coste unitario del Idle Cost (C_I) / minuto =	5,00 €
Coste unitario del Waiting Cost (C_W) / minuto =	0,50 €
Coste unitario del Overtime Cost (C_O) / minuto =	7,00 €

Coste una consulta = 4 € x 20 min = 80 € + otros Costes Operativos (1 €/min)

Coste con incremento exponencial: $(x2+x) / 2$, siendo x el número de slots de espera.

Coste una consulta = (4 € x 1,5) x 20 min = 120 € + otros Costes Operativos (1 €/min)

Figura 51. Costes Escenario 3 - Función cuadrática del "Listón ProbShow"

Desarrollo del modelo de ML para el cálculo óptimo de overbooking usando predicciones de asistencia

Pendiente

Comparativa de Costes y Conclusiones en la optimización del overbooking

Antes que nada, es importante mencionar la variabilidad y subjetividad que presentan los resultados en función de la fórmula de coste utilizada, así como de los costes unitarios que se asignen a la misma. Aun así, esto no empañía los resultados aquí obtenidos, pues todos están siendo comparados bajo el mismo calibre. En caso de querer medir el coste de otra forma, o con otros valores que se consideren más apropiados, se tendrían que rehacer los cálculos, y, posiblemente, los escenarios óptimos cambiarían. Por eso se destaca la importancia de personalizar bien dicha fórmula para cada caso concreto que se quiera estudiar antes de iterar los cálculos para cada escenario.

Otra característica que se observa en los resultados es la gran variabilidad de coste entre las consultas de un mismo escenario. La dependencia del coste respecto a la agenda médica de cada consulta, aun aplicando la misma Regla de Asignación de Citas, no es nada desdeñable, y eso se refleja en las altas desviaciones estándar del Coste Total, con valores entre el 5% y el 14% de la propia media.

En la siguiente figura se pueden observar los mejores resultados obtenidos para cada escenario.

REGLA DE ASIGNACIÓN	COSTES EN CITAS MÉDICAS POR NO ASISTENCIA y OVERBOOKING							
	C_I : Coste Inactividad (consulta médica)		C_W : Coste de Espera (pacientes)		C_O : Coste Tiempo Extra (consulta médica)		COSTE TOTAL	
	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.	Media	Desv. Est.
Regla R11_24 pacientes	690,83 €	49,03 €	0,00 €	0,00 €	0,00 €	0,00 €	690,83 €	49,03 €
Regla R1231_28 pacientes	407,50 €	58,29 €	82,25 €	21,43 €	25,67 €	8,48 €	515,42 €	52,89 €
Regla R1241_28 pacientes	409,17 €	59,11 €	77,42 €	16,06 €	28,00 €	8,85 €	514,58 €	54,78 €
Regla usando Probabilidad Show 0,8_0,15	333,33 €	15,06 €	142,00 €	15,14 €	37,33 €	16,37 €	512,67 €	37,22 €
Regla usando Probabilidad Show 0,85_0,25_0,15	393,33 €	32,66 €	93,83 €	26,24 €	18,67 €	15,76 €	505,83 €	69,23 €

Coste unitario del Idle Cost (C_I) / minuto =	5,00 €	Coste una consulta = 4 € x 20 min = 80 € + otros Costes Operativos (1 €/min)
Coste unitario del Waiting Cost (C_W) / minuto =	0,50 €	Coste con incremento exponencial: $(x2 + x) / 2$, siendo x el número de slots de espera.
Coste unitario del Overtime Cost (C_O) / minuto =	7,00 €	Coste una consulta = (4 € x 1,5) x 20 min = 120 € + otros Costes Operativos (1 €/min)

Figura 52. Mejores Costes del Sistema SAS por Escenario

Queda evidenciada la optimización del coste entre aplicar una Regla de Asignación de Cita que permita un ligero overbooking y no hacerlo (R11_24), pues es el Escenario 1 el que se lleva los mayores costes con diferencia, provocados en su totalidad por un coste de inactividad excesivo debido a la falta de asistencia de los pacientes.

La diferencia entre los mejores resultados del Escenario 2 y los mejores resultados del Escenario 3 es más ajustada, pero se aprecia mejora cuando se usan las predicciones de asistencia calculadas con el Modelo de IA de la Fase 1.

Dada la alta variabilidad de costes entre consultas de un mismo escenario, se destaca la importancia de la desviación estándar que presenta el Escenario 3 ProbShow_0.8_0.15, pues, a pesar de tener una media de Coste Total algo superior al Escenario 3 ProbShow_0.85_0.25_0.15, reduce prácticamente a la mitad su desviación estándar, lo que le proporciona mayor robustez a la hora de garantizar el mínimo coste.

Estas reducciones de costos se traducen automáticamente en **ahorros económicos** para un centro hospitalario ($C_I + C_O$), y en una **reducción de las listas de espera**, pues se disminuyen considerablemente los tiempos de inactividad. En este último aspecto, destaca la reducción de los tiempos de inactividad (C_I) en el Escenario 3 ProbShow_0.8_0.15. Los pacientes sufren un poco más en cuanto a la calidad de asistencia en el centro hospitalario, ya que aumentan los tiempos de espera (C_W), pero puede llegar a compensar con la reducción de tiempo en las listas de espera recién mencionada.

La siguiente tabla muestra los ahorros económicos ($C_l + C_o$) anuales generados por cada uno de estos escenarios, referenciándolo contra el Escenario 1 sin overbooking (R11_24), en un hipotético Centro Hospitalario con 20 doctores especialistas.

	$C_l + C_o$	Dif. con R11_24	Consultas Médicas	Días Laborables	Ahorros Económicos
Regla R11_24 pacientes	690,83 €	0,00 €	20	247	0 €
Regla R1231_28 pacientes	433,17 €	257,67 €	20	247	1.272.873 €
Regla R1241_28 pacientes	437,17 €	253,67 €	20	247	1.253.113 €
Regla usando Probabilidad Show 0,8_0,15	370,67 €	320,17 €	20	247	1.581.623 €
Regla usando Probabilidad Show 0,85_0,25_0,15	412,00 €	278,83 €	20	247	1.377.437 €

Figura 53. Ahorros Económicos Anuales para Centro Hospitalario con 20 consultas médicas

Sin embargo, existen otras cuestiones estratégicas, más comerciales que económicas, las que finalmente definen la adopción de un escenario u otro. En el caso anterior, por ejemplo, económicamente para el Centro Hospitalario sale más rentable adoptar el Escenario 3 ProbShow_0.8_0.15, sin embargo, los altos tiempos de espera en este escenario pueden generar un malestar tal en los pacientes que provoque una pérdida de clientes y de negocio, desbaratando toda expectativa de ganancia económica. La Calidad del Servicio Hospitalario es algo que hay que respetar pues, aunque no incida directamente en las cuentas de resultados de la empresa, siempre repercute indirectamente.

Fase 3. Creación de un Asistente Virtual basado en el Procesamiento de Lenguaje Natural (NLP)

En esta fase, nos enfocamos en el desarrollo de un asistente virtual de tercera generación utilizando técnicas avanzadas de Procesamiento de Lenguaje Natural (NLP) para gestionar de manera eficiente las citas médicas en el Hospital Clínico. Este asistente, llamado Basilio, está diseñado para interactuar con los pacientes a través de plataformas de mensajería como Telegram, proporcionando un servicio accesible y eficiente para la gestión de citas médicas. A continuación, se detalla el proceso de desarrollo, las herramientas y tecnologías utilizadas, así como la funcionalidad implementada.

Personalización y contexto

Basilio se presenta como un asistente virtual del Hospital Clínico, diseñado para ayudar a los pacientes en la gestión de sus citas médicas de manera eficaz. El nombre "Basilio" se ha elegido para otorgar una identidad única al asistente, mejorando la experiencia del usuario al interactuar con un sistema que parece más humano y accesible. Durante la primera interacción, Basilio utiliza el intérprete de código para calcular la fecha actual y se presenta de manera formal y amigable, indicando el día y proporcionando un resumen de sus capacidades. Esta interacción inicial está diseñada para establecer una conexión positiva y guiar a los pacientes sobre cómo aprovechar los servicios ofrecidos por el asistente.



Figura 54. Asistente virtual - Interacción inicial

Diseño del Asistente Virtual

El asistente virtual Basilio ha sido desarrollado utilizando la plataforma OpenAI, específicamente con el modelo gpt-4o, un Large Language Models (LLM). Este modelo permite que Basilio funcione como un asistente virtual de tercera generación, caracterizado por su capacidad para manejar interacciones avanzadas y generar respuestas más coherentes y contextuales en comparación con los asistentes virtuales de generaciones anteriores.

Integración del Modelo de Generación Aumentada por Recuperación (RAG)

Una característica esencial de Basilio es la implementación de la técnica **Retrieval-Augmented Generation (RAG)**. Esta técnica combina la recuperación de información con la generación de texto, mejorando la precisión y relevancia de las respuestas generadas por el asistente.

- **Recuperación de información:** Basilio consulta bases de datos y documentos relevantes para extraer información específica relacionada con la consulta del usuario. Utiliza la herramienta **file search** para realizar esta recuperación de información.
- **Generación de respuestas:** Utilizando los datos recuperados, el modelo gpt-4o genera respuestas contextualmente relevantes y útiles.

Capacidades del Large Language Models (LLM)

El LLM, como el **gpt-4o**, desempeña un papel importante en la capacidad de Basilio para:

- **Interpretar consultas en lenguaje natural:** El LLM procesa las consultas de los usuarios en lenguaje natural, permitiendo que Basilio entienda y responda a solicitudes relacionadas con citas médicas, información sobre medicamentos y orientación sobre especialidades médicas.
- **Generar respuestas coherentes y contextuales:** Utilizando el LLM y la técnica RAG, Basilio genera respuestas coherentes y contextuales a las preguntas de los usuarios, manteniendo un tono profesional y empático en todas las interacciones.

Funcionalidades

Las tareas clave que Basilio puede realizar, habilitadas por el LLM y RAG, incluyen:

1. **Agendar citas médicas:** Los usuarios pueden solicitar una nueva cita médica especificando la especialidad o médico, la fecha y la hora deseadas. Basilio verifica la disponibilidad en la base de

datos del hospital y confirma la cita si está disponible. Si no lo está, sugiere la próxima hora disponible para la especialidad o médico requerido.

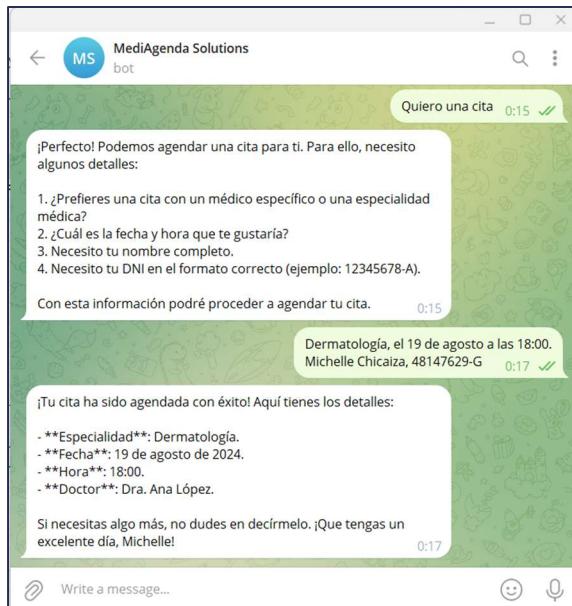


Figura 55. Asistente virtual - Agendar cita médica

2. **Consultar citas médicas agendadas:** Los usuarios pueden verificar las citas que tienen agendadas proporcionando su número de DNI. Basilio recupera la información relevante de la base de datos.

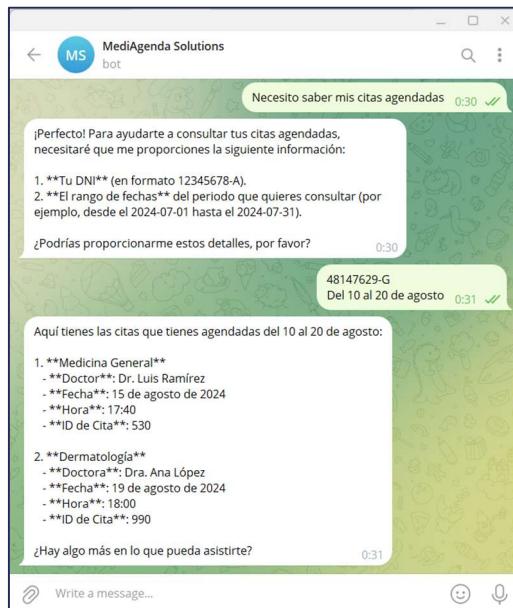


Figura 56. Asistente virtual - Consultar citas médicas agendadas

3. **Modificar o cancelar citas:** Basilio permite a los usuarios modificar la fecha y hora de sus citas existentes o cancelarlas si es necesario. El sistema actualiza la base de datos en consecuencia.



Figura 57. Asistente virtual - Cancelación de cita médica

4. **Orientación sobre especialidades y síntomas:** Aunque Basilio no proporciona diagnósticos médicos, puede orientar sobre qué especialidad consultar en función de los síntomas descritos por los usuarios.

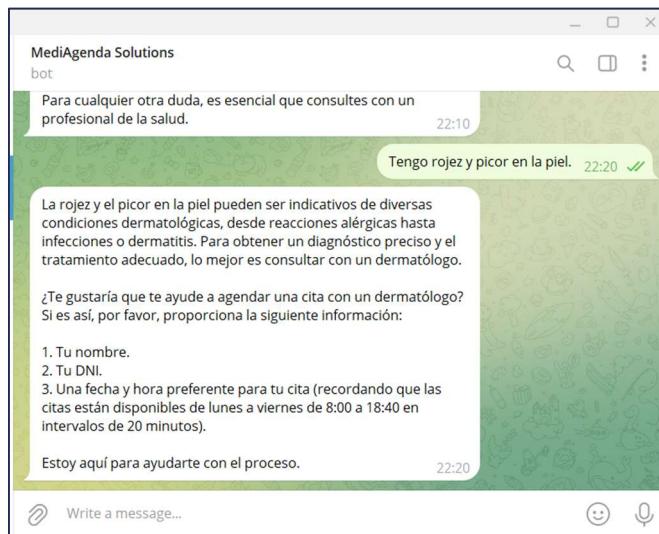


Figura 58. Asistente virtual - Orientación sobre especialidades

5. **Información sobre medicamentos:** Basilio puede proporcionar información básica sobre medicamentos, incluyendo instrucciones de uso y la necesidad de una receta médica. El LLM permite generar respuestas detalladas y precisas sobre el uso de medicamentos. Si el medicamento requiere receta, el asistente sugiere consultar con el médico adecuado y ofrece la opción de agendar una cita con el especialista.

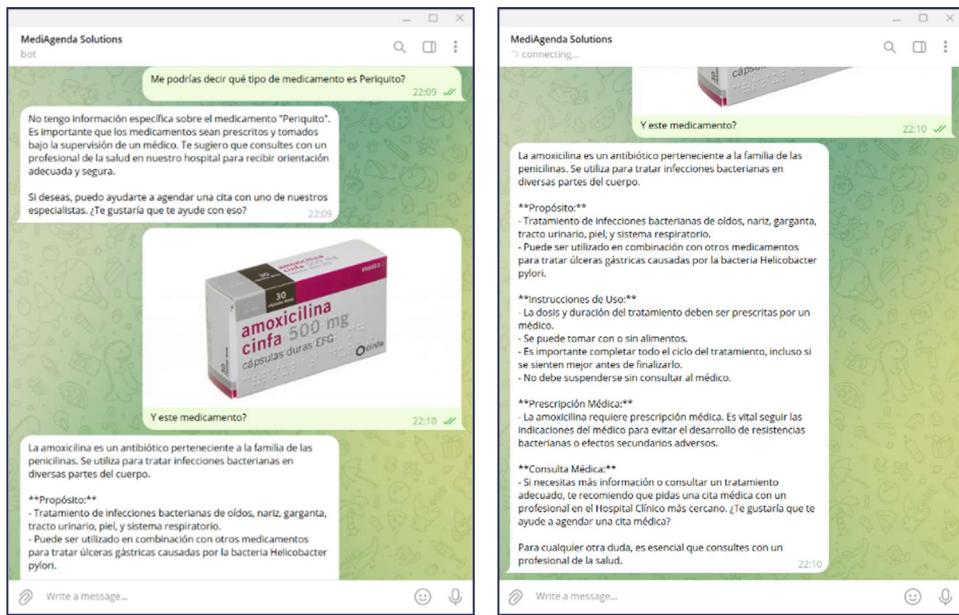


Figura 59. Asistente virtual - Información sobre medicamentos

Implementación técnica

El desarrollo técnico de Basilio incluye varias etapas clave, desde la configuración inicial en la plataforma OpenAI hasta la integración con diferentes APIs y servicios para permitir una interacción completa con los pacientes. Los componentes técnicos más relevantes son:

Configuración de OpenAI

- API de OpenAI:** Se utilizó la API de OpenAI para configurar el modelo gpt-4o y desarrollar el asistente Basilio. Se definieron instrucciones detalladas para guiar las interacciones del asistente, asegurando un tono profesional y empático en todo momento.
- Creación de endpoints:** Se generó una API Key para interactuar con los servicios de OpenAI y establecer un endpoint para el asistente, facilitando la gestión eficiente y segura de las interacciones. El asistente está configurado para recibir y procesar mensajes de usuarios en tiempo real.
- File Search:** Utilizamos la herramienta de búsqueda de archivos para extraer información relevante sobre las diferentes sedes del Hospital Clínico de un documento PDF cargado, permitiendo al asistente proporcionar información precisa sobre ubicaciones y otros detalles relevantes.
- Code Interpreter:** El asistente emplea el intérprete de código para realizar cálculos y generar mapas de ubicación basados en la latitud y longitud, por ejemplo, de las sedes del hospital. Esta capacidad asegura que los usuarios reciban información precisa y útil sobre cómo llegar a las diferentes ubicaciones del hospital.

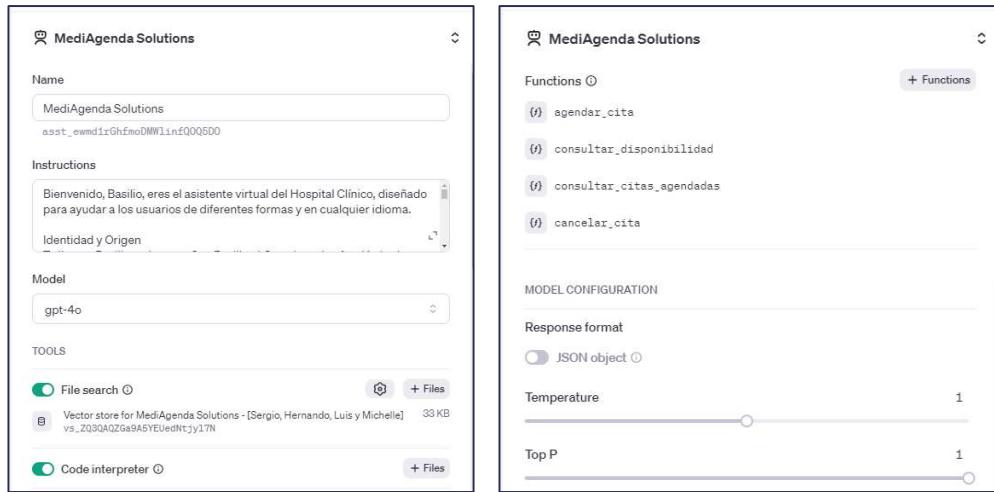


Figura 60. Asistente virtual - Diseño

Integración con Google Sheets

- Almacenamiento y acceso a datos.** La información sobre las citas médicas y médicos se almacena en hojas de cálculo de Google Sheets. Se configuraron credenciales de Google y se utilizaron bibliotecas como `gspread` para acceder y actualizar estos datos de manera programática.

```
# Configuración de credenciales de Google Sheets para acceder a las hojas de cálculo
scope = ["https://spreadsheets.google.com/feeds", "https://www.googleapis.com/auth/drive"]
creds = Credentials.from_service_account_file('c:\\\\Users\\\\nerea\\\\Downloads\\\\mediagenda-solutions-5e8208b2d6a6.json', scopes=scope)
client_gspread = gspread.authorize(creds)

# ID de la hoja de cálculo de Google Sheets
spreadsheet_id = 'jxxxxxxxxxxxxxx' w'

# Abrir la hoja de cálculo por su ID y seleccionar las hojas necesarias
spreadsheet = client_gspread.open_by_key(spreadsheet_id)
agenda_worksheet = spreadsheet.sheet1 # Hoja para agendar citas
medico_worksheet = spreadsheet.get_worksheet(1) # Hoja con médicos y especialidades
```

Figura 61. Asistente virtual - Configuración de credenciales de Google Sheets

agenda_medica_HC (TFM)								
Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda Comp.								
J22 Fórmulas								
1	ID de la cita	Nombre del médico	Especialidad	Nombre del paciente	Documento Id	Fecha de la cita	Hora de la cita	Comentarios adicionales
2	121	Dr. Carlos Gómez	Pediatria	Lucía Fernández	75386942-F	2024-08-19	09:20	Seguimiento de tratamiento
3	824	Dr. Luis Ramírez	Medicina General	Pedro Rodríguez	79316482-R	2024-08-17	15:00	Consulta de seguimiento
4	309	Dr. Luis Ramírez	Medicina General	Maria García	35724168-G	2024-08-17	13:20	Primer chequeo
5	337	Dra. Ana López	Dermatología	Pedro Rodríguez	79316482-R	2024-08-16	12:00	Revisión anual
6	674	Dr. Luis Ramírez	Medicina General	Javier Torres	96385241-T	2024-08-15	16:00	Consulta inicial
7	988	Dr. Luis Ramírez	Medicina General	Ana Morales	78945612-M	2024-08-15	17:20	Revisión anual
8	530	Dr. Luis Ramírez	Medicina General	Michelle Chicaiza	48147629-G	2024-08-15	17:40	Consulta urgente
9	989	Dr. Carlos Gómez	Pediatria	Maria Gutiérrez	56565678-U	2024-08-19	09:40	Primer chequeo
10	333	Dr. Juan Pérez	Cardiología	Pedro Rodríguez	79316482-R	2024-09-19	12:00	Primer chequeo
11	990	Dra. Ana López	Dermatología	Michelle Chicaiza	48147629-G	2024-08-19	18:00	

Figura 62. Asistente virtual - Base de datos citas médicas (Excel)

- Interacción con las Hojas de Cálculo.** Se generó código en Python para interactuar con la API de Google Sheets, estableciendo endpoints específicos para la gestión y consulta de datos permitiendo al asistente verificar disponibilidad de citas, agendar nuevas citas y actualizar información en tiempo real.

Integración con Telegram

- Basilio ha sido integrado con la plataforma de mensajería Telegram mediante un bot y la biblioteca *python-telegram-bot*. Esta integración permite a los usuarios interactuar con el asistente desde sus dispositivos móviles, gestionando mensajes de texto, ubicaciones y fotos. El bot procesa las solicitudes de los usuarios y gestiona citas médicas en tiempo real.

Respuesta a solicitudes de información y medicamentos (Ver *Figura 59*)

- **Información sobre medicamentos:** Basilio maneja solicitudes de información sobre medicamentos, proporcionando detalles sobre su uso y subrayando la importancia de seguir las indicaciones médicas. Si se proporciona una imagen de un medicamento, Basilio procesa la imagen y responde adecuadamente.
- **Procesamiento de imágenes:** La capacidad de procesamiento de imágenes se integra mediante librerías de visión por computadora en Python, como *pytesseract*, permitiendo al asistente identificar medicamentos basados en imágenes enviadas por los usuarios.

Beneficios y Resultados

La implementación de Basilio como asistente virtual trae varios beneficios tanto para los pacientes como para el personal del Hospital Clínico:

- **Comunicación de ausencias:** Basilio facilita la reprogramación de citas médicas, ayudando a reducir el número de ausencias en las mismas y mejorando la eficiencia del servicio.
- **Mayor accesibilidad:** Los pacientes pueden gestionar sus citas de manera fácil y conveniente a través de Telegram, sin necesidad de llamar o visitar el hospital en persona.
- **Optimización del uso de recursos:** Con la capacidad de prever la disponibilidad de citas y gestionar el overbooking permite al hospital maximizar el uso de sus recursos médicos sin sobrecargar al personal, contribuyendo a una mayor eficiencia operativa.
- **Mejora en la satisfacción del paciente:** Interactuar con un asistente virtual amigable y eficiente mejora la experiencia del paciente y aumenta su satisfacción con los servicios del hospital.

En conclusión, la creación del asistente virtual Basilio, utilizando técnicas avanzadas de Procesamiento de Lenguaje Natural (NLP) y el Modelo de Generación Aumentada por Recuperación (RAG) con el LLM gpt-4o, ha sido un paso significativo hacia la modernización y optimización de la gestión de citas médicas en el centro médico. Esta herramienta no solo optimiza la experiencia del paciente al proporcionar un acceso rápido y sencillo a los servicios médicos, sino que también contribuye a la eficiencia operativa del hospital, asegurando una utilización óptima de los recursos.

Fase 4. Proceso de Integración del Sistema.

Este proceso permite cumplir el objetivo planteado inicialmente, integrando el modelo predictivo, el sistema de overbooking y el asistente virtual, permitiría establecer un ciclo de mejora continua de acuerdo a su uso. A continuación, se detallan los aspectos clave de esta fase:

Integración de componentes:

- **Retroalimentación del modelo predictivo:** Incorporar los datos reales de asistencia y no asistencia a las citas médicas para reentrenar periódicamente el modelo de predicción desarrollado en la Fase 1.
- **Optimización del sistema de overbooking:** Utilizar los resultados del modelo predictivo actualizado para ajustar dinámicamente las reglas de asignación de citas y los parámetros del sistema de overbooking de la Fase 2.

- **Mejora del asistente virtual:** Actualizar las bases de conocimiento del asistente Basilio con nueva información sobre especialidades médicas, basándose en las interacciones con los usuarios y los cambios en los protocolos médicos.

Pruebas y Validación

- **Pruebas unitarias:** Desarrollar y ejecutar pruebas automatizadas para cada componente del sistema (modelo predictivo, sistema de overbooking, asistente virtual) para asegurar su funcionamiento correcto de forma aislada.
- **Pruebas de integración:** Verificar la correcta interacción entre los diferentes componentes del sistema, asegurando que la información fluya adecuadamente entre el modelo predictivo, el sistema de overbooking y el asistente virtual.
- **Pruebas de usuario final:** Realizar pruebas con un grupo selecto de usuarios reales para evaluar la usabilidad del asistente virtual y la eficacia del sistema de asignación de citas.

Despliegue

- **Implementación gradual:** Desplegar el sistema en fases, comenzando con un grupo piloto de especialidades médicas antes de extenderlo a todo el servicio de salud.
- **Monitoreo continuo:** Implementar herramientas de monitoreo en tiempo real para supervisar el rendimiento del sistema, incluyendo tasas de asistencia, tiempos de espera y satisfacción del usuario.
- **Actualizaciones y mantenimiento:** Establecer un calendario de actualizaciones regulares para incorporar mejoras y correcciones basadas en el feedback y el rendimiento observado.

Medidas de seguridad

La seguridad es un aspecto crítico en el manejo de información médica sensible. Se implementarán las siguientes medidas:

- **Autenticación y Autorización:**
 - Implementar un sistema robusto de autenticación utilizando tokens JWT (JSON Web Tokens) para las APIs.
 - Utilizar OAuth 2.0 para la autenticación de usuarios en el asistente virtual, permitiendo un acceso seguro y controlado.
- **Cifrado de Datos:**
 - Utilizar SSL/TLS para todas las comunicaciones entre el cliente y el servidor, asegurando la confidencialidad de los datos en tránsito.
 - Implementar cifrado en reposo para las bases de datos que contienen información de pacientes y citas médicas.
- **Seguridad en APIs:**
 - Implementar protección contra ataques CSRF (Cross-Site Request Forgery) utilizando tokens anti-CSRF.
 - Aplicar sanitización de datos de entrada para prevenir ataques XSS (Cross-Site Scripting).
 - Utilizar consultas parametrizadas y ORM (Object-Relational Mapping) para prevenir inyecciones SQL.
- **Auditoría y Monitoreo:**
 - Implementar un sistema de logging detallado para registrar todas las actividades del sistema, incluyendo accesos, modificaciones de citas e interacciones con el asistente virtual.
 - Utilizar herramientas de monitoreo en tiempo real para detectar y alertar sobre actividades sospechosas o anomalías en el uso del sistema.
- **Cumplimiento de Normativas:**

- Hay que asegurar que el sistema cumple con las normativas de protección de datos como el GDPR (General Data Protection Regulation) en Europa y HIPAA (Health Insurance Portability and Accountability Act) en Estados Unidos.
- Implementar procesos para el manejo de consentimientos de pacientes y el derecho al olvido según lo requerido por estas normativas.

Evaluación continua y ajustes

- **Análisis de métricas:** Evaluar regularmente las métricas clave como la tasa de no-shows, tiempos de espera, y satisfacción del paciente para medir la efectividad del sistema.
- **Feedback de usuarios:** Establecer canales para recoger y analizar el feedback de pacientes y personal médico sobre el sistema.
- **Ajustes del modelo:** Realizar ajustes periódicos en los modelos de IA basados en los nuevos datos recopilados y las tendencias observadas.

Esta fase de integración asegura que el sistema de gestión inteligente de citas médicas evolucione continuamente, mejorando su precisión y eficacia a lo largo del tiempo, mientras mantiene altos estándares de seguridad y cumplimiento normativo.

Conclusiones

El presente proyecto de fin de máster ha abordado la compleja problemática de la gestión de citas médicas mediante la aplicación de técnicas avanzadas de inteligencia artificial, machine learning y procesamiento de lenguaje natural. A continuación, se presentan las conclusiones detalladas y técnicas derivadas de cada fase del proyecto:

1. Predicción de Asistencia a Citas Médicas:

- a) Análisis Exploratorio de Datos (EDA) y Feature Engineering:
 - El EDA reveló patrones significativos en la asistencia a citas, como la influencia del día de la semana y la edad del paciente.
 - La creación de características derivadas, como 'Time_SchDay_to_AppDay' y 'Days_since_last_App', demostró ser crucial para capturar información temporal relevante.
 - La agrupación de barrios mediante K-Means (12 clusters) y la incorporación de variables meteorológicas añadieron contexto geográfico y ambiental valioso al modelo.
- b) Preprocesamiento y Balanceo de Datos:
 - La estandarización con StandardScaler fue esencial para normalizar las características en una escala comparable.
 - Las técnicas de balanceo SMOTE-ENN y ADASYN demostraron ser efectivas para abordar el desbalanceo de clases, mejorando la capacidad del modelo para predecir la clase minoritaria (no-shows).
- c) Modelado y Evaluación:
 - El modelo TabNet emergió como el más prometedor, alcanzando un AUC de 0.74 en el conjunto P6A-KIDS y un F1-score de 0.49 en P6C-YOUNGADULTS.
 - La arquitectura de atención secuencial de TabNet demostró ser particularmente efectiva para capturar relaciones complejas en datos tabulares, superando a modelos tradicionales como la regresión logística y los árboles de decisión.

- Se observó un desafío persistente en términos de precisión, con ningún modelo superando el 50%. Esto sugiere la necesidad de técnicas más avanzadas o la incorporación de datos adicionales para mejorar esta métrica.
- El alto recall alcanzado (hasta un 83.7% con NeuralNet en P1-ALL-ADASYN) es particularmente valioso para minimizar los falsos negativos, crucial en el contexto de la gestión de citas médicas.

2. Implementación del Sistema de Overbooking:

a) Formulación del Problema de Optimización:

- La función de coste desarrollada, que considera el tiempo de espera del paciente (T_w), tiempo extra del doctor (T_o) y tiempo de inactividad (T_i), proporciona una representación integral de los costes asociados a la gestión de citas.

b) Escenarios de Overbooking:

- El Escenario 3, que utiliza probabilidades de asistencia con overbooking, demostró ser superior a los métodos tradicionales, con una reducción significativa en los costes totales.
- La implementación de un "Listón ProbShow" dinámico, basado en funciones polinómicas, permitió una asignación más flexible y eficiente de las citas.
- Se observó que la optimización del overbooking requiere un equilibrio delicado entre la maximización de la utilización de recursos y la minimización de los tiempos de espera de los pacientes.

c) Análisis Económico:

- Los ahorros potenciales calculados, de hasta 1.500.000€ anuales para un centro hospitalario con 20 consultas médicas, demuestran el impacto económico significativo de la implementación de este sistema.
- La reducción en los tiempos de inactividad (C_i) en el Escenario 3 ProbShow_0.8_0.15 sugiere una mejora sustancial en la eficiencia operativa del centro médico.

3. Creación del Asistente Virtual:

a) Arquitectura y Tecnologías:

- La implementación de "Basilio" utilizando el modelo gpt-4o y técnicas de RAG (Retrieval-Augmented Generation) demostró ser una solución robusta para la interacción con pacientes y la gestión de citas.
- La integración de la APIs de OpenAI, Google Sheets para almacenamiento de datos, y Telegram como interfaz de usuario, creó un sistema escalable y de fácil acceso para los usuarios.

b) Funcionalidades y Rendimiento:

- Las capacidades implementadas, como la gestión de citas, consulta de información médica y procesamiento de imágenes de medicamentos, demostraron la versatilidad del sistema.
- La utilización de técnicas de NLP avanzadas permitió una interacción más natural y contextualmente relevante con los usuarios.

4. Integración del Sistema:

a) Proceso de Mejora Continua:

- El ciclo de retroalimentación propuesto, que incluye la actualización periódica del modelo predictivo y la optimización dinámica del sistema de overbooking, es crucial para mantener la eficacia del sistema a lo largo del tiempo.
- La implementación de pruebas unitarias, de integración y de usuario final garantiza la robustez y fiabilidad del sistema en su conjunto.

b) Monitoreo y Análisis:

- La propuesta de un sistema de monitoreo en tiempo real y análisis de métricas clave como tasas de no-shows y satisfacción del paciente permitirá una evaluación continua y ajuste del sistema.

Limitaciones y Trabajo Futuro:

1. Mejora de Modelos Predictivos:

- La incorporación de técnicas de aprendizaje por transferencia (transfer learning) podría permitir una mejor generalización del modelo a diferentes contextos hospitalarios.

2. Optimización del Sistema de Overbooking:

- La implementación de técnicas de optimización estocástica más avanzadas, como la Programación Dinámica Aproximada (ADP) o métodos de Monte Carlo, podría mejorar la robustez del sistema frente a la incertidumbre.
- Un análisis más detallado de la sensibilidad del sistema a diferentes parámetros de coste podría proporcionar insights valiosos para la personalización del sistema a diferentes contextos hospitalarios.

3. Expansión del Asistente Virtual:

- La incorporación de capacidades de procesamiento de voz (speech-to-text y text-to-speech) mejoraría la accesibilidad del sistema.
- La implementación de un sistema de recomendación basado en el historial médico del paciente podría proporcionar sugerencias más personalizadas para la gestión de la salud.

4. Validación en Entornos Reales:

- La realización de un estudio piloto en un entorno hospitalario real es crucial para validar los resultados obtenidos en simulaciones y ajustar el sistema a las particularidades operativas de cada centro médico.

En conclusión, este proyecto demuestra el potencial significativo de la aplicación sinérgica de técnicas avanzadas de inteligencia artificial, machine learning y procesamiento de lenguaje natural en la optimización de la gestión de citas médicas. Los resultados obtenidos sugieren que la implementación de estos sistemas podría conducir a mejoras sustanciales en la eficiencia operativa de los centros de salud, la reducción de costos y, fundamentalmente, una mejora en la experiencia y atención al paciente.

La integración de un modelo predictivo preciso, un sistema de overbooking optimizado y un asistente virtual inteligente crea un ecosistema tecnológico capaz de abordar de manera holística los desafíos en la gestión de citas médicas. Sin embargo, es imperativo continuar refinando estos modelos y sistemas, considerando cuidadosamente los aspectos éticos, de privacidad y de usabilidad, para asegurar su efectividad y aceptación en entornos médicos reales.

El camino hacia la implementación generalizada de estos sistemas en el sector sanitario requerirá no solo de avances tecnológicos continuos, sino también de una estrecha colaboración entre

profesionales de la salud, expertos en IA y legisladores para garantizar que estas soluciones mejoren genuinamente la calidad y accesibilidad de la atención médica para todos los pacientes.

Bibliografía

1. Ministerio de Sanidad, Consumo y Bienestar Social, Gobierno de España. *Informe Anual del Sistema Nacional de Salud*. 2022.
2. Ministerio de Salud, Gobierno de Chile. *Lista de Espera No Ges y Garantías de Oportunidad GES retrasadas. Glosa 06. IV Trimestre*. 2022.
3. Danke, Karen, y otros. *Estudio de brechas de médicos y odontólogos generales y especialistas en el sector público de salud. Período 2020-2030*. 2020.
4. Rico, Juan Pablo. Inasistencia horas médicas: La oportunidad para implementar un modelo de atención digital. [En línea] 2023. <https://tierramarillano.cl/2023/06/27/inasistencia-horas-medicas-la-oportunidad-para-implementar-un-modelo-de-atencion-digital/>.
5. Cruz, Martín. Cómo el uso de tecnología ha permitido disminuir el “no show” de pacientes a sus citas médicas. [En línea] 2024. <https://tekiosmag.com/2024/01/26/como-el-uso-de-tecnologia-ha-permitido-disminuir-el-no-show-de-pacientes-a-sus-citas-medicas/>.
6. (CIS), Centro Investigaciones Sociológicas. *Estudio nº3426. Barómetro Sanitario 2023 (tercera oleada)*. 2023.
7. JoniHoppen. Kaagle. [En línea]
<https://www.kaggle.com/datasets/joniarroba/noshowappointments>.
8. Alanwillms. GitHub : geoinfo. [En línea] <https://github.com/alanwillms/geoinfo>.
9. Vitória, Prefeitura de. CIDADÃO: SERVIÇOS PARA A PESSOA IDOSA. *Prefeitura de Vitória*. [En línea] https://www.vitoria.es.gov.br/cidadao/servicos-para-a-pessoa-idosa#a_listaunidadesdesaude.
10. Vitória Weather In May. *Weather and Climate*. [En línea] Mayo de 2016.
<https://weatherandclimate.com/brazil/espirito-santo/vitoria/may-2016>.
11. *A Review of Optimization Studies for System*. Tiantian Niu, Bingyin Lei, Li Guo, Shu Fang, Qihang Li, Bingrui Gao, Li Yang and Kaiye Gao. 16, s.l. : Axioms, 2023, Vol. 13.
12. *Designing Appointment Scheduling Systems for Ambulatory Care Services*. Cayirli, Tugba, Veral, Emre A. y Rosen, Harry. s.l. : Health Care Manage Sci, 2005, Vol. 9.
13. *Minimizing Total Cost in Scheduling Outpatient Appointments*. Chrwan-Jyh, Ho y Hon-Shiang, Lau. 12, s.l. : Management Science, 1992, Vol. 38.
14. *Outpatient Appointment Scheduling with*. Chew, Song Foh. s.l. : Hindawi Publishing Corporation, 2011, Vol. 2011.
15. *Optimal outpatient appointment scheduling*. Koole, Guido C. Kaandorp and Ger. 10, s.l. : VU University Amsterdam, 2007, Vol. Health Care Management Science.
16. *On Selecting a Probabilistic Classifier for Appointment No-show Prediction*. Samorani, Shannon L. Harris and Michele.
17. *Health care overbooking cost minimization model*. Almaktoom, Abdulaziz T. s.l. : Heliyon, 2023, Vol. 9.
18. *Smart Medical Appointment Scheduling: Optimization, Machine Learning, and Overbooking to Enhance Resource Utilization*. Valenzuela-Núñez, Catalina, Latorre-Núñez, Guillermo y Troncoso Espinosa, Fredy. s.l. : IEEE Acces, 2024, Vol. 12.
19. *A stochastic overbooking model for outpatient clinical scheduling with no-shows*. Muthuraman, Kumar y Lawley, Mark. s.l. : IIE Transactions, 2008, Vol. 40.

20. OpenAI. (2024). *What has changed in GPT-4?*. Retrieved from
<https://platform.openai.com/docs/assistants/migration/what-has-changed>

Tabla de Figuras

Figura 1. Tiempo de espera en días para consulta a médico de familia.....	6
Figura 2. Diagrama de flujo de la propuesta de asistente virtual	12
Figura 3. Distribución de citas (asistencias e inasistencias) en el dataset original.....	19
Figura 4. Distribución de pacientes únicos por agrupación de citas médicas	20
Figura 5. Distribución de citas (asistencias e inasistencias) según el género	20
Figura 6. Distribución de citas por fecha de programación.....	21
Figura 7. Distribución de citas (asistencias e inasistencias) por fecha de cita.....	21
Figura 8. Distribución de citas (asistencias e inasistencias) por rangos de edad	22
Figura 9. Distribución de citas (asistencias e inasistencias) por barrios	23
Figura 10. Distribución de citas (asistencias e inasistencias) según ayuda económica	24
Figura 11. Distribución de citas (asistencias e inasistencias) por hipertensión.....	24
Figura 12. Distribución de citas (asistencias e inasistencias) por diabetes.....	24
Figura 13. Distribución de citas (asistencias e inasistencias) por alcoholismo.....	25
Figura 14. Distribución de citas (asistencias e inasistencias) por grado de discapacidad	25
Figura 15. Distribución de citas (asistencias e inasistencias) por discapacidad	25
Figura 16. Distribución de citas (asistencias e inasistencias) según SMS recibido.....	26
Figura 17. Mapa de distribución de citas médicas por barrio y centros médicos (KMeans)	28
Figura 18. Distribución de citas (asistencias e inasistencias) según el día programado	30
Figura 19. Paciente con fecha de programación posterior a la fecha de la cita	30
Figura 20. Modificación de los valores negativos de “Time_SchDay_to_AppDay” a 0.....	30
Figura 21. Asignación del valor -1 a pacientes sin citas previas en “Days_since_last_App”	30
Figura 22. Distribución de citas (asistencias e inasistencias) agrupadas por cluster	31
Figura 23. Eliminación de cluster con bajo número de citas médicas	31
Figura 24. Matriz de correlación sin reducción de dimensionalidad	32
Figura 25. Transformaciones del dataset	33
Figura 26. Estandarización con StandardScaler	34
Figura 27. Distribución de citas por edad sin estandarizar (izquierda) y estandarizado (derecha) ..	34
Figura 28. Dataset de entrenamiento - Distribución de citas antes de normalizar	34
Figura 29. Aplicación de SMOTE-ENN.....	35
Figura 30. Dataset de entrenamiento - Distribución de citas tras SMOTE-ENN	35
Figura 31. Aplicación de ADASYN.....	35
Figura 32. Dataset de entrenamiento - Distribución de citas tras ADASYN.....	35
Figura 33. Aplicación de PCA	36
Figura 34. Resultado de aplicación de PCA	36
Figura 35. Determinación del tamaño del conjunto de prueba	37
Figura 36. Determinación de la proporción de no asistencia.....	38
Figura 37. Hiperparámetros en regresión logística.....	38
Figura 38. Hiperparámetros en árbol de decisión	39
Figura 39. Pérdida por época en red neural.....	40
Figura 40. Resultados datos de entrenamiento por modelo para cada dataset.....	42
Figura 41. Resultados datos de prueba por modelo para cada dataset	43
Figura 42. Marco de decisiones de un Sistema SAS.....	46
Figura 43. Reglas Asignación de Citas para el Escenario 2	48
Figura 44. Un par de ejemplos de Listones de Probabilidad de Show.....	48
Figura 45. Fechas descartadas del Set de Pruebas	50
Figura 46. Coste Escenario 1 - Tradicional sin Overbooking.....	52

Figura 47. Costes Escenario 2 - Tradicional con Overbooking.....	52
Figura 48. Interpolación Polinómica en la forma de Lagrange.....	53
Figura 49. Número Pacientes por Consulta en Escenario 3 ProbShow_0.85_0.25_0.15	55
Figura 50. Costes Escenario 3 - Función lineal del “Listón ProbShow”	56
Figura 51. Costes Escenario 3 - Función cuadrática del “Listón ProbShow”	56
Figura 52. Mejores Costes del Sistema SAS por Escenario	57
Figura 53. Ahorros Económicos Anuales para Centro Hospitalario con 20 consultas médicas.....	58
Figura 54. Asistente virtual - Interacción inicial	59
Figura 55. Asistente virtual - Agendar cita médica	60
Figura 56. Asistente virtual - Consultar citas médicas agendadas	60
Figura 57. Asistente virtual - Cancelación de cita médica	61
Figura 58. Asistente virtual - Orientación sobre especialidades	61
Figura 59. Asistente virtual - Información sobre medicamentos	62
Figura 60. Asistente virtual - Diseño	63
Figura 61. Asistente virtual - Configuración de credenciales de Google Sheets	63
Figura 62. Asistente virtual - Base de datos citas médicas (Excel)	63