



MediAgenda Solutions

IA EN LA GESTIÓN DE CITAS MÉDICAS:
INNOVANDO EN OPTIMIZACIÓN DE AGENDAS
MÉDICAS Y ATENCIÓN AL CLIENTE

**TFM - Máster en IA para la
Transformación de Negocios**

Grupo 3
Hernando Acevedo Aguilar
Michelle Alexandra Chicaiza Anrrango
Luis Marcelo Ortiz Carinzo
Sergio Valdueza Lozano

17 Octubre 2024

Contenido

- 1. INTRODUCCIÓN.**
- 2. OBJETIVOS y ESTRUCTURA del PROYECTO.**
 - **FASE 1. Predicción Asistencia a citas médicas**
 - **FASE 2. Optimización Agendas Médicas con Overbooking.**
 - **FASE 3. Asistente Virtual.**
- 3. CONCLUSIONES.**
- 4. DESAFÍOS FUTUROS.**
- 5. DEMO.**





Introducción

Problema de asignación - cumplimiento de citas médicas

Problema

Tiempos de espera para una cita médica.

Causa identificada

No Show.

Consecuencias

Consultas desocupadas.
Aumento de costos.

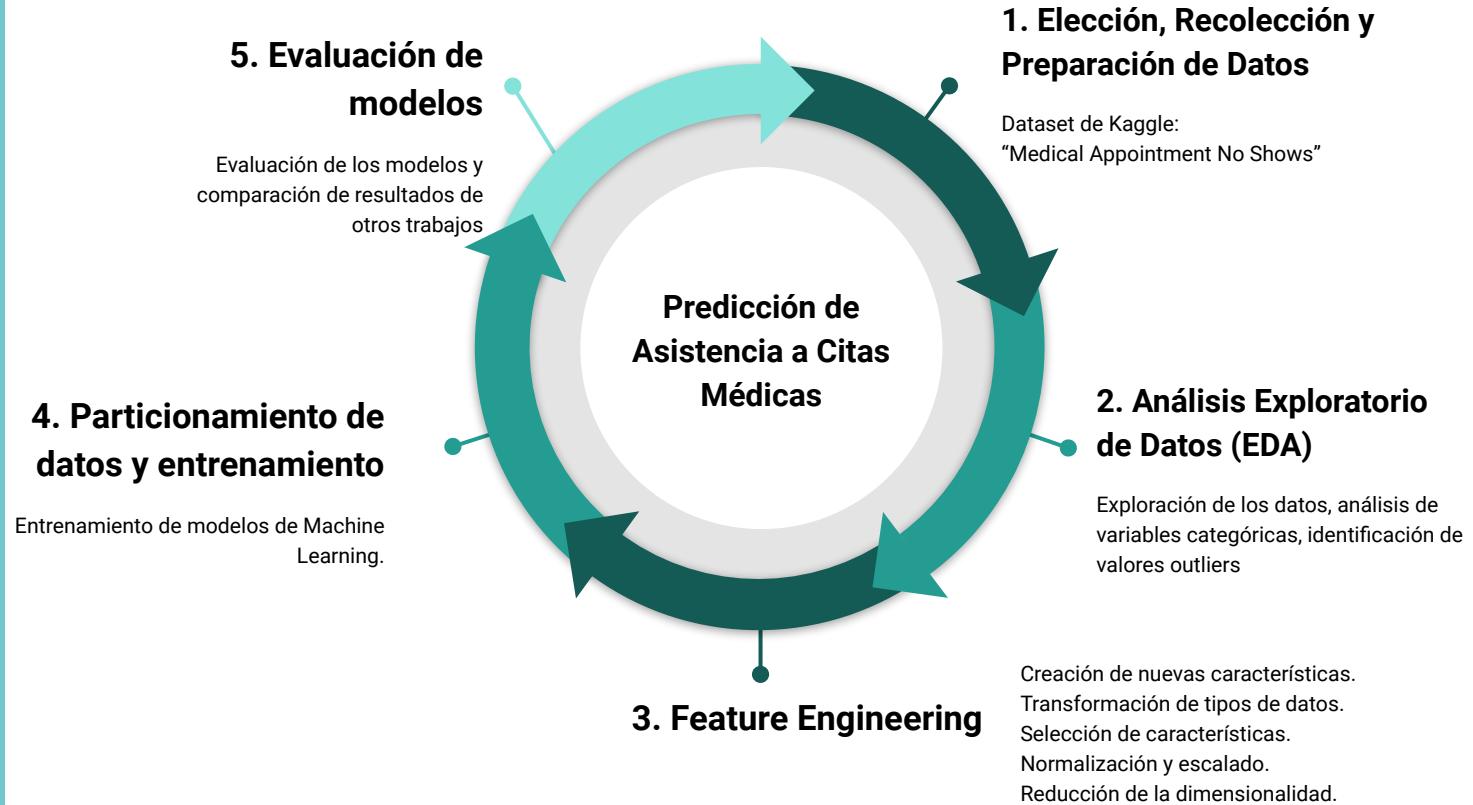


Objetivo del proyecto

Mejorar la eficiencia y reducir los costos asociados a la gestión de citas médicas mediante el uso de técnicas de Inteligencia Artificial.



Fase 1. Predicción de Asistencia a Citas Médicas





Fase 1. Predicción de Asistencia a Citas Médicas



1. Elección, Recolección y Preparación de Datos

Dataset de Kaggle:
"Medical Appointment No Shows"

Fase 1. Predicción de Asistencia a Citas Médicas

1. Elección, Recolección y Preparación de Datos

Se identifican datos relevantes:

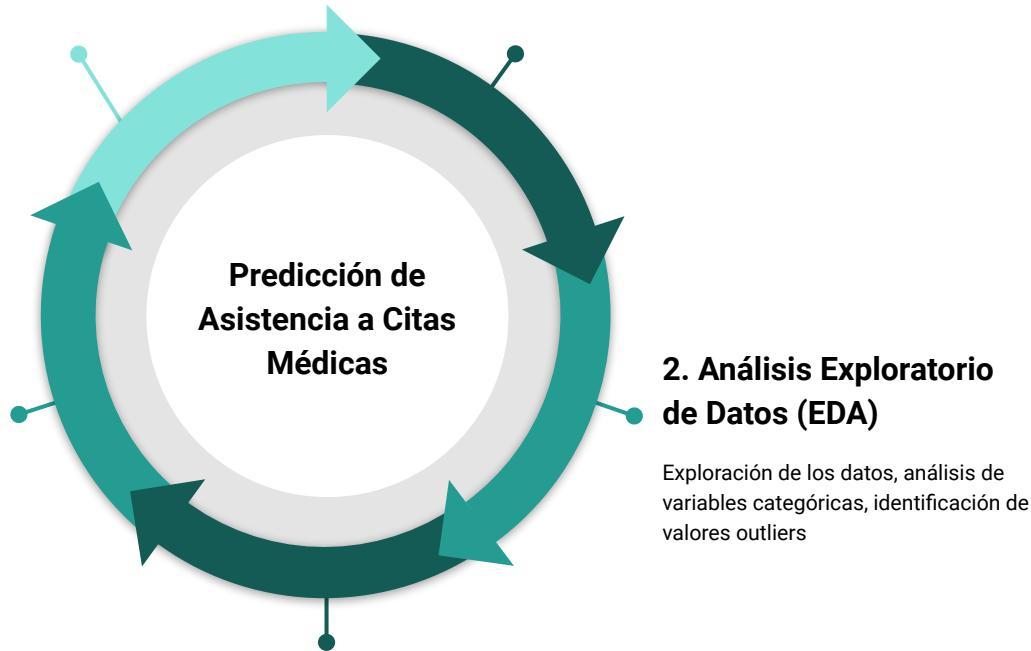
- Conjunto de datos total: **110.527**
- Pacientes distintos: **62.299**
- Variable target: “No Show”:
 - **20.19% inasistencia**

DATASET ORIGINAL		
1	PatientId	float64
2	AppointmentID	int64
3	Gender	object
4	ScheduledDay	object
5	AppointmentDay	object
6	Age	int64
7	Neighbourhood	object
8	Scholarship	int64
9	Hypertension	int64
10	Diabetes	int64
11	Alcoholism	int64
12	Handicap	int64
13	SMS_received	int64
14	NoShow	object





Fase 1. Predicción de Asistencia a Citas Médicas



Fase 1. Predicción de Asistencia a Citas Médicas

2. Análisis Exploratorio de Datos (EDA)

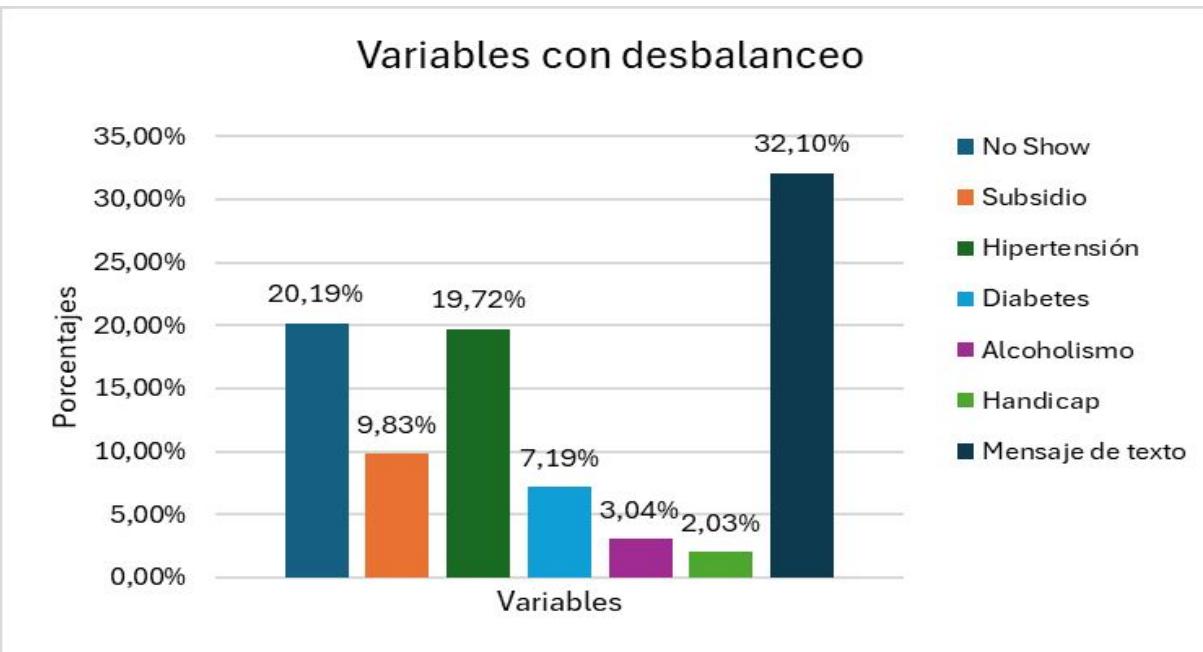
Identificación de datos relevantes:

- 60.87% de los pacientes cita única
- 65% mujeres
- Solicitud del 10 de noviembre 2015 al 08 de junio 2016
- Citas asignadas del 29 de abril al 08 de junio de 2016
- Edades desde 1 a 115 años
- Los pacientes entre 5 y 30 años tuvieron mayor tasa de inasistencia
- 81 barrios distintos



Fase 1. Predicción de Asistencia a Citas Médicas

2. Análisis Exploratorio de Datos (EDA)





Fase 1. Predicción de Asistencia a Citas Médicas



3. Feature Engineering

Creación de nuevas características.
Transformación de tipos de datos.
Selección de características.
Normalización y escalado.
Reducción de la dimensionalidad.

Fase 1. Predicción de Asistencia a Citas Médicas



3. Feature Engineering

DATASET ORIGINAL	
1	PatientId float64
2	AppointmentID int64
3	Gender object
4	ScheduledDay object
5	AppointmentDay object
6	Age int64
7	Neighbourhood object
8	Scholarship int64
9	Hypertension int64
10	Diabetes int64
11	Alcoholism int64
12	Handicap int64
13	SMS_received int64
14	NoShow object

DATASET TRATADO	
1	PatientId int64
2	AppointmentID int64
3	Gender int32
4	ScheduledDay datetime64[ns]
5	AppointmentDay datetime64[ns]
6	App_DayOfWeek int32
7	Time_SchDay_to_AppDay int64
8	Days_since_last_App int64
9	Age int64
10	Neighbourhood int32
11	Neigh_Cluster int64
12	Health_Centre int64
13	Temperature float64
14	WindSpeed float64
15	Precipitation float64
16	Scholarship int64
17	Hypertension int64
18	Diabetes int64
19	Alcoholism int64
20	Handicap int64
21	Presence_Health_Conds int64
22	SMS_received int64
23	Prior_Apps_byPatient int64
24	Prior_NoShows_byPatient int64
25	Prob_NoShow_byPatient float64
26	NoShow bool

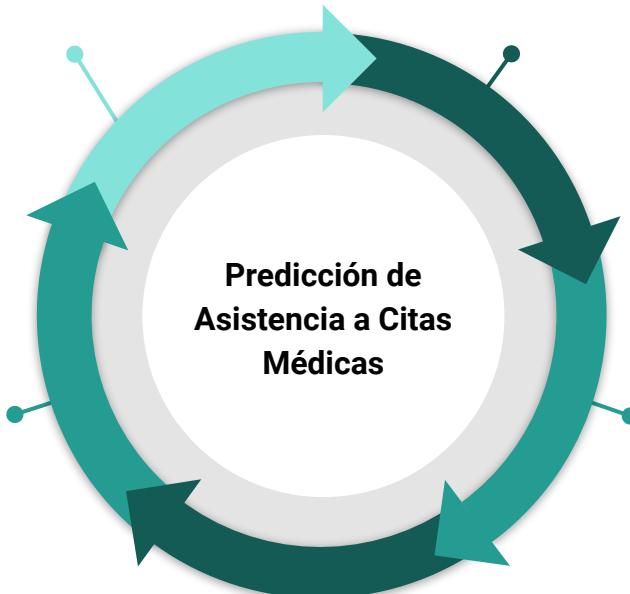
DATASET FINAL	
1	PatientId int64
2	AppointmentID int64
3	Gender int32
4	ScheduledDay datetime64[ns]
5	AppointmentDay datetime64[ns]
6	App_DayOfWeek int32
7	Time_SchDay_to_AppDay int64
8	Days_since_last_App int64
9	Age int64
10	Neighbourhood int32
11	Neigh_Cluster int64
12	Health_Centre int64
13	Temperature float64
14	WindSpeed float64
15	Precipitation float64
16	Scholarship int64
17	Hypertension int64
18	Diabetes int64
19	Alcoholism int64
20	Handicap int64
21	Presence_Health_Conds int64
22	SMS_received int64
23	Prior_Apps_byPatient int64
24	Prior_NoShows_byPatient int64
25	Prob_NoShow_byPatient float64
26	NoShow bool



Fase 1. Predicción de Asistencia a Citas Médicas

4. Particionamiento de datos y entrenamiento

Entrenamiento de modelos de Machine Learning.



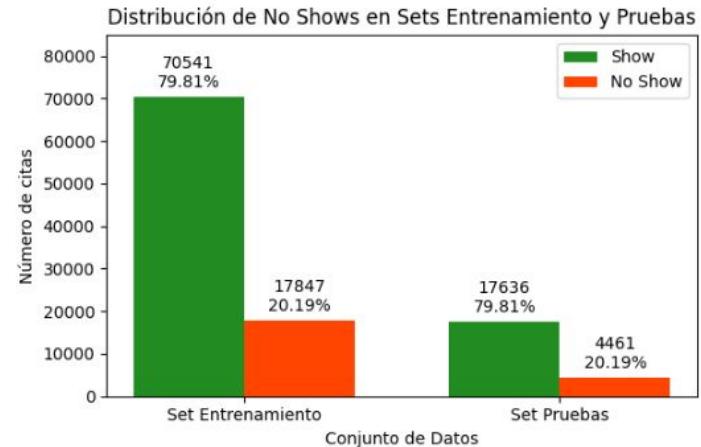
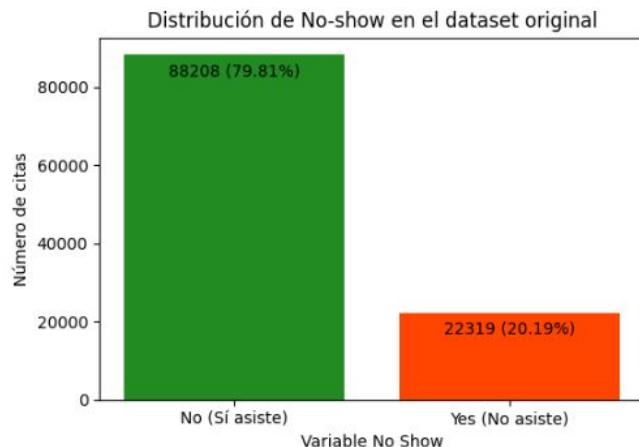
Fase 1. Predicción de Asistencia a Citas Médicas



Particionamiento de datos

Conjunto de pruebas del 20% de los datos

Misma estratificación

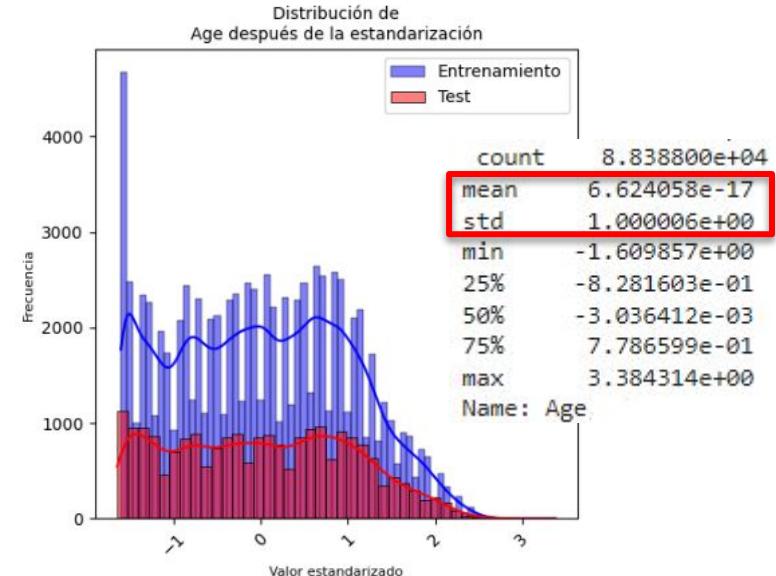
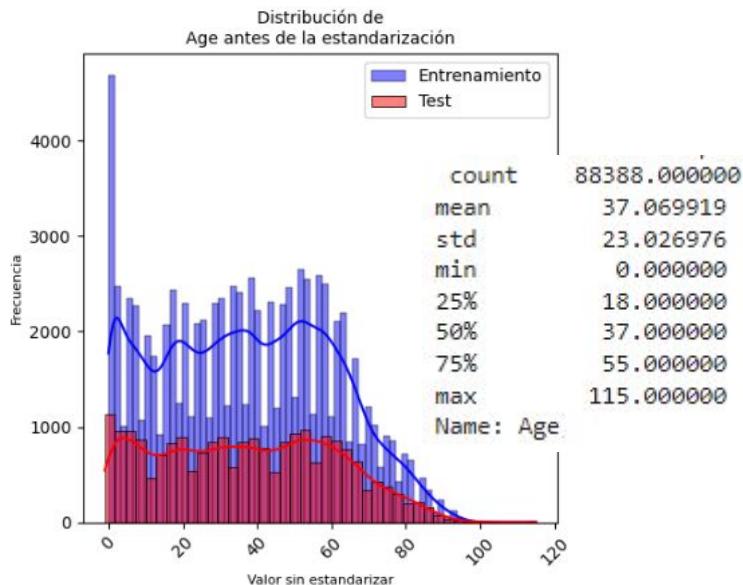


Fase 1. Predicción de Asistencia a Citas Médicas

Estandarización de variables predictivas

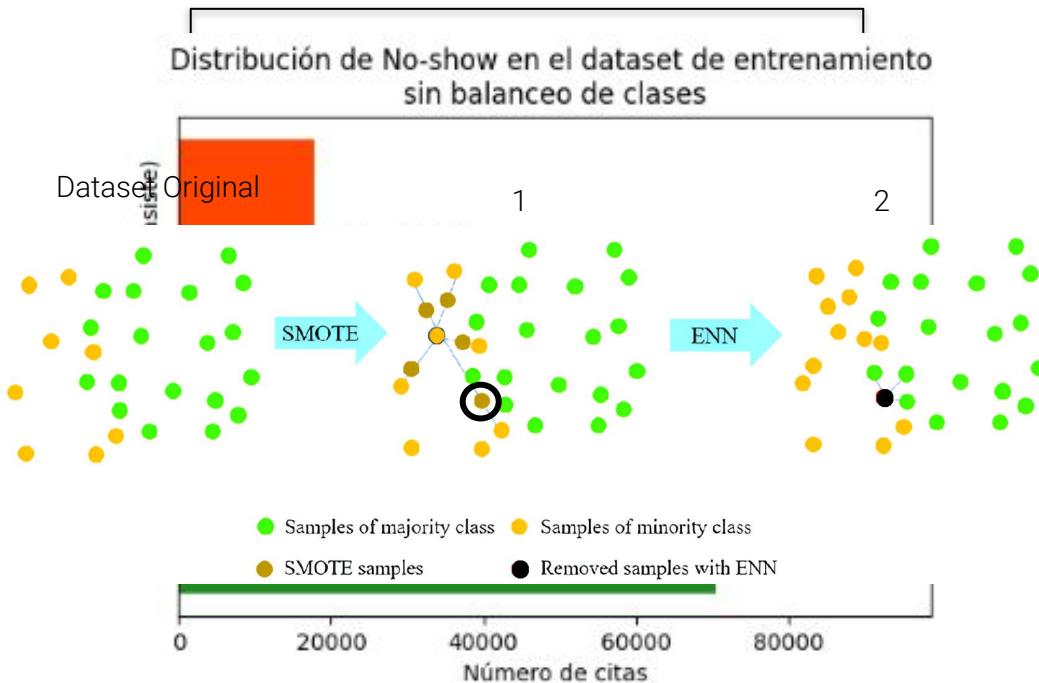


Método StandardScaler



Fase 1. Predicción de Asistencia a Citas Médicas

Técnicas de balanceo de clases - Dataset de entrenamiento



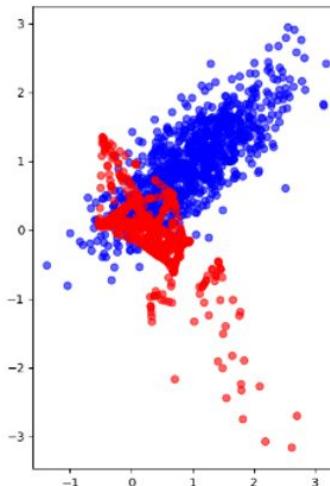
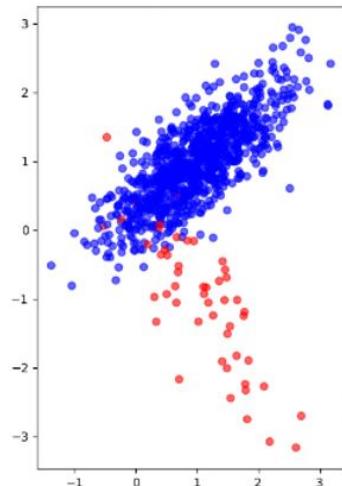
Fase 1. Predicción de Asistencia a Citas Médicas

Técnicas de balanceo de clases - Dataset de entrenamiento



Método ADASYN

Dataset Original

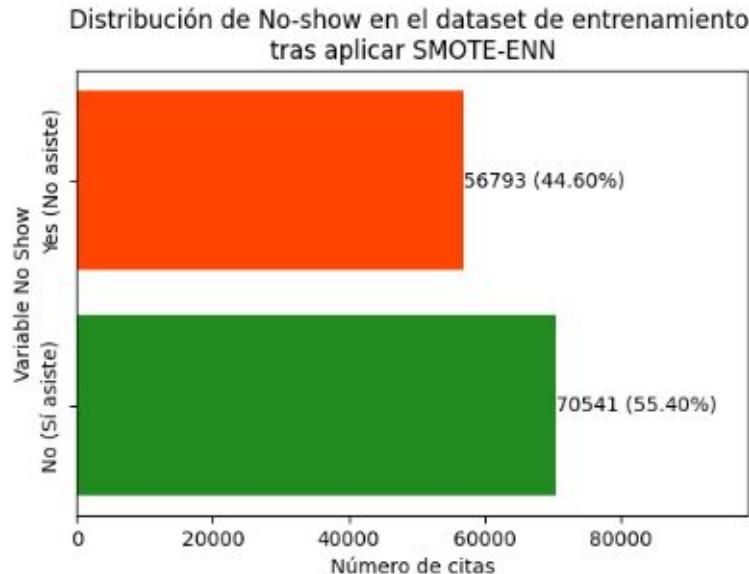


Fase 1. Predicción de Asistencia a Citas Médicas

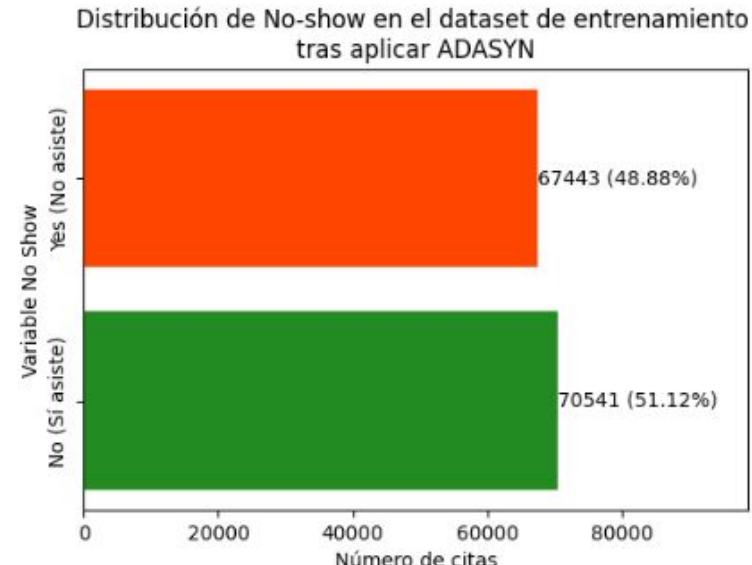
Técnicas de balanceo de clases - Dataset de entrenamiento



Dataset 1



Dataset 2

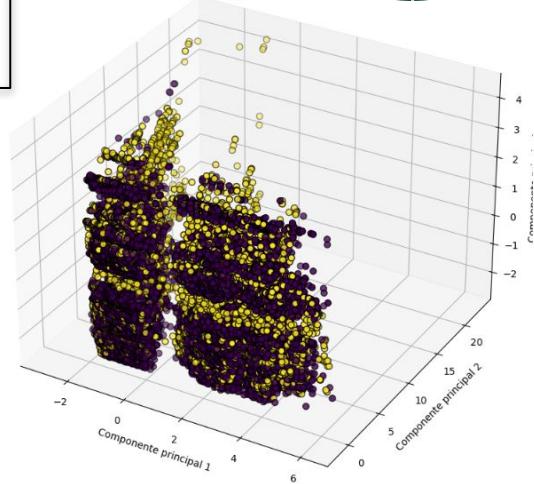


Fase 1. Predicción de Asistencia a Citas Médicas

Técnica de reducción de la dimensionalidad



- Objetivo**
 - Simplificar el conjunto de datos y mejorar la eficiencia de los modelos.
 - Conservar al menos el 95% de la varianza total.
- Resultados**
 - Variables originales: 19
 - Componentes principales seleccionados: 16
 - Varianza explicada: 96.66%
- Decisión**
 - Mantenimiento de todas las variables originales.
 - La eliminación de 3 variables no justificaba la posible pérdida de calidad en las predicciones.



Número de componentes: 16
Varianza explicada por cada componente: [0.13707614 0.12212461 0.09235221 0.0689051
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



Fase 1. Predicción de Asistencia a Citas Médicas

Dataset para modelos predictivos de inasistencia



Fase 1. Entrenamiento de modelos

El **objetivo** fue obtener un modelo capaz de predecir la **probabilidad de inasistencia** de los pacientes.

Para **cada conjunto de datos** se entrena los diferentes modelos y se utilizan diferentes métricas de evaluación: Accuracy, Precision, Recall, F1-Score y **AUC**



Se entregaron diferentes modelos de **Machine Learning**, incluyendo:

- Regresión Logística
- Árbol de Decisión

Y **Redes Neuronales de Deep Learning**:

- NeuralNet
- TabNet

Fase 1. Modelos de Machine Learning

Se entranan modelos utilizando la librería de Sklearn de Python.

1. Para cada modelo de Sklearn, se entranan tres instancias de **GridSearchCV**.
 - a. Accuracy, Precision y Recall

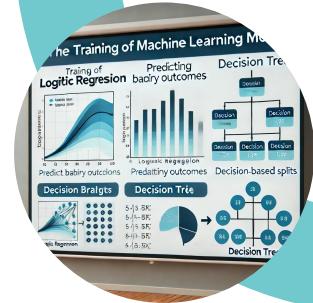
```
# Se define el rango de hiperparámetros para la búsqueda en cuadricula
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
    'max_iter': [100, 200, 300, 500, 1000]
}
```

Hiperparámetros en regresión logística

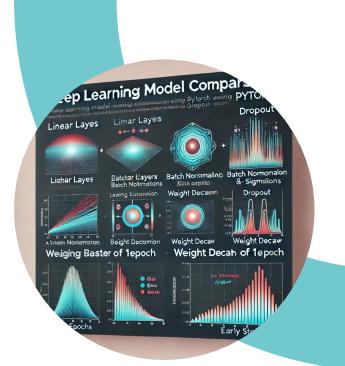
```
# Se define el rango de hiperparámetros para la búsqueda en cuadricula
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [10, 15, 20],
    'min_samples_split': [2, 3, 5],
    'min_samples_leaf': [2, 3, 4]
}
```

Hiperparámetros en árbol de decisión

2. Se realiza validación Cruzada **K-Fold Cross-Validation** con **5 particiones** y se barajaron los datos.
3. Se evaluó el rendimiento en un **conjunto de validación**.
4. El modelo con la mejor puntuación de validación se guarda y utiliza en el **conjunto de prueba**.



Fase 1. Modelos de Deep Learning



Se utiliza la librería PyTorch de Python para entrenar los modelos en local mediante GPU.

Se entrena dos modelos distintos de red neuronal profunda NeuralNet y TabNet.

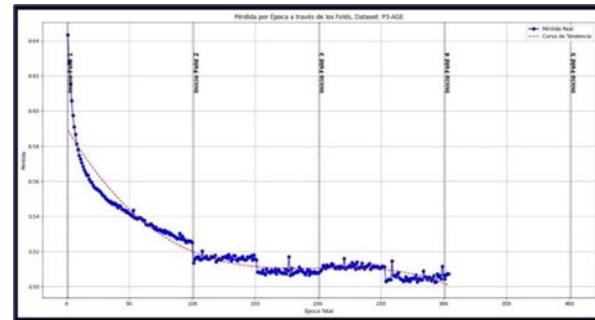
Para la red neuronal de NeuralNet se define:

- 3 Capas lineales
- 2 capas de Batch Normalization
- 2 Funciones de activación ReLU
- 1 Capa de Dropout
- 1 Función de activación Sigmoid

Validación con StratifiedKFold con 5 particiones

Y parámetros de entrenamiento equivalentes

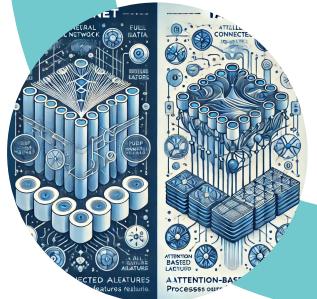
- Epoch = 100
- Optimizador (optim.Adam) con lr=1e-4 y weight_decay=1e-5, Scheduler
- Early Stopping



Perdida por época en NeuralNet

Fase 1. NeuralNet y TabNet

Arquitecturas de redes neuronales adaptadas a Datos Tabulares.



NeuralNet

Modelo tradicional de red neuronal profunda (DNN).

Procesa todas las características de entrada al mismo tiempo.

Utiliza capas completamente conectadas (lineales)

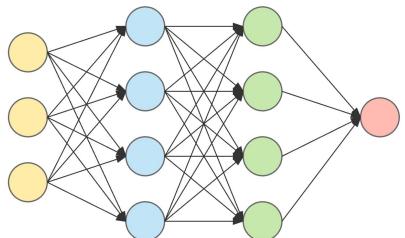


Diagrama NeuralNet

TabNet

Modelo inspirado en transformers, especialmente diseñado para datos tabulares.

Emplea un enfoque de atención secuencial, que le permite a la red centrarse en las características más relevantes de los datos en cada paso del proceso.

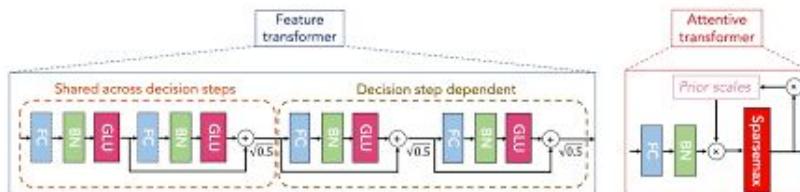


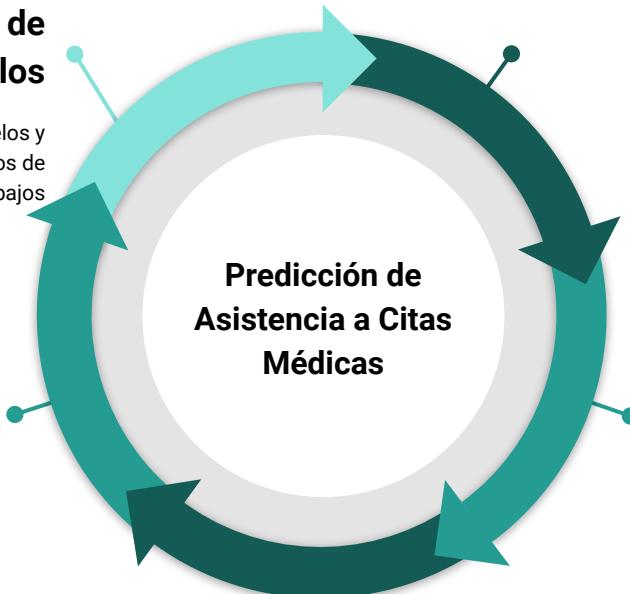
Diagrama TabNet



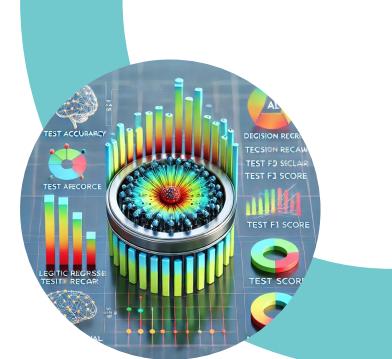
Fase 1. Predicción de Asistencia a Citas Médicas

5. Evaluación de modelos

Evaluación de los modelos y comparación de resultados de otros trabajos



Fase 1. Resultados modelos



Se entrenaron un total de 8 modelos para cada conjunto de datos, contabilizando un total de **88 modelos**

Fortalezas:

- Rendimiento superior al azar en todos los conjuntos de datos
 - Todos los modelos logran AUC > 0,5
 - Consistencia entre los resultados obtenidos en los distintos conjuntos
 - Mejor resultado en el dataset correspondiente a la totalidad del conjunto de datos con ADASYN.

Limitaciones:

- Precisión < 50% en todos los modelos

Destaca:

- TabNet alcanzó el mejor AUC: 0.73

Dataset	Model	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test AUC
Dataset completo con ADASYN	Logistic Regression (Accuracy)	64,57%	30,70%	60,03%	0,41	0,66
	Decision Tree (Recall)	61,40%	30,50%	71,33%	0,43	0,72
	Neural Network (NeuralNet)	53,57%	28,15%	83,70%	0,42	0,69
	Neural Network (TabNet)	58,90%	30,19%	78,95%	0,44	0,73

Resultados datos de Test destacados

Fase 1. Análisis comparativo de resultados

Se compara los resultados obtenidos con **Alshaya et al. 2019**, que realiza un análisis de la misma base de datos de Kaggle.

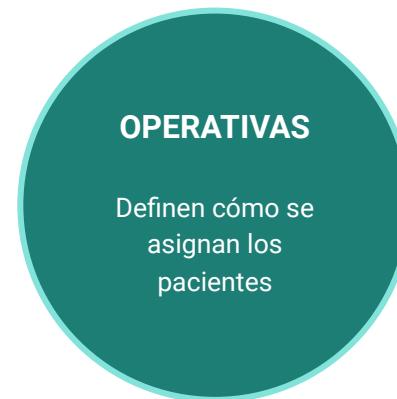
Modelo	Técnica de Balanceo	AUC ROC	F1 Score
NeuralNet (TFM)	ADASYN	0,69	0,42
Decision Tree (Recall)	ADASYN	0,72	0,43
TabNet (TFM)	ADASYN	0,73	0,44
Random Forest (Alshaya et al., 2019)	SMOTEENN	0,65	0,36
Support Vector Classifier (Alshaya et al., 2019)	ADASYN, SMOTEENN, AIKNN, RandomUnderSampler	0,68	0,44
Stochastic Gradient Descent	SMOTE, SMOTEENN, SMOTETomek	0,68	0,44
Logistic Regression (Alshaya et al., 2019)	ADASYN, SMOTEENN, RandomUnderSampler	0,71	0,43

Resultados TFM



Fase 2. Optimización Agendas Médicas

Sistemas de Programación de Citas



+ Factores Ambientales

=

Incertidumbre y Variabilidad



Fase 2. Optimización Agendas Médicas



Escenarios de Estudio

Bloque mañana (9:00 a 13:00)

Descanso (2h)

Bloque tarde (15:00 a 19:00)

Escenario 1 (tradicional sin Overbooking):



Escenario 2 (tradicional con Overbooking):



Slots o Consultas Médicas de 20 min

1 Paciente

2 Pacientes

3 Pacientes

Slots disponibles

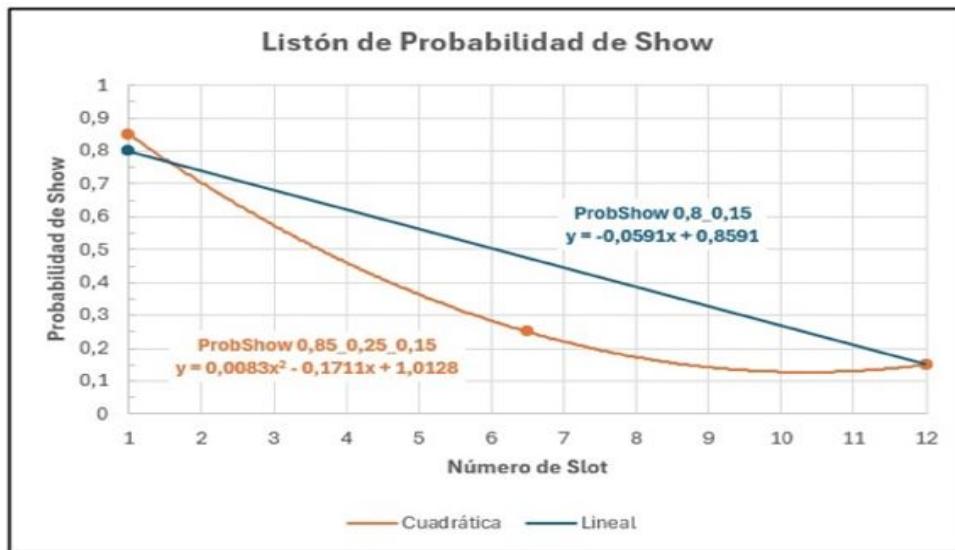
Slots Prohibidos

Fase 2. Optimización Agendas Médicas

Escenarios de Estudio

Escenario 3 (con Overbooking) - Listones de Probabilidad de Asistencia

- Definidos con función lineal (24 casos).
- Definidos con función cuadrática (64 casos).



Fase 2. Optimización Agendas Médicas



Función del Coste a Minimizar

- T_I = Tiempo de inactividad en el centro médico (Idle Cost).
- T_W = Tiempo de espera del paciente (Wait Cost).

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

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

- T_O = Tiempo extra a realizar por el doctor (Overtime Cost).

$$\begin{aligned} \text{Coste Total} = & T_I(x) * 16 \text{ €/min} * 20\text{min} + \\ & + \sum_{i=1}^N T_{w_i}(y) * 0,5 \text{ €/min} * 20\text{min} + \\ & + T_O(z) * 3,5 \text{ €/min} * 20\text{min} \end{aligned}$$

$T_I(x) = x$ = 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.

Fase 2. Optimización Agendas Médicas

Resultados (I):



COSTES EN CITAS MÉDICAS POR NO ASISTENCIA y OVERBOOKING

ESCENARIOS	Media nº slots afectados para sobrecoste		C_i : Coste Inactividad	C_w : Coste de Espera	C_o : Coste Tiempo Extra	COSTE TOTAL
	Inactividad	Tiempo Extra				

Escenario 1 (RBA sin Overbooking):

Regla R11_24 pacientes	3,83	0,00	1.224,20	0,00	0,00	1.224,20
------------------------	------	------	----------	------	------	----------

Escenario 2 (RBA con Overbooking):

Regla R1241_28 pacientes	2,19	1,21	701,69	189,45	84,52	975,67
--------------------------	------	------	--------	--------	-------	--------

Escenario 3 (OBA con Predicción de Asistencia):

Probabilidad Show 0,9_0,15	1,75	0,92	560,41	304,08	64,30	928,79
Probabilidad Show 0,9_0,2	1,62	0,96	518,50	331,95	67,32	917,77
Probabilidad Show 0,9_0,5_0,25	1,95	0,68	622,96	258,96	47,47	929,40
Probabilidad Show 0,9_0,55_0,25	1,70	0,94	544,21	319,83	65,88	929,92

Fase 2. Optimización Agendas Médicas



Resultados (II):

AHORROS PURAMENTE ECONÓMICOS

ESCENARIOS	$C_l^{(*)} + C_o$	Dif. con R11_24	Ahorros Económicos
Escenario 1 (RBA sin Overbooking):			
Regla R11_24 pacientes	841,64	0 €	0 €
Escenario 2 (RBA con Overbooking):			
Regla R1241_28 pacientes	566,94	275 €	1.357.033 €
Escenario 3 (OBA con Predicción de Asistencia):			
Probabilidad Show 0,9_0,15	449,58	392 €	1.936.785 €
Probabilidad Show 0,9_0,2	423,79	418 €	2.064.171 €
Probabilidad Show 0,9_0,5_0,25	475,76	366 €	1.807.448 €
Probabilidad Show 0,9_0,55_0,25	440,02	402 €	1.983.983 €

20 consultas - 265 días laborables

Conclusiones:

Éxito del Escenario 3

(uso de Probabilidades de Asistencia)

- **Reducción Inactividad** (Objetivo cumplido!)
 - Más Servicios:
 - > 55% respecto RBA sin Overbooking
 - > 25% respecto RBA con Overbooking
- **Mejora Económica**
 - > 2mill € respecto RBA sin Overbooking
 - > 700mil € respecto RBA con Overbooking

¡ MUY SENSIBLE A LA PERSONALIZACIÓN DE CADA CENTRO MÉDICO !

Fase 3. Asistente virtual

Funcionalidades



1. Gestión de citas médicas

- Solicitar, cancelar, modificar, consultar citas médicas

2. Orientación sobre especialidades o síntomas

3. Información breve sobre medicamentos

- Imagen o texto

4. Accesibilidad multicanal

- Integración plataformas de mensajería (ej. Telegram) para facilitar la gestión de citas médicas sin necesidad de llamadas o visitas presenciales.

5. Respuestas coherentes y contextuales

- Capacidad de generar respuestas que mantengan el contexto de la conversación.
- Evitar respuestas desconectadas o confusas, asegurando una interacción fluida.
- Seguimiento lógico de la conversación sin necesidad de reiniciar flujos, adaptándose a cambios o aclaraciones del usuario.

Fase 3. Asistente virtual

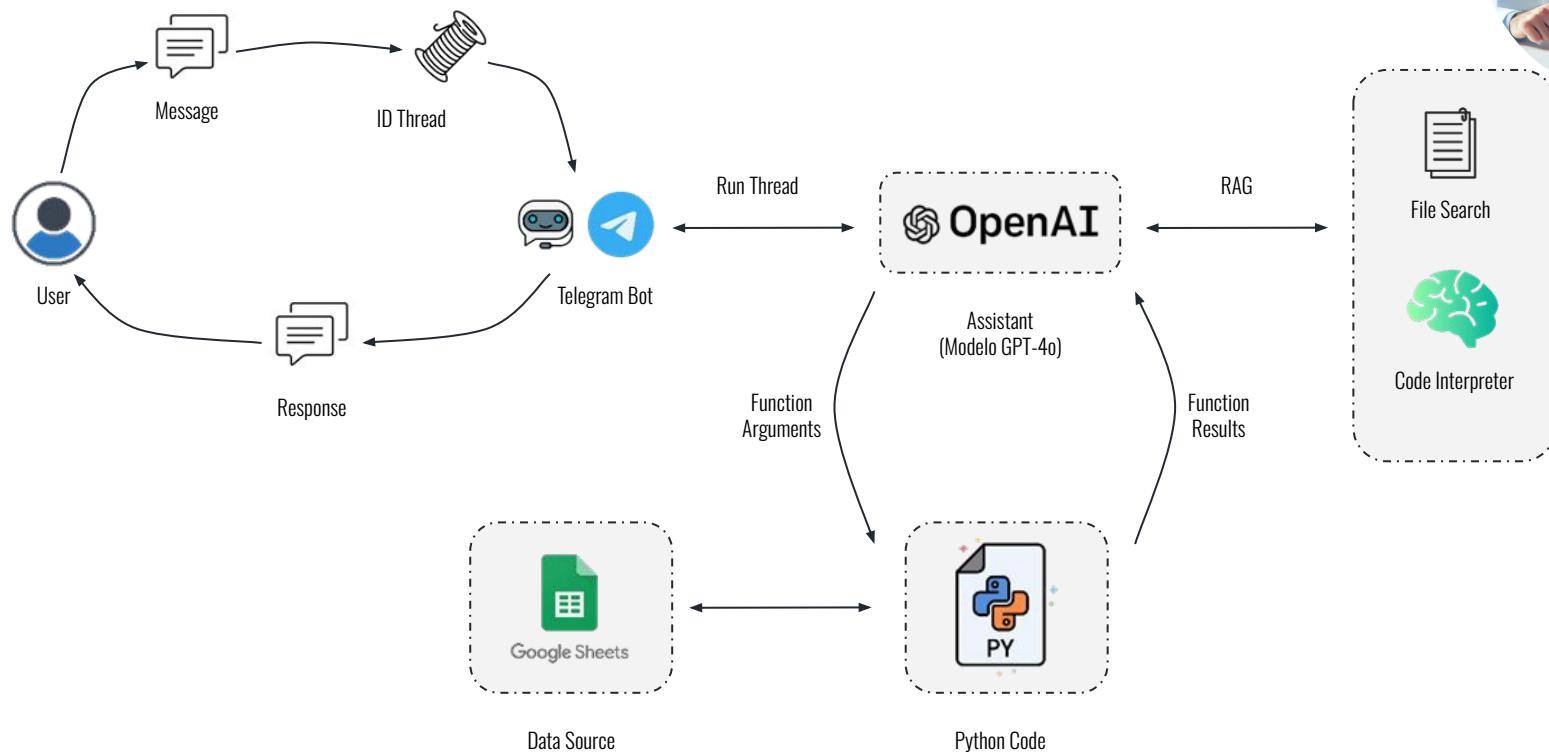
Comparativa 3ra generación vs. Generación anterior

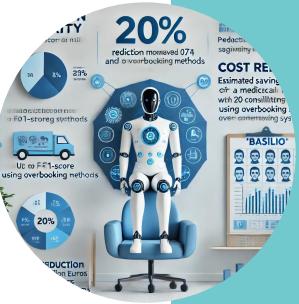
	Dialogflow (Generaciones anteriores)	OpenAI (GPT-4, 3ra Generación)
Algoritmo	Usa algoritmos simples, con respuestas predefinidas	Basado en modelos avanzados (LLMs), genera respuestas más complejas y flexibles.
Flujos de conversación	Limitado a flujos conversacionales rígidos y configurados manualmente.	Responde dinámicamente a cambios en la conversación, manteniendo una interacción fluida.
Personalización	Personalización limitada y manual para cada caso de uso.	Aprende de interacciones anteriores y ajusta automáticamente las respuestas.
Adaptabilidad al Contexto	Baja adaptabilidad al contexto.	Responde mejor al contexto, ofreciendo respuestas más coherentes.
Consistencia	Las respuestas son consistentes pero más rígidas y predecibles.	Las respuestas pueden variar, pero son más naturales e intuitivas.
Costo de Implementación	Menor inversión inicial y menores recursos técnicos.	Requieren una mayor inversión y recursos técnicos.



Fase 3. Asistente virtual ‘Basilio’

Arquitectura





Conclusiones

Predicción superior:

- El modelo TabNet alcanzó el mejor AUC, mejorando los resultados respecto a métodos de Machine Learning Tradicionales.

Eficiencia operativa:

- Se estima reducir **los tiempos de inactividad en un 55%** de los servicios médicos.
- "**Basilio**", asistente virtual de tercera generación, al compararlo con la gestión de citas con personal humano, **reduce los costes operativos y permite una respuesta inmediata a usuarios**.

Reducción de costos:

- Ahorro estimado de hasta **2 millones de euros anuales** para un centro médico con 20 consultas que no utiliza sistemas de overbooking.
- Ahorro de hasta **650 mil euros anuales** para centros médicos que ya usan sistemas tradicionales de overbooking.



Conclusiones

La integración de técnicas de inteligencia artificial, junto con un sistema de overbooking inteligente, puede generar un impacto significativo en la gestión de citas médicas.

- Los resultados indican mejoras en la eficiencia operativa, **reducción de costos operativos y del tiempo de espera para la asignación de una cita.**
- La implementación del asistente virtual ha mostrado ser una solución que permite **automatizar y optimizar la interacción con los usuarios**, con el potencial de mejorar la experiencia del paciente.



Desafíos Futuros

- **Mejora Modelos Predictivos.**
 - Incorporar técnicas de aprendizaje por transferencia (Transfer Learning).
 - Incorporar un 2º modelo predictivo del clima.
- **Depurar el Sistema de Overbooking.**
 - Realizar un análisis más detallado de la sensibilidad del sistema a diferentes parámetros de coste.
 - Entrenar un modelo de IA de optimización.
- **Expansión del Asistente Virtual.**
 - Incorporar capacidades de procesamiento de voz (speech-to-text y text-to-speech).
- **Validación en Entornos Reales.**
 - Realización de estudio piloto en un centro médico.



muchas *Gracias*



MediAgenda Solutions

