UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

IMPLEMENTACIÓN DE ALGORITMOS DE IA PARA LA DETECCIÓN DE FRAUDE EN SERVICIOS DE SALUD

POR

Víctor Ricardo Contreras Valderrama

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Electrónico

Profesor Guía

Rosa Figueroa Iturrieta

Marzo 2024
Concepción (Chile)

©2024 Víctor Ricardo Contreras Valderrama

©2024 Víctor Ricardo Contreras Valderrama
Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Agradecimientos

Me gustaría agradecer a mi familia, especialmente mis padres que siempre han estado ahí para apoyarme y alentarme en este proceso en mi vida. Agradecer también a mi nueva familia, Natalia y Blanca, que siempre han estado para ayudarme en lo que necesite. Nada de esto sería posible sin ustedes, las quiero mucho.

Agradecer a la profesora Rosa Figueroa que siempre estuvo disponible para guiarme y responder mis dudas cuando lo necesité.

Sumario

El fraude en la salud es un tema muy relevante hoy. El fraude en seguros médicos en Estados Unidos (Medicare) causa pérdidas superiores a los miles de millones de dólares por año. Adicional a las pérdidas económicas se suma el aumento de burocracia del proceso de cuenta médica, producto de los fraudes, que dificultan y hacen engorrosos los procesos de justificación y cobro de cuentas médicas. Los actores principales de estos procesos son los proveedores de salud (hospitales y otros centros de salud), doctores y los pacientes que utilizan este servicio.

Esta memoria de título está enfocada en atacar este problema utilizando datos tabulares de código abierto que contienen la información de pacientes registrados en el programa federal de seguro médico Medicare con el proveedor de salud respectivo asignado a cada paciente. En total se utilizó información de 558.211 pacientes. Primero, se realizó un análisis exploratorio de los datos para procesarlos seguido de una ingeniería de características. Posteriormente se hizo una sintonización de hiperparámetros con validación cruzada para asegurar la robustez de los parámetros seleccionados. Paralelamente, se evaluó el error de cada clasificador para monitorear el rendimiento de los modelos. Finalmente, se entrenaron y evaluaron tres clasificadores (AdaBoost, Support Vector Machine (SVM) y TabNet), con el objetivo de clasificar los ejemplos del set de datos en las clases “fraude” y “no fraude”. En términos de sensibilidad, el mejor clasificador fue SVM (sensibilidad=0.62), seguido por AdaBoost (sensibilidad = 0.61). Por otro lado, en términos de área bajo la curva de operación, el clasificador que presento los mejores resultados fue AdaBoost con un puntaje de 0.7.

Este estudio revela la capacidad de técnicas avanzadas de aprendizaje automático para identificar posibles fraudes en programas de seguro médico, demostrando que, aunque el SVM tuvo la mejor sensibilidad, AdaBoost ofreció un equilibrio más robusto entre sensibilidad y especificidad, como se refleja en su AUC-ROC superior. Este enfoque no solo aporta a la detección eficiente de fraudes sino también subraya la importancia de la selección y optimización adecuadas de modelos para abordar problemas complejos de datos.

Abstract

Health fraud is a highly relevant issue today. Insurance fraud within the United States Medicare system results in losses amounting to billions of dollars annually. In addition to financial losses, there is an increase in the bureaucracy of the medical billing process, a direct consequence of fraud, which complicates and encumbers the processes of justification and billing for medical services. The main actors in these processes are healthcare providers (hospitals and other health centers), doctors, and the patients who use these services.

This thesis focuses on addressing this problem using open-source tabular data containing information on patients registered in the federal Medicare insurance program, along with the respective healthcare provider assigned to each patient. Information from a total of 558,211 patients was used. First, an exploratory data analysis was conducted to process the data, followed by feature engineering. Subsequently, hyperparameter tuning was performed with cross-validation to ensure the robustness of the selected parameters. Concurrently, the error of each classifier was evaluated to monitor the models' performance. Finally, three classifiers (AdaBoost, Support Vector Machine (SVM), and TabNet) were trained and evaluated, aiming to classify the dataset examples into “fraud” and “non-fraud” categories. In terms of sensitivity, the best classifier was SVM (sensitivity=0.62), followed by AdaBoost (sensitivity=0.61). On the other hand, in terms of the area under the receiver operating characteristic curve, AdaBoost presented the best results with a score of 0.7.

This study reveals the capability of advanced machine learning techniques to identify potential frauds in medical insurance programs, showing that, although SVM had the best sensitivity, AdaBoost offered a more robust balance between sensitivity and specificity, as reflected in its superior AUC-ROC. This approach not only contributes to the efficient detection of fraud but also underscores the importance of proper model selection and optimization to address complex data problems.

Tabla de contenidos

Lista de tablas

Lista de figuras

1. Introducción

1.1. Antecedentes del problema

El problema de la detección de fraude en los servicios de salud representa una preocupación crítica a nivel global en el ámbito de la atención médica [1] [2]. Dicho problema consiste en la obtención de beneficios o pagos indebidos a través de reclamaciones falsas o engañosas dentro del sistema de atención médica. Estos actos fraudulentos pueden variar desde la emisión de facturas por servicios médicos no prestados hasta la manipulación de registros de pacientes para obtener pagos no merecidos. Este fenómeno no sólo compromete la integridad del sistema de salud, sino que también implica significativas pérdidas financieras para las aseguradoras y, lo más alarmante, puede afectar la calidad de la atención que reciben los pacientes reales [3].

El fraude en el sector de la salud es un desafío complejo que impacta a gobiernos, aseguradoras, proveedores de atención médica y pacientes. El incremento en los costos de atención médica y la complejidad creciente de los sistemas de salud han facilitado el surgimiento de actividades fraudulentas [4]. Aunque este problema tiene décadas de existencia, ha evolucionado con la digitalización y la implementación de la atención médica electrónica. Con la adopción de registros médicos electrónicos y métodos de facturación digital, los defraudadores han encontrado nuevas oportunidades para explotar vulnerabilidades en estos sistemas para su beneficio personal. En este contexto, la detección eficaz de fraude se vuelve crucial para preservar la integridad y sostenibilidad de los sistemas de salud a nivel mundial.

Los daños financieros que deben asumir los administradores de estos servicios hacen imperativa la intervención de instituciones judiciales y financieras para prevenir el fraude. Aunque es difícil estimar el monto exacto de las pérdidas, se calcula que oscilan entre 13 mil millones de euros en Europa y entre 20 y 70 mil millones de dólares en Estados Unidos anualmente [3]. Se estima que hasta un 10% del presupuesto anual del gobierno de EE.UU. se pierde debido al fraude, lo que equivale a aproximadamente 300 mil millones de dólares[5].

1.2. Definición del problema

Este estudio se centra en detectar fraudes utilizando información de pacientes registrados en el programa federal de seguro médico Medicare para clasificar a los proveedores como fraudulentos o no fraudulentos, con datos etiquetados.

1.3. Objetivos generales

Diseñar un modelo de clasificación de fraude empleando datos tabulares de pacientes registrados en el programa federal de seguro médico Medicare.

1.4. Objetivos específicos

- Aplicar metodologías de Ingeniería de Características para identificar y seleccionar las mejores variables para formar parte del conjunto de características a ser utilizadas en el proceso de clasificación.
- Implementar modelos de clasificación que permitan realizar la clasificación de los datos, teniendo en cuenta la naturaleza de los datos y estudios previos en la temática.
- Evaluar los algoritmos de clasificación implementados con el fin de obtener métricas que permitan identificar el mejor algoritmo para el problema.

1.5. Alcances y limitaciones

Este trabajo se enfoca en alcanzar los objetivos previamente mencionados, considerando ciertas limitaciones:

- El costo computacional del código debe ser razonable, dado que el manejo de grandes volúmenes de datos puede incrementar significativamente el tiempo de procesamiento de algunos algoritmos con el aumento de la dimensionalidad del conjunto de datos.
- Es esencial que los resultados sean reproducibles, lo que requiere una definición precisa de las bibliotecas utilizadas, el conjunto de datos finales y los hiperparámetros de los distintos modelos implementados.

2. Revisión bibliográfica y marco teórico

2.1. Estado del arte en clasificación de fraude.

Una amplia gama de investigaciones se centra en los algoritmos de clasificación basados en el aprendizaje automático (ML) y el aprendizaje profundo (DL), con un enfoque significativo en la detección de fraude, tanto en el sector bancario como en el de la salud. Un estudio innovador de Yoo et al. [6] adopta una perspectiva basada en la teoría de grafos, implementando dos enfoques distintos: el primero emplea redes neuronales basadas en grafos, mientras que el segundo aplica métodos clásicos de ML con información de grafos. Utilizar la teoría de grafos permite aprovechar las relaciones entre los datos para mejorar el desempeño, aunque a menudo estas relaciones no son evidentes, lo que requiere un profundo conocimiento de los datos y sus interconexiones. A pesar de sus excelentes resultados, estos métodos demandan un tiempo de ejecución hasta 300 veces mayor en comparación con las técnicas convencionales. Hancock y Taghi [7], por otro lado, se centraron en el modelo CatBoost, un algoritmo basado en árboles de decisión eficaz tanto para datos categóricos como numéricos. CatBoost ha demostrado ser particularmente eficaz en la identificación de patrones de fraude, lo que permite ahorrar tiempo al obviar etapas de preprocesamiento. En comparación con XGBoost, CatBoost mostró un mejor desempeño en términos de curva ROC, con valores de 0.7851 frente a 0.7615, respectivamente, utilizando datos de fraude obtenidos del sitio oficial de Medicare[8].

Altaher y Malebary [9] presentaron una optimización del modelo LightGBM (OLightGBM), el cual superó en desempeño a otras técnicas en casi todas las métricas principales, incluyendo la máquina de vectores de soporte (SVM), la regresión logística, el método de los k vecinos más cercanos, los clasificadores bayesianos, el bosque aleatorio y el árbol de decisión. OLightGBM también superó al modelo base LightGBM y a CatBoost. En [10], se propone un modelo basado en LightGBM centrado en los "ejemplos difíciles", es decir, aquellos casos en los que el modelo enfrenta mayores dificultades para clasificar. Esta estrategia es útil en conjuntos de datos altamente desbalanceados, como suele detectar fraude. Estudios comparativos, como los presentados en [11], [12] y [13], emplearon el mismo conjunto de datos disponible en Kaggle que se utilizará en este proyecto. Se exploraron modelos basados en árboles, incluyendo el bosque aleatorio, el árbol de decisiones, XGBoost, AdaBoost y un clasificador basado en el potenciador de gradiente, junto con métodos más tradicionales como SVM y la regresión logística. Los resultados de estos estudios servirán como referencia para comparar las métricas de desempeño de nuestro proyecto.

Una revisión sistemática realizada por Ashtiani en 2022 [14] ofrece una visión de las tendencias

actuales en la identificación de fraude financiero. Aunque los datos de fraude financiero pueden diferir en estructura de los datos de fraude en servicios de salud, este estudio proporciona información valiosa sobre los modelos implementados. Los tres modelos más utilizados en los estudios analizados son SVM, árboles de decisión y clasificadores bayesianos, presentes en más del 80% de las investigaciones revisadas. Las redes neuronales artificiales también son un método popular.

Yashraj et al. [15] se centran en la detección de fraude en servicios de salud utilizando datos desbalanceados, aplicando técnicas de muestreo populares como el sobremuestreo sintético de minorías (SMOTE) y el submuestreo. Destacan los modelos basados en árboles como XGBoost, LightGBM, GBM, árboles de decisión y bosques aleatorios, siendo XGBoost y LightGBM los que mostraron desempeños muy similares.

2.2. Marco teórico.

2.2.1. Información mutua

En teoría de información, la información mutua es una técnica que evalúa la dependencia entre dos variables; en este contexto, se refiere a las características y la variable objetivo [16]. Esta métrica cuantifica cuánta información aporta una variable sobre otra. La relación entre dos variables X e Y se define como:

$$I(X\,;\,Y) \;=\; H(X) - H(X \mid Y) \qquad \text{[Ec. 1]}$$

Donde $I(X;Y)$ corresponde a la ganancia mutua entre X e Y, $H(X)$ es la entropía de X y $H(X \mid Y)$ es la entropía condicional de X dado un Y. Un valor alto de ganancia mutua indica una mayor relación entre las variables, sugiriendo que la característica proporciona información valiosa para el modelo. Por el contrario, un valor bajo implica que la característica aporta poca información, lo que puede justificar su eliminación del modelo. Esta técnica es crucial para identificar y seleccionar las características más significativas que influyen en la precisión de un modelo de aprendizaje automático, optimizando así su rendimiento y eficiencia.

2.2.2. Validación cruzada

La validación cruzada (CV) es una técnica esencial en proyectos de inteligencia artificial para garantizar la confiabilidad y coherencia del modelo durante el proceso de ajuste de hiperparámetros [17]. Uno de los métodos más populares dentro de la validación cruzada es el *k-folds*. Este procedimiento divide el conjunto de datos en *k* segmentos de igual tamaño, utilizando cada segmento una vez como conjunto de prueba mientras que los restantes sirven como conjunto de entrenamiento. Este proceso se repite *k* veces, con cada segmento usado como conjunto de prueba. Los resultados

obtenidos se promedian para obtener una evaluación final más precisa del modelo. La Figura 2.1 ilustra el proceso de validación cruzada con *k*=4, donde el conjunto de datos se divide en cuatro partes, y cada parte se utiliza como conjunto de prueba en una iteración diferente.

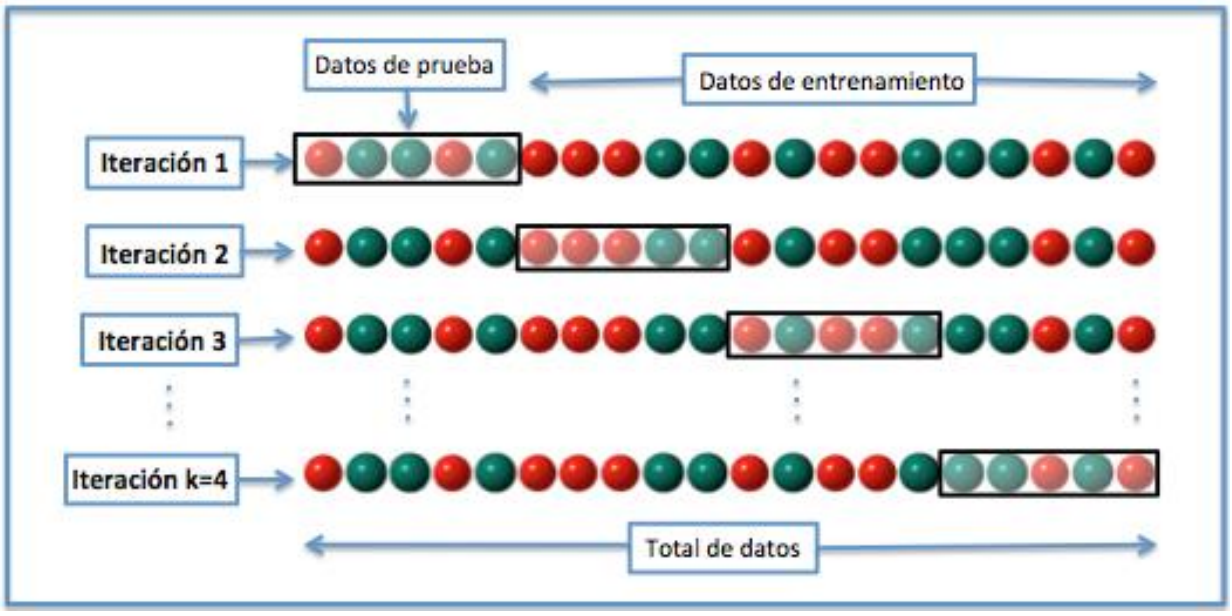


Figura 2. 1. Validación cruzada para *k=4 [18]*

Es crucial subrayar que la aplicación de la validación cruzada se lleva a cabo exclusivamente dentro del conjunto de datos de entrenamiento. De esta manera, se asegura que la evaluación del modelo se realice con un conjunto de datos de prueba totalmente nuevo y no expuesto previamente durante el proceso de ajuste. Adoptar esta estrategia promueve una evaluación del modelo sólida y confiable, reduciendo significativamente el riesgo de sobreajuste y garantizando una capacidad de generalización efectiva hacia datos no vistos. Este procedimiento subraya la importancia de preservar la integridad del conjunto de datos de prueba, reservándolo únicamente para la evaluación final del modelo, con el fin de obtener una medida fidedigna de su desempeño en situaciones reales.

2.2.3. Clasificador Support Vector Machine (SVC)

El modelo de Máquinas de Vectores de Soporte (SVM) es una herramienta poderosa y versátil en el campo del aprendizaje automático supervisado, aplicable tanto a tareas de regresión como de clasificación [19]. La eficacia de las SVM reside en su capacidad para trabajar en espacios de alta dimensionalidad y en su flexibilidad para adaptarse a diferentes tipos de datos mediante la selección de funciones kernel adecuadas. Estas funciones transforman el espacio de características a uno de mayor dimensión donde es más fácil lograr una separación lineal entre las clases. Los kernels más

comunes incluyen el lineal, polinomial, radial (RBF) y sigmoidal.

La implementación práctica de modelos SVM puede realizarse utilizando Scikit-learn, una de las bibliotecas de Python más populares en el ámbito del aprendizaje automático. Dentro de Scikit-learn, el modelo Support Vector Classifier (SVC) se basa en los principios de las SVM y ofrece una interfaz conveniente para su aplicación en problemas de clasificación [20]. Una de las ventajas de trabajar con SVC es la fácil sintonización de hiperparámetros, ya que sólo es necesario elegir factor de regularización y un kernel apropiado, que pueden ser lineal, polinomial, radial o sigmoidal.

2.2.4. TabNet

TabNet es un innovador modelo de red neuronal desarrollado por Google AI, diseñado específicamente para mejorar el rendimiento en el manejo de datos tabulares [21]. Datos de tipo tabular son los más comunes en el mundo digital [22]. A pesar de esto, modelos de DL enfocados en datos tabulares han sido poco explorados. La estructura de TabNet se caracteriza por integrar un transformador de características, un transformador atento y una máscara de características, como se ilustra en la figura 2.2. Esta arquitectura facilita una representación procesada que se divide mediante un bloque separador, la cual es luego utilizada tanto por el transformador atento como para generar la salida del modelo. Un aspecto distintivo de TabNet es la capacidad de la máscara para ofrecer insights interpretables sobre el funcionamiento del modelo, permitiendo una agregación de información para identificar las características más significativas.

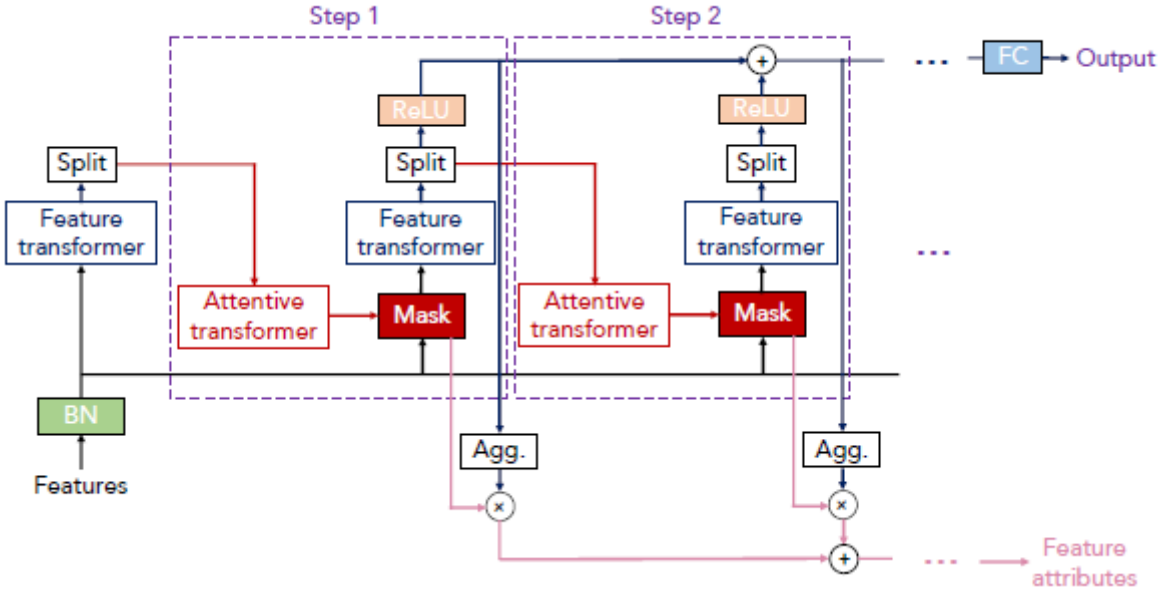


Figura 2. 2. Arquitectura general codificador TabNet

La figura 2.3 desglosa el funcionamiento de los bloques transformadores. Se muestra un

ejemplo de un bloque transformador de características, compuesto por una red de cuatro capas, donde dos son comunes a todos los pasos de decisión y las dos restantes son específicas de cada paso. Cada capa incluye una capa completamente conectada (FC), normalización por lotes (BN) y la función de activación GLU, que introduce no linealidad. En cuanto al bloque transformador atento, se presenta un mapeo de una sola capa modulado por información de escala previa, la cual indica la utilización previa de cada característica. El modelo emplea sparsemax para la normalización, resultando en una selección enfocada de las características más relevantes.

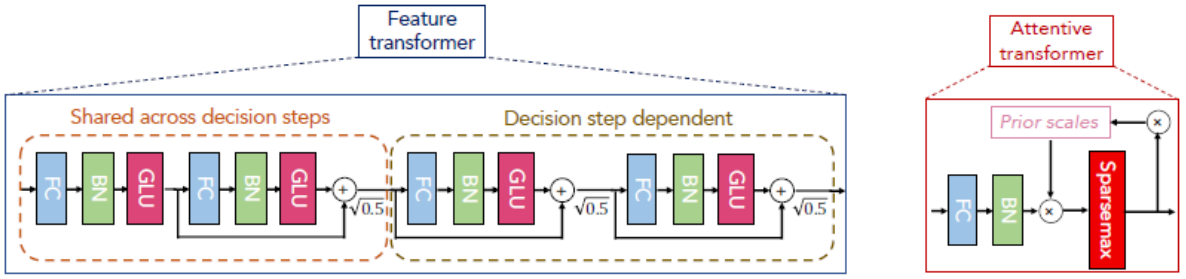


Figura 2. 3. Bloques de características y atención.

Los creadores de TabNet han reportado que este modelo supera en rendimiento a otros enfoques convencionales. La innovadora estrategia de TabNet para la selección de características, junto con el uso de mecanismos de atención, permite un aprendizaje más eficaz de conjuntos de datos complejos. Esto abre nuevas vías para aplicaciones de aprendizaje automático en diversos campos. El modelo se implementa utilizando la biblioteca PyTorch, destacando por su flexibilidad y eficiencia en la configuración de hiperparámetros clave, detallados en la tabla 2.1. En resumen, TabNet representa un avance significativo en el procesamiento de datos tabulares mediante aprendizaje profundo.

Tabla 2. 1. Hiperparámetros a sintonizar del modelo de aprendizaje profundo TabNet

Parámetro	Descripción
Dimensión de entrada	Dimensión de los embedding de decisión
Dimensión de salida	Dimensión de los embedding de atención, suele configurarse con el mismo valor que la dimensión de entrada
Numero de pasos	Este parámetro define el número de pasos de decisión o bloques que se utilizan para hacer sus predicciones.
Factor de relajación (gamma)	Este parámetro ayuda a controlar la cantidad de nuevas características que se seleccionan en cada paso de decisión.
Coeficiente de dispersión	Este parámetro controla la fuerza de la regularización aplicada al mecanismo de atención para fomentar la dispersión

2.2.5. AdaBoost

AdaBoost es un modelo de basado en arboles de decisiones [23]. Este método destaca por transformar varios clasificadores débiles, es decir lo que apenas superan el rendimiento al azar, para formar un clasificador fuerte. El entrenamiento es secuencial donde cada clasificador débil va corrigiendo a sus predecesores para luego ser combinado mediante una votación ponderada. La metodología de AdaBoost se inicia asignando un peso uniforme a todas las instancias en el conjunto de datos de entrenamiento, lo que refleja la importancia de cada una en el proceso de aprendizaje. A medida que el algoritmo avanza, estos pesos se ajustan para enfocar el aprendizaje en las instancias más difíciles de clasificar correctamente, lo que permite una mejora continua del modelo. Las etapas clave en el funcionamiento de AdaBoost son:

- Inicialización: AdaBoost comienza asignando pesos iguales a todas las instancias del conjunto de entrenamiento. Estos pesos son utilizados para indicar la importancia de cada instancia durante el entrenamiento del modelo. El funcionamiento general de Adaboost se puede separar en etapas:

- Entrenamiento: En cada iteración, AdaBoost entrena un clasificador débil utilizando el conjunto de entrenamiento ponderado actual. Después del entrenamiento, el algoritmo evalúa el desempeño del clasificador en el conjunto de entrenamiento.

- Actualización de pesos: finalizada cada iteración, se aumentan los pesos de las instancias mal

clasificadas por el clasificador actual, de modo que, en la siguiente iteración, el nuevo clasificador débil se enfoque más en los casos difíciles y trate de corregirlos.

- Combinación de clasificadores: Cada clasificador débil se asigna a un peso en la votación final basado en su precisión. Los clasificadores con mejor rendimiento tienen más influencia en la votación final. Los pesos de votación se calculan a partir del error de cada clasificador débil, donde el error es la fracción ponderada de clasificaciones incorrectas sobre el conjunto total.

- Resultado final: El proceso añade clasificadores hasta alcanzar un número predeterminado de iteraciones o hasta lograr una precisión perfecta en el conjunto de entrenamiento. El modelo final hace predicciones basándose en la suma ponderada de las predicciones de todos los clasificadores débiles.

Se utilizó la biblioteca AdaBoostClassifier de Scikit-learn. Los hiperparámetros críticos a ajustar incluyen el número de estimadores, que determina el número máximo de estimadores en los que se detiene el proceso de boosting, y la tasa de aprendizaje, que influye en la contribución de cada clasificador en cada iteración. El número de estimadores regula la complejidad del modelo combinado: un valor excesivamente alto puede causar sobreajuste. Por otro lado, la tasa de aprendizaje equilibra la velocidad y precisión del ajuste del modelo; una tasa demasiado baja puede resultar en una convergencia lenta hacia la solución óptima, mientras que una tasa demasiado alta puede sobrepasar el mínimo óptimo. La selección cuidadosa de estos hiperparámetros es crucial para optimizar el rendimiento del modelo AdaBoost, equilibrando así la precisión y la generalización del modelo.

2.2.6. Métricas de evaluación

Las métricas de desempeño juegan un papel crucial en la evaluación de clasificadores. Se centran particularmente en el análisis de la matriz de confusión, presentada en la Tabla 2.2, que categoriza los resultados de la clasificación en correctos e incorrectos. Esta matriz compila las mediciones esenciales para una evaluación exhaustiva del rendimiento del proceso de clasificación. Se examinarán cinco métricas principales, que son:

Tabla 2. 2. Matriz de confusión

		Predicción	
		Positivo	Negativo
Real	Positivo	Verdadero positivo (TP)	Falso negativo (FN)
	Negativo	Falso positivo (FP)	Verdadero negativo (TN)

- Exactitud (*accuracy*): es la fracción de predicciones correctas entre el número total de casos analizados. Permite conocer una medida general de la cantidad de predicciones correctas hechas por el modelo. Esta métrica es especialmente útil cuando existe un balance de clases.

$$Exactitud = \frac{TP+TN}{TP+FP+FN+TN} \quad \text{[Ec. 2]}$$

- Área bajo la curva de características operativas del receptor (AUC): es una métrica que permite evaluar el rendimiento de un modelo de clasificación calculando el área bajo la curva ROC. Esta curva se traza en un plano que tiene la tasa de verdaderos positivos (sensibilidad) en el eje Y frente a la tasa de falsos positivos (1-especificidad) en el eje X para diferentes puntos de corte o umbrales de decisión. Esta métrica proporciona una medida agregada del rendimiento en todos los umbrales de clasificación, y su valor varía de 0 a 1. Un AUC de 1 indica un modelo perfecto que clasifica correctamente todas las instancias positivas y negativas. Un AUC de 0.5 sugiere un rendimiento equivalente al azar.

- Precisión: La precisión es la fracción de predicciones positivas que fueron correctas. Se centra en la calidad de las predicciones positivas del modelo. Se calcula como el número de verdaderos positivos dividido por la suma de los verdaderos positivos y los falsos positivos. Se calcula de la siguiente manera:

$$Precisión = \frac{TP}{TP+FP} \quad \text{[Ec. 3]}$$

- Sensibilidad (*Recall*): La sensibilidad mide la capacidad del modelo para identificar de manera correcta todas las instancias importantes, es decir, la fracción de los verdaderos positivos identificados del total de casos positivos reales. Esta medida es de mucha utilidad donde no reconocer valores positivos tiene consecuencias graves. Se calcula de la siguiente manera:

$$Sensibilidad = \frac{TP}{TP+FN} \qquad \text{[Ec. 4]}$$

- Puntaje F1 (F1-score): es la media armónica de la precisión y la sensibilidad. Es una medida que combina estas dos métricas en un solo número, siendo útil cuando se necesita un balance entre estas dos métricas. Esta se calcula de la siguiente manera:

$$F1 = 2 \times \frac{Precision \times Recall}{(Precision+Recall)} \qquad \text{[Ec. 5]}$$

2.2.7. Perdida logarítmica (Log loss)

Esta es una métrica de evaluación para modelos de clasificación que se basa en la entropía cruzada. Mide la diferencia de distribución de probabilidad real de las clases y la distribución de probabilidad predicha por el modelo y está en el rango entre 0 y 1. Esta métrica de perdida es relevante en modelos de clasificación binaria ya que esta castiga predicción seguras y equivocadas, lo que ayuda a identificar si el modelo está demasiado confiado en sus decisiones incorrectas. Otra razón por la que es útil utilizar perdida logarítmica es que permite comparar distintos modelos. Para un problema de clasificación binaria se calcula de la siguiente manera:

$$Perdida\ logarítmica\ = -(y \cdot log(p) + (1-y) \cdot log(1-p)) \qquad \text{[Ec. 6]}$$

Donde:

- y es la etiqueta verdadera (0 o 1).
- p es la probabilidad predicha para la clase con etiqueta 1.

3. Metodología

Para la implementación del modelo de clasificación de fraude, se desarrollará una metodología detallada en la Figura 3.1. El proceso comienza con la adquisición del set de datos de acceso público, que consta de cuatro conjuntos distintos. Estos se unificarán en un único dataset basado en características comunes, lo que permitirá un análisis uniforme.

La siguiente fase involucra un análisis exploratorio de datos. Este análisis es fundamental para obtener un entendimiento profundo del conjunto de datos, permitiendo identificar su distribución, detectar valores atípicos y comprender las interacciones entre diferentes variables. Este paso es crucial para guiar las decisiones del proyecto.

A continuación, se realiza la ingeniería de características. Este proceso incluye la reducción de la dimensionalidad que puede llevar a un sobreajuste del modelo con el set de entrenamiento, resultando en un rendimiento deficiente en el set de prueba[24]. La ingeniería de características también puede incluir la selección de características relevantes, transformaciones de variables, y la creación de nuevas características que pueden mejorar la capacidad predictiva del modelo.

El siguiente paso es el entrenamiento de tres modelos de clasificación diferentes. Sus hiperparámetros serán sintonizados para maximizar su desempeño. La sintonización de hiperparámetros se realiza a través del método búsqueda en cuadrícula (gridsearch) con validación cruzada, con el objetivo de encontrar la configuración óptima que ofrezca los mejores resultados.

Finalmente, se evaluarán los modelos utilizando métricas de desempeño. Métricas relevantes como la exactitud, precisión, la sensibilidad, el puntaje F1 y el área bajo la curva ROC (AUC-ROC). Estas métricas proporcionarán una visión detallada del rendimiento de cada modelo, permitiendo identificar el más efectivo para la detección de fraude.

Esta metodología garantiza un enfoque sistemático y riguroso para la clasificación de fraudes, asegurando que el modelo final sea robusto, preciso y capaz de generalizar bien a nuevos datos.

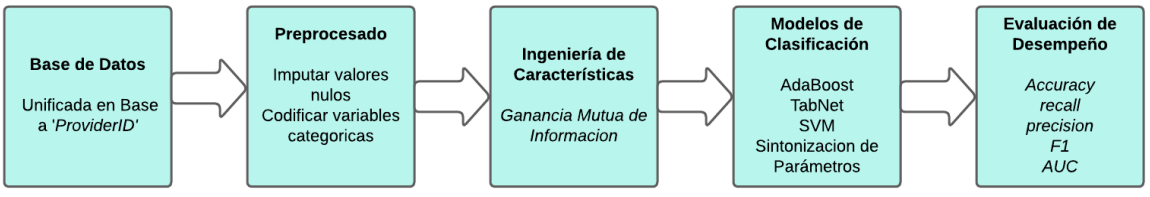


Figura 3. 1. Modelo propuesto

3.1. Descripción de set de datos

El conjunto de datos utilizado es de tipo tabular de código abierto relacionado a proveedores fraudulentos asociados al servicio de Medicare. El set fue obtenido de un repositorio de *Kaggle* [25]. Este set de datos consiste en 4 subsets.

3.1.1. Pacientes hospitalizados

Este set contiene información relacionada a los pacientes que requerían hospitalización. En total son 40.473 muestras y 30 características que corresponden a reclamos. Algunas de las características principales se pueden apreciar en la tabla 3.1.

Tabla 3. 1. Características principales del set de datos de pacientes hospitalizados

Característica	Descripción
BeneID	Código único relacionado al benefactor
ClaimID	Código único relacionado al reclamo
Provider	Proveedor a cargo del servicio
AttendingPhysician	Doctor a cargo de la atención
OperatingPhysician	Doctor a cargo de la operación
OtherPhysician	Otro doctor, en caso de ser necesario
ClmDiagnosisCode_1	Primer código de diagnóstico, este cumple con el formato de ICD-9, que está bajo el estatuto de la organización mundial de la salud. Por reclamo puede haber hasta 10 diagnósticos posibles
ClmProcedureCode_1	Primer código de procedimiento, este cumple con el formato de ICD-9. Por reclamo puede haber hasta 6 procedimiento realizados
DeductibleAmtPaid	Monto pagado relacionado al deducible
InscClaimAmtReimbursed	Monto reembolsado por el seguro
AdmissionDt/ DischargeDt	Fechas de admisión y liberación.

3.1.2. Pacientes no hospitalizados

Este conjunto de datos incluye información sobre pacientes que no requirieron hospitalización. Consta de 517,737 registros y 27 características distintas, todas ellas relacionadas con reclamaciones médicas. Las características principales son consistentes con las de los datos de pacientes hospitalizados, con la principal diferencia de que, dado que no se incluyen datos sobre fechas de

admisión y liberación. Esta consistencia en las características permite realizar comparaciones y análisis integrados entre pacientes hospitalizados y no hospitalizados.

3.1.3. Datos de beneficiario

Este conjunto de datos contiene información personal detallada de los pacientes, abarcando un total de 138,556 registros con 25 características distintas. Es importante destacar que el número total de registros de pacientes hospitalizados y no hospitalizados excede la cantidad de registros en este conjunto de datos. Esto se debe a que algunos pacientes pueden haber requerido múltiples atenciones médicas, lo que implica que un solo paciente puede estar asociado a varios registros de atención médica, aunque solo aparezca una vez en el conjunto de datos de información personal.

Tabla 3. 2. Características principales del set de datos de beneficiarios

Característica	Descripción
BeneID	Código único relacionado al benefactor
DOB/DOD	Fecha de nacimiento y de defunción
Gender	Genero del paciente
Race	Raza del paciente
ChronicCond_Alzheimer	Presencia de condición crónica, en este caso de alzheimer
IPAnnualReimbursementAmt	Monto anual reembolsado por hospitalización
IPAnnualDeductibleAmt	Monto anual relacionado al deducible por hospitalización
OPAnnualReimbursementAmt	Monto anual reembolsado por no hospitalización
OPAnnualDeductibleAmt]	Monto anual relacionado al deducible por no hospitalización

3.1.4. Datos de proveedores

Este conjunto de datos incluye información específica sobre el fraude asociado con los proveedores de servicios de salud. Cada proveedor está etiquetado como fraudulento o no fraudulento, y todos los reclamos vinculados a ese proveedor se etiquetan de acuerdo con esta clasificación.

Para consolidar la información dispersa en los distintos conjuntos de datos mencionados, se realizó un proceso de unificación ilustrado en la Figura 3.2. Primero, los conjuntos de datos de pacientes hospitalizados y no hospitalizados se fusionaron en base a dos características comunes: el identificador del beneficiario (BeneID) y el identificador del proveedor (ProviderID). Este conjunto fusionado se combinó después con el conjunto de datos de beneficiarios utilizando el identificador del beneficiario como clave de unión. Por último, se integró la información del conjunto de datos de

proveedores usando el identificador único de proveedores como referencia.

El resultado de este meticuloso proceso de unificación es un conjunto de datos integral con 558,221 registros y abarca 56 características distintas.

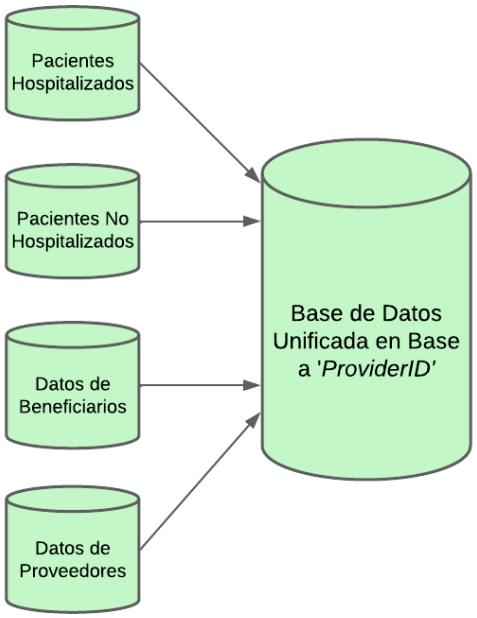


Figura 3. 2. Proceso de unión de los sets de datos

3.2. Preprocesado

En esta fase, se realizó una evaluación exhaustiva para identificar y abordar diversas irregularidades presentes en los conjuntos de datos. La presencia de ruido en los datos puede tener un impacto significativo en la eficacia de los modelos de aprendizaje automático que se entrenarán posteriormente. Los principales tipos de ruido identificados en esta etapa incluyen:

- Valores faltantes: Se detectó una cantidad considerable de datos faltantes en varias características, particularmente en los códigos de procedimiento y diagnóstico, donde hasta el 99% de los datos estaban ausentes. Este patrón sugiere que en muchos casos no se llevaron a cabo procedimientos, lo que requiere un tratamiento cuidadoso de estos valores faltantes para evitar sesgos en el modelo.
- Codificación de etiquetas: implica asignar un valor numérico específico a cada categoría única dentro de una variable no numérica. Este proceso simplifica la manipulación y análisis de los datos en etapas subsecuentes del modelado.
- Etiquetas erróneas: Las inconsistencias en las etiquetas pueden llevar a entrenar modelos con

información incorrecta. Se revisaron las etiquetas para corregir errores y garantizar la precisión de la clasificación.

En el set se encontró un gran número de valores faltantes, en características como los códigos de procedimiento y diagnostico se encontraron hasta un 99% de datos faltantes. Esto indica que en esos casos no se realizaron procedimientos. En estas características también se encontraron datos no numéricos, esto se debe a los códigos respectivos a la norma ICD-9. La estrategia adoptada fue la codificación de etiquetas, asignando un valor numérico a cada código único según la clasificación ICD-9. Esta técnica permite transformar atributos categóricos en formatos numéricos adecuados para el análisis y modelado computacional. Para características numéricas, los valores faltantes se reemplazaron por ceros.

Además, se llevó a cabo la estandarización de las características numéricas, como los montos deducibles y los reembolsos, que se encontraban en escalas variadas. Aplicando una estandarización z, se ajustaron los datos para tener una media de cero y una desviación estándar de uno. Aunque los modelos pueden tener un rendimiento similar sin esta estandarización, se considera una práctica recomendada para facilitar el aprendizaje y mejorar la interpretación de los resultados en modelos de clasificación.[26].

3.3. Ingeniería de características

La etapa de ingeniería de características es un componente esencial en el desarrollo de modelos de inteligencia artificial, ya que implica la transformación y selección meticulosa de variables para mejorar la efectividad y el rendimiento del modelo. A través de este proceso, se pueden crear nuevas características que enriquecen el conjunto de datos original con información valiosa, o eliminar aquellas que aportan poco y son redundantes, optimizando así la dimensionalidad de los datos y mitigando el riesgo de sobreajuste.

Particularmente en contextos críticos como el médico y económico, donde las decisiones basadas en los modelos pueden tener consecuencias significativas, la interpretabilidad del modelo adquiere una importancia primordial. La claridad en cómo las características influyen en las predicciones del modelo facilita la toma de decisiones informadas.

El análisis inicial se centró en identificar oportunidades para agregar, transformar o eliminar características, con especial atención en aquellas de naturaleza financiera, dada la relevancia del fraude en este ámbito. Además, se exploró la relación entre los doctores y los reclamos para descubrir

patrones o indicadores potenciales de fraude. Esto llevó a la introducción de nuevas variables que representan, por ejemplo, el promedio de variables numéricas significativas asociadas a cada identificador único de los proveedores o doctores involucrados.

En la práctica, se generaron 105 nuevas variables, ampliando el conjunto de datos a un total de 171 características distintas. Este enriquecimiento del conjunto de datos no solo mejora la capacidad del modelo para detectar patrones complejos y sutiles asociados con el fraude, sino que también contribuye a una mejor interpretación de los resultados, apoyando así la toma de decisiones basada en evidencia. Se hace un promedio de cada una de las variables numéricas asociadas a características relevantes como BeneID, AttendingPhysician, entre otras. La Figura 3.3 ilustra esta relación entre características y cómo se integran en el análisis.

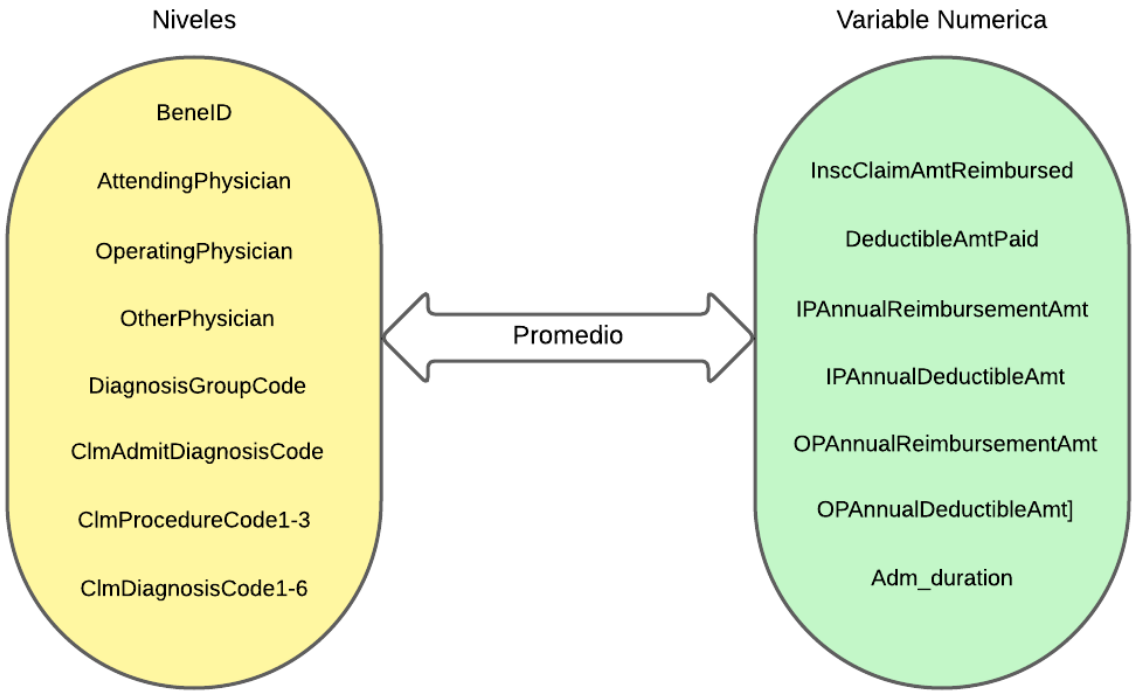


Figura 3. 3. Diagrama de relación de nuevas características

Después de seleccionar el conjunto de datos adecuado para el análisis, se aplicó la medida de informacion mutua para evaluar la dependencia entre las características del conjunto de datos y la variable objetivo, que en este caso es la detección de fraude. La validación cruzada se empleó como herramienta para asegurar la consistencia de los resultados obtenidos de este análisis.

Los resultados revelaron que las características relacionadas con los doctores encargados de los pacientes eran las que presentaban una mayor influencia en la identificación fraude. Este hallazgo subraya la importancia del papel que juegan los profesionales médicos en el contexto del fraude en los servicios de salud. La Tabla 3.3 muestra en detalle las características que demostraron un desempeño sobresaliente en este análisis.

El impacto significativo de estas características sugiere que los patrones de conducta o las prácticas de los doctores pueden ser indicadores críticos de posibles actividades fraudulentas.

Tabla 3. 3. Características con mayor relación con la variable objetivo

Característica	Información Mutua
PerAttendingPhysicianAvg/OPAnnualReimbursementAmt	0,176508
PerAttendingPhysicianAvg/InscClaimAmtReimbursed	0,176094
AttendingPhysician	0,169101
PerAttendingPhysicianAvg/IPAnnualReimbursementAmt	0,167862
PerAttendingPhysicianAvg/IPAnnualDeductibleAmt	0,167096
PerAttendingPhysicianAvg/OPAnnualDeductibleAmt	0,164858
PerAttendingPhysicianAvg/DeductibleAmtPaid	0,163505
OtherPhysician	0,044973
PerOtherPhysicianAvg/Claim_Duration	0,043373

3.4. Modelos de clasificación

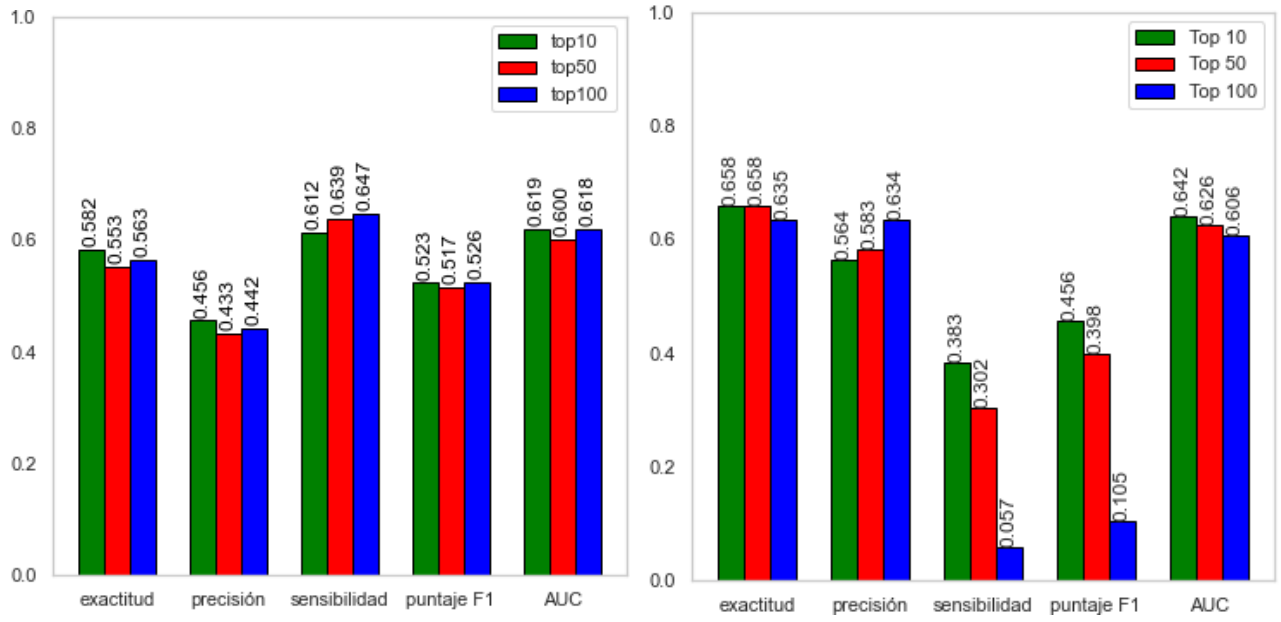
3.4.1. Selección de subset a utilizar

Para evaluar los modelos, se tomó el conjunto de datos definido antes y se procedió a evaluarlos con los clasificadores en sus configuraciones predeterminadas, sin sintonizar hiperparámetros. Este paso es fundamental para comprender cómo varía el desempeño de los modelos al trabajar con distintos subconjuntos de datos. Es importante mencionar que, como se ha señalado anteriormente, una alta dimensionalidad en el conjunto de datos puede llevar a un fenómeno de sobreajuste. Por eso, se decidió trabajar con subconjuntos con un número variable de características seleccionadas según su importancia determinada por el ranking de información mutua.

Para seleccionar un número adecuado de características para construir los clasificadores se comparó el desempeño de clasificadores utilizando los top 10, 50 y 100 características. Los resultados de esta evaluación inicial revelaron que el desempeño mejora al utilizar las 10 características más

relevantes. Este hallazgo es evidenciado en la figura 3.4, donde se observa una ligera ventaja en el desempeño en todas las métricas relevantes al utilizar el clasificador SVC. Por otro lado, en el caso de TabNet, se destacó en dos métricas específicas, mientras que en las demás, sus resultados fueron comparables a los obtenidos con el conjunto reducido de características.

Este análisis sugiere que la selección cuidadosa de características basadas en su relevancia puede influir en la eficacia de los modelos de clasificación, optimizando así el rendimiento y mitigando el riesgo de sobreajuste.



(a)SVC (b) TabNet

Figura 3. 4. Métricas de desempeño para distintos subsets

La cantidad de muestras se ajustó de 558.221 a 10.000, en línea con la estrategia de concentrar el análisis en las 10 características más influyentes identificadas. Según las recomendaciones de buenas prácticas en modelado estadístico, se sugiere emplear un número de muestras que sea al menos 10 veces mayor al número de características evaluadas [27]. Para llevar a cabo el muestreo aleatorio, se preservó la proporción original entre casos de fraude y no fraude, que es del 62% y 38%, respectivamente. Este enfoque garantiza que el subconjunto de datos mantenga la representatividad del conjunto original, permitiendo que los hallazgos y modelos desarrollados reflejen de manera precisa las tendencias y patrones inherentes al conjunto de datos completo. De esta manera, la estrategia de reducción de muestras contribuye a mantener un equilibrio entre la riqueza informativa de los datos y la eficiencia del proceso de modelado, evitando el riesgo de sobreajuste y promoviendo una generalización efectiva en las predicciones del modelo.

Posteriormente, el conjunto de datos se dividió en dos segmentos: un 70% destinado al

entrenamiento y el otro 30% reservado exclusivamente para pruebas. Este último conjunto de prueba se utilizará al final del proyecto, asegurando que la evaluación final del modelo se haga en datos nuevos y no vistos durante el entrenamiento. Este enfoque es fundamental para evitar cualquier sesgo en la evaluación del modelo y garantizar una estimación fiable de su capacidad para generalizar a nuevos datos.

3.4.2. Sintonización de hiperparámetros

La fase de ajuste de hiperparámetros es un paso crítico en el desarrollo de modelos de machine learning, ya que los valores seleccionados pueden tener un impacto significativo en el rendimiento del modelo. Para el Support Vector Classifier (SVC), los hiperparámetros clave incluyen el tipo de kernel y el hiperparámetro de regularización (C). Si se opta por un kernel no lineal, como base radial, polinomial o sigmoidal, un hiperparámetro adicional, gamma, se vuelve relevante y requiere sintonización. Para abordar esto, se utiliza un enfoque sistemático empleando un diccionario de posibles valores para estos hiperparámetros junto con la función GridSearchCV, que realiza una búsqueda exhaustiva a través de las combinaciones de hiperparámetros especificadas, utilizando la validación cruzada para evaluar el desempeño de cada configuración. En este caso se utilizó *k-folds* con $k=5$, esto es para no aumentar de sobremanera el costo computacional manteniendo una relación adecuada entre entrenamiento y validación.

Adaboost, por otro lado, se centra en dos hiperparámetros: el número de estimadores y la tasa de aprendizaje. Estos determinan, respectivamente, el número máximo de iteraciones del proceso de boosting antes de que se detenga y la contribución de cada clasificador al modelo final.

TabNet, siendo un modelo más complejo con una estructura de red neuronal profunda diseñada específicamente para datos tabulares, ofrece un espectro más amplio de hiperparámetros para ajustar, reflejando su arquitectura y mecanismos de aprendizaje únicos. La tabla 2.1 presenta una lista detallada de estos hiperparámetros, incluyendo dimensiones de entrada y salida, número de pasos de decisión, factores de relajación, y más, todos los cuales influyen en cómo TabNet aprende de los datos y toma decisiones.

La sintonización exhaustiva de estos hiperparámetros a través de métodos como GridSearchCV para SVC y Adaboost, y un enfoque más personalizado para TabNet debido a su complejidad, culmina en la selección de configuraciones óptimas que mejoran el rendimiento de cada modelo. Los resultados de este proceso de ajuste se resumen en la tabla 3.4, donde se comparan las configuraciones finales y su efectividad.

Tabla 3. 4. Hiperparámetros sintonizados para los distintos clasificadores

Modelo	Hiperparámetro	Valor
SVC	Kernel	'rbf'
	Parámetro regularización (C)	100
	Coeficiente kernel (gamma)	0,1
AdaBoost	Numero estimadores (n_estimators)	400
	Tasa aprendizaje (learning_rate)	1,0
TabNet	Dimension de entrada(n_d)	128
	Dimension de salida (n_a)	32
	Numero de pasos(n_steps)	1
	Factor de relajación (gamma)	1,0
	Coeficiente de dispersión (lambda_sparse)	0,001

3.4.3. Entrenamiento de modelo

Con los hiperparámetros ya sintonizados, se procede a entrenar los modelos que es la última etapa antes de evaluar rendimiento. Se utilizará el set de prueba que no ha sido utilizado a lo largo del proceso. Este corresponde al 30% del set original asegurando que la evaluación del modelo se realice en muestras no vistas, lo que proporciona una medida fiable de su capacidad para generalizar a nuevos datos. En el capítulo siguiente se dedicará a un análisis detallado de las métricas de rendimiento y los errores asociados a cada clasificador. Esta evaluación permitirá determinar la eficacia de cada modelo en la tarea de clasificación, identificando sus fortalezas y debilidades en el contexto específico de la detección de fraude en servicios de salud.

4. Resultados

Para evaluar la calidad de los clasificadores es necesario analizar sus métricas con hiperparámetros sintonizados y no sintonizados.

4.1. SVC

Es esencial analizar el rendimiento del SVC con hiperparámetros tanto optimizados como no optimizados, además de examinar el error en cada división del conjunto de datos. Esto permitirá evaluar el comportamiento del modelo con los conjuntos de entrenamiento y prueba, identificando posibles casos de sobreajuste, subajuste o un desempeño adecuado.

En la figura 4.1, se presentan las métricas obtenidas, donde se observa una ligera mejora en todas las métricas. Este cambio marginal era esperado en este clasificador, dado el número limitado de hiperparámetros disponibles para ajustar sacrificando costo computacional permite obtener mejores resultados.

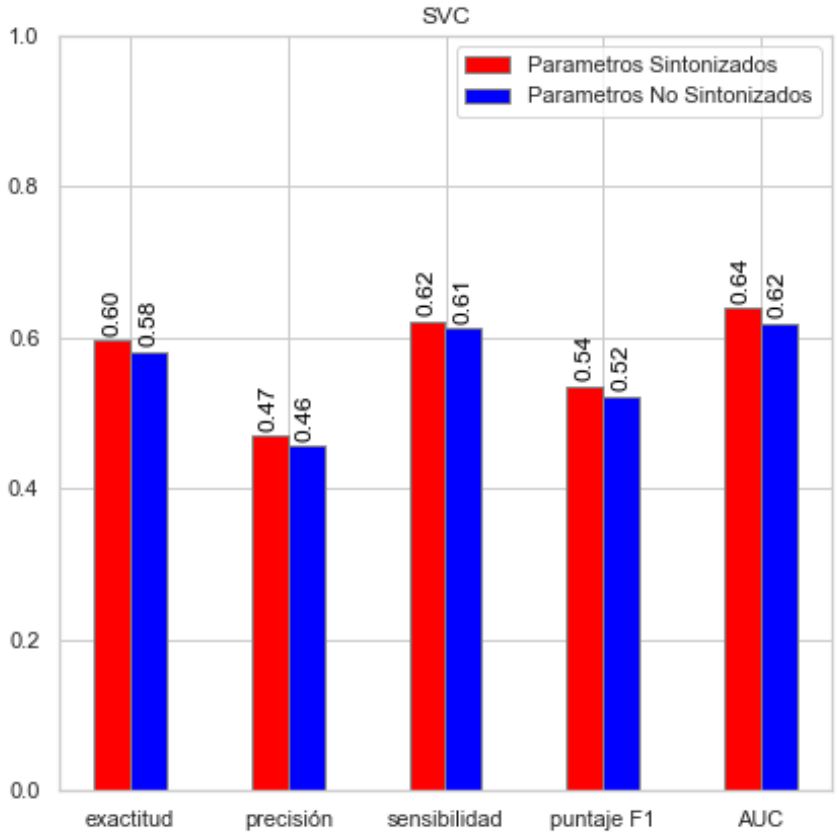


Figura 4. 1. Métricas de modelo SVC con hiperparámetros sintonizados y no sintonizados.

Al analizar el factor de perdida logarítmica en la figura 4.2, correspondiente a diferentes tamaños del conjunto de datos, se nota que el error tiende a estabilizarse a medida que aumenta el tamaño del conjunto. En los conjuntos de datos más pequeños, se observa una variabilidad mayor en la pérdida, lo que puede ser indicativo de sobreajuste. Este análisis subraya la importancia del tamaño del conjunto de datos en el entrenamiento del modelo y destaca la necesidad de equilibrar la complejidad del modelo y la adecuación de los datos para evitar el sobreajuste y asegurar un rendimiento robusto del modelo.

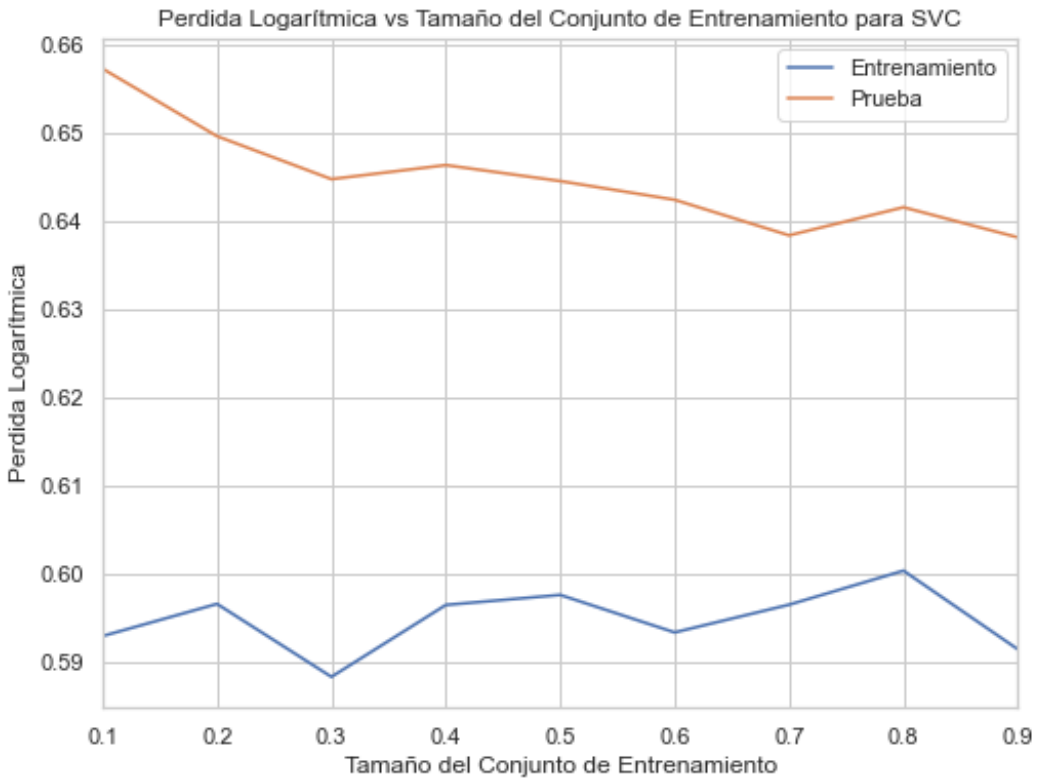


Figura 4. 2. Perdida versus tamaño del conjunto de entrenamiento en el clasificador SVC

4.2. AdaBoost

Al examinar las métricas presentadas en la figura 4.3, se destacan cambios significativos en las métricas de sensibilidad y puntaje f1 de 6 y 3 centésimas respectivamente. Estos cambios se atribuyen al mayor impacto que tienen los hiperparámetros en AdaBoost, lo cual induce variaciones más marcadas en su desempeño. Se observa un ligero cambio en AUC, mientras que el exactitud y precisión permanecen prácticamente inalterados.

AdaBoost muestra una particular susceptibilidad al sobreajuste, especialmente porque su

hiperparámetro más crucial, el número de aprendices débiles (estimadores), puede incrementarse para optimizar el rendimiento, lo cual incrementa la complejidad del modelo. Sin embargo, muchos estimadores hacen que el modelo se ajuste demasiado a los datos de entrenamiento, por lo que se tomó precaución para no seleccionar un número demasiado elevado de estos.

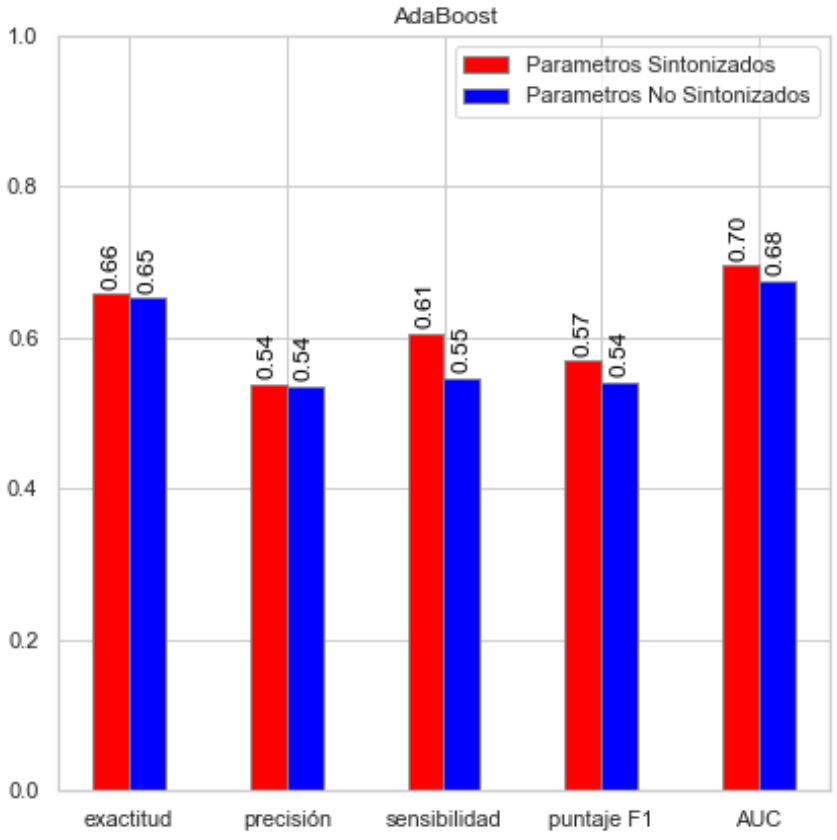


Figura 4. 3. Métricas de modelo AdaBoost con hiperparámetros sintonizados y no sintonizados

Al analizar el factor de perdida logarítmica en la figura 4.4 para diferentes tamaños del conjunto de datos, se observa que el error tiende a estabilizarse a medida que aumenta el tamaño del conjunto. Esto sugiere un ajuste adecuado del modelo al conjunto de datos. El punto óptimo parece ser al usar el 60% de los datos; más allá de este punto, el incremento en la cantidad de datos no mejora significativamente el desempeño del modelo, lo que lleva a una estabilización del error tanto para los datos de entrenamiento como para los de prueba. Esta estabilización del error indica que el modelo es capaz de generalizar bien y manejar nuevos datos de manera efectiva, minimizando el riesgo de sobreajuste.

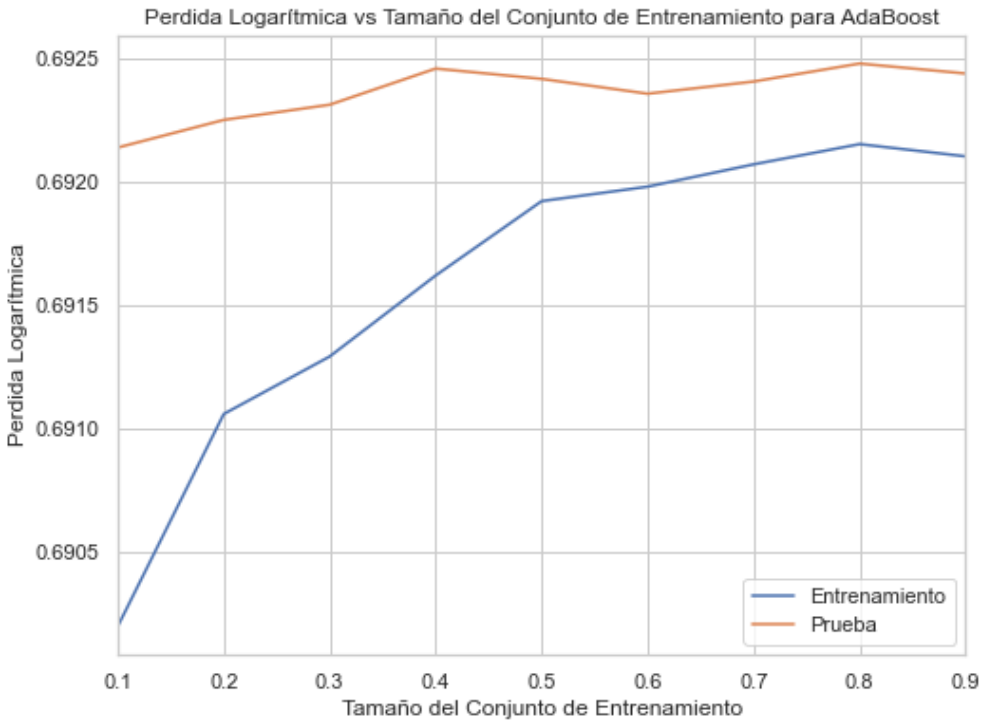


Figura 4. 4. Perdida versus tamaño del conjunto de entrenamiento en AdaBoost

4.3. TabNet

Al examinar las métricas presentadas en la figura 4.5, se observan mejoras significativas en todas las métricas evaluadas. Este resultado subraya la dependencia crítica de TabNet, un modelo de aprendizaje profundo, de sus hiperparámetros para un rendimiento óptimo. La sintonización de estos hiperparámetros es un proceso complejo y desafiante, dado el amplio abanico de posibles combinaciones y el elevado costo computacional que implica explorarlas todas exhaustivamente para encontrar la configuración óptima.

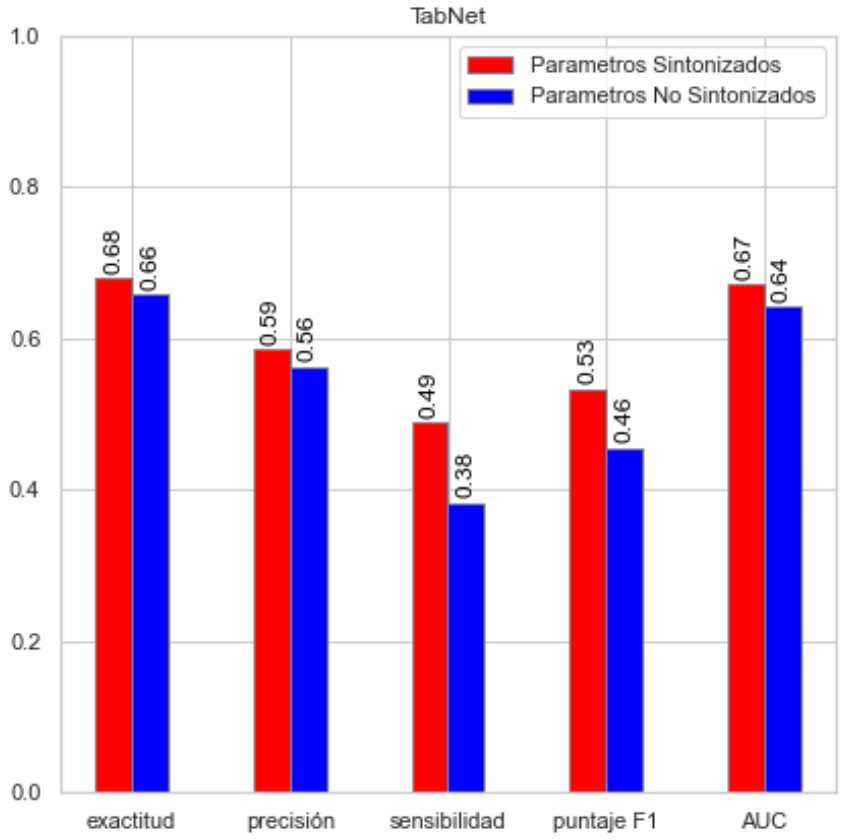


Figura 4. 5. Métricas de modelo de modelo TabNet con hiperparámetros sintonizados y no sintonizados.

En contraste con los modelos anteriores, el análisis del error en TabNet se realiza en función de las épocas de entrenamiento (epoch), en lugar de variar el tamaño del conjunto de datos. Una "época" se define como un ciclo completo de entrenamiento utilizando todos los datos disponibles en el conjunto. Según se detalla en la figura 4.6, el error a lo largo de las épocas muestra una convergencia rápida, estabilizándose después de la catorceava época. Esta estabilización es indicativa de un ajuste adecuado del modelo, evidenciado por la convergencia y estabilidad subsiguiente de las curvas de error. La estabilidad de estas curvas sugiere que el modelo es capaz de generalizar efectivamente, adaptándose a nuevos datos sin incurrir en sobreajuste.

La implementación de una estrategia de detención temprana ayuda a evitar el sobreajuste, deteniendo el entrenamiento cuando se ve que el modelo ya no mejora significativamente su desempeño en un conjunto de validación.

En resumen, el análisis de las métricas y el comportamiento del error a lo largo del entrenamiento para TabNet resalta la importancia de una cuidadosa selección y ajuste de hiperparámetros en modelos de aprendizaje profundo. Además, demuestra la efectividad de estrategias como la detención temprana

para mantener la capacidad de generalización del modelo, asegurando que este brinde un desempeño robusto y confiable.

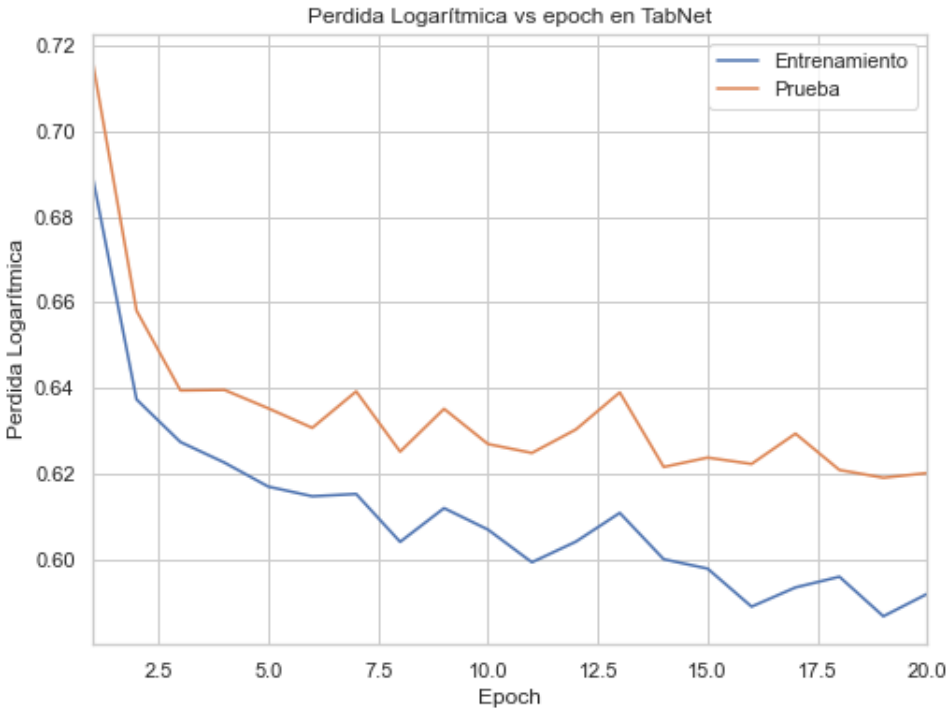


Figura 4. 6. Pérdida versus etapas de entrenamiento con hiperparámetros sintonizados

4.4. Resumen de resultados

La figura 4.7 muestra un análisis comparativo del rendimiento entre los tres modelos estudiados, destacando el desempeño de TabNet en términos de exactitud y precisión, con valores de 0.68 y 0.59, respectivamente. Por otro lado, AdaBoost resalta en las métricas de AUROC y el puntaje F1,con valires de 0,70 y 0,57 respectivamente. SVC por su parte, destaca en la métrica de sensibilidad, con un valor de 0.62. Esto sugiere un desempeño general equilibrado en los tres clasificadores. Como el objetivo principal es detectar fraudes, SVC emerge como el candidato más prometedor, lo que indica su habilidad para identificar correctamente los casos positivos de fraude. La tabla 4.1 resume los resultados, destacando en negrita los desempeños superiores. Este análisis subraya la importancia de seleccionar el modelo adecuado según el objetivo específico del proyecto. Aunque la precisión y la exactitud son importantes, en contextos donde la detección de casos positivos es crítica, la sensibilidad se convierte en una métrica clave.

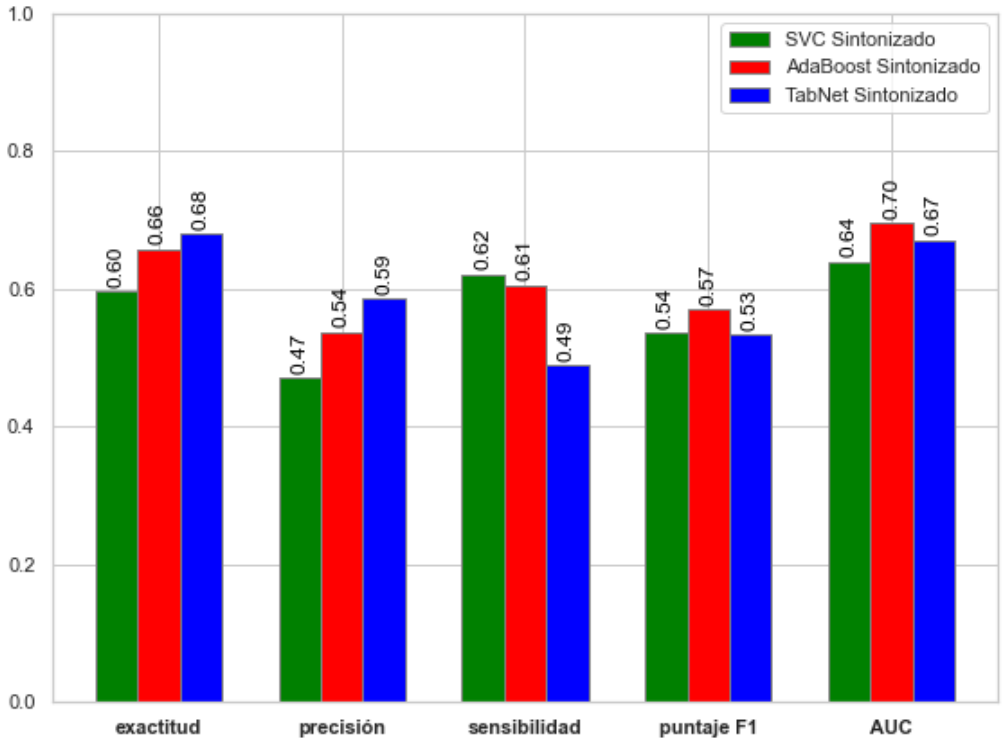


Figura 4. 7. Métricas de los tres modelos con hiperparámetros sintonizados

Tabla 4. 1. Métricas de los tres modelos con hiperparámetros sintonizados y no sintonizados.

Modelo	Exactitud	Precisión	Sensibilidad	Puntaje F1	AUC
SVC	0,58	0,46	0,61	0,52	0,62
SVC Sintonizado	0,6	0,47	**0,62**	0,54	0,64
AdaBoost	0,65	0,54	0,55	0,54	0,68
AdaBoost Sintonizado	0,66	0,54	0,61	**0,57**	**0,7**
TabNet	0,66	0,56	0,38	0,46	0,64
TabNet Sintonizado	**0,68**	**0,59**	0,49	0,53	0,67

5. Conclusiones

5.1. Discusión

5.1.1. Datos utilizados

El estudio se basó en datos de código abierto actualizados hasta el año 2009, derivado de las complejidades inherentes a la adquisición de datos más recientes de Medicare, tales como la necesidad de autorización y largos procesos de aprobación por parte de los proveedores de servicios de salud. La obtención de datos en Chile, donde el sistema de salud se divide entre servicios públicos y privados, enfrenta desafíos similares, incluyendo procedimientos de acreditación minuciosos, preocupaciones sobre la confidencialidad y privacidad de los pacientes, y la digitalización insuficiente de los registros médicos.

5.1.2. Resultados

La evaluación comparativa de los tres modelos reveló un desempeño generalmente homogéneo entre ellos. Esta constatación alinea con la literatura existente y la práctica común en proyectos de inteligencia artificial (IA), donde estos modelos son renombrados por su versatilidad y eficacia en una amplia gama de tareas de clasificación. Sin embargo, es fundamental destacar que el "mejor modelo" es una noción relativa, determinada intrínsecamente por las especificidades y requisitos únicos del problema en cuestión. En este estudio, enfocado en la detección de fraude en servicios médicos, el modelo SVM demostró ser excepcionalmente competente en identificar de manera precisa los casos de fraude, marcándolo como el candidato óptimo para esta aplicación específica.

La implementación de la validación cruzada en la sintonización de hiperparámetros jugó un papel pivotal en la optimización de la robustez de los modelos. Este método de evaluación riguroso asegura que los modelos no solo se ajusten a los datos con los que fueron entrenados, sino que también mantengan un alto grado de generalización cuando se enfrentan a datos nuevos y no vistos. Esta capacidad de generalización es crucial para el despliegue efectivo de sistemas de IA en entornos reales, donde la variabilidad y la incertidumbre de los datos son la norma.

5.2. Trabajos futuros

Durante el desarrollo de este proyecto, se identificaron varias áreas susceptibles de mejora o ampliación. En primer lugar, la adquisición de datos más recientes ya sea relacionados con Medicare o con información del Ministerio de Salud de Chile, potencia la relevancia de los resultados al abordar desafíos actuales en la detección de fraude. Además, la creación de un sistema en línea que permita detectar fraudes en tiempo real o en intervalos cortos puede representar un avance significativo en la

lucha contra esta problemática.

La aplicación de modelos más complejos, como las redes neuronales basadas en grafos o modelos embebidos, aprovechar los últimos avances tecnológicos y arquitecturas de inteligencia artificial. La creciente inversión de grandes empresas en IA subraya la importancia de mejorar la efectividad y optimización de estos modelos para obtener resultados superiores.

Otro aspecto interesante es realizar un análisis más detallado del conjunto de datos disponible. Los códigos de procedimiento y diagnóstico presentes en los datos ofrecen una oportunidad única para agrupar la información basándose en su similitud y desarrollar un nuevo modelo que se beneficie de estos insights. Este enfoque puede revelar patrones y conexiones previamente no detectados, contribuyendo a una comprensión más profunda del fraude en el ámbito de la salud y a la creación de soluciones más efectivas.

A pesar de los resultados prometedores obtenidos, es importante reconocer que siempre existe margen para la mejora mediante una sintonización más profunda y exhaustiva de los hiperparámetros. Sin embargo, esta tarea es notoriamente demandante desde el punto de vista computacional, requiriendo extensos recursos y tiempo. Es esencial considerar el equilibrio entre la complejidad del modelo y la interpretabilidad; modelos excesivamente complejos pueden ser difíciles de analizar y comprender, lo cual es un aspecto crítico en aplicaciones sensibles como la detección de fraude en servicios médicos.

Referencias

[1] V. Rawte and G. Anuradha, "Fraud detection in health insurance using data mining techniques," in *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, 15-17 Jan. 2015, pp. 1-5, doi: 10.1109/ICCICT.2015.7045689.

[2] J. Li, Q. Lan, E. Zhu, Y. Xu, and D. Zhu, "A study of health insurance fraud in China and recommendations for fraud detection and prevention," *Journal of Organizational and End User Computing (JOEUC)*, vol. 34, no. 4, pp. 1-19, 2022.

[3] M. Z. A. Mayaki and M. Riveill, *Multiple Inputs Neural Networks for Medicare fraud Detection*. 2022.

[4] M. Herland, T. M. Khoshgoftaar, and R. A. Bauder, "Big Data fraud detection using multiple medicare data sources," *Journal of Big Data*, vol. 5, no. 1, p. 29, 2018, doi: 10.1186/s40537-018-0138-3.

[5] N. Kurani, J. Ortaliza, E. Wager, L. Fox, and K. Amin, "How Has U.S. Spending on Healthcare Changed Over Time?," *Health Spending*, 2022. [Online]. Available: http://www.healthsystemtracker.org/chartcollection/u-s-spending-healthcare-changed-time.

[6] Y. Yoo, D. Shin, D. Han, S. Kyeong, and J. Shin, "Medicare fraud detection using graph neural networks," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 20-22 July 2022, pp. 1-5, doi: 10.1109/ICECET55527.2022.9872963.

[7] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *Journal of Big Data*, vol. 7, no. 1, p. 94, 2020, doi: 10.1186/s40537-020-00369-8.

[8] U. S. C. f. M. M. S. U.S. Government, "The Official U.S. Government Site for Medicare," 2020. [Online]. Available: https://www.medicare.gov/.

[9] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," *IEEE Access*, vol. 8, pp. 25579-25587, 2020, doi: 10.1109/ACCESS.2020.2971354.

[10] W. Yang, W. Hu, Y. Liu, Y. Huang, X. Liu, and S. Zhang, "Research on Bootstrapping Algorithm for Health Insurance Data Fraud Detection Based on Decision Tree," in *2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference*

on Intelligent Data and Security (IDS), 15-17 May 2021, pp. 57-62, doi: 10.1109/BigDataSecurityHPSCIDS52275.2021.00021.

[11] N. Agrawal and S. Panigrahi, "A Comparative Analysis of Fraud Detection in Healthcare using Data Balancing & Machine Learning Techniques," in *2023 International Conference on Communication, Circuits, and Systems (IC3S)*, 26-28 May 2023, pp. 1-4, doi: 10.1109/IC3S57698.2023.10169634.

[12] S. Lavanya, S. Manoj Kumar, and P. Mohan Kumar, "Machine Learning Based Approaches for Healthcare Fraud Detection: A Comparative Analysis," *Annals of the Romanian Society for Cell Biology*, pp. 8644–8654, 2021. [Online]. Available: http://annalsofrscb.ro/index.php/journal/article/view/2409.

[13] H. Shah, D. Pandya, K. Panchal, and N. P. More, "Classification of Machine and Deep learning Techniques for Financial Fraud Detection of Healthcare Industry," in *2022 International Conference on Futuristic Technologies (INCOFT)*, 2022, pp. 1-7, doi: 10.1109/INCOFT55651.2022.10094538.

[14] M. N. Ashtiani and B. Raahemi, "Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review," *IEEE Access*, vol. 10, pp. 72504-72525, 2022, doi: 10.1109/ACCESS.2021.3096799.

[15] R. Y. Gupta, S. S. Mudigonda, and P. K. Baruah, "A Comparative Study of Using Various Machine Learning and Deep Learning-Based Fraud Detection Models For Universal Health Coverage Schemes.," *International Journal of Engineering Trends and Technology*, vol. 69, no. 3, pp. 96-102, 2021.

[16] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized Mutual Information Feature Selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189-201, 2009, doi: 10.1109/TNN.2008.2005601.

[17] D. Berrar, "Cross-validation," ed, 2019.

[18] Wikipedia. "Validación Cruzada." https:// es.wikipedia.org/wiki/Validación_cruzada (accessed 02-02-2024.

[19] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18-28, 1998, doi: 10.1109/5254.708428.

[20] S.-l. developers, "sklearn.svm.SVC," Web Page 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.

[21] S. Ö. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pp. 6679-6687, 2021, doi: 10.1609/aaai.v35i8.16826.

[22] J. Bughin, J. Seong, J. Manyika, M. Chui, and R. Joshi, "Notes from the AI frontier: Modeling the impact of AI on the world economy," *McKinsey Global Institute*, vol. 4, 2018.

[23] T. K. An and M. H. Kim, "A New Diverse AdaBoost Classifier," in *2010 International Conference on Artificial Intelligence and Computational Intelligence*, 23-24 Oct. 2010 2010, vol. 1, pp. 359-363, doi: 10.1109/AICI.2010.82.

[24] N. Altman and M. Krzywinski, "The curse(s) of dimensionality," *Nature Methods*, vol. 15, no. 6, pp. 399-400, 2018/06/01, doi: 10.1038/s41592-018-0019-x.

[25] R. A. Gupta. "Kaggle Healthcare Provider Fraud detection Datasets." http://www.kaggle.com/datasets/rohitrox/ healthcare-provider-fraud-detection-analysis (accessed 29-08-2023).

[26] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105524, 2020, doi: https://doi.org/10.1016/j.asoc.2019.105524.

[27] A. Alwosheel, S. van Cranenburgh, and C. G. Chorus, "Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis," *Journal of Choice Modelling*, vol. 28, pp. 167-182, 2018, doi: https://doi.org/10.1016/j.jocm.2018.07.002.

UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA
RESUMEN DE MEMORIA DE TITULO

Departamento	**:** Departamento de Ingeniería Eléctrica
Carrera	: Ingeniería civil electrónica
Nombre del memorista	: Victor Ricardo Contreras Valderrama
Título de la memoria	**:** Implementación de algoritmos de ia para la detección de fraude en servicios de salud
Fecha de la presentación oral	: 28/03/2024
Profesor(es) guía	: Rosa Figueroa Iturrieta
Profesor(es) revisor(es)	: Daniel Sbárbaro Hofer y Mario Medina Carrasco
Concepto	**:**
Calificación	**:**

Resumen (máximo 200 palabras)

El fraude en la salud es un tema muy relevante hoy. El fraude en seguros médicos en Estados Unidos (Medicare) causa pérdidas superiores a los miles de millones de dólares por año.

Esta memoria de título está enfocada en atacar este problema utilizando datos tabulares de código abierto que contienen la información de pacientes registrados en el programa federal de seguro médico Medicare con el proveedor de salud respectivo asignado a cada paciente. En total se utilizó información de 558.211 pacientes. Primero, se realizó un análisis exploratorio de los datos para procesarlos seguido de una ingeniería de características. Posteriormente se hizo una sintonización de hiperparámetros con validación cruzada para asegurar la robustez de los parámetros seleccionados. Paralelamente, se evaluó el error de cada clasificador para monitorear el rendimiento de los modelos. Finalmente, se entrenaron y evaluaron tres clasificadores (AdaBoost, Support Vector Machine (SVM) y TabNet), con el objetivo de clasificar los ejemplos del set de datos en las clases “fraude” y “no fraude”. En términos de sensibilidad, el mejor clasificador fue SVM (sensibilidad=0.62), seguido por AdaBoost (sensibilidad = 0.61). Por otro lado, en términos de área bajo la curva de operación, el clasificador que presento los mejores resultados fue AdaBoost con un puntaje de 0.7.