Máster en Dirección Aseguradora Profesional

Curso académico 2022-2023

Memoria Fin de Máster

Beneficios de la inteligencia artificial y el aprendizaje automático en la detección de fraude en el seguro

Autor: PhD Eduardo Ramos Pérez

Tutor: Fernando León de Santos Fernández

Firmado: Eduardo Ramos Pérez

Índice

1. OBJETIVOS

La misión de esta memoria es analizar los beneficios de la aplicación de inteligencia artificial y aprendizaje automático en la identificación del fraude en el seguro. Debido al potencial que han demostrado estos algoritmos gracias a la creciente capacidad computacional, estos son ampliamente utilizados en ámbitos tan diferentes como puedan ser la diagnosis médica (Wu et al. 2018), conducción autónoma (Howard et al. 2019), procesamiento del lenguaje natural (Devlin et al. 2018 y Brown et al. 2020) e instituciones financieras (Dixon et al. 2020, Hastie et al. 2009 y Ramos-Pérez et al. 2021). Para la consecución de la misión de esta memoria, se establecen los siguientes objetivos secundarios (Gráfico 1):

- Estimar un modelo de detección de fraude, basado en algoritmos del ámbito de la inteligencia artificial y el aprendizaje automático, que sea estadísticamente más preciso que las reglas automáticas tradicionales. Para ello será necesario adaptar los algoritmos de inteligencia artificial y aprendizaje automático con un mayor poder predictivo para poder ser usados dentro del marco de la identificación del fraude en el seguro. Dentro del marco de la inteligencia artificial se puede diferenciar tres diferentes tipos de aprendizaje: supervisado, no supervisado y por refuerzo. En la aplicación a tratar en esta memoria se utilizarán las técnicas y algoritmos pertenecientes al aprendizaje supervisado ya que, debido a las características del problema y de las bases de datos, se dispone de un histórico de siniestros con las características de los mismos y un indicador de si fue fraude o no. Los algoritmos más ampliamente utilizados dentro de este aprendizaje debido a su capacidad predictiva son las redes neuronales (Mcculloch and Pitts 1943), gradient boosting con árboles de clasificación o regresión (Freund and Schapire 1997, Friedman 2000 y Chen and Guestrin 2016) y bosques aleatorios (Breiman 2001). Estos algoritmos serán utilizados para construir un modelo que, en función de las características del siniestro, permita evaluar la probabilidad de que se trate de un fraude.

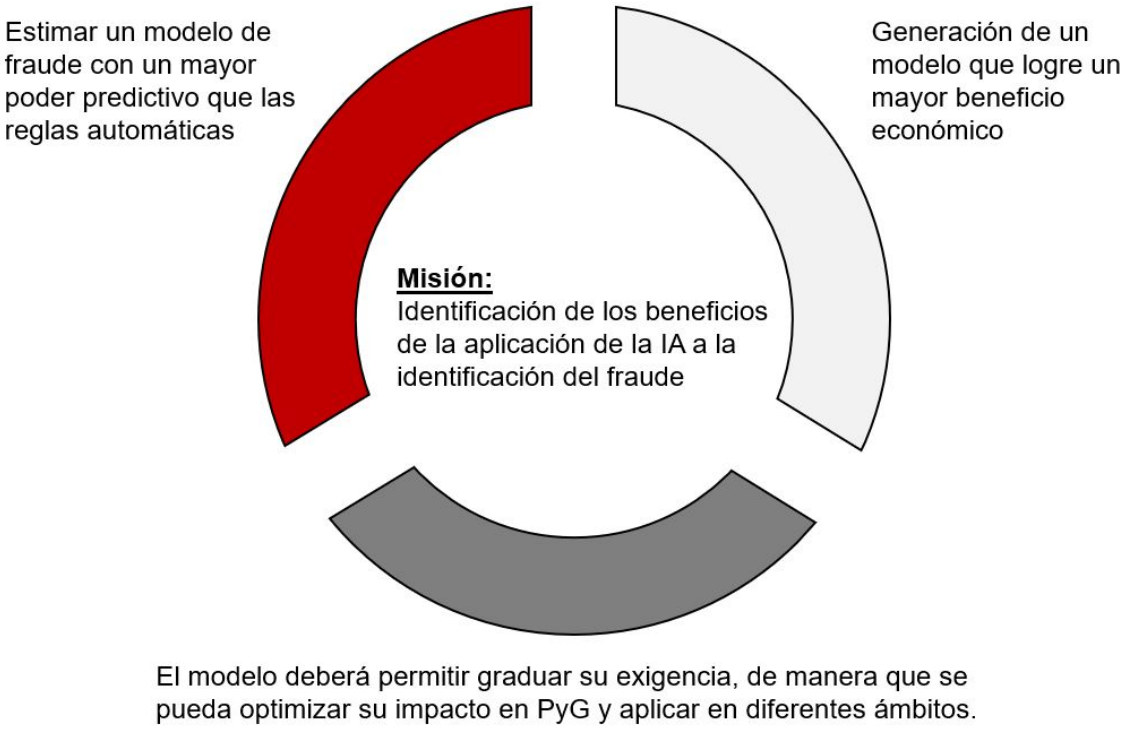
  Es relevante mencionar que la identificación del fraude, dentro del ámbito estadístico, es un problema de clasificación con datos desbalanceados. Esto significa que, hay muchos más siniestros que no son fraude en comparación con aquellos que si lo son. Por tanto, si se quieren rehusar casi todos los fraudes, se identificarán como tal muchos siniestros que en realidad no lo son, generando insatisfacción en los clientes. Por tanto, se aplicarán técnicas estadísticas que permitan que los algoritmos reconozcan que se encuentran ante un problema con datos desbalanceados, mejorando así su capacidad predictiva y minimizando el problema anteriormente mencionado.

- Generación de un modelo que permita graduar su exigencia en la detección de fraude. De esta manera, se podrá optimizar el impacto de este en la cuenta de resultados y permitir su uso en diferentes ámbitos de la entidad aseguradora. Para ello, el modelo que se presentará en esta memoria deberá estimar la probabili-

dad de fraude de cada siniestro. Regulando el umbral de probabilidad a partir del cual un siniestro de considera fraudulento se podrá optimizar el uso del modelo en función de la variable que más interese a la compañía de seguros.

- Generar un modelo de fraude que, gracias a la capacidad predictiva del aprendizaje automático, logre un beneficio económico mayor que los métodos tradicionales. Se analizará tanto el porcentaje de siniestros que habría que revisar bajo ambos enfoques, como la precisión de los mismos identificando siniestros que realmente son un fraude. Ambas dimensiones son significativas. La primera nos indica el esfuerzo que deberá realizar el equipo de fraude y, por tanto, el gasto de administración que la empresa deberá asumir. La segunda, nos indicará el potencial ahorro en coste siniestral que la empresa tendrá gracias a una correcta gestión e identificación del fraude. Ambas dimensiones tienen un impacto directo en la competitividad de la empresa, tanto mejorando márgenes como permitiendo una tarificación y selección de riesgos más precisa.

Figura 1: Misión y objetivos secundarios



Fuente: Elaboración propia

2. INTRODUCCIÓN

2.1. El fraude en la industria del seguro

Se entiende por fraude en el ámbito asegurador aquellos casos en los que el asegurado, perjudicado o beneficiario de una prestación, oculta información o engaña a la entidad aseguradora con la intención última de obtener un beneficio de un siniestro encuadrado dentro de un contrato de seguro.

Por su tipología, el fraude que sufren las entidades aseguradoras españolas puede agruparse en tres grandes familias (AXA 2023). El fraude ocasional es la familia o tipología de fraude más comúnmente extendida en el mercado asegurador. Se tratan de aquellos clientes que, aprovechando la ocurrencia de un siniestro, reclaman la reparación de daños preexistentes o magnifican las consecuencias del mismo con el fin de obtener un beneficio o indemnización extraordinario. La frecuencia de este tipo de fraude aumenta durante los periodos de desaceleración económica mientras que su coste medio reduce. En segundo lugar, se encuentra la familia del fraude premeditado. En esta tipología, los daños que se reclaman a la entidad aseguradora son fruto de una acción predeterminada cuyo objetivo último es la obtención de una indemnización por parte de la aseguradora. Por último, se encuentra la familia del fraude organizado. En este caso se trata de bandas cuyo objetivo último es industrializar el fraude asegurador.

Tal y como se indica en ICEA (2023) (informe basado en 36 compañías que representan un 58.7 % de la cuota de mercado), el fraude en el sector asegurador español ha ido creciendo en los últimos años. El Cuadro 1 muestra la evolución del impacto del fraude en el coste siniestral de Autos, Multirriesgo Hogar y Responsabilidad Civil. Se muestra que, a pesar del ruido que pudiera generar en la serie el Covid-19, el peso del fraude en el coste siniestral es mayor en 2022 que en 2017 para todos los ramos.

Cuadro 1: Impacto del fraude en el coste de los siniestros

Ramo	2017	2018	2019	2020	2021	2022
Autos RC Daños Corporales	6.35 %	6.99 %	7.77 %	7.32 %	9.88 %	10.22 %
Autos RC Daños Materiales	2.41 %	2.90 %	2.60 %	2.26 %	2.72 %	3.16 %
Autos Robo	6.15 %	7.54 %	7.08 %	9.76 %	10.87 %	6.98 %
Autos Resto de Garantías	1.09 %	1.66 %	1.40 %	1.60 %	1.38 %	1.50 %
Multirriesgo Hogar	2.02 %	2.22 %	2.09 %	2.00 %	2.17 %	2.96 %
Responsabilidad Civil	8.40 %	9.42 %	8.71 %	4.42 %	10.02 %	13.30 %

Fuente: ICEA (2023)

Este mismo informe muestra que, en términos de coste, el ramo de Automóviles representa un 66.51 % de todo el fraude cometido en el sector asegurador, seguido por Diversos (PYME, Comercio, Hogar, Comunidades) y Responsabilidad Civil con un 19.56 %. Por tanto, la mayoría del fraude se comete en No Vida, siendo un 11.56 % el peso de Vi-

da, Accidentes y Salud. Las ratios de fraude evitado fluctúan ampliamente entre ramos. A pesar de que automóviles es el ramo que más peso tiene en el fraude, su ratio de fraude evitado es uno de los menores con 52.90 %. En Diversos y Responsabilidad Civil se evita más del 80 % del fraude mientras que en Vida, Accidentes y Salud ese porcentaje asciende a más de un 85 %.

Todos estos datos muestran la relevancia de la detección del fraude para la mejora de las ratios de siniestralidad y selección de riesgos. Por ello, las compañías aseguradoras siguen realizando esfuerzos e invirtiendo con el objetivo mejorar la detección del fraude. Esto se ve adicionalmente refrendando por el rendimiento de las inversiones en fraude que se muestran en el Cuadro 2.

Cuadro 2: Rendimiento del gasto en detección de fraude

Ramo	Importe medio del fraude evitado	Gasto medio de investigación	Ratio de Rendimiento
Automóviles	1,872.8€	61.3€	30.5€
Diversos y RC General	2,252.5€	45.7€	49.3€
Vida, Accidentes y Salud	8,294.0€	52.9€	156.6€
Otros Ramos	7,011.4€	159.4€	44.0€

Fuente: ICEA (2023)

Con independencia del ramo que se trate, se observa que el gasto medio de investigación se encuentra muy por debajo del importe medio de los fraudes evitados. A pesar de estos resultados positivos en términos de rentabilidad, la detección del fraude es una tarea compleja debido a la variedad que existe términos de tipología.

En lo relativo al tipo de fraude, El Cuadro 3 muestra que la tipología de fraude más común depende mucho del ramo. Adicionalmente, se observa que no hay una tipología de fraude dominante, siendo la excepción el negocio de Vida, Accidentes y Salud dónde las lesiones preexistentes suponen más de un 64 % del fraude total. En el caso del ramo de Automóviles, que tal y como se ha comentado previamente representa el 66.51 % del coste fraudulento en el negocio asegurador, los principales tipos de fraude son la presencia de lesiones preexistentes, la petición de una indemnización desproporcionada o por una cobertura excluida.

En el caso del ramo de Diversos y RC general, hay aún más diversidad que en Automóviles en lo relativo a las principales causas de fraude, habiendo 6 tipos de fraude por encima del 10 %. Adicionalmente a las tipologías ya mencionadas para Automóviles, cabe destacar la alta frecuencia que representan el fraude en la suscripción y la simulación del siniestro en el ramo de Diversos y RC general. Esto se debe a que las características de los riesgos asegurados y tomadores es muy diferente. Al tratarse de un seguro con un carácter no tan minorista como el de Automóviles y, por tanto, con una gran variedad de riesgos asegurados dependiendo del sector de actividad económica, las posibilidades que tiene el tomador para simular un siniestro o engañar en el proceso

de suscripción son mayores.

Cuadro 3: Tipos de fraude

Tipología	Automóviles	Diversos y RC General	Vida, Accidentes y Salud	Otros Ramos
Fraude Suscripción	0.33 %	17.11 %	9.92 %	13.12 %
Siniestro simulado	13.22 %	23.64 %	8.21 %	26.05 %
Exclusión cobertura	19.07 %	14.55 %	5.62 %	16.53 %
Lesión preexistente	26.17 %	14.64 %	64.37 %	4.74 %
Falsedad documental	7.25 %	3.99 %	5.85 %	15.49 %
Desproporción	26.63 %	10.92 %	0.66 %	18.62 %
Falta nexo causal	6.78 %	3.37 %	1.46 %	2.70 %
Otros	0.55 %	11.78 %	3.91 %	2.75 %

Fuente: ICEA (2023)

El Cuadro 4 muestra que la variedad observada en la tipología no se da en el caso del defraudador directo. Con independencia del ramo de aseguramiento, el defraudador directo más común es el propio asegurado. Cabe esperar que esto sea así ya que, como se mostraba en El Cuadro 3, algunas de las tipologías de fraude más comunes con aquellas en las que es necesaria una participación activa del tomador o asegurado como, por ejemplo, el fraude en la suscripción, la simulación de un siniestro o la exclusión de cobertura.

El segundo defraudador directo más común es el contrario, el cual tiene un peso especialmente elevado en el ramo de Automóviles. En este caso, el contrario puede pedir una indemnización desproporcionada por los daños o lesiones sufridas o incluso ocultar una lesión preexistente con el objetivo de lograr una mayor compensación.

Cuadro 4: Defraudadores directos

Ramo	Asegurado	Contrario	Mediador	Reparador	Otros
Automóviles	70.90 %	22.99 %	0.07 %	4.21 %	1.83 %
Diversos y RC General	87.67 %	7.98 %	1.66 %	1.01 %	1.68 %
Vida, Accidentes y Salud	93.69 %	3.94 %	0.70 %	0.89 %	0.79 %
Otros Ramos	69.81 %	15.40 %	6.31 %	3.88 %	4.59 %

Fuente: ICEA (2023)

Resumiendo todo lo comentado en esta sección, se podría decir que el fraude en el sector del seguro español está fuertemente dominado por Automóviles. Este ramo representa más de un 65 % de todo el coste fraudulento del sector. Adicionalmente, hay una gran variedad en la tipología de fraude pero, sin embargo, el defraudador principal es casi siempre el asegurado y el contrario. Las tasas de fraude suelen estar fuertemente influenciadas por el contexto socioeconómico, provocando un aumento de las tasas de fraude entre 2017 y 2022. Por último, es relevante remarcar la alta rentabilidad (más de

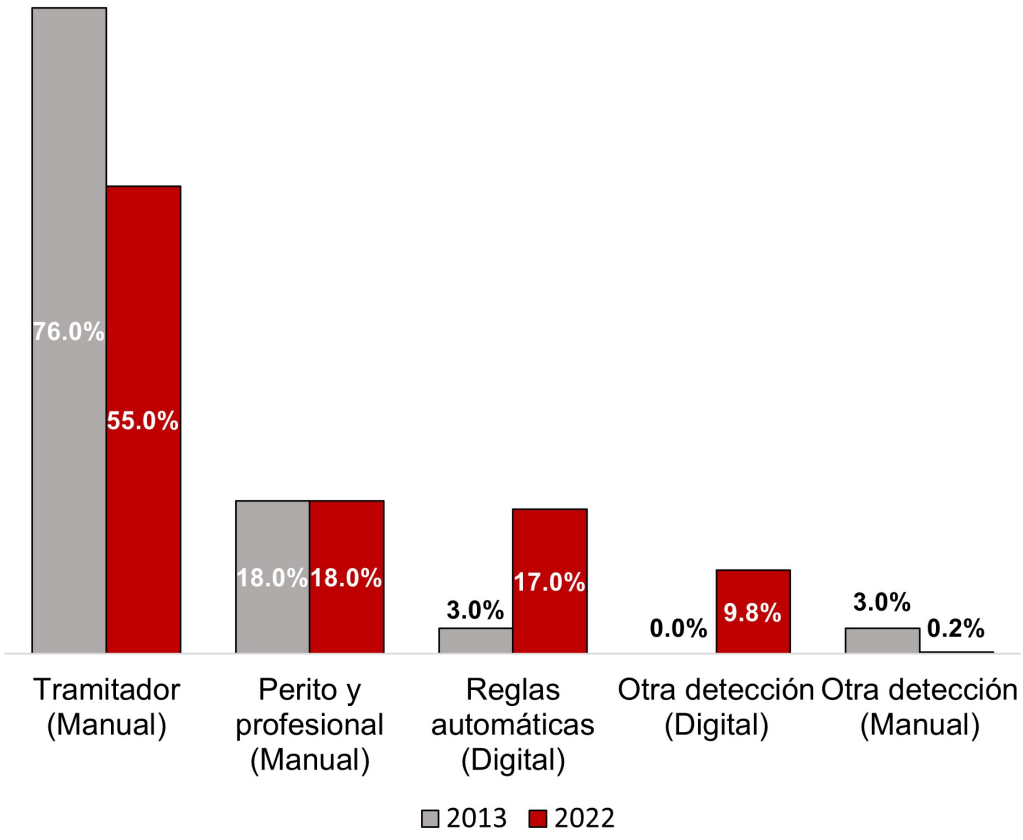
30 euros con independencia del ramo) que las compañías aseguradoras obtienen por cada euro destinado a investigar fraude.

2.2. Medios de detección del fraude

Los medios de detección del fraude han evolucionado significativamente en los últimos años tal y demuestra AXA (2023). En el año 2013, sólo el 3 % de la detección del fraude de AXA era digital (a través de reglas automáticas), siendo el peso de la detección manual del 97 %. Tal y como se puede observar en la Figura 2, el 76 % del fraude era detectado manualmente a través de mediadores y el 18 % mediante peritos.

En 2022, los tramitadores y peritos siguen siendo la principal forma de detección de fraude. Sin embargo, su peso relativo sobre el total del fraude detectado ha bajado del 94 % en 2013, al 73 % en 2022. Esta tendencia a la baja se debe al incremento de la detección digital que en 2022 asciende a un 27 %, mientras que en 2013 apenas alcanzaba un peso del 3 % sobre el total de métodos de detección del fraude.

Figura 2: Evolución de los medios de detección del fraude



Fuente: AXA 2023. X Mapa del fraude en España

En línea con los datos mostrados en la Figura 2, el aumento en la detección digital en AXA se debe al desarrollo de reglas automáticas más avanzadas. Merece también la pena destacar la contribución positiva de otros métodos digitales de detección del fraude como la ciencia de datos internos y externos, cuyo peso en 2022 es relevante mientras que en 2013 era inexistente.

Esta tendencia mostrada para AXA, aunque con diferentes magnitudes, es aplicable al resto de mercado asegurador español. Desde 2013 hasta el día de hoy, la capacidad para gestionar, almacenar y analizar datos ha crecido significativamente. Esto ha permitido que los métodos automáticos hayan aumentado su capacidad para detectar fraude, tomando cada vez una mayor relevancia en comparación con los métodos de detección manuales.

2.3. La inteligencia artificial y su aplicación en la detección de fraude

La mayoría de los algoritmos más ampliamente utilizados en aplicaciones de IA como las redes neuronales, arboles aleatorios o gradient boosting fueron desarrollados mucho antes del auge de popularidad que este campo está experimentando en los últimos años. Esto también ocurre con los principales métodos utilizados para la optimización de estos modelos: back-propagation (Rumelhart et al. 1986 y Rumelhart and Zipser 1986), R-prop (Riedmiller and Braun 1993), iRprop (Igel and Hüsken 2003), RMSpro (Zeiler 2012) o ADAM (Kingma and Ba 2014). De hecho, hasta las principales estructuras de IA en las que se basa el conocido Chat GPT, llamados transformer, fueron desarrolladas mucho antes de este auge por (Vaswani et al. 2017).

Aunque la base teórica de estos modelos quedó definida hace ya años, su implementación y llegada al gran público es algo que ha sucedido gradualmente debido a las siguientes razones:

- Para explotar toda la potencia de estos algoritmos, es necesario entrenarlos con una gran cantidad de datos y variables. El sector privado ha ido adaptando sus sistemas informáticos y aumentando gradualmente su capacidad para recolectar y almacenar datos de sus clientes y operaciones.
- El entrenamiento de este tipo de algoritmos es computacionalmente costoso debido a la gran cantidad de información que necesitan para ser entrenados y a la complejidad de sus estructuras. A partir de 2008, estos algoritmos pueden ser entrenados en la GPU, acelerando su proceso de entrenamiento y haciendo posible la estimación de algoritmos con un mayor número de parámetros.

A pesar de la reciente atención y auge de la IA, este es un camino que se seguirá recorriendo en los próximos años e incluso décadas. En el futuro, la capacidad de computación y la cantidad de datos que podremos recolectar serán aún mayores. Adicionalmente, la investigación está trabajando de manera continua en encontrar maneras más

eficientes computacionalmente para optimizar estos algoritmos. Por tanto, todas las herramientas de IA tienen por delante un largo recorrido por caminar y mucho campo de mejora. A pesar de ello, estos algoritmos ya han provocado cambios significativos en la detección del fraude:

- Reducción del trabajo manual. Tal y como se ha mencionado anteriormente, la detección manual del fraude ha ido reduciendo su peso significativamente en los últimos años en favor de la detección automática. Las herramientas de IA y aprendizaje automático han jugado un rol fundamental en este proceso gracias a su capacidad para gestionar una cantidad de datos y realizar análisis complejos. Esta reducción del trabajo manual ha permitido aumentar la eficiencia de los analistas de fraude y ahorrar costes a las entidades aseguradoras. Adicionalmente, tener algoritmos de IA o aprendizaje automático capaces de identificar siniestros sospechosos como el mejor de los tramitadores o peritos, permite mejorar significativamente las capacidades de aquellos que tienen una menor experiencia y reducir el esfuerzo de investigación.

- Mejora de la experiencia del cliente y distribuidor. Las herramientas de aprendizaje automático han mejorado la precisión de los modelos de detección de fraude. Esto significa que hay un menor número de casos que serán marcados erróneamente como fraude. En línea con lo comentado en la Sección 2.1, la mayoría del fraude es cometido por el propio asegurado. Por tanto, un modelo de fraude impreciso tendrá un impacto muy negativo en la experiencia que tengan clientes y distribuidores con la compañía aseguradora.

- Prevención del fraude y gestión del riesgo. Los modelos basados en aprendizaje automático o inteligencia artificial pueden ayudar no solo a detección del fraude, sino también a su prevención. Al ser entrenados con datos relativos al perfil del cliente y los bienes asegurados, los modelos de detección de fraude basados en IA pueden ser utilizados para saber qué perfil de cliente es más propenso a cometer fraude. Incluir esta información en el proceso de suscripción ayuda a la gestión del riesgo y a la prevención del fraude.

En el ámbito académico y corporativo se han desarrollado numerosos enfoques basados en inteligencia artificial y aprendizaje automático para la detección del fraude en las instituciones financiera. Con el objetivo de detectar el fraude en transacciones realizadas con tarjetas de crédito, Fu et al. (2016) propuso un modelo basado en redes neuronales convolucionales (CNN, por sus siglas en inglés). Este algoritmo, generalmente utilizado en el campo del reconocimiento de imágenes, también fue utilizado por Zhang et al. (2018) con el objetivo de detectar transacciones online fraudulentas. Las CNNs también han sido recientemente utilizadas en el campo del fraude en el seguro de automóviles por Xia et al. (2022). Estos autores proponen combinar las CNNs con 'long-short term memory' (LSTM), que se trata de un tipo de red neuronal ampliamente utilizada para tratar con series temporales. Las redes neuronales feed-forward o prealimentadas también han sido ampliamente utilizadas para la detección del fraude bancario (Bouchti et al. 2017) y

con tarjetas de crédito (Ghobadi and Rohani 2016 y Srivastava et al. 2016).

Adicionalmente a las redes neuronales y sus diversas variantes, los algoritmos basados en árboles de decisión han sido también ampliamente utilizados para la detección del fraude. Este enfoque es especialmente preciso tratando con datos estructurados, que son la tipología de la que normalmente se dispone en el ámbito financiero. De hecho, estudios comparativos como los realizados por Elsayed et al. (2021) y Borisov et al. (2022) demuestran que este tipo de algoritmos basados en árboles de regresión son capaces de superar el rendimiento de redes neuronales de aprendizaje profundo en problemas con datos estructurados. Zouboulidis and Kotsiantis (2012) aplicó arboles aleatorios para la detección del fraude en los estados financieros de compañías griegas. Randhawa et al. (2018) y Tongesai et al. (2022) también aplicaron algoritmos basados en árboles de decisión para la identificación del fraude. En la primera de las investigaciones anteriormente nombradas, los autores utilizaron Adaboost para la detección del fraude en tarjetas de crédito. Por otro lado, el segundo estudio científico se centró en aplicar XGBoost para la detección del fraude en el sector asegurador.

3. DETECCIÓN DE FRAUDE BASADO EN IA Y APRENDIZAJE AUTOMÁTICO

El modelo de detección de fraude que se presenta en esta memoria es una regla automática basada en inteligencia artificial y aprendizaje automático. Por tanto, el objetivo es generar una herramienta basada en este tipo de algoritmos que sea capaz de indicar a la compañía aseguradora la probabilidad de que un siniestro sea fraudulento y que mejore la precisión de un sistema de reglas automáticas más tradicional. Para ello, los algoritmos utilizados tomarán como referencia una serie de características del siniestro y del asegurado. Tal y como se ha puesto de manifiesto en la sección 2.1, en la mayoría de las ocasiones el defraudador es el propio asegurado y, por tanto, las características del mismo son clave para la precisión del modelo. La base de datos utilizada es de dominio público y será analizada a lo largo de esta sección.

A lo largo de las diferentes subsecciones que componen esta sección se explicará, en primer lugar (subsección 3.1), los fundamentos técnicos de los algoritmos de inteligencia artificial y aprendizaje automático que se aplicarán en el modelo de detección del fraude que se propone en esta memoria. La arquitectura del modelo que se propone será explicada en la subsección 3.2. Por último, la base de datos utilizada, los modelos de reglas automáticas utilizados como referencia para la comparativa contra los modelos basados en aprendizaje automático y los resultados empíricos serán expuestos en la subsección 3.3.

3.1. Fundamentos de las principales técnicas aplicadas

En esta subsección de realizará una breve introducción de los diferentes algoritmos que serán aplicados en el modelo de predicción del fraude que se propone es esta memoria.

3.1.1. Redes neuronales feedforward

Este algoritmo está inspirado en la biología humana (Mcculloch and Pitts 1943) y sus principales componentes son:

- Neuronas. Tal y como queda representado por los círculos rojos de la Figura 3, las redes neuronales feedforward están compuestas por numerosas neuronas organizadas en capas. Las neuronas cuyo input no viene de ninguna otra neurona se denominan neuronas de entrada. Estas son las encargadas de realizar una primera interpretación de los datos que alimentan a la red. Aquellas que a continuación no tienen ninguna otra neurona se consideran neuronas de salida. El resultado de esta última clase de neurona determina la predicción realizada por la red neuronal.
- Pesos y conexiones. Éstas últimas quedan representadas en la Figura 3 por las líneas que conectan las diferentes neuronas y son las encargadas de transferir

la información entre ellas. En el caso de las redes neuronales feedforward, la información recibida por la neurona i de la capa j es el resultado de la siguiente combinación lineal:

$$X_{i,j} = \sum_{i=1}^{I} w_i x_i + w_0 \tag{1}$$

Siendo x_i el resultado de la neurona i, w_i el peso asociado a x_i y w_0 el término constante asociado a la combinación lineal de conjunto de las neuronas y pesos. Con el objetivo de mantener la simplicidad de la Figura 3, en ella no se han representado los pesos y sus combinaciones lineales. La estimación de los pesos de las redes neuronales es normalmente realizada a través del algoritmo de backpropagation (Rumelhart et al. 1986 y Rumelhart and Zipser 1986) en combinación de algún método de optimización de pesos, como RMSpro (Zeiler 2012) o ADAM (Kingma and Ba 2014).

- Funciones de activación. Como se acaba de indicar en el punto anterior, las neuronas reciben una combinación lineal de los resultados procedentes de las neuronas predecesoras (denominado como $X_{i,j}$). Antes de que esta combinación lineal sea liberada por la siguiente neurona, se debe aplicar una función de activación a $X_{i,j}$. La función sigmoidal y ReLU son las activaciones más comúnmente utilizadas para problemas de clasificación y regresión respectivamente. La función sigmoidal tiene la siguiente expresión:

$$h(X_{i,j}) = \frac{1}{1 + \exp(-X_{i,j})} \tag{2}$$

Mientras que la activación ReLU es:

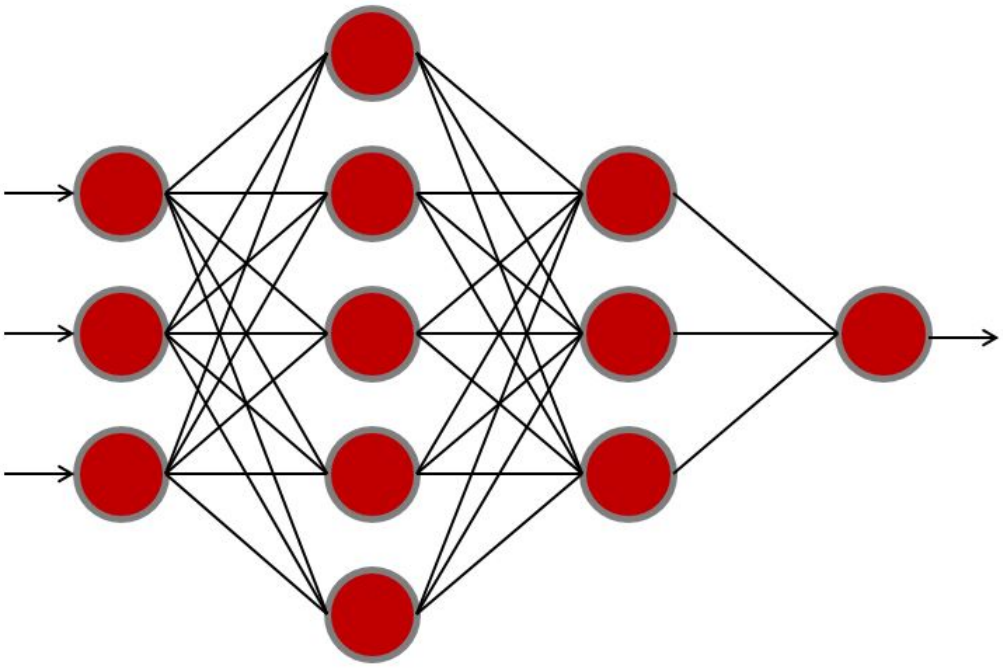
$$h(X_{i,j}) = max(0, X_{i,j}) \tag{3}$$

Por tanto, el resultado de una neurona sería $h\big(\sum_{i=1}^{I} w_i x_i + w_0\big)$. Aunque la expresión que define el resultado de una neurona es sencilla, la notación matemática de una red neuronal suele ser compleja debido al gran número de capas y neuronas que la compone. Siguiendo la notación propuesta por Bishop (2006), la expresión matemática de una red neuronal compuesta por dos capas sería la siguiente:

$$\hat{f}(X) = h^{(3)}\left(\sum_{k=1}^{T} w_{p,k}^{(3)} h^{(2)}\left(\sum_{j=1}^{M} w_{k,j}^{(2)} h^{(1)}\left(\sum_{i=1}^{D} w_{j,i}^{(1)} x_i + w_{j,0}^{(1)}\right) + w_{k,0}^{(2)}\right) + w_{p,0}^{(3)}\right) \tag{4}$$

Siendo $h^{(n)}$ la función de activación de la capa n, $w_{z,v}^{(n)}$ es el *v* asociado a la neurona z que se encuentra dentro de la capa n y x_i es la variable independiente i. La variable p define el número de resultados que generará la red neuronal por cada conjunto de variables independientes. A lo largo de esta memoria se trabajará con redes neuronales cuyo

Figura 3: Estructura de una red neuronal feedforward



Fuente: Elaboración propia

objetivo es predecir la probabilidad de fraude y, por tanto, p será igual a 1.

Tal y como se ha comentado anteriormente, el algoritmo de back-propagation juega un rol fundamental en la estimación de los pesos de las redes neuronales. Este mecanismo es el responsable de asignar a cada peso el gradiente de la función de error seleccionada para entrenar la red neuronal. Los pasos del algoritmo de back-propagation son:

1. Tomando las variables independientes y los pesos iniciales de la red neuronal se calcula el resultado que generaría el algoritmo, $\hat{f}(X)$.

2. Comparando la predicción de la red neuronal ($\hat{f}(X)$) con el valor real de la variable independiente (y) se calcula el error: $E_d = L(\hat{f}(X)_d, y_d)$, siendo d el número de observación de la base de datos utilizada para entrenar la red neuronal y L la función de error seleccionada. El error cuadrático, $L_{se} = (y - \hat{y})^2$, y la entropía cruzada, $L_{log} = -(y \log(p) + (1 - y) \log(1 - p))$, son las funciones de error más comúnmente utilizadas para los problemas de regresión y clasificación respectivamente.

3. Se calcula el gradiente de la función de error para cada uno de los pesos de la red. Para ello, es necesario aplicar la regla de la cadena y calcular las siguientes derivadas parciales:

   - $\partial E_d / \partial \hat{f}(X)_d$. La expresión de esta derivada dependerá de la función de error seleccionada.

- $\partial \hat{f}(X)_d / \partial \sum_{i=1}^{I} w_i x_{i,d}$. Como $\hat{f}(X) = h(\sum_{i=1}^{I} w_i x_i)$, la expresión matemática de esta derivada parcial dependerá de la función de activación seleccionada (h).
- $\partial \sum_{i=1}^{I} w_i x_{i,d} / \partial w_i$, que es la derivada de la combinación lineal de los valores de entrada y sus pesos.

Habiendo calculado estas derivadas parciales, la regla de la cadena debe ser aplicada para obtener el gradiente de la función de error asociado al peso i:

$$\frac{\partial E_d}{\partial w_i} = \frac{\partial E_d}{\partial \hat{f}(X)_d} \frac{\partial \hat{f}(X)_d}{\partial \sum_{i=1}^{I} w_i x_{i,d}} \frac{\partial \sum_{i=1}^{I} w_i x_{i,d}}{\partial w_i} \tag{5}$$

4. Los gradientes de cada observación son combinados de la siguiente manera: $g_i = \left(\sum_{d=1}^{D} \partial E_d / \partial w_i\right) / D$, siendo D el número de observaciones considerado para el paso por el algoritmo de back-propagation.

5. Una vez se ha asignado el gradiente del error a cada uno de los pesos, estos deben ser actualizados. El criterio de actualización de pesos utilizado en esta memoria es ADAM (Kingma and Ba 2014):

$$w_{i,t} = w_{i,t-1} - \delta \frac{\hat{m}_{i,t}}{\sqrt{\hat{v}_{i,t}} + \epsilon} \tag{6}$$

$$\hat{m}_{i,t} = \frac{\beta_1 m_{i,t-1} + (1-\beta_1) g_{i,t}}{1-\beta_1^t} \tag{7}$$

$$\hat{v}_{i,t} = \frac{\beta_2 v_{i,t-1} + (1-\beta_2) g_{i,t}^2}{1-\beta_2^t} \tag{8}$$

$g_{i,t}$ es el gradiente del peso i en la época t del proceso de entrenamiento de la red neuronal y δ es ratio de aprendizaje. Los autores de ADAM proponen que $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ y $\epsilon = 10^{-8}$. En esta memoria se aplica esta parametrización. Con el objetivo de reducir el sobreajuste, se puede penalizar los pesos que toman un valor elevado de la siguiente manera:

$$w_{i,t} = w_{i,t-1} - \delta \left(\frac{\hat{m}_{i,t}}{\sqrt{\hat{v}_{i,t}} + \epsilon} + \lambda w_{i,t-1} \right). \tag{9}$$

Siendo λ el parámetro responsable de la penalización de los pesos

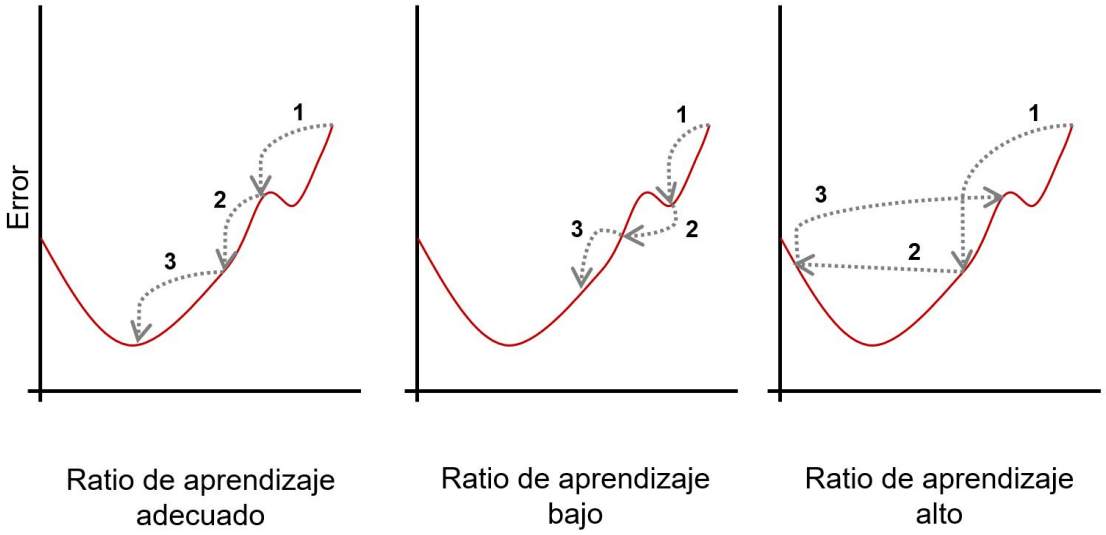
Durante el proceso de entrenamiento de este algoritmo habitualmente se optimizan, entre otros, algunos de los siguientes hiperparámetros o características del mismo:

- Ratio de aprendizaje δ. A mayor sea este valor, más rápidamente se actualizarán los pesos en cada iteración del algoritmo de back-propagation y, por tanto, el proceso de entrenamiento será más corto. Un número excesivamente alto de este número provocará que la red neuronal no se pueda aproximar al error mínimo, mientras que un valor muy bajo hará que el algoritmo se actualice tan lentamente que no llegue a minimizar el error. Por tanto, es relevante encontrar un valor para

este parámetro que permita una correcta aproximación al mínimo del error. La Figura 4, donde la línea roja representa el error cometido por la red neuronal, ilustra las diferentes casuísticas nombradas en este párrafo.

- El nivel de regularización del algoritmo puede controlarse a través de dos hiperparámetros: el porcentaje de regularización dropout y, en caso de que se utilice ADAM para la actualización de los pesos, γ. El dropout trata de prevenir el sobreajuste de la red neuronal omitiendo un cierto porcentaje de neuronas. En cada iteración de la fase de entrenamiento se seleccionan aleatoriamente las neuronas que se quedan fuera. En lo relativo a γ, el impacto que tiene en la actualización de los pesos queda reflejado en las fórmulas dónde se explica ADAM.
- Iteraciones de la red neuronal o épocas son el número de veces que, el total de la base de datos de entrenamiento pasa por el algoritmo de back-propagation.

Figura 4: Optimización de la ratio de aprendizaje



Fuente: Elaboración propia

3.1.2. Bosques aleatorios

Los bosques aleatorios están compuestos por números árboles de regresión o clasificación. En el caso de esta memoria, debido a las características del problema a tratar, se utilizarán árboles de clasificación. Dado un conjunto de variables independientes x_i y una variable dependiente y, un árbol realizará subconjuntos de información basados en x_i de manera iterativa de manera que agrupe observaciones con una y similar. Cada subconjunto se realiza evaluando diferentes candidatos ($\theta = (j, t_m)$), los cuales consisten en aplicar a la variable independiente j el umbral t_m. Por tanto, asumiendo que Q_m es

el subconjunto de la base de datos inicial formado por n_m observaciones en la iteración m del árbol, cada candidato separará la información de la siguiente manera:

$$Q_m^{izq}(\theta) = \{(x, y)|x_j \leq t_m\} \tag{10}$$

$$Q_m^{dcha}(\theta) = \{(x, y)|x_j > t_m\} \tag{11}$$

La calidad de cada candidato será evaluada de la siguiente manera:

$$G(Q_m, \theta) = \frac{n_m^{izq}}{n_m} H(Q_m^{izq}(\theta)) + \frac{n_m^{dcha}}{n_m} H(Q_m^{dcha}(\theta)) \tag{12}$$

Siendo H la función de error seleccionada. Finalmente, se seleccionará el candidato θ^* que minimice $G(Q_m, \theta)$. Las funciones de error más comúnmente utilizadas son Gini y entropía:

$$Gini = H_g(Q_m) = p_m(1 - p_m) \tag{13}$$

$$Entropy = H_e(Q_m) = -p_m \log(p_m) \tag{14}$$

Siendo $p_m = \frac{1}{n_m} \sum_{y \in Q_m} I(y = 1)$.

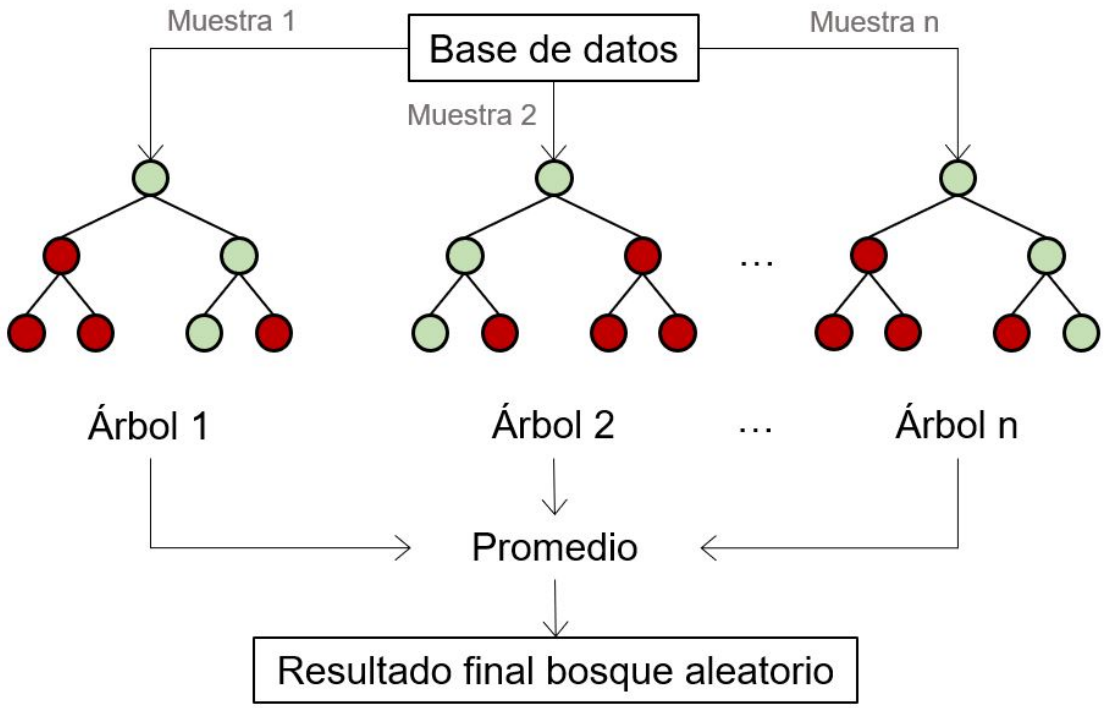
Los árboles de regresión y clasificación son algoritmos con un bajo sesgo, pero alta varianza. Es decir, se trata de un enfoque que tiende a sobreajustar los datos utilizados para el entrenamiento. Los bosques aleatorios tratan de solventar este punto débil de los árboles aleatorizando las variables independientes y las observaciones de la base de datos a ser consideradas en cada ramificación del árbol. Siguiendo este enfoque, los bosques aleatorios consisten en el promedio de n árboles aleatorizados. Por tanto, un bosque aleatorio (Figura 5) consiste en:

1. Se toman n muestras aleatorias de la base de datos, n_{tree}.
2. Para cada muestra aleatoria de la base de datos (n_{tree}) se estima un árbol de clasificación o regresión. Como se ha indicado anteriormente, en cada ramificación del árbol se considerarán al azar m_{try} variables independientes. Del conjunto de variables aleatoriamente seleccionadas, para la ramificación del árbol se seleccionará aquella o aquellas que reduzcan la medida de error en una mayor manera. En el caso de los problemas de clasificación, como el de la identificación del fraude en el seguro, las medidas de error habitualmente utilizadas para evaluar las ramificaciones de los árboles son la impureza de Gini o la ganancia de información basada en el concepto de entropía.
3. Una vez han sido estimados todos los árboles, la predicción del bosque aleatorio será el promedio de todos ellos.

Algunos de los principales hiperparámetros que se suelen optimizar durante el proceso de entrenamiento de los árboles aleatorios son:

- La profundidad o número de ramificaciones máxima de cada uno de los árboles que contienen el bosque aleatorio. Cuanto mayor sea la profundidad de los árboles, mejor serán capaces de describir el problema y mayor será su precisión. Sin embargo, si se estiman árboles demasiado profundos, el bosque aleatorio terminará sobreajustando y, por tanto, siendo un modelo con una baja capacidad predictiva.
- El número de árboles a ser estimado n_{tree}. Este debe ser lo suficientemente elevado como para que la media de todos ellos sea lo más estable posible. Obviamente, los hiperparámetros relativos a la configuración de los árboles tienen un impacto significativo en el número de ellos que es necesario.
- Al igual que la profundidad de los árboles, el número de variables seleccionadas al azar para cada ramificación (m_{try}) también podrá ser utilizada para mantener bajo control el potencial sobreajuste del modelo.

Figura 5: Estructura de un bosque aleatorio



Fuente: Elaboración propia

3.1.3. Gradient Boosting: XGBoost

Al igual que los bosques aleatorios, pero con un enfoque diferente, gradient boosting trata de solventar el problema de sobreajuste que sufren los árboles de regresión y clasificación. El algoritmo consiste en lo siguiente (Figura 6):

1. Se inicializa $\hat{f}_0(x)$ con un valor constante que cumple el siguiente criterio:

$$\hat{f}_0(x) = \underset{\rho}{\operatorname{argmax}} \sum_{i=1}^{n} L(y_i, \rho) \tag{15}$$

   Siendo $L(y_i, \rho)$ la función de error y n el número de observaciones.

2. desde $t = 1$ hasta T:

   2.1. Se calcula el gradiente de la función de error:

$$z_{it} = -\frac{\partial L(y_i, \hat{f}_{t-1}(x_i))}{\partial \hat{f}_{t-1}(x_i)} \tag{16}$$

   siendo $i = 1, \ldots, n$

   2.2. Z observaciones son seleccionadas aleatoriamente.

   2.3. Se ajusta un árbol, $\hat{h}_t(x)$, al gradiente calculado en $2{,}1$. Las observaciones a ser consideradas son aquellas seleccionadas en el paso $2{,}2$.

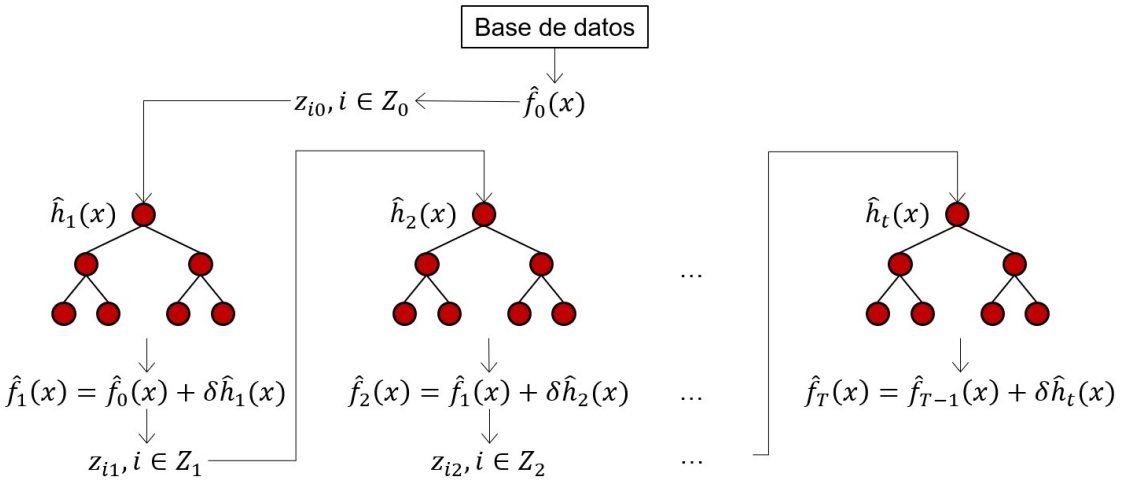
   2.4. Se actualiza el modelo de la siguiente manera:

$$\hat{f}_t(x) = \hat{f}_{t-1}(x) + \delta \hat{h}_t(x) \tag{17}$$

   δ es el factor de aprendizaje que determina a la velocidad e intensidad con la que el modelo se actualiza en cada iteración.

3. Tras las T iteraciones del paso 2, se obtendría el modelo final $\hat{f}_T(x)$.

Figura 6: Estructura de un gradient boosting



Fuente: Elaboración propia

XGBoost (Chen and Guestrin 2016) es una extensión de gradient boosting que es computacionalmente menos costosa y que permite incluir regularización L1 y L2 para así controlar el sobreajuste. L1 aplica una penalización igual al valor absoluto de los coeficientes

mientras que, L2 penaliza con el cuadrado de los coeficientes. Este enfoque se define matemáticamente de la siguiente manera:

$$Obj_{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_{it}) + \sum_{i=1}^{t} \Omega_i \tag{18}$$

Siendo Ω el término de regularización, T el número de árboles totales que tendrá XGboost, n el número de observaciones de la base datos y $\hat{y}_i^{(t)}$ la estimación en la iteración t. Los autores de este algoritmo realizan la siguiente definición matemática de los árboles de clasificación y regresión:

$$h_t(x) = w_{q(x)}, w \in \mathbb{R}^J, q : \mathbb{R}^d \to \{1, 2, \ldots, J\} \tag{19}$$

Siendo q la función que asigna cada observación a su ramificación, J el número de ramificaciones y w la puntuación de cada una de las ramificaciones (más alto cuanto más preciso es). Asumiendo esta definición, se aplica la regularización de la siguiente manera:

$$\Omega_t = \gamma J + \frac{1}{2}\lambda \sum_{j=1}^{J} w_j^2 \tag{20}$$

Los parámetros que determinan la intensidad de la regularización L2 y L1 son λ y γ respectivamente. Entonces, la función de error de XGboost queda definida por la siguiente expresión:

$$Obj_{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_{i,t-1} + \hat{h}_t(x_i)) + \Omega_t + constant \tag{21}$$

Aplicando la expansión de Taylor de segundo orden se obtiene que:

$$Obj_{(t)} \simeq \sum_{i=1}^{n} [L(y_i, \hat{y}_{i,t-1}) + g_i \hat{h}_t(x_i) + \frac{1}{2} e_i \hat{h}_t^2(x_i)] + \Omega_t + constant \tag{22}$$

$$g_i = \partial_{\hat{y}_{i,t-1}} L(y_i, \hat{y}_{i,t-1}) \tag{23}$$

$$e_i = \partial^2_{\hat{y}_{i,t-1}} L(y_i, \hat{y}_{i,t-1}) \tag{24}$$

Eliminando las constantes, la función de error en la iteración t será:

$$Obj_{(t)} \simeq \sum_{i=1}^{n} [g_i \hat{h}_t(x_i) + \frac{1}{2} e_i \hat{h}_t^2(x_i)] + \Omega_t \tag{25}$$

Siendo I_d el subconjunto de datos asignados a la ramificación d del árbol y expandiendo

la expresión Ω_t, la función objetivo de XGBoost seguiría la siguiente expresión:

$$Obj_{(t)} \simeq \sum_{i=1}^{n}[g_i\hat{h}_t(x_i) + \frac{1}{2}e_i\hat{h}_t^2(x_i)] + \gamma J + \frac{1}{2}\lambda\sum_{j=1}^{J}w_j^2 \tag{26}$$

$$= \sum_{j=1}^{J}[(\sum_{i\in I_d} g_i)w_j + \frac{1}{2}(\sum_{i\in I_d} e_i + \lambda)w_j^2] + \gamma J \tag{27}$$

El segundo sumatorio de la expresión previa puede simplificarse ya que puntuación asignada de las observaciones es igual si caen dentro de una misma ramificación. Por tanto, el w óptimo de la ramificación j sería aquel que maximice:

$$w_j^* = -\frac{\sum_{i\in I_d} g_i}{\sum_{i\in I_d} e_i + \lambda} \tag{28}$$

Por tanto, la función objetivo para optimizar XGBoost sería:

$$Obj_{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{J}\frac{(\sum_{i\in I_d} g_i)^2}{\sum_{i\in I_d} e_i + \lambda} + \gamma J \tag{29}$$

Con el objetivo de reducir el coste computacional e incluir regularización L1 y L2 en gradient boosting, XGBoost toma como referencia la función objetivo anterior para evaluar iterativamente las ramificaciones óptimas de los árboles que componen el algoritmo. Los autores de XGboost sugieren utilizar la maximización de la siguiente expresión (basada en la función objetivo anteriormente definida) para decidir la ramificación óptima:

$$Obj_{(t)}^{Split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} e_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} e_i + \lambda} + \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} e_i + \lambda}\right] - \gamma \tag{30}$$

I_L y I_R son los subconjuntos en los que se separa I tras la ramificación del árbol. Si no se encuentra una solución en la que γ sea menor que el resto de la expresión, no se abrirá una nueva ramificación.

Durante el proceso de entrenamiento de este algoritmo, y al igual que en el caso de los bosques aleatorios, la profundidad máxima de los árboles y el número de árboles a ser estimados son dos de los principales hiperparámetros a ser optimizados. Las razones son las mismas que las ya mencionadas para los bosques aleatorios al final de la subsección 3.1.2.

El factor de aprendizaje δ juega también un papel fundamental en este algoritmo ya que controla la velocidad con la que el mismo se actualiza. Un δ muy elevado o bajo puede provocar que el algoritmo no minimice todo lo que podría el error (ver Figura 4). Por último, el factor de regularización L2 γ también suele ser optimizado durante la fase de entrenamiento de Xgboost. La correcta utilización de este factor permite fijar una elevada profundidad para los árboles que, luego, será recortada por γ si fuera necesario.

3.1.4. Gradient Boosting: Adaboost

Al igual que XGBoost, Adaboost (Freund and Schapire 1997) es una variante de gradient boosting. El objetivo de este algoritmo es ir dando más peso durante las iteraciones del gradient boosting a aquellas observaciones que producen un mayor grado de error. El enfoque aplicado por este algoritmo es especialmente interesante para el caso de la detección de fraude ya que de manera automática se irá centrando en aquellos casos que son más difíciles de identificar como fraudulentos. Matemáticamente, el algoritmo consiste en lo siguiente:

1. Dada una secuencia de N observaciones y siendo x el conjunto de variables independientes e y la variable dependiente, se inicializan los pesos de cada observación de la siguiente manera: $w_i^1 = 1/N$ para $i = 1, \ldots, N$.

2. desde $t = 1$ hasta T:

   2.1. Se calcula la distribución de pesos de la siguiente manera:

   $$p^t = \frac{w^t}{\sum_{i=1}^{N} w_i^t} \tag{31}$$

   2.2. Se ajusta un árbol de regresión o clasificación $\hat{h}_t$ con la distribución de pesos p^t.

   2.3. Se calcula el error de $\hat{h}_t$: $\epsilon_t = \sum_{i=1}^{N} p_i^t |\hat{h}_t(x_i) - y_i|$.

   2.4. Se estima $\beta_t = \epsilon_t/(1 - \epsilon_t)$.

   2.5. Los pesos para la siguiente iteración serían igual a la siguiente expresión:

   $$w_i^{t+1} = w_i^t \beta_t^{1-|\hat{h}_t(x_i)-y_i|} \tag{32}$$

3. Tras las T iteraciones del paso 2, el resultado final del modelo $\hat{h}_f(x)$ sería:

$$\hat{h}_f(x) = \begin{cases} 1 & \text{Si } \sum_{t=1}^{T} (\log 1/\beta_t) \hat{h}_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \log 1/\beta_t \\ 0 & \text{En caso contrario} \end{cases} \tag{33}$$

Los principales hiperparámetros que se suelen optimizar durante el proceso de entrenamiento de Adaboost son:

- La ratio de aprendizaje δ. Tal y como se ha comentado en el caso de Xgboost y redes neuronales, este hiperparámetro controla la velocidad con la que Adaboost se actualiza. Un δ no adecuado provocará el modelo no minimice el error todo lo que podría (Figura 4)

- El número máximo de ramificaciones de los árboles y el número de iteraciones son también dos de los principales hiperparámetros de Adaboost. El papel que juegan es el mismo que el ya mencionado en las subsecciones 3.1.1 y 3.1.2 para las redes neuronales y los bosques aleatorios respectivamente.

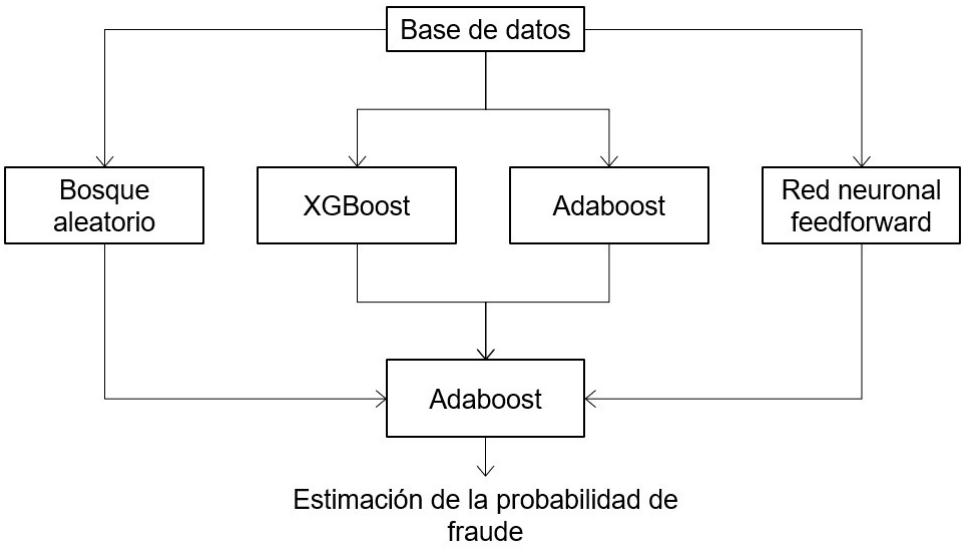
- Por último, merece la pena destacar la función de error seleccionada para crear nuevas ramificaciones en los árboles que componen Adaboost. Como se ha mencionado anteriormente, gini o entropía son los principales criterios utilizados en los problemas de clasificación. Gini favorece ramificaciones con mayor número de observaciones y es computacionalmente más eficiente que la entropía. Normalmente, las diferencias en la precisión por el uso de gini o entropía como funciones de error son menores y, por tanto, es un hiperparámetro que no se suele tratar de optimizar. En cualquier caso, la función de error también podría ser optimizada en el resto de los algoritmos basados en árboles (bosques aleatorios y Xgboost).

3.2. Arquitectura del modelo propuesto

Como ya se ha comentado previamente, el modelo de fraude que se propone en esta memoria trata de ofrecer la probabilidad de que un siniestro sea fraude en función de sus características y las del asegurado. Con estas probabilidades, la compañía podrá decidir a través del establecimiento de diferentes umbrales como de estricta es a la hora de detectar fraude e incluso hallar un umbral que maximice el beneficio obtenido en el proceso de detección del fraude. Adicionalmente, permitiría mejorar la selección de riesgos de la compañía aseguradora a través de la inclusión de estas probabilidades en el proceso de suscripción de nuevas pólizas y aplicando cierto malus o restringiendo los descuentos comerciales a aquellos riesgos con una gran probabilidad de tener siniestros y cometer fraude. Todas estas posibilidades que ofrece un modelo de fraude probabilístico no son aplicables de manera tan directa con los modelos tradicionales de reglas automáticas. Estos últimos marcan los siniestros como fraudulento o no, sin ofrecer la probabilidad asociada a cada una de las opciones.

Con el objetivo de intentar conseguir un nivel de precisión superior a las reglas automáticas tradicionales, el modelo que se propone en esta memoria apila diferentes algoritmos del ámbito del aprendizaje automático y la inteligencia artificial. La técnica de apilado de algoritmos ha sido ampliamente utilizada en numerosos ámbitos de estudio ya que, a través de la unificación de diferentes algoritmos, permite mejorar la precisión que tendrían los modelos por separado. Dentro del ámbito financiero y actuarial, este enfoque ha sido ampliamente utilizado para la predicción de volatilidad de mercados financieros o de materias primas (Lu, Que, and Cao 2016, Kim and Won 2018, Kristjanpoller and Minutolo 2018, Ramos-Pérez et al. 2019, Vidal and Kristjanpoller 2020 y Ramos-Pérez et al. 2021), donde se requiere de modelos con un gran poder predictivo por lo intenso y repentino que suele ser el comportamiento de esta variable. Desde el punto de vista puramente técnico el apilamiento de algoritmos consiste en la construcción de capas, de manera que los algoritmos de la segunda capa se alimentarán de las predicciones realizadas por los algoritmos presentes en la primera.

Figura 7: Estructura del modelo de fraude propuesto



Fuente: Elaboración propia

Las capas del modelo de predicción de fraude que se propone en esta memoria están compuestas por los siguientes algoritmos (Figura 7):

- Capa 1. Los algoritmos incluidos en esta parte del modelo tienen como input la base de datos. Las características de la base de datos serán explicadas en las siguientes secciones pero, resumiendo, contiene como variable dependiente si el siniestro fue fraudulento o no y como variables independientes características del cliente y el siniestro. Los algoritmos utilizados en esta capa son:

    - Bosque aleatorio. Los hiperparámetros que se optimizarán serán el número de árboles que contiene el bosque aleatorio y la profundidad o número de ramificaciones de los árboles que componen el bosque aleatorio. Para encontrar esta configuración óptima se aplicará validación cruzada con 10 iteraciones, siendo la función de error utilizada la entropía cruzada. La definición de las predicciones de este algoritmo sería la siguiente:

    $$\hat{f}_{rf}(x) = \frac{1}{N}\sum_{i=1}^{N}\hat{h}_i(x) \tag{34}$$

    Siendo $\hat{h}_i$ los árboles que componen el bosque aleatorio y N el número de ellos que serán estimados.

    - XGBoost. En este caso, los hiperparámetros a ser optimizados serán γ y el número máximo de ramificaciones permitidas para los árboles que componen en el algoritmo. Tal y como se ha explicado en secciones anteriores, el primer parámetro permite determinar el nivel de regularización L2, mientras que el segundo da mayor o menor complejidad a cada uno de los árboles. La con-

figuración óptima se obtendrá aplicando validación cruzada (10 iteraciones). El número de iteraciones se fija en 500 y la ratio de aprendizaje en $\delta = 0{,}01$. Tal y como se ha indicado anteriormente, la definición de las predicciones realizadas por este modelo sigue esta expresión:

$$\hat{f}_{xg}(x) = \hat{f}_{t-1}(x) + \delta\hat{h}_t(x) = \hat{f}_0(x) + \delta\hat{h}_1(x) + \cdots + \delta\hat{h}_t(x) \tag{35}$$

- Adaboost. Al igual que en los casos anteriores, se aplicará validación cruzada con 10 iteraciones. Los hiperparámetros que se optimizarán serán: el criterio para crear nuevas ramificaciones en los árboles (Gini o entropía), el número máximo de ramificaciones de cada uno de los árboles, el número de iteraciones T y la ratio de aprendizaje δ. Este modelo tiene la siguiente expresión:

$$\hat{f}_{ad}(x) = \hat{f}_{t-1}(x) + \delta\hat{h}_t(x) = \hat{f}_0(x) + \delta\hat{h}_1(x) + \cdots + \delta\hat{h}_t(x) \tag{36}$$

  Las predicciones finales, tal y como se ha comentado en la sección 3.1.4, se obtendrán de la siguiente manera:

$$\hat{f}_{ad}(x) = \begin{cases} 1 & \text{Si } \sum_{t=1}^{T}(\log 1/\beta_t)\hat{h}_t(x) \geq \frac{1}{2}\sum_{t=1}^{T}\log 1/\beta_t \\ 0 & \text{En caso contrario} \end{cases} \tag{37}$$

- Red neuronal feedforward. Se aplicarán 10 iteraciones de validación cruzada para optimizar el nivel de regularización dropout y la ratio de aprendizaje δ. La regularización aplicada consiste en apartar aleatoriamente ciertas conexiones durante cada una de las iteraciones de la fase de aprendizaje. La red neuronal que se utilizará tendrá 2 capas intermedias con 8 neuronas cada. Por tanto, la expresión matemática para obtener las predicciones sería:

$$\hat{f}_{rn}(x) = h^{(3)}\left(\sum_{k=1}^{8} w_{p,k}^{(3)} h^{(2)}\left(\sum_{j=1}^{8} w_{k,j}^{(2)} h^{(1)}\left(\sum_{i=1}^{D} w_{j,i}^{(1)} x_i + w_{j,0}^{(1)}\right) + w_{k,0}^{(2)}\right) + w_{p,0}^{(3)}\right) \tag{38}$$

  La explicación de cada uno de los parámetros de este algoritmo ya ha sido facilitada en la sección 3.1.1.

- Capa 2. El algoritmo presente en esta capa toma como input las predicciones de los algoritmos presentes en la primera capa. Su objetivo es, por tanto, tomar los atributos más precisos de los algoritmos de la primera capa para así obtener una mayor precisión que cada uno de los algoritmos individuales. En línea con lo mostrado en la Figura 7, Adaboost será el algoritmo que apile las predicciones de la primera capa. Por tanto, la expresión final del modelo de predicción de fraude propuesto sería:

$$\hat{f}_{ad2}(X_{capa1}) = \hat{f}_{t-1}(X_{capa1}) + \delta\hat{h}_t(X_{capa1}) \tag{39}$$

Siendo X_{capa1} el vector que contiene las predicciones de todos los modelos de la primera capa: $\hat{f}_{rf}(x)$, $\hat{f}_{xg}(x)$, $\hat{f}_{ad}(x)$ y $\hat{f}_{rn}(x)$. Las predicciones finales del modelo propuesto serán:

$$\hat{f}_{ad2}(X_{capa1}) = \begin{cases} 1 & \text{Si } \sum_{t=1}^{T}(\log 1/\beta_t)\hat{h}_t(X_{capa1}) \geq \frac{1}{2}\sum_{t=1}^{T}\log 1/\beta_t \\ 0 & \text{En caso contrario} \end{cases} \tag{40}$$

El problema de la detección de fraude es un problema de clasificación no balanceado. Esto quiere decir que, dentro de las bases de datos, el caso de fraudes es significativamente menor que los casos no fraudulentos. En el ámbito del aprendizaje automático hay diferentes maneras de afrontar este problema. Una de las maneras de estimar algoritmos con bases de datos no balanceadas es dar diferentes pesos al error cometido dependiendo del valor de la variable dependiente. Se da un peso mayor al error cometido con la clase menos representada en la base de datos, de manera que los errores de ambas clases tengan el mismo peso durante el entrenamiento del algoritmo. Una forma habitual de calcular los pesos es la siguiente:

$$w_i = \begin{cases} \text{Si } y_i = 0 & \text{entonces } 1/\sum_{i=1}^{N} I(y=0) \\ \text{Si } y_i = 1 & \text{entonces } 1/\sum_{i=1}^{N} I(y=1) \end{cases} \tag{41}$$

Siendo $I(y = z)$ una variable que toma el valor de 1 cuando se cumple la condición $y = z$ y 0 en caso contrario. N es el número de observaciones de la base de datos de entrenamiento.

Otras dos maneras de tratar con problemas no balanceados es a través de la reducción o aumento de la base de datos de entramiento. Estos mecanismos consisten en reducir (eliminando aleatoriamente observaciones de la clase más común) o aumentar (simulando nuevas observaciones de la clase menos común con algún algoritmo como el propuesto por Chawla et al. 2002) la base de datos con el objetivo de que la variable dependiente esté balanceada. En esta memoria, se aplicará la reducción de la base de datos para estimar los algoritmos que componen el modelo de detección de fraude.

Adicionalmente, la utilización de Adaboost en la segunda capa del modelo es especialmente apropiada ya que, debido a la construcción del mismo, este realizará especial énfasis en aquellos casos que son especialmente complicados de clasificar correctamente. De esta manera, la segunda capa del algoritmo se centrará, prácticamente desde el primer momento, en aquellos casos en los que los algoritmos de la primera capa han tenido dificultades para realizar la clasificación.

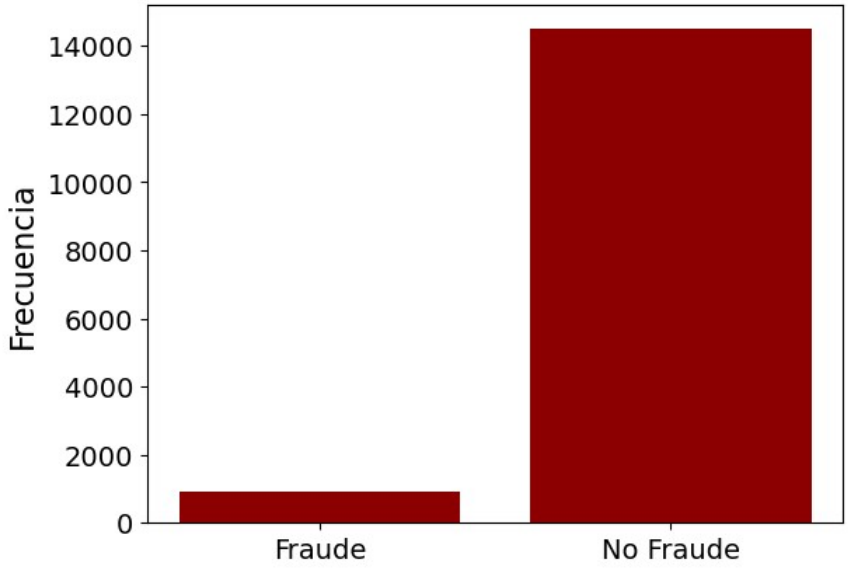
3.3. Resultados empíricos

En esta sección se analizará la base de datos utilizada para estimar los diferentes modelos de detección de fraude (subsección 3.3.1), se presentarán brevemente los modelos tradicionales utilizados para las comparativas (subsección 3.3.2) y, por último, se mostrarán los resultados empíricos obtenidos (subsección 3.3.3).

3.3.1. Base de datos

La base de datos de siniestros de autos que se utilizará en esta memoria contiene las características del siniestro, el asegurado y la identificación del siniestro como fraudulento o no fraudulento. Esta base de datos es dominio público y fue subida por Oracle a Github (para descargar la información hacer click aquí). Como se ha comentado en secciones anteriores, la detección de fraude es un problema de clasificación no balanceado. Tal y como puede observarse en la Figura 8, esto se refleja en la base datos utilizada para esta memoria que contiene 923 casos de fraude por más de $14,000$ que no lo son.

Figura 8: Fraude en la base de datos



Fuente: Elaboración propia

La base de datos, con el objeto de poder perfilar las características del fraude, tiene numerosas variables relativas a las características del cliente y el siniestro:

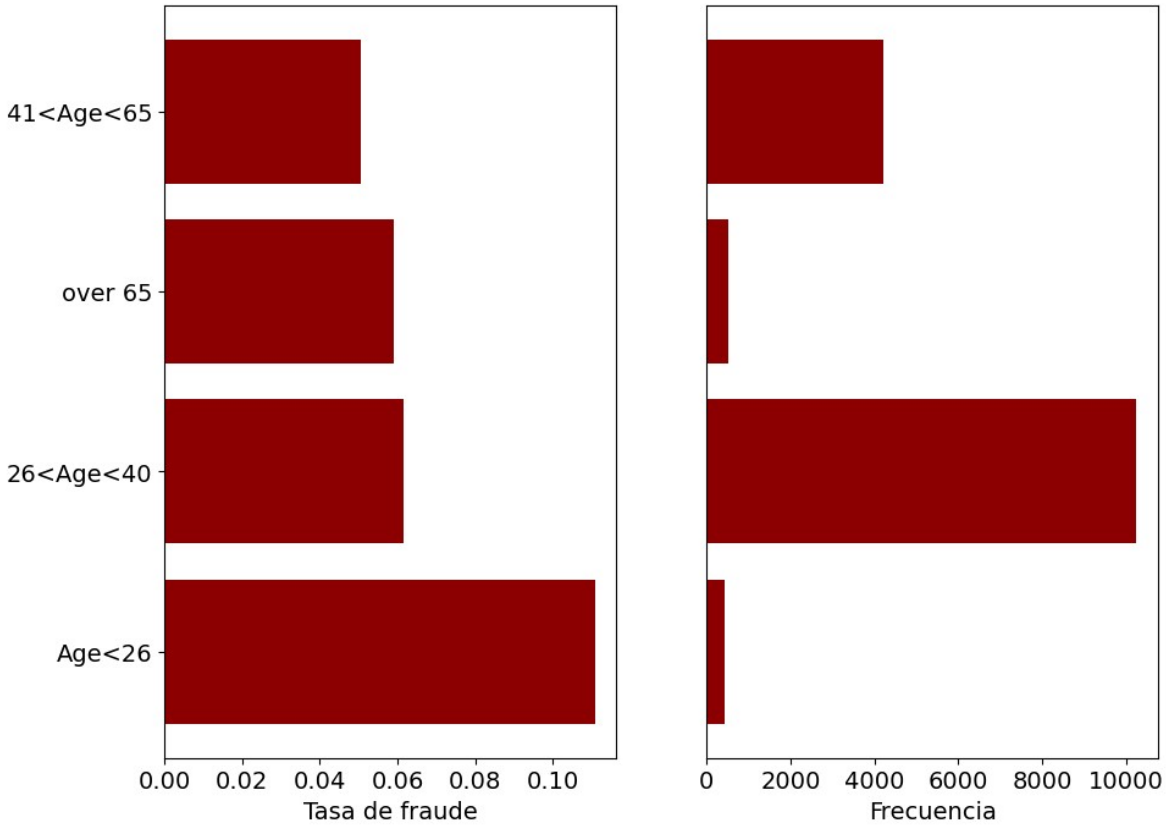
- Marca del coche. Hay 19 marcas presentes en la base de datos pero, algunas de ellas como Porche o Ferrari, apenas son 5 observaciones de la base de datos. Por ello, para la estimación de los algoritmos, las marcas se agruparán en las siguientes categorías: marcas de lujo, americanas, alemanas, japonesas y resto de marcas.
- Área en la que sucedió el accidente. Esta variable diferencia entre si el accidente sucedió en un entorno urbano o rural.

- Mes en el que ocurrió el siniestro
- Sexo del asegurado
- Culpa. Puede tomar dos valores: Terceros o asegurado.
- Categoría del vehículo. Se diferencia entre coches deportivos, berlinas o sedan y utilitarios.
- Precio del vehículo. No se trata de una variable continua ya que se han agrupado en 6 categorías: $x < 20,000, 20,000 \leq x < 30,000, 30,000 \leq x < 40,000, 40,000 \leq x < 50,000, 50,000 \leq x < 60,000, 60,000 \leq x < 70,000$ y $x \geq 70,000$.
- Estado civil del asegurado.
- Rating del conductor asegurado. Esta variable separa en 4 categorías a los asegurados dependiendo de su historial siniestral.
- Antigüedad del vehículo asegurado. Al igual que el precio del vehículo, no se trata de una variable continua ya que se ha categorizado en 3 clases: $x < 2, 2 \leq x < 6$ y $x \geq 6$.
- Informe policial. Esta variable indica si la policía realizó o no un informe sobre el siniestro.
- Edad del asegurado. Contiene las siguientes categorías: $x < 26, 26 \leq x < 40, 40 \leq x < 65$ y $x \geq 65$.
- Contratación. Determina si la póliza se contrató con un agente o con un mediador de seguros.
- Testigos. Indica si hubo testigos o no del siniestro.
- Número de suplementos contratados en la póliza.
- Tipo de póliza. Esta variable informa el tipo de póliza contratada: responsabilidad civil, colisión o todo riesgo.
- Número de siniestros que el asegurado ha tenido en el pasado. Esta variable se agrupa en cuatro categorías: 0, 1, de 2 a 4 y más de 4.
- Día de la semana en la que ocurrió el siniestro.
- Franquicia fijada en la póliza.
- Diferencia de días entre que ocurrió el siniestro y la entrada en vigor de la póliza. Contiene 3 categorías: $x < 15, 15 \leq x < 30$ y $x \geq 30$.
- Número de años desde el último cambio de domicilio de la póliza.

Estas variables serán utilizadas como variables independientes en los algoritmos de la primera capa del modelo de detección de fraude que se propone en esta memoria. Analizando la base de datos, se descubren ciertas categorías de algunas variables que son especialmente proclives a cometer fraude. Por ejemplo, tal y como se observa en la Figura 9, la franja de edad de aquellos asegurados por debajo de los 26 años muestra una probabilidad de fraude muy superior al resto de franjas de edad. La tasa de fraude (número de siniestros fraudulentos dividido entre el número total de siniestros) se encuentra por encima del $10\,\%$ por lo que, a pesar de que la frecuencia total en la base de datos no es elevada, al menos uno de cada diez siniestros de estos asegurados es fraudulento.

Otro ejemplo de categoría especialmente proclive al fraude se puede encontrar en la variable que define el tipo de vehículo. La Figura 10 muestra grandes diferencias en las tasas de fraude dependiendo de si se trata de un vehículo deportivo, un sedan o un utilitario. Al igual que en el caso de aquellos asegurados por debajo de 26 años, los utilities muestran una tasa de fraude por encima del $10\,\%$.

Figura 9: Fraude por edad del asegurado



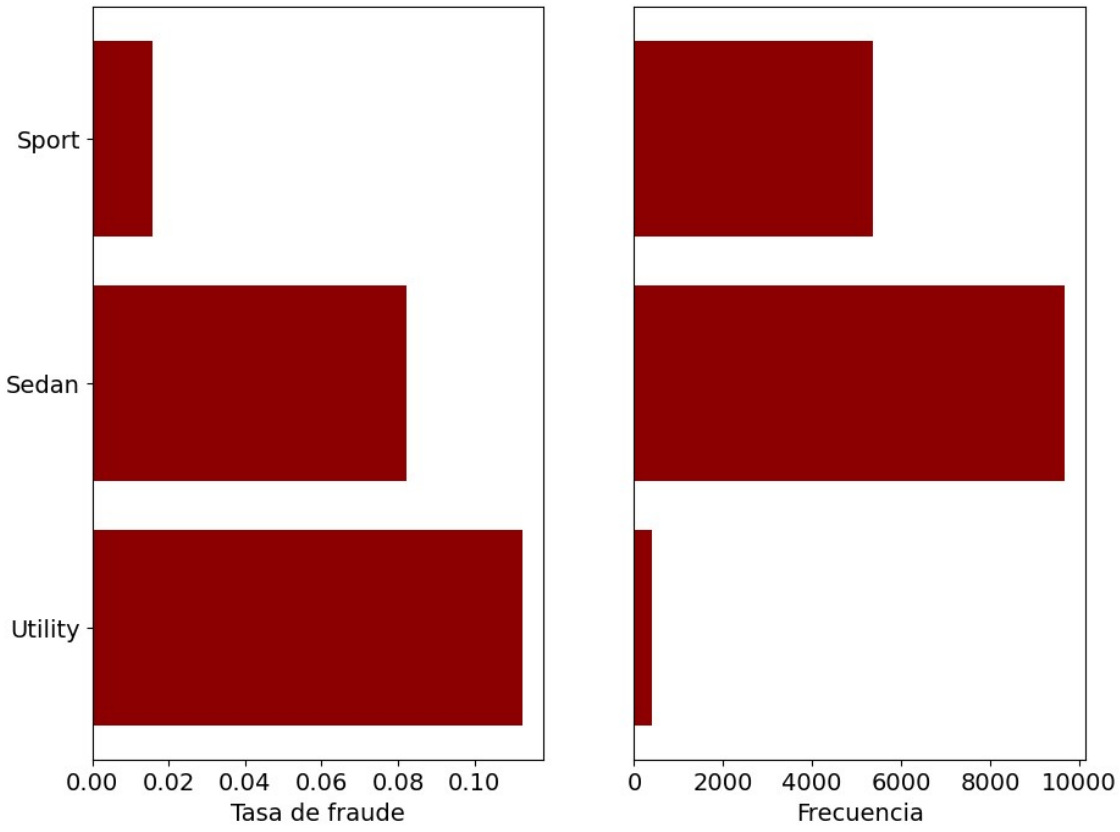
Fuente: Elaboración propia

De cara a establecer las reglas automáticas y estimar el modelo de detección de fraude

presentado en la subsección 3.2, la base de datos se separa aleatoriamente en dos: entrenamiento y validación. La base de datos de entrenamiento será utilizada para calibrar las reglas automáticas y el modelo que se propone en esta memoria y estará compuesta por el $75\,\%$ de las observaciones de la base de datos. El resto de las observaciones $(25\,\%)$ formará parte de la base de datos de validación y será utilizada para comparar la precisión del modelo basado en aprendizaje automático e IA con los modelos tradicionales de reglas automáticas para la detección del fraude.

La validación de cualquier tipo de modelo nunca debe ser llevado a cabo con la misma información con la que se ha estimado el mismo. Hacer esto llevaría a una sobreestimación la capacidad predictiva ya que el modelo ha sido especialmente optimizado tener un alto nivel de precisión con la base de datos de entrenamiento. Lo relevante es evaluar la capacidad de generalizar del modelo y, por tanto, la precisión de los modelos en secciones de información que no hayan sido consideradas durante su entrenamiento.

Figura 10: Fraude por tipo de póliza



Fuente: Elaboración propia

3.3.2. Modelos de referencia

Para evaluar el modelo propuesto en la subsección 3.2, se comparará su precisión con modelos tradicionales de reglas automáticas para la predicción del fraude. Estas reglas automáticas, que darán al analista de la compañía de seguros una señal de que el siniestro podría ser un fraude, se calibrarán con la base de datos de entramiento. En esta subsección se explicarán los dos modelos de reglas automáticas utilizados como referencia. La diferencia entre ambos será el nivel de exigencia requerido a las reglas automáticas y el número de reglas que deben darse al mismo tiempo para que el siniestro sea marcado como fraude. A lo largo de esta memoria, se nombrará como regla automática conservadora (RAC) al modelo que prefiere ser más restrictivo en las probabilidades de fraude requeridas pero que exige que menos de ellas se den al mismo tiempo para marcar el siniestro como fraude. Por otro lado, el modelo menos selectivo en las probabilidades fraude se le denominará regla automática estándar (RAE).

La RAC se basa en marcar como fraude aquellos siniestros que, al menos, contengan dos categorías cuya probabilidad de fraude es mayor al $8\,\%$. Tomando como referencia la base de datos de entramientos mencionada en la anterior sección, estas categorías son:

- Área en la que sucedió el accidente. Los ocurridos en zonas rurales tienen un porcentaje de fraude mayor al $8\,\%$.
- Los vehículos sedan y utilitarios tienen un fraude mayor al umbral establecido por RAE mientras que, los deportivos, se encuentran por debajo de este nivel.
- Automóviles con precios mayores o iguales a $70{,}000$ y menores de $20{,}000$.
- La probabilidad de fraude también se encuentra por encima del $8\,\%$ en aquellos asegurados con una edad menor a los 26 años.
- Tipo de póliza: todo riesgo.
- La diferencia entre los días entre los que ocurrió el siniestro y la entrada en vigor de la póliza es menor o igual a los 30 días.
- Pólizas cuyo domicilio ha sido modificado en los últimos 3 años.
- Franquicias desde 300 hasta 500.

La RAE, tal y como se ha nombrado anteriormente, tiene un umbral de probabilidad de fraude ($7{,}5\,\%$) menor que el RAC ($8\,\%$). Con el objetivo de no aumentar en exceso el número de siniestros marcados como fraude por la RAE, se requerirá que se cumplan al mismo tiempo 4 categorías que superen el umbral. Adicionalmente a las categorías nombradas para el RAC, RAE considerará también:

- La culpa del siniestro debe ser del asegurado y no de un tercero.

- Antigüedad del coche inferior a los 2 años.
- La marca del coche no debe ser ni americana, ni alemana, ni japonesa y no debe ser considerada de lujo.
- El número de siniestros anteriores debe ser igual a cero.
- Las tasas de fraude también son mayores al $7{,}5\,\%$ en aquellos siniestros ocurridos en marzo, mayo y agosto.

Tanto estas dos reglas automáticas como el modelo de fraude propuesto han sido estimados en función de las observaciones contenidas en la base de datos de entrenamiento.

3.3.3. Comparativa de resultados

A lo largo de esta subsección, se comparará la precisión de las reglas automáticas y el modelo de fraude basado en aprendizaje automático sobre la base de datos de validación. Esto permitirá la evaluación de los modelos sobre observaciones no consideradas durante su fase de entrenamiento, de manera que dará una estimación de cómo se comportarían los diferentes enfoques si se implementarán dentro de las operaciones diarias de una compañía aseguradora.

Antes de comenzar con las comparativas entre los diferentes modelos, a continuación de nombran los hiperparámetros seleccionados para cada uno de los algoritmos del modelo propuesto en esta memoria. Tal y como se ha mencionado en la subsección 3.2, se ha aplicado validación cruzada para encontrar el valor óptimo de los mismos. Los resultados son los siguientes:

- Bosque aleatorio - Capa 1. Tras aplicar 10 iteraciones de validación cruzada se obtiene que el número óptimo de árboles que componen el bosque aleatorio son $N = 250$ y que la profundidad de los mismos o su número máximo de ramificaciones será de 9.
- XGBoost - Capa 1. La configuración óptima de este algoritmo, de acuerdo al proceso de calibración mencionado en la subsección 3.2 es el siguiente: el parámetro responsable de fijar el nivel de regularización L2 será $\gamma = 1{,}5$, mientras que la profundidad o el número de ramificaciones de cada uno de los árboles que compone el algoritmo será igual a 9.
- Adaboost - Capa 1. Tras 10 iteraciones de validación cruzada, se obtiene que la configuración óptima es la siguiente: entropía como función de error para generar nuevas ramificaciones de los árboles, la ratio de aprendizaje será $\delta = 0{,}05$, el número de iteraciones $T = 750$ y la máxima profundidad de cada árbol será igual a 5 niveles o ramificaciones.
- Red neuronal - Capa 1. La ratio de aprendizaje óptimo es $\delta = 0{,}025$ y el porcentaje de regularización dropout del 20 %.

- Adaboost - Capa 2. Tal y como queda definido a lo largo de la subsección 3.2, este algoritmo es el responsable de apilar las predicciones obtenidas en la capa 1 de la arquitectura del modelo. Tras las 10 iteraciones de validación cruzada se obtiene la siguiente configuración óptima: entropía como función de error para generar nuevas ramificaciones de los árboles, la ratio de aprendizaje será $\delta = 0{,}05$, el número de iteraciones $T = 250$ y el máximo de ramificaciones de cada árbol será igual a 2.

Una vez mostrada la configuración óptima del modelo propuesto y con el objetivo de dar una mayor robustez a la comparativa de resultados, la evaluación de los modelos considerará diferentes métricas ampliamente utilizadas en los problemas de clasificación (Ver Figura 11 y Cuadro 5 para seguir el cálculo de éstas). Son las siguientes:

- Exactitud: $(TP + TN)/(TP + TN + FP + FN)$. Este indicador nos muestra cómo de cerca están los modelos estadísticos de estimar correctamente tanto los casos que son fraude como los que no lo son.
- Especificidad. Esta métrica indica el porcentaje de fraudes correctamente detectados sobre el total de fraudes marcados por el modelo estadístico. Su cálculo es el siguiente: $TP/(TP + FP)$
- Sensibilidad o % de fraude detectado: $TP/(TP + FN)$. Indica el porcentaje de casos que, siendo realmente fraude, han sido detectados como tal por el modelo estadístico.
- F1- Score: $2TP/(2TP + FP + FN)$. Es la media armónica de la sensibilidad y la especificidad.
- % Marcas. Porcentaje de observaciones marcadas como fraude por el modelo estadístico: $TP + FP/(TP + TN + FP + FN)$

Al modelo de fraude propuesto en esta memoria se le denominará como S-ADA en referencia a las siglas de las principales técnicas utilizadas. La letra S se refiere a la técnica de apilación (Stacking en inglés) de algoritmos que se aplica en el modelo. ADA hace referencia a Adaboost, el modelo utilizado para apilar el resto de los algoritmos utilizados en el modelo. Para más detalles del modelo, ir a la subsección 3.2.

Antes de profundizar en el análisis de las métricas anteriores, las matrices de confusión (Cuadro 5) muestran que S-ADA incurre en un número muy inferior de falsos positivos. Esto quiere decir que el número de veces que se marca incorrectamente un siniestro como fraudulento es mucho menor con S-ADA que con RAE o RAC. Normalmente, este menor grado de error suele suponer una menor detección de fraude. Sin embargo, el enfoque que se propone en esta memoria es capaz de identificar más casos de fraude (Verdadero positivo) que los modelos que no utilizan aprendizaje automático o IA dentro de su estructura.

Figura 11: Matriz de confusión

Fuente: Elaboración propia

Los modelos (bosque aleatorio, Adaboost, Xgboost y red neuronal) que componen a S-ADA están especialmente preparados para encontrar patrones o relaciones en los datos que para un ser humano serían muy difícil de identificar sin hacer uso de ellos. La complejidad de estos algoritmos, su capacidad para trabajar soluciones no lineales o multidimensionales y la gran cantidad de variables a considerar en el proceso de identificación de fraude hacen que sean capaces de encontrar soluciones no intuitivas que, sin embargo, son las que hacen que este enfoque logre un mayor nivel de verdaderos positivos con menos falsos positivos.

Cuadro 5: Resultados empíricos: Matrices de confusión

Modelo	Verdadero Positivo	Falso Positivo	Verdadero Negativo	Falso Negativo
RAC	195	2305	1325	30
RAE	158	1546	2084	67
S-ADA	208	563	3067	17

Elaboración propia

El Cuadro 6 muestra la comparativa entre los resultados obtenidos con los diferentes modelos para cada una de las métricas de precisión que se van a evaluar.

Los resultados obtenidos sugieren dos principales conclusiones. La primera es que S-ADA es capaz de detectar más fraude marcando como fraudulento un número de siniestros significativamente menor que RAC y RAE. La Sensibilidad nos indica que S-ADA es capaz de detectar el 92.4 % del fraude mientras que RAC y RAE son capaces de identificar el 86.7 % y el 70.3 % respectivamente. RAC tiene un nivel de sensibilidad muy

Cuadro 6: Resultados empíricos: Métricas de precisión

Modelo	Exactitud	Especificidad	Sensibilidad	F1-Score	% Marcas
RAC	0.394	0.078	0.867	0.143	64.9 %
RAE	0.582	0.093	0.703	0.164	44.2 %
S-ADA	0.850	0.270	0.924	0.418	20.0 %

Elaboración propia

similar al de S-ADA. Sin embargo, para lograr este nivel de sensibilidad, RAC marca como siniestro fraudulento el 64.9 % de los siniestros mientras que S-ADA identifica sólo el 20.0 % como tal. De acuerdo con los datos de rentabilidad de fraude de ICEA (explicados en la subsección 2.1), la alta precisión de S-ADA podría llegar a mejorar significativamente las ratios de siniestralidad, permitiendo a la entidad aseguradora identificar más fraude con un número menor de investigaciones.

En segundo lugar, S-ADA logra el % de identificación de fraude anterior revisando muchos menos siniestros que RAC y RAE. La Especificidad muestra que RAC y RAE sólo realizan una marca de fraude correcto en el 7.8 % y 9.3 % de las ocasiones respectivamente. Esto quiere decir que, en el caso del RAC, por cada 10 siniestros marcados como fraudulento, más de 9 en realidad no lo son. Sin embargo, las marcas realizadas por S-ADA son correctas en el 27 % de las ocasiones. Para conseguir mayores ratios, los modelos basados en reglas automáticas tienen que ser aún más estrictos provocando que el % de fraude detectado o Sensibilidad se vea negativamente afectado. S-ADA permitiría a la entidad aseguradora revisar menos siniestros manteniendo un elevado nivel de detección de fraude, teniendo un impacto significativo en la eficiencia de la compañía (se requiere un menor nivel de revisión humana) y en la satisfacción de los clientes (se reduce el número de siniestros incorrectamente marcados como fraudulentos).

Por tanto, S-ADA permite tener un alto nivel de fraude detectado sin la necesidad de marcar como fraudulento un gran número de ellos. Esto se ve refrendado por la Exactitud el F1-Score, métricas en las que el modelo propuesto consigue resultados significativamente superiores a RAC y RAE.

Los modelos basados en reglas automáticas no ofrecen una probabilidad de fraude asociada a cada siniestro. Simplemente marcan los siniestros como fraude o no fraude. Las predicciones de los modelos basados en aprendizaje automático o IA son probabilidades de fraude. Normalmente, aquellos registros con una probabilidad inferior al $50\,\%$ son marcados como no fraude mientras que, el resto de ellos, son marcados como fraude. De hecho, el Cuadro 6 se ha construido considerando este umbral, que es el utilizado como regla general en los problemas de clasificación tratados con algoritmos de aprendizaje automático o IA.

Tomando como referencia el importe medio del fraude evitado (1,872.8€), el gasto medio

de investigación (61.3€) y la ratio de rendimiento (30.5€) mostrado en la subsección 2.1, se ha optimizado el umbral de S-ADA para obtener el máximo beneficio posible. El gasto medio de la investigación se ha penalizado de manera que se incremente en función del número de siniestros identificados como fraude por parte del modelo. A mayor sea este número, mayor será el gasto medio de investigación. Esta penalización trata de representar el daño que tendría el descontento de los clientes y la falta de eficiencia en los procesos asociados a la revisión de un gran número de siniestros. Su formulación es la siguiente:

$$Penalty(\,\%) = \theta * \frac{TP + FP}{TP + FP + TN + FN} * 100 \tag{42}$$

A mayor sea θ, mayor la penalización aplicada a los modelos por marcar como fraude siniestros que, en realidad, no lo son. En esta memoria se ha fijado $\theta = 0{,}2$. Por tanto, la cuenta de resultados de fraude a optimizar se construye de la siguiente manera:

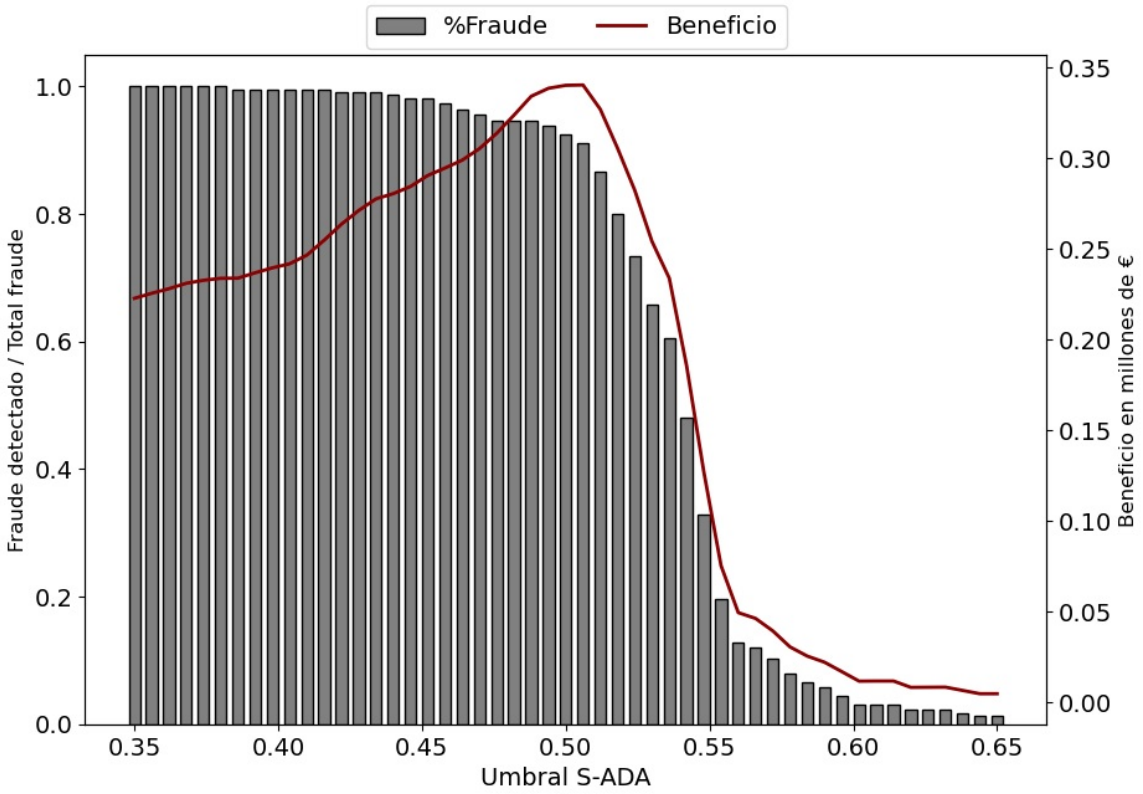
$$\begin{aligned} P\&L &= \sum_{i=1}^{N} 1.872{,}8 * I(y_i = \hat{y}_i = 1) - 61{,}3 * Penalty(\,\%) * I(\hat{y}_i = 1) = \\ &= \sum_{i=1}^{N} Ahorro_i - Coste_i \end{aligned} \tag{43}$$

Siendo $I(x = i)$ una variable que toma el valor de 1 cuando la condición $x = i$ se cumple y 0 en caso contrario.

La optimización del umbral de S-ADA se ha realizado calculando su $P\&L$ para diferentes umbrales. La Figura 12 muestra los resultados de este proceso de optimización. El umbral que maximiza el $P\&L$ es $0{,}506$, muy cercano al $0{,}5$ utilizado por defecto en el ámbito del aprendizaje automático y la IA. Tal y como se puede apreciar, aquellos porcentajes muy alejados del entorno del $0{,}5$ devuelven unos resultados significativamente menores. Esto se debe a que, cuando se baja mucho el umbral el modelo marca cómo fraude un gran número de siniestros que, en realidad, no lo son aumentando los costes. Por otro lado, cuando el umbral es muy elevado, el modelo deja de marcar como fraude muchos siniestros que sí lo son, minorando significativamente el beneficio asociado a la detección de estos siniestros fraudulentos.

Una vez hallado el umbral que optimiza los resultados económicos obtenidos por S-ADA, se compara sus resultados con los de RAC y RAE. Tal y como se ha comentado anteriormente, los modelos basados en reglas automáticas (sin el uso de IA o aprendizaje automático) no ofrecen un rango de probabilidad y, por tanto, no se puede llevar a cabo la optimización realizada para S-ADA. El Cuadro 7 muestra que, en términos de $P\&L$, los resultados de los modelos tradicionales son sensiblemente menores a los obtenidos con S-ADA. Los algoritmos y métodos aplicados en este modelo permiten que sea

Figura 12: Optimización del umbral de S-ADA



Fuente: Elaboración propia

capaz de identificar un alto porcentaje de fraude sin incurrir en un número elevado de falsos positivos (FP). Por tanto, es capaz de absorber la mayoría del beneficio asociado a la detección del fraude sin ser penalizado significativamente por el número de falsos positivos obtenidos.

Cuadro 7: Cuenta de resultados de fraude (en millones de €)

Modelo	Ahorro	Coste	P&L
RAC	0.365	0.382	-0.017
RAE	0.296	0.215	0.081
S-ADA	0.384	0.043	0.341

Elaboración propia

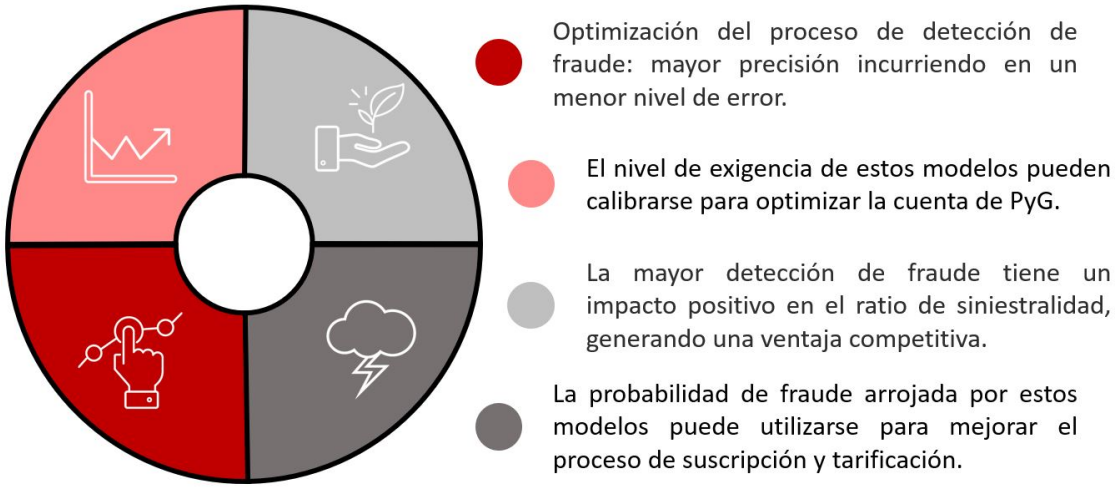
4. CONCLUSIONES

En línea con los resultados empíricos obtenidos en esta memoria, en la subsección 4.1 se presentarán los beneficios de la aplicación de IA y machine en el ámbito de la detección del fraude. Posteriormente (subsección 4.2), se profundizará en extensiones al trabajo presentado en esta memoria y en sus potenciales. beneficios.

4.1. Beneficios de la aplicación de IA en la detección de fraude

La Figura 13 muestra un resumen de las cuatro principales conclusiones o beneficios que se extrapolan de los resultados empíricos mostrados en la subsección 3.3.3. A lo largo de esta sección se profundizará en cada uno de ellos.

Figura 13: Conclusiones y beneficios del aprendizaje automático e IA



Fuente: Elaboración propia

La mejora en el porcentaje de fraude detectado es el primer y más inmediato de los beneficios que muestra el uso de algoritmos de aprendizaje automático e IA. Las diferentes métricas utilizadas en el análisis empírico muestran que el modelo basado en este tipo de algoritmos no sólo detecta más fraude, sino que además incurre en un nivel de error muy inferior a las reglas automáticas tradicionales. Esto permitirá a las entidades aseguradoras ser más precisos en la detección del fraude, de manera que no solamente se identificará más fraude, sino que además la cantidad de siniestros marcados incorrectamente como fraudulentos será menor. Esto permitirá que la revisión de estos siniestros

por el departamento de fraude sea más eficiente y que, adicionalmente, la satisfacción del cliente sea mayor ya que un menor número de ellos será investigado erróneamente. Es relevante remarcar que, la capacidad predictiva de los algoritmos utilizados en esta memoria permite incorporar rápidamente en los modelos de detección de fraude nuevas tendencias y comportamientos de los defraudadores.

En segundo lugar, y tal y como ya se ha comentado a lo largo de esta memoria, los modelos de fraude basados en IA y aprendizaje automático predicen, en función de sus características, la probabilidad de que sea fraude. Penalizando el error que comenten los modelos y haciendo uso de los gastos de investigación y del importe de los siniestros fraudulentos, se puede calibrar un umbral de probabilidad de fraude tal que optimice la cuenta de pérdidas y ganancias de la entidad aseguradora. Tal y como se muestra en los resultados empíricos mostrados en secciones anteriores, los modelos de detección de fraude basados en algoritmos de aprendizaje automático pueden llegar a repercutir unos beneficios significativamente superiores a los registrados por aquellos modelos que no hacen uso de ellos.

La tercera conclusión es que el uso de IA o el aprendizaje automático para la detección del fraude puede suponer una ventaja competitiva, ya que la mejora en las ratios de siniestralidad puede llegar a ser significativa. Tal y como se ha demostrado con los resultados empíricos de esta memoria, el beneficio obtenido es notablemente mayor gracias al mayor poder predictivo y al menor grado de error a la hora de predecir siniestros fraudulentos. Esto dará la posibilidad a la empresa aseguradora de repercutir este impacto a sus asegurados o a sus inversores. Por un lado, la entidad podría rebajar la tarifa repercutiendo este impacto en sus asegurados. Por otro lado, si la compañía mantiene las tarifas obtendrá un mayor beneficio por acción, teniendo un impacto positivo en los inversores a través de la cotización bursátil o el dividendo.

Por último, la probabilidad de fraude que devuelven los modelos basados en algoritmos como los presentados en esta memoria pueden utilizarse para mejorar los procesos de suscripción y tarificación. Las entidades aseguradoras podrían establecer un umbral de probabilidad de fraude a partir del cual no estarían dispuestas a suscribir un nuevo riesgo. En este caso, se está evaluando la probabilidad de fraude de un potencial nuevo cliente de la compañía aseguradora. Por tanto, las variables independientes que determinarán la probabilidad de fraude serán aquellas relativas a las características del asegurado y del bien cubierto. Al no existir siniestro, las variables que definen las características del cliente no estarán disponibles.

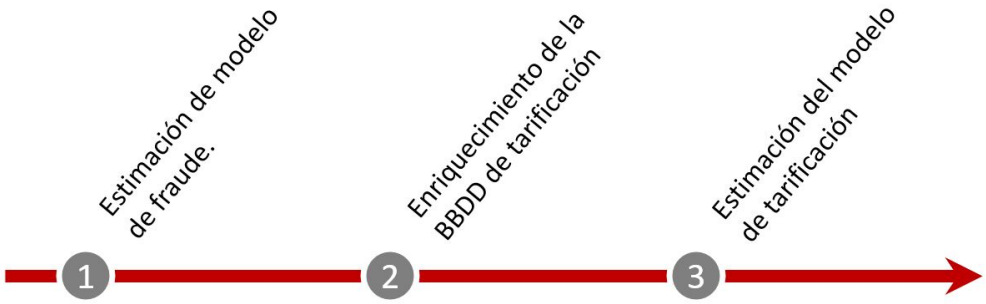
De manera adicional, los modelos de tarificación podrían incluir la probabilidad de fraude como una variable independiente más. Este último punto sobre la tarificación puede considerarse también como una futura línea de investigación y, por tanto, se profundiza más sobre ella en la siguiente subsección. En definitiva, el objetivo último de los departamentos de fraude de las entidades aseguradoras es que no haya fraude. Las medidas

preventivas focalizadas en la tarificación y suscripción tienen la ambición de minimizar este riesgo y optimizar los recursos destinados al fraude (a menor sea, menores gastos de gestión e investigación).

4.2. Futuras líneas de investigación

Tomando como base los resultados obtenidos en esta memoria, en esta subsección se comentarán brevemente dos posibles futuras líneas de investigación. La primera consistiría en analizar el potencial incremento de precisión del modelo propuesto al incluir como variables independientes algunos datos externos como pueda ser la información censal, la renta media del área o la situación macroeconómica cuando sucedió el siniestro. Este tipo de enriquecimiento de bases de datos ya lo hacen algunas entidades aseguradoras para mejorar la estimación de las tarifas de seguros de No Vida. Al igual que ocurre en el ámbito de la tarificación, la situación económica del asegurado y el contexto macroeconómico podrían ser datos relevantes para detectar o estimar la probabilidad de fraude asociada a los siniestros y, por tanto, tener un impacto positivo en la capacidad predictiva de los modelos.

Figura 14: Uso de la probabilidad de fraude en la tarificación



Fuente: Elaboración propia

La segunda futura línea de investigación, tal y como ya se ha adelantado en la previa subsección, sería la inclusión de las probabilidades de fraude como variable independiente de los modelos de tarificación. De manera muy simplificada, el proceso tendría los siguientes pasos:

1. Se estimará un modelo de fraude basado en algoritmos de aprendizaje automático e IA. Este modelo estimará la probabilidad de fraude de los asegurados y no

de siniestros individuales. Por tanto, las variables independientes que nutrirán este modelo serán, principalmente, las características del bien asegurado y cliente. Al no tratarse de un siniestro, las variables independientes que muestran las características del mismo no existirán.

2. La base de datos para la estimación de los modelos de tarificación se enriquecerá con la probabilidad de fraude estimada para cada asegurado. El modelo estimado en el punto anterior será el encargado de proveer esta información.

3. Por último, el modelo de tarificación será estimado considerando como variable independiente la probabilidad de fraude. De esta manera, la tarifa estimada para cada uno de los asegurados considerará implícitamente la probabilidad de que cometa fraude.

Referencias

AXA (2023). X mapa axa del fraude.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag.

Borisov, V., T. Leemann, K. Sessler, J. Haug, M. Pawelczyk, and G. Kasneci (2022). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21.

Bouchti, A. E., A. Chakroun, H. Abbar, and C. Okar (2017). Fraud detection in banking using deep reinforcement learning. In *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*, pp. 58–63.

Breiman, L. (2001, Oct). Random forests. *Machine Learning 45, 5–32.*

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Advances in Neural Information Processing Systems*, Volume 33, pp. 1877–1901. Curran Associates, Inc.*

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). Smote. synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16.

Chen, T. and C. Guestrin (2016). XGBoost: A scalable tree boosting system. pp. 785–794.

Devlin, J., M. Chang, K. Lee, and K. Toutanova (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805.*

Dixon, M., I. Halperin, and P. Bilokon (2020, 05). Machine Learning in Finance: From Theory to Practice*.*

Elsayed, S., D. Thyssens, A. Rashed, H. S. Jomaa, and L. Schmidt-Thieme (2021). Do we really need deep learning models for time series forecasting?

Freund, Y. and R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139.

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics 29, 1189–1232.*

Fu, K., D. Cheng, Y. Tu, and L. Zhang (2016). Credit card fraud detection using convolutional neural networks. In Neural Information Processing: 23rd International Conference,

ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part III 23*, pp. 483–490. Springer.*

Ghobadi, F. and M. Rohani (2016). Cost sensitive modeling of credit card fraud using neural network strategy. In 2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1–5.*

Hastie, T., R. Tibshirani, and J. Friedman (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York.*

Howard, A., M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le (2019). Searching for mobilenetv3. pp. 1314–1324.

ICEA (2023). El fraude al seguro español. estadística año 2022.

Igel, C. and M. Hüsken (2003). Empirical evaluation of the improved Rprop learning algorithm. Neurocomputing 50, 105–123.

Kim, H. and C. Won (2018). Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with applications 103, 25–37.*

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. CoRR abs/1412.6980.

Kristjanpoller, W. and M. Minutolo (2018). A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. *Expert Systems with Applications 109, 1–11.*

Lu, X., D. Que, and G. Cao (2016). Volatility forecast based on the hybrid artificial neural network and garch-type models. Procedia Computer Science 91, 1044 – 1049.

Mcculloch, W. and W. Pitts (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics 5, 127–147.*

Ramos-Pérez, E., P. Alonso-González, and J. Núñez-Velázquez (2019). Forecasting volatility with a stacked model based on a hybridized Artificial Neural Network. Expert Systems with Applications 129, 1–9.

Ramos-Pérez, E., P. J. Alonso-González, and J. J. Núñez Velázquez (2021). Multi-transformer: A new neural network-based architecture for forecasting s&p volatility. *Mathematics 9.*

Randhawa, K., C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi (2018). Credit card fraud detection using adaboost and majority voting. IEEE Access 6, 14277–14284.

Riedmiller, M. and H. Braun (1993). A direct adaptive method for faster backpropagation learning: The Rprop algorithm. pp. 586–591.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. pp. 318–362.

Rumelhart, D. E. and D. Zipser (1986). Feature discovery by competitive learning. pp. 151–193.

Srivastava, A., M. Yadav, S. Basu, S. Salunkhe, and M. Shabad (2016). Credit card fraud detection at merchant side using neural networks. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 667–670.

Tongesai, M., G. Mbizo, and K. Zvarevashe (2022). Insurance fraud detection using machine learning. In *2022 1st Zimbabwe Conference of Information and Communication Technologies (ZCICT)*, pp. 1–6.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. *CoRR abs/1706.03762.*

Vidal, A. and W. Kristjanpoller (2020). Gold volatility prediction using a cnn-lstm approach. Expert Systems with Applications 157.

Wu, J., B. Zhou, D. Peck, S. Hsieh, V. Dialani, L. Mackey, and G. Patterson (2018, 05). Deepminer: Discovering interpretable representations for mammogram classification and explanation.

Xia, H., Y. Zhou, and Z. Zhang (2022). Auto insurance fraud identification based on a cnn-lstm fusion deep learning model. *International Journal of Ad Hoc and Ubiquitous Computing 39, 37–45.*

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. ArXiv abs/1212.5701.

Zhang, Z., X. Zhou, X. Zhang, L. Wang, and P. Wang (2018). A model based on convolutional neural network for online transaction fraud detection. *Security and Communication Networks 2018.*

Zouboulidis, E. and S. Kotsiantis (2012). Forecasting fraudulent financial statements with committee of cost-sensitive decision tree classifiers. In I. Maglogiannis, V. Plagianakos, and I. Vlahavas (Eds.), Artificial Intelligence: Theories and Applications*, Berlin, Heidelberg, pp. 57–64. Springer Berlin Heidelberg.*