uc3m | Universidad Carlos III de Madrid

Máster Universitario en Ciencias Actuariales y Financieras
2024-2025

Trabajo Fin de Máster

"Más allá de las cajas negras: algoritmos interpretables de aprendizaje automático en la práctica actuarial"

Jorge Montejano Mariscal

Tutor/es

Raquel Pérez Calderón

Madrid, junio de 2025

RESUMEN

El incremento en la capacidad computacional, junto con la mayor disposición y calidad de datos ha favorecido el desarrollo de algoritmos de aprendizaje automático con una capacidad predictiva muy superior a los modelos tradicionales. No obstante, este incremento ha ido acompañado en la mayoría de los casos por un aumento de la complejidad de los modelos. Por tanto, en sectores críticos como la salud, finanzas y seguros se ha producido una demanda creciente por modelos que no sean sólo precisos, sino también interpretables y transparentes.

Así, el presente trabajo compara algoritmos de aprendizaje automático interpretativos con modelos *black-box* en el contexto de la práctica actuarial, evaluando tanto su rendimiento predictivo como su capacidad explicativa. Para ello, se ejecuta una aplicación práctica desarrollada en Python, utilizando datos reales del ramo de autos para predecir la frecuencia siniestral de los asegurados.

A través de este ejemplo se demuestra que los modelos interpretativos alcanzan una precisión competitiva mientras ofrecen explicaciones claras que facilitan la confianza, la adopción y la toma de decisiones informadas.

Palabras clave: Interpretabilidad; aprendizaje automático; modelización; transparencia.

ABSTRACT

The increase in computational power, along with greater availability and quality of data, has fostered the development of machine learning algorithms with predictive capabilities far superior to traditional models. However, this improvement has in most cases been accompanied by an increase in model complexity. Therefore, in critical sectors such as healthcare, finance, and insurance, there has been a growing demand for models that are not only accurate but also interpretable and transparent.

This work compares interpretable machine learning algorithms with black-box models in the context of actuarial practice, evaluating both their predictive performance and explanatory power. To this end, a practical application developed in Python is carried out, using real data from the auto insurance sector to predict policyholders' claim frequency.

Through this example, it is demonstrated that interpretable models achieve competitive accuracy while providing clear explanations that enhance trust, adoption, and informed decision-making.

Keywords: Interpretability; machine learning; modeling; transparency.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

1.1. Interpretabilidad en el aprendizaje automático

La irrupción de los algoritmos de aprendizaje automático y su adopción en los procesos de modelado de variables ha supuesto una eficiencia y precisión exponencialmente superior a lo observado previamente. No obstante, esto ha conllevado a su vez un incremento en la complejidad y opacidad en la operativa de los modelos, puesto que, a diferencia del modelado estadístico tradicional, el paradigma del aprendizaje automático ha estado dominado por un enfoque de maximización de la capacidad predictiva, dejando en un segundo plano la capacidad del modelo de extraer nuevo conocimiento en el contexto del campo de análisis sobre el que se aplica.

Esta opacidad implica que la adopción de estos procedimientos en sectores donde resulta indispensable que los procesos sean transparentes y justificables, debido al marco regulatorio en el que se encuadran, la complejidad inherente en la gestión del riesgo y las implicaciones éticas que subyacen las decisiones resulte cuestionada y no se hayan explotado sus ventajas como en otros sectores. Entre ellos se encuentra el sector asegurador, donde los modelos de aprendizaje automático pueden ofrecer estimaciones muy precisas de las probabilidades de siniestro y fraude, o la determinación de primas, entre otras muchas aplicaciones, pero con el riesgo de que los modelos que carecen de interpretabilidad pueden perpetuar inadvertidamente sesgos, discriminar a ciertos grupos demográficos y, en consecuencia, no respetar estándares éticos y regulatorios.

Esta interpretabilidad se define como la capacidad de los interesados de comprender el razonamiento detrás de las decisiones tomadas por modelos de aprendizaje automático, pudiendo determinar las causas que lo justifican, así como predecir y reproducir de forma inequívoca los resultados. (Kim, Khanna, & Koyejo, 2016) (Biran & Cotton, 2017)

La última década ha supuesto una revolución como campo de estudio del aprendizaje automático interpretativo, publicándose una amalgama de métodos que destacan su naturaleza como herramienta fundamental a garantizar por los modelos utilizados. Tres objetivos principales impulsan la necesidad de interpretabilidad: justificación de las decisiones, mejora del modelo y descubrimiento de nuevos conocimientos.

La interpretabilidad es crucial para garantizar que las decisiones automatizadas no se realicen de manera errónea o injustificada, facilitando la transparencia y responsabilidad en las interacciones entre las partes interesadas y proporcionando información que permite justificar los resultados obtenidos, asegurando un proceso auditable y defendible como ético y justo. Esto no solo genera confianza, sino que también facilita el cumplimiento regulatorio y la eliminación de potenciales sesgos.

Asimismo, la capacidad de explicar los modelos facilita considerablemente su mejora continua. Un modelo comprensible permite a los usuarios identificar claramente por qué el sistema produce determinados resultados, lo que a su vez facilita su optimización. Por

tanto, al disponer de un entendimiento claro de las decisiones del modelo, los usuarios pueden aportar ajustes específicos y eficaces, estableciendo un ciclo iterativo de mejora constante y cooperación entre humanos y modelos automatizados. Esta capacidad también permite ejercer un mayor control sobre ellos, puesto que entender el comportamiento del sistema facilita la detección temprana de vulnerabilidades y errores, posibilitando una rápida corrección en situaciones de baja criticidad. De este modo, la interpretabilidad contribuye directamente a la prevención de fallos, proporcionando herramientas para el diagnóstico y la depuración efectiva del modelo, y otorgando un control más exhaustivo sobre su funcionamiento.

Finalmente, la interpretabilidad contribuye al descubrimiento y generación de conocimientos más profundos sobre los datos, respaldando la toma de decisiones estratégicas más allá de la simple precisión predictiva. Esta capacidad tiene un gran potencial, puesto que entender las relaciones causales y los factores de riesgo clave que influyen en resultados como la pérdida de clientes o la siniestralidad en el caso del sector asegurador permite optimizar la asignación de recursos, las estrategias y potenciar la retención y satisfacción de los clientes.

1.2. Motivación y objetivos del trabajo

En el presente trabajo se estudian y proponen alternativas de métodos y modelos interpretativos, con objeto de contribuir al desarrollo de metodologías que aúnen el potencial predictivo del aprendizaje automático con la transparencia necesaria para garantizar su uso en el contexto del sector asegurador.

Para ello, en primer lugar, se establecerán las definiciones teóricas pertinentes de los métodos comprendidos en la metodología a aplicar, para posteriormente aplicar esta metodología a un caso práctico desarrollado en Python, cuyo código está presente en el anexo, utilizando datos reales de naturaleza aseguradora.

A continuación, se expondrán los resultados obtenidos bajo las distintas propuestas, sobre los que se procederá a analizar si constituyen alternativas efectivamente interpretativas y que pueden competir en desempeño con los modelos potentes pero que carecen de dicha transparencia.

2. METODOLOGÍA Y MARCO TEÓRICO

2.1. Modelo inicial (GLM)

Como base de la aplicación práctica del presente trabajo se ha optado por un modelo lineal generalizado (Generalized Linear Models, o GLM, por sus siglas en inglés). Introducidos por Nelder y wedderburn en 1972, los modelos lineales generalizados constituyen una herramienta ampliamente utilizada y estudiada en la modelización de datos. Representan una extensión flexible de los modelos lineales clásicos al eliminar la restricción de que la variable respuesta se distribuya como una Normal, permitiendo modelar también variables de respuesta que siguen otras distribuciones de la familia exponencial. (Nelder & Wedderburn, 1972)

Por tanto, los GLMs representan un marco que unifica varios modelos estadísticos, como son la regresión lineal, la logística, la regresión de Poisson o la Gamma, entre otros. Esta flexibilidad se sustenta en los tres elementos base que conforman todo modelo lineal generalizado:

I. Componente aleatorio: representa la distribución de la variable respuesta, que, aún encontrándose bajo el alcance de la familia exponencial, puede variar significativamente, desde ser continua y estrictamente positiva (distribución Gamma) a ser binaria (distribución de Bernoulli) o discreta y estrictamente positiva (distribución de Poisson), entre otras.

II. Componente sistemático: es la combinación lineal de las variables independientes escaladas por los parámetros Beta desconocidos, expresado como:

$$\eta = X\beta \ = \ \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

III. Función de enlace: expresada como $g(\cdot)$, relaciona el valor esperado de la variable respuesta ($\mu = E[Y|X]$) con el predictor lineal (η). Algunos ejemplos de funciones de enlace y las distribuciones con las que se relacionan habitualmente se incluyen en la tabla 1

Tabla 1. Funciones de enlace comunes en los modelos lineales generalizados

Distribución	Función de enlace	Forma de la función	Respecto al valor esperado
Normal	Identidad	$\mu = \eta$	$\mu = \eta$
Bernoulli	Logit	$ln\left(\frac{\mu}{1-\mu}\right) = \eta$	$\mu = \frac{1}{1+exp^{-\eta}}$
Poisson	Logarítmica	$ln(\mu) = \eta$	$\mu = exp^{\eta}$

Por tanto, un modelo lineal generalizado se define formalmente como:

$$g(E[Y|X]) = \eta = X\beta = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

O, en función del valor esperado de la variable respuesta:

$$E[Y|X] = \mu = g^{-1}(\eta)$$

El hecho de que se pueda analizar de forma directa el impacto marginal de la variable explicativa xi sobre la respuesta esperada, a través del coeficiente βi y la función de enlace empleada, convierte a los GLM en una herramienta no solamente versátil, como hemos expuesto, sino altamente interpretativa y ampliamente utilizada en las disciplinas más reguladas.

Por otro lado, este tipo de modelos presenta ciertas desventajas: es sensible a valores extremos en las observaciones, la estimación de los parámetros resulta severamente impactada por la presencia de multicolinealidad en las variables independientes, y requiere una correcta especificación de la distribución subyacente y la función de enlace.

Además, la facilidad interpretativa que supone la estimación de coeficientes fijos representa a su vez su mayor limitación, ya que no es capaz de capturar relaciones no-lineales entre las variables independientes y la variable respuesta. Si bien podríamos aplicar transformaciones a las variables independientes o categorizarlas para recoger esta relación no lineal, generalmente no supone una solución ´óptima ya que desvirtúa la interpretabilidad y sencillez del modelo, siendo esta su mayor virtud.

También podríamos destacar como inconveniente el que no considere de forma nativa el impacto de las interacciones entre variables independientes, aunque esto se puede solventar aplicando ingeniería de características (*feature engineering*) para incorporar al modelo nuevas variables que representen dichas interacciones.

2.2. Modelos de aprendizaje automático (Black-box)

El avance exponencial en las técnicas y potencia de computación, junto con un acceso a fuentes de datos de volúmenes cada vez mayores, ha propiciado la aparición de multitud de modelos altamente flexibles, lo que les permite recoger de forma efectiva relaciones no lineales, aumentando en gran medida el poder predictivo respecto a los algoritmos existentes previamente. Dada la naturaleza de este trabajo, resulta imposible abarcar una aplicación práctica de todos estos modelos, y tampoco es su fin, por lo que se trabajará sobre los algoritmos de modelado que trataremos en los sucesivos apartados.

2.2.1. Random Forest

El algoritmo de *Random forest,* introducido por Leo Breiman en 2001, representa un método de ensamblado (*ensembling*) útil tanto para tareas de regresión como clasificación. Se basa en la combinación de las predicciones realizadas por multitud de

árboles de decisión simples entrenados sobre submuestras aleatorias y con reemplazo del conjunto de datos. A su vez, en cada árbol de decisión se selecciona aleatoriamente un subconjunto determinado de las variables independientes. (Breiman, 2001)

En el caso de tareas de clasificación, la predicción final se deriva de la categoría en que la instancia haya sido clasificada en más ocasiones, mientras que para regresión (contexto para el que se utilizarán en el presente trabajo) se realiza un promedio de las estimaciones de todos los árboles, de acuerdo con la fórmula:

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

La dinámica de entrenamiento de múltiples árboles simples con muestra de datos y variables independientes aleatorias supone que cada árbol por separado no resulte fiable, pero al agregarlos en conjunto se logra abordar los problemas de sobreajuste (*overfitting*) y sensibilidad a la varianza de los datos que presentan otros algoritmos de aprendizaje automático, conduciendo a una generalización más estable sobre datos nuevos y, por tanto, a una mayor potencia predictiva.

No obstante, esta mejora sustancial en la capacidad predictiva respecto a algoritmos básicos pero interpretativos como los árboles de decisión simple (en los que se basa) supone una reducción en la interpretabilidad intrínseca del modelo.

2.2.2. Redes Neuronales

Las Redes Neuronales Artificiales (*Neural Networks*, *NN*, o *Artificial Neural Networks*, *ANN*, por sus siglas en inglés), inspiradas en el comportamiento del cerebro humano, son algoritmos computacionales que destacan por su versatilidad, permitiendo modelizar relaciones muy complejas y abstractas entre variables. Se trata, en esencia, de aproximadores universales de funciones, basándose en la actualización recursiva de los pesos de los parámetros mediante un proceso empírico por el que se minimiza la función de pérdida determinada (*backpropagation*).

Si bien el concepto de las redes neuronales artificiales no es reciente, son los algoritmos más beneficiados del incremento computacional y de acceso a grandes volúmenes de datos experimentados este siglo, puesto que se ha hecho posible reducir el coste computacional de entrenar estas estructuras complejas, demostrando su superior capacidad predictiva en la gran mayoría de problemas.

Se componen de nodos interconectados organizados en capas, donde cada nodo transforma la entrada que recibe mediante la aplicación de una función de activación, pasando esta salida a ser el input (o uno de ellos) del próximo nodo. La forma más básica de estos nodos, conocida como perceptrón, presenta un funcionamiento que se puede expresar matemáticamente como:

$$a = \sigma(\sum_{i=1}^{n} w_i x_i + b)$$

Donde *a* representa el output de la neurona, x_i son los diferentes inputs que recibe, w_i los pesos (entrenables) asignados a estos inputs, *b* el sesgo (también entrenable) y σ la función de activación aplicada.

Por tanto, una red neuronal con una capa oculta puede expresarse como:

$$\hat{y} = f(x) = \sigma(\sum_{j=1}^{m} w_j^{(2)} * \sigma\left(\sum_{i=1}^{n} w_{ij}^{(1)} x_i + b_j^{(1)}\right) + b^{(2)})$$

Su versatilidad proviene de la capacidad de personalización de las capas, funciones de pérdida y funciones de activación utilizadas, por lo que existen numerosos tipos de arquitecturas aplicadas generalmente a un tipo concreto de problemas o estructuras de datos.

Por su constitución y funcionamiento, representan una herramienta capaz de adaptarse a naturalezas de datos muy distintos, aprendiendo patrones complejos no explícitos, y con una gran robustez, puesto que generalizan con resultados muy positivos sobre nuevas muestras, especialmente si se aplican técnicas de regularización.

No obstante, como todo modelo, no están exentas de limitaciones, puesto que requieren grandes volúmenes de datos para mejorar las predicciones de algoritmos más sencillos, suelen conllevar recursos computacionales más costosos en tiempo y esfuerzo, y, a medida que se incrementa su complejidad, se vuelven más opacas y difíciles de interpretar.

2.3. Interpretabilidad y aprendizaje automático

Por lo general, una vez aplicados y entrenados los modelos más complejos, siempre que estos sean adecuados para el problema a tratar, se obtienen mejores predicciones que con algoritmos más simples, pero a cambio de renunciar a la capacidad de poder interpretar y explicar su funcionamiento y el cómo llegan a estas mejores predicciones.

Como se ha adelantado previamente, en contextos muy regulados y/o aquellos en los que la incertidumbre en torno al funcionamiento del modelo supone un riesgo no admisible, resulta crucial cambiar este paradigma.

Al ser un campo incipiente que no se había explorado en gran medida previamente, las innovaciones y descubrimientos en el ámbito de la interpretabilidad del aprendizaje automático son frecuentes y cada vez más prometedoras. Esto supone que existen ya numerosos métodos disponibles para el estudio (y otros surgirán mientras este trabajo se realiza), por lo que en el presente trabajo se evaluarán algunos de ellos, bien ampliamente aceptados o bien prometedores de cara a su evolución futura.

Si bien no existe un consenso claro en las definiciones de interpretabilidad aplicadas al aprendizaje automático, sí se observa un patrón común en la clasificación de las técnicas disponibles para facilitarla, que sigue la taxonomía expuesta en la figura 1.

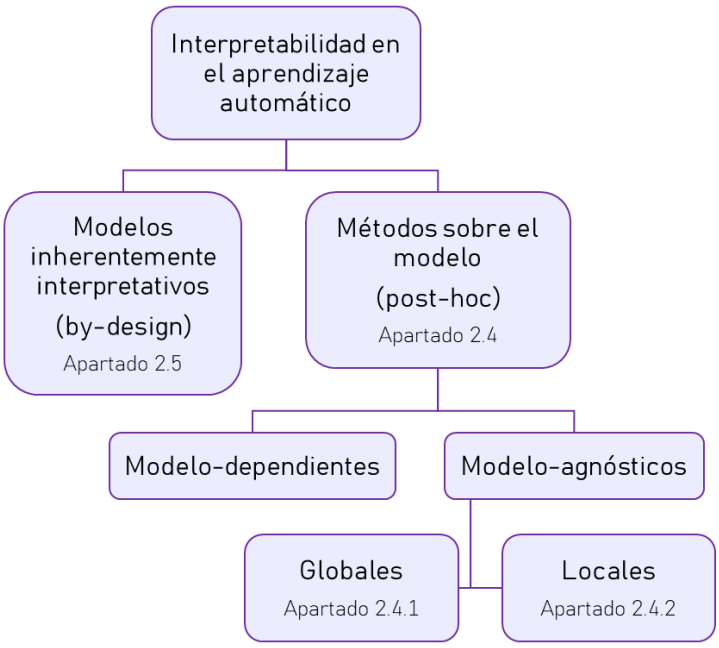

Figura 1. Clasificación de las herramientas de interpretabilidad. Fuente: Elaboración propia

En los siguientes apartados se abordan las características propias de estas divisiones y los métodos que se van a aplicar en el presente trabajo.

2.4. Métodos de interpretabilidad sobre el modelo (*post-hoc*)

La interpretabilidad sobre el modelo (generalmente referida en la literatura como *post-hoc*) implica que las herramientas que se engloban en esta categorización son aplicadas una vez el modelo (en este caso no interpretativo, o *black-box*, como se les suele denominar) ya ha sido entrenado, por lo que son útiles para entender y explicar los resultados, aunque no aportan conocimiento sobre el proceso que ha conducido a dichos resultados.

Los métodos *post-hoc* se pueden dividir en dos categorías: modelo-dependientes, que son específicos de un tipo de modelo, analizando partes concretas del mismo que impactan en mayor o menor medida al resultado; y modelo-agnósticos, donde lo que se analiza es cómo varían las predicciones del modelo cuando se efectúan cambios en las variables independientes, ignorando el tipo de modelo empleado en el proceso.

En este trabajo se ha optado por incluir métodos modelo-agnósticos, puesto que proporcionan una mayor flexibilidad y, a su vez, una base comparativa al poder efectuarse sobre cualquier tipo de modelo, estableciendo un procedimiento común.

A su vez, estos métodos modelo-agnósticos se pueden clasificar en locales y globales.

2.4.1. Métodos modelo-agnósticos globales

Aquí se engloban los métodos que permiten describir el comportamiento promedio del modelo, por lo que resultan útiles para entender y mejorar el modelo en su conjunto.

2.4.1.1. Importancia de las variables

Este método clasifica por orden de importancia sobre las predicciones del modelo las variables independientes. Introducidos por Aaron Fisher, Cynthia Rudin y Francesca Dominici en 2019, se basa en la idea de la medida de la importancia de variable por permutación para los bosques aleatorios propuesta por Leo Breiman (Breiman, 2001), extendiéndola a una versión independiente del modelo utilizado. Además, permite utilizar diferentes métricas como comparativa y que se ejecute sobre datos no utilizados para el entrenamiento (como el *test set*), lo que potencia la robustez de los resultados especialmente en modelos con sobreajuste. (Fisher, Rudin, & Dominici, 2019)

El algoritmo se ejecuta a través de tres pasos diferenciados:

I. En primer lugar, se evalúa el error original del modelo sobre el conjunto de datos (recomendable usar datos que no hayan sido usados para el entrenamiento) a través de la métrica elegida, ya sea el coeficiente de determinación, el error cuadrático medio, la precisión, etcétera (en función de la naturaleza de los datos y el tipo de tarea en cuestión).
II. A continuación, se itera sobre cada variable independiente j, permutando sus valores de forma aleatoria y manteniendo el resto de las variables constantes. Esto permite romper su relación con la variable dependiente y generar una nueva matriz de datos con las que se obtienen nuevas predicciones, sobre las que se evalúa el error en función de la métrica elegida. Este nuevo error se utiliza para estimar el cambio (como diferencia o porcentaje) que supone respecto al error original, lo que representa la importancia relativa de la variable.
III. Por último, se ordenan las variables en orden descendiente en función de su importancia relativa computada en el paso previo.

Así, las variables que suponen un error sustancialmente mayor al original al permutarlas tienen una importancia relativa elevada, mientras que, si este error no cambia materialmente, la variable en cuestión no es importante en la predicción del modelo.

Este método es sencillo de interpretar e implementar y captura de forma automática las interacciones, puesto que, si dos variables tienen un efecto conjunto elevado sobre la predicción, al alterar cualquiera de ellas el error aumentará considerablemente.

2.4.1.2. PDP

Los gráficos de dependencia parcial (*Partial Dependency Plots*, o PDP, por sus siglas en inglés), introducidos por Jerome Friedman en 2001, muestran los efectos marginales de las variables independientes sobre la predicción del modelo, manteniendo constantes las demás. (Friedman, Greedy Function Approximation: A Gradient Boosting Machine, 2001)

Para una variable independiente (X_i) determinada, su PDP se calcula como:

$$PDP(X_i) = E\left[\hat{f}(X_i, X_{-i})\right] = \frac{1}{n}\sum_{j=1}^{n} \hat{f}(x_{ij}, x_{-ij})$$

Donde X_i es la variable en cuestión, X_{-i} el conjunto del resto de variables, $\hat{f}()$ es la predicción del modelo, x_{ij} el valor que toma variable X_i para la instancia j, y x_{-ij} el valor j-ésimo del resto de variables independientes.

Es decir, se trata de obtener la curva que representa el valor promedio de las n curvas obtenidas al calcular las predicciones que realiza el modelo al iterar sobre cada instancia cambiando el valor de la variable i por el rango de dicha variable, manteniendo el resto constante.

Representan un método muy intuitivo y sencillo de implementar, aunque según el modelo puede ser computacionalmente ineficiente, además de asumir una independencia entre las variables que no necesariamente se cumple. Permite determinar si existe una relación monótona, compleja o si no existe relación (si la curva observada es constante) entre la variable en cuestión y las predicciones.

En cualquier caso, en conjunto con la visualización de la importancia de las variables permite formar una imagen de qué variables tienen un mayor impacto en las predicciones del modelo.

2.4.2. Métodos modelo-agnósticos locales

Este tipo de herramientas se enfoca en explicar predicciones concretas del modelo, lo que resulta útil para analizar casos de interés, como un perfil determinado o casos extremos.

Dentro de este grupo encontramos métodos que permiten explicar las predicciones como la suma de efectos de las variables independientes, y otros que se centran en una única variable independiente en concreto y estudian la sensibilidad del proceso predictivo ante sus variaciones. En el desarrollo de este trabajo se aplicarán herramientas recogidas en la primera definición.

2.4.2.1. LIME

El método de explicaciones interpretativas locales modelo-agnósticas (*Local Interpretable Model-agnostic Explanations* o LIME, por sus siglas en inglés), fue publicado en 2016 por Marco Tulio ribeiro, Sameer Singh y Carlos Guestrin. En esencia, se trata de entrenar un modelo simple para interpretar la predicción de una instancia concreta. (Ribeiro, Singh, & Guestrin, 2016)

El algoritmo que sigue se puede dividir en cuatro pasos:

I. Una vez seleccionada la instancia para la que se quiere explicar la predicción del modelo no interpretativo (*black-box*), se genera un *dataset* de muestras similares

aleatorias, generadas a partir de una distribución normal con media y desviación típica derivadas de cada variable independiente, y se calcula la predicción del modelo para cada una de estas pseudo-muestras.

II. A continuación, las pseudo-muestras generadas se ponderan a través de un *kernel* que se reduce conforme la distancia respecto a la instancia original decrece.

III. Después, se entrena el modelo interpretable (por ejemplo, una regresión Lasso o un árbol de decisión) sobre el nuevo conjunto de datos (nuevas muestras y las predicciones respectivas) minimizando la siguiente función de pérdida:

$$L(f, g, \pi_{x_0}) = \sum_i \pi_{x_0}(x_i)(f(x_i) - g(x_i))^2$$

Donde $\pi_{x_0}(x_i)$ representa las ponderaciones por distancia a cada pseudo-muestra, $f(x_i)$ y $g(x_i)$ son las predicciones del modelo no interpretativo y del modelo simple a entrenar para cada pseudo-muestra, respectivamente.

IV. Una vez entrenado el modelo simple, se obtienen los coeficientes o reglas, permitiendo interpretar el impacto de cada variable sobre la predicción del modelo *black-box* sobre la instancia original.

El output de este método es sencillo de interpretar, además de flexible, puesto que se pueden usar diferentes modelos interpretativos, y funciona para diferentes tipos de datos. No obstante, la elección de la función de distancia es crucial, pudiendo alterar significativamente los resultados observados.

2.4.2.2. SHAP

El método SHAP (*SHapley Additive exPlanations*), propuesto por Scott Lundberg y Su-In Lee en 2017 a partir de los valores óptimos derivados de la teoría de juegos de Shapley de 1953, estima la contribución que tiene cada variable independiente sobre la predicción final del modelo para una instancia concreta. (Lundberg & Lee, 2017)

Intuitivamente, la predicción del modelo representa la recompensa del juego en el que cada variable independiente actúa como un jugador, y SHAP distribuye esta recompensa de forma justa en función de cuánto ha contribuido cada jugador, considerando todas las combinaciones posibles.

La fórmula que lo define es la siguiente:

$$f(x) \simeq g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

Donde $f(x)$ es la predicción del modelo no explicativo para una instancia concreta, $g(z')$ es la predicción del modelo explicativo que aproxima el método, $\phi_0 = E[f(x)]$ es la media (esperanza) de las predicciones sobre las que se ha entrenado el modelo, z' es un vector binario que indica la presencia o ausencia de la variable (1 o 0), y ϕ_j es la contribución de la variable j.

La aproximación, y no igualdad entre *f* y *g* deriva de que el cálculo de los valores exactos de Shapley es computacionalmente muy costoso para un número de variables no pequeño, por lo que en la implementación de SHAP en la mayoría de software estos valores se estiman a través de muestreo con Monte-Carlo siguiendo la fórmula:

$$\phi_j = \frac{1}{N}\sum_{n=1}^{N}(\hat{f}\left(x_{+j}^{(n)}\right) - \hat{f}\left(x_{-j}^{(n)}\right))$$

En la que, dada una instancia específica *x*, $\hat{f}\left(x_{+j}\right)$ es la predicción del modelo original para esta, pero habiendo sustituido un número aleatorio de variables por los valores de dichas variables de otra instancia aleatoria *z*, salvo por el valor que toma la variable *j* que se mantiene. El vector de valores x_{-j} es casi idéntico a x_{+j} salvo porque el valor de la variable *j* también se sustituye por el de la instancia *z*, y se calcula su predicción. Se computa la diferencia entre ambas predicciones y se repite el proceso *N* veces, calculando el valor promedio de las diferencias, que es el estimador del valor de Shapley para esa variable independiente. (Molnar, 2025)

Este método proporciona explicaciones precisas y consistentes para instancias específicas, identificando qué variables tienen impactos positivos o negativos sobre la estimación promedio que se toma como *baseline*.

Además, la implementación en Python permite agregar los valores Shapley de numerosas instancias y obtener una visión global del comportamiento del modelo, derivada de un estudio de las predicciones individuales. Destaca la funcionalidad "*SHAP summary plot*", que combina los valores Shapley de cada instancia para cada variable y los agrega en un gráfico que permite interpretar la importancia de las variables y sus efectos simultáneamente, haciendo que este método se solape en cierta medida con el subgrupo descrito anteriormente.

2.4.3. Modelo sustituto

Si se tiene un modelo que presenta un poder predictivo elevado, pero que no es interpretativo (*black-box*), se puede entrenar un modelo más sencillo y que sea inherentemente interpretativo sobre las predicciones del modelo complejo.

Esto permite explicar las predicciones que realiza el modelo principal de forma sencilla, en función del modelo utilizado como sustituto. Simplemente se entrena este minimizando el error sobre las predicciones del modelo no interpretativo, en lugar de sobre los valores reales de la variable dependiente. A través de métricas como el coeficiente de determinación (R^2) se puede medir fácilmente la capacidad del modelo sustituto de aproximar las predicciones del modelo complejo.

La flexibilidad de este método es una de sus ventajas principales, junto con su sencillez, aunque siempre sujetas a la elección que se haga de modelo sustituto.

2.5. Modelos inherentemente interpretativos (*by-design*)

En esta categoría se recogen diversos modelos, cuya característica común es que permiten entender de forma simple y exacta a juicio humano el proceso que lleva a una serie de inputs *i* (una instancia) a convertirse en un *output* (una predicción) que se puede explicar bajo un mismo criterio para todos los *inputs*.

Los modelos interpretativos por diseño presentan ventajas a la hora de corregir sus fallos y mejorarlos al ser conscientes de cómo funcionan sus mecanismos internos. De igual forma, resultan fácilmente justificables y revisables, ofreciendo a los expertos en la materia de estudio pertinente la capacidad de determinar si el proceso es consistente con el conocimiento que se tienen del campo. (Molnar, 2025)

2.5.1. EBM

El Algortimo EBM (*Explainable Boosting Machine*), introducido junto al paquete de Python InterpretML por parte del equipo de investigación de Microsoft en 2019, es un tipo de modelo aditivo generalizado que utiliza árboles de decisión impulsados por gradientes para aprender las funciones de cada variable. Además, la segunda mejora sustancial respecto a los modelos aditivos clásicos es que detecta e incluye automáticamente el efecto de interacciones por pares de variables independientes que más reducen el error residual, aunque en la implementación en Python esta característica se puede descartar. (Nori, Jenkins, Koch, & Caruana, 2019)

Así, combina el poder predictivo de los algoritmos modernos de aprendizaje automático manteniendo la interpretabilidad implícita de los modelos aditivos, definiéndose como:

$$g(E[y]) = \beta_0 + \sum f_j(x_j) + \sum f_{ij}(x_i, x_j)$$

Cada función de las variables independientes se estima mediante un proceso de entrenamiento iterativo de árboles de decisión impulsados por gradientes, tal y como se expone en la figura 2.

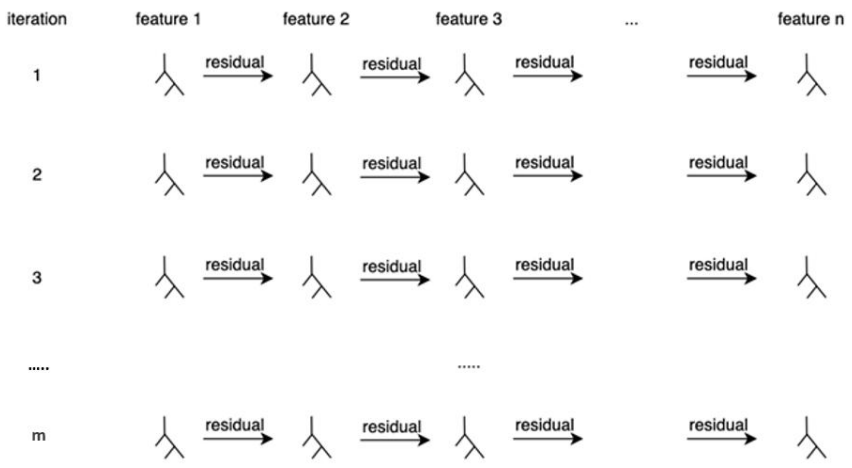

Figura 2. Esquema del algoritmo de entrenamiento del modelo EBM. Fuente: Elaboración propia

Es decir, se comienza la primera iteración entrenando un árbol de decisión sencillo que únicamente puede utilizar la primera variable independiente, procediéndose a continuación a actualizar el residuo (siguiendo la mecánica *gradient boosting*) y a entrenar un nuevo árbol de decisión sencillo que solo puede usar la siguiente variable, así hasta haber entrenado *m* árboles de decisión sencillos, uno por cada variable. Este proceso se repite hasta completar el número de iteraciones especificado, por lo que al final del proceso que se observa en la figura 3 se tendrán *m* árboles de decisión para cada variable. Para cada uno de estos árboles, se extraen los outputs que representan para cada instancia (rango completo de la variable) y se agregan, sustituyéndose así por una función discreta, obteniendo una por cada variable:

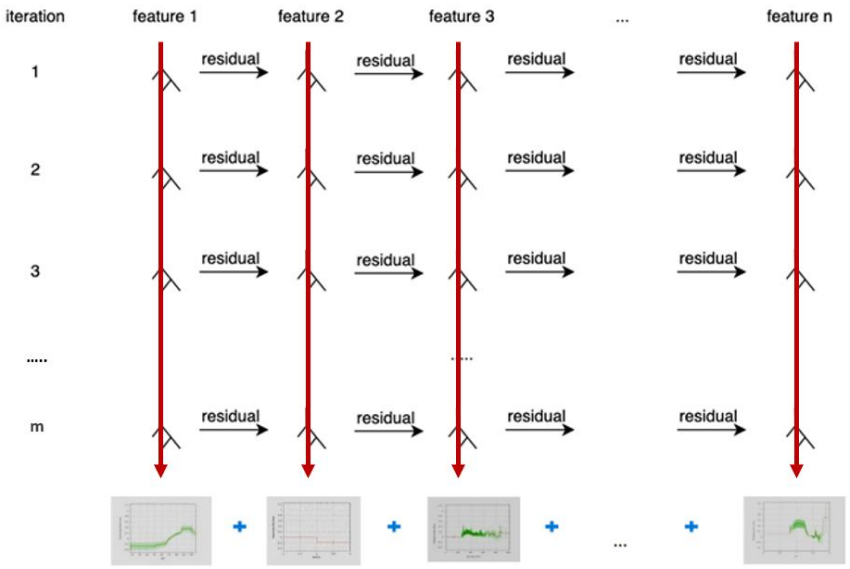

Figura 3. Proceso de entrenamiento del modelo EBM. Fuente: Elaboración propia

Por tanto, para cada variable se obtiene una función que actúa de forma independiente en la predicción, permitiendo la plena interpretabilidad del modelo y actuando como una tabla de consulta que devuelve el valor de contribución a la predicción dado el valor que toma la instancia correspondiente, como se puede observar en la figura 4.

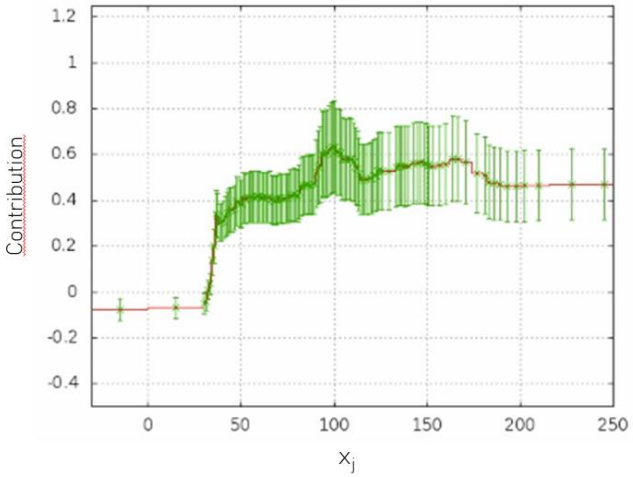

Figura 4. Contribución marginal de la variable independiente j en el EBM. Fuente: Elaboración propia

Cabe remarcar que el orden de las variables es indiferente gracias a que se utiliza una tasa de aprendizaje muy baja, por lo que se tienen que utilizar numerosas iteraciones en cualquier caso y al realizarse este alto número de iteraciones focalizando cada variable, se mitiga también el posible efecto de la colinealidad entre variables.

Tras haber obtenido las funciones para cada variable independiente, se evalúan y seleccionan los pares de interacciones más relevantes en función de los errores residuales aún presentes, que contienen patrones no explicados por los efectos de las variables independientes.

$$r = y - \hat{y}_{additive} = y - (\beta_0 + \sum_j f_j(x_j))$$

Sobre estos residuos se entrenan árboles simples impulsados por gradientes, usando solo pares de variables x_i y x_j, evaluando si se produce una mejora (reducción) en la función de pérdida, computando para cada par de variables su indicador de importancia:

$$s_{j,k} = L_{additive} - L_{additive+f_{j,k}}$$

Siendo L la función de pérdida del modelo. Se ordenan por importancia las parejas de variables y se seleccionan sólo el top k parejas (siendo k un hiperparámetro que calibra la complejidad del modelo).

Una vez se han seleccionado las principales parejas, se repite el proceso de *gradient boosting* pero en un espacio bidimensional, obteniendo las funciones pertinentes que se incluyen también como componente aditivo al modelo, manteniendo la interpretabilidad puesto que se pueden visualizar y estudiar como un mapa de calor en 2D, representado en la figura 5, gráficos de contorno o una función en tres dimensiones.

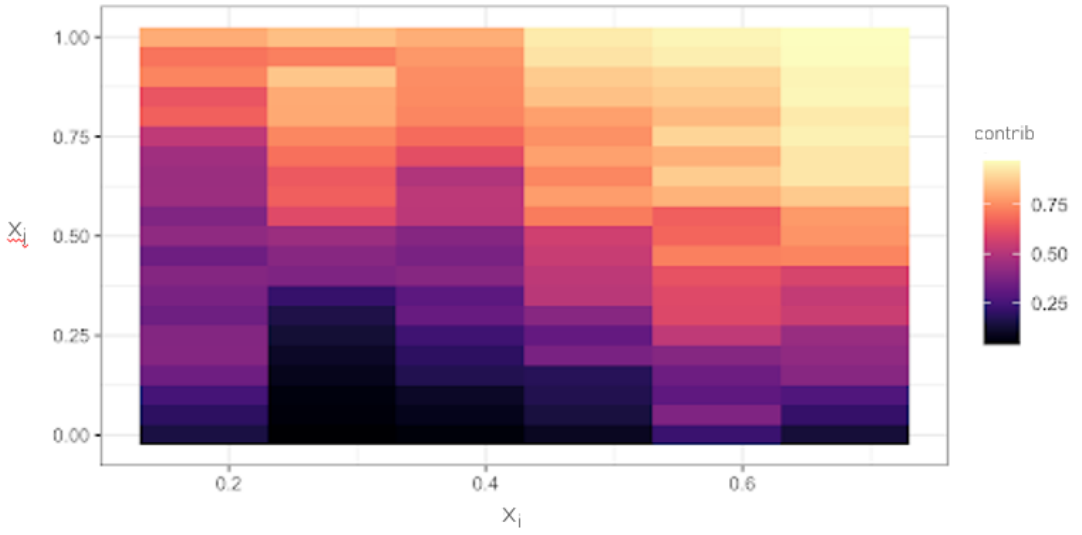

Figura 5. Contribución marginal de interacciones en el EBM. Fuente: Elaboración propia

2.5.2. NAM

De forma análoga a los EBM y el *gradient boosting*, los modelos neuronales additivos (Neural Additive Models, o NAM, por sus siglas en inglés) combinan la potencia de las redes neuronales junto con la interpretatividad de los modelos generales aditivos. Este modelo fue propuesto en 2021 por un conjunto de investigadores compuesto, entre otros, por varios de los miembros que desarrollaron el EBM.

Así, como se representa en la figura 6, sobre cada variable independiente se entrena a la vez una red neuronal profunda, obteniéndose el output (predicción del modelo) tras aplicar la función de activación σ pertinente a la agregación de los distintos términos independientes de estas redes neuronales junto con el sesgo β.

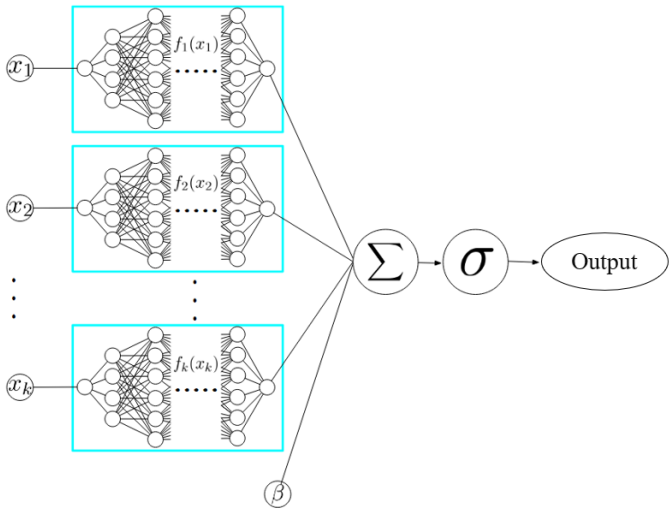

Figura 6. Esquema del modelo NAM. Fuente: Elaboración propia

De forma que se minimiza la función de pérdida actualizando los parámetros de todas las redes en conjunto, en lugar de ir entrenando árbol por árbol como en el modelo anterior. Una vez entrenadas las redes, estas se sustituyen por las funciones obtenidas:

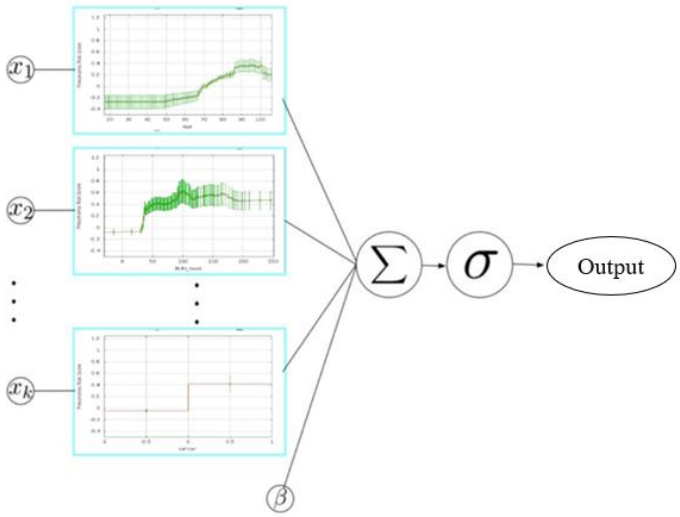

Figura 7. Funciones de contribución por variable en el modelo NAM. Fuente: Elaboración propia

Si se quiere incluir términos de interacción estos se pueden entrenar en conjunto con el resto de las redes de las variables independientes o tras haber obtenido las funciones entrenadas para estas.

Como ventajas de este modelo cabe destacar que las redes neuronales presentan mayor escalabilidad que los modelos basados en árboles, permitiendo, por ejemplo, el desarrollo de modelos de computación paralela de altas prestaciones. Además, son extremadamente flexibles, permitiendo modificar cualquier aspecto intermedio para adaptarse a datos o problemáticas diversos y complejos.

2.5.3. RuleFit

Publicado en 2008 por Jerome Friedman y Bogdan Popescu, el algoritmo RuleFit se basa en un modelo de regresión con restricciones o escaso (*sparse linear model*), que incluye tanto las variables originales como otras nuevas que capturan las interacciones a través de reglas binarias extraídas de árboles de decisión. (Friedman & Popescu, 2008)

Para generar estas nuevas variables independientes, se entrenan algoritmos como Random Forest para predecir la variable a explicar a través de la generación de muchos árboles de decisión, unificándose los caminos que toman cada uno de estos árboles hasta llegar al nodo de predicción final y transformándolos así en reglas de decisión. La figura 6 recoge de manera visual el proceso de creación de reglas expuesto.

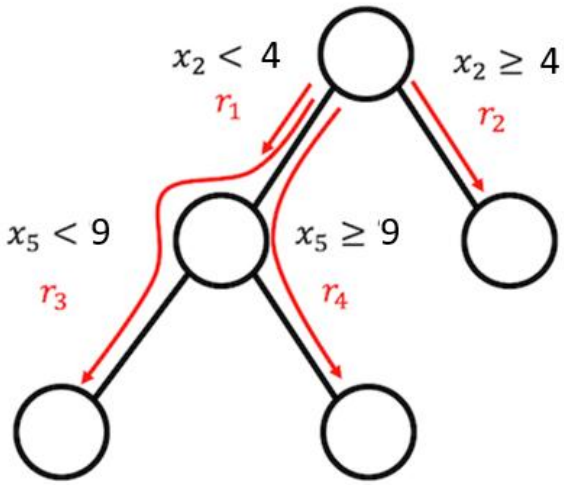

Figura 8. Derivación de reglas a través de un árbol de decisión. Fuente: Elaboración propia

De este árbol de ejemplo, por tanto, extraeríamos cuatro reglas de decisión (r1a r4) que siguen la lógica de, en el caso de r4: SI $X_2 < 4$ Y $X_5 >= 9$ ENTONCES 1 SI NO 0

Este proceso se repite para cada uno de los árboles entrenados hasta obtener *K* reglas binarias (la variable solo puede tomar el valor 1 si se cumplen todas las condiciones de la regla o 0 en caso contrario). Las predicciones de los árboles originales no se tienen en cuenta, ya que el fin es utilizar las reglas generadas como nuevas variables a incluir en un modelo de regresión junto con las variables originales.

La inclusión de las variables originales resulta importante dado que los modelos basados en árboles tienen el inconveniente de no representar bien las relaciones lineales simples existentes entre la variable respuesta y las explicativas. No obstante, antes de incluirlas en el modelo, estas se *winsorizan* en función de la siguiente fórmula:

$$l_j^*(x_j) = \min\left(\delta_j^+, \max\left(\delta_j^-, x_j\right)\right)$$

Donde, fijado un valor de delta, por ejemplo 0.01, aquellas observaciones que se encuentren por encima del percentil 99% o por debajo del 1%, tomarán el valor de dicho percentil, mitigando el efecto de los *outliers*. Tras esto, también se normalizan las variables lineales para ajustarse a la importancia relativa de las reglas de decisión, de acuerdo con la fórmula:

$$l_j(x_j) = 0{,}4 \times \frac{l_j^*(x_j)}{\text{std}\left(l_j^*(x_j)\right)}$$

Así, el procedimiento anterior completa la primera parte del proceso, que realmente solo es una transformación de las variables. La segunda parte del proceso consiste en la aplicación de un modelo de regresión Lasso (*Least Absolute Shrinkage and Selection Operator*, por sus siglas en inglés) de acuerdo con la siguiente fórmula:

$$\phi_j\, \hat{f}(X) = \hat{\beta}_0 + \sum_{k=1}^{K} \hat{\alpha}_k\, r_k\,(X) + \sum_{j=1}^{p} \hat{\beta}_j\, l_j\,(x_j\,)$$

Donde $\hat{\alpha}_k$ es el peso estimado para la regla de decisión k y $\hat{\beta}_j$ el de la variable independiente j. Se utiliza este tipo de regresión dado el elevado número de reglas que se obtienen en el primer paso, puesto que introduce restricciones a la función de pérdida aplicada, permitiendo que determinados parámetros (pesos) estimados tomen el valor 0, reduciendo en la práctica las variables y reglas efectivas, además de prevenir el sobreajuste.

La principal ventaja de este modelo es que mantiene la interpretabilidad sencilla de los modelos lineales, pero mejorando su poder predictivo, gracias a la inclusión de los términos de interacción en base a reglas, que resultan fácilmente integrables en cualquier software actuarial.

Por otro lado, también representa un algoritmo muy flexible, pudiéndose establecer diversas restricciones en la generación de los árboles para establecer el número y la complejidad deseada de las reglas.

3. BASE DE DATOS

3.1. Descripción y obtención

Los datos que se utilizarán para el entrenamiento y evaluación de los distintos modelos propuestos en el presente trabajo provienen del paper "*Dataset of an actual motor vehicle insurance portfolio*", de Jorge Segura-Gisbert, Josep Lledó y Jose M. Pavía, de la Universidad de Valencia, publicado en 2025 en la *European Actuarial Journal*. La base de datos, conformada por 105,555 filas y 30 variables, proviene de una muestra real (anonimizada de acuerdo con los estándares de protección de datos) de una aseguradora de no vida española para el período comprendido entre finales de 2015 y 2018, abarcando tres años completos. (Segura-Gisbert, Lledó, & Pavía, 2025)

Al acceder al repositorio con el enlace provisto en el artículo se puede descargar un zip que contiene tres archivos con la información descrita en dicho *paper*. El primero de ellos, denominado "*Descriptive of the variables*" contiene dos pestañas, cada una de las cuales describe qué representan las variables contenidas en los otros dos ficheros de la carpeta, "*Motor vehicle insurance data*" y "*sample type claim*". En las tablas 2 y 3 se recogen las variables encontradas en estos archivos, respectivamente.

Tabla 2. Variables originales del dataset

Variable	**Descripción**
ID	Número interno de identificación asignado a cada contrato anual formalizado por un asegurado. Cada tomador puede tener varias filas en el conjunto de datos, representando diferentes anualidades del producto.
Date_start_contract	Fecha de inicio del contrato del tomador (DD/MM/AAAA).
Date_last_renewal	Fecha de la última renovación del contrato (DD/MM/AAAA).
Date_next_renewal	Fecha de la próxima renovación del contrato (DD/MM/AAAA).
Distribution_channel	Clasifica el canal a través del cual se contrató la póliza. 0 para agente y 1 para corredores de seguros.
Date_birth	Fecha de nacimiento del asegurado declarada en la póliza (DD/MM/AAAA).
Date_driving_licence	Fecha de expedición del permiso de conducir del asegurado (DD/MM/AAAA).
Seniority	Número total de años que el asegurado ha estado asociado con la entidad aseguradora, indicando su nivel de antigüedad.
Policies_in_force	Número total de pólizas que el asegurado tiene en vigor en la entidad aseguradora durante el período de referencia.

Max_policies	Número máximo de pólizas que el asegurado ha tenido en vigor con la entidad aseguradora.
Max_products	Número máximo de productos que el asegurado ha tenido simultáneamente en algún momento.
Lapse	Número de pólizas que el cliente ha cancelado o han sido canceladas por impago en el año actual de vigencia, excluyendo aquellas que han sido reemplazadas por otra póliza.
Date_lapse	Fecha de cancelación. Fecha de terminación del contrato (DD/MM/AAAA).
Payment	Último método de pago de la póliza de referencia. 1 representa un proceso administrativo semestral y 0 indica un método de pago anual.
Premium	Importe neto de la prima asociado a la póliza durante el año en curso.
Cost_claims_year	Coste total de siniestros de la póliza durante el año en curso.
N_claims_year	Número total de siniestros ocurridos en la póliza durante el año en curso.
N_claims_history	Número total de siniestros presentados a lo largo de toda la duración de la póliza.
R_Claims_history	Ratio del número de siniestros presentados para la póliza específica respecto a la duración total (en años completos) de la póliza en vigor. Indica la frecuencia histórica de siniestros.
Type_risk	Tipo de riesgo asociado a la póliza. Cada valor corresponde a un tipo de riesgo: 1 para motocicletas, 2 para furgonetas, 3 para turismos y 4 para vehículos agrícolas.
Area	Variable dicotómica que indica la zona. 0 para rural y 1 para urbana (más de 30 000 habitantes) en términos de condiciones de tráfico.
Second_driver	1 si hay varios conductores habituales declarados, o 0 si solo se declara un conductor.
Year_matriculation	Año de matriculación del vehículo (AAAA).
Power	Potencia del vehículo medida en caballos de fuerza.
Cylinder_capacity	Cilindrada del vehículo.
Value_vehicle	Valor de mercado del vehículo al 31/12/2019.
N_doors	Número de puertas del vehículo.

Type_fuel	Tipo específico de combustible usado para alimentar el vehículo. Gasolina (P) o Diésel (D).
Length	Longitud del vehículo, en metros.
Weight	Peso del vehículo, en kilogramos.

Este archivo constituye la base de datos descrita, compuesta por variables típicas en la operativa aseguradora, puesto que corresponde a un conjunto de datos real, cuyo contenido servirá al objeto de estudio del presente trabajo. Por otro lado, se encuentra el archivo "*sample type claim.csv*", comprendido por las variables recogidas en la tabla 3.

Tabla 3. Variables del archivo de muestras por tipos de siniestro

Variable	**Descripción**
ID	Número interno de identificación asignado a cada contrato anual formalizado por un asegurado. Cada tomador puede tener varias filas en el conjunto de datos, representando diferentes anualidades del producto.
Cost_claims_year	Coste total de siniestros incurridos para la póliza de seguro durante el año en curso.
Cost_claims_by_type	Coste total de siniestros por tipo para la póliza de seguro durante el año en curso.
Claims_type	Nueve tipos: asistencia en viaje, lunas rotas, reclamación, negligencia, robo, incendio, todo riesgo, lesiones, otros.

Si bien este archivo podría suponer un análisis muy interesante, ya que separa por naturaleza los siniestros y sus respectivos costes, lo que permitiría analizar el riesgo a nivel granular (por cobertura), su aplicación efectiva no se contempla en este trabajo, puesto que se trata de una muestra pequeña de estos siniestros, lo que dificulta la obtención de modelos significativos.

De cualquier forma, la carencia de datos públicos no sintéticos de la operativa del sector asegurador, junto con el amplio abanico de posibilidades de estudio y análisis que aporta la diversidad de variables contenidas, hacen de esta base de datos una fuente de conocimiento que sin duda aportará nuevos conocimientos conforme comience a ser utilizada por más investigadores, dada su reciente publicación.

3.2. Análisis exploratorio de los datos

Se observan valores incompletos en la variable *Type_fuel*, los cuales se deben en su totalidad a un subgrupo de tipo de riesgo particular, en este caso el de las motocicletas. Debido a esto, y dado que los subgrupos de riesgo presentan, como se puede observar

en la figura 9, un comportamiento notoriamente distinto en el número de siniestros esperado, se opta por analizar solo el subgrupo de los automóviles, que representa el grueso de la cartera.

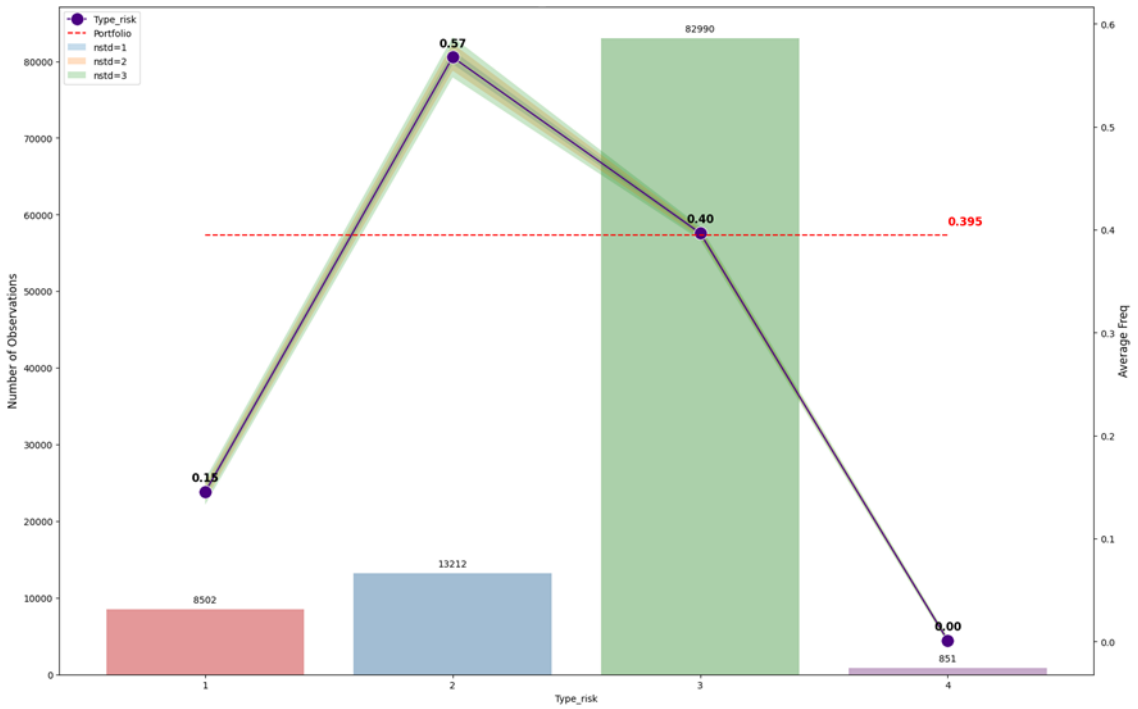

Figura 9. Frecuencia promedio y distribución por tipología de riesgo. Fuente: Elaboración propia

Donde 1 representa motocicletas, 2 furgonetas, 3 turismos y 4 vehículos agrícolas.

Una vez se tienen los datos filtrados por este tipo de riesgo, se generan ciertas variables de relevancia a través de las originales de fechas, tales como los años de carné del asegurado, su edad o la antigüedad del vehículo. Tras esto, el compendio principal de variables presenta los siguientes rangos:

Tabla 4. Descripción por variable de media, máximo, mínimo y percentiles principales

Variable	**Media**	**Mínimo**	**25%**	**Mediana**	**75%**	**Máximo**
Seniority	6.636	1	3	4	9	40
Policies_in_force	1.404	1	1	1	2	17
Max_policies	1.770	1	1	1	2	17
Max_products	1.047	1	1	1	1	4
Lapse	0.219	0	0	0	0	6
Premium	333.969	40.4	253.9	300.23	371.2	2993.34
N_claims_history	2.769	0	0	1	4	52
R_Claims_history	0.441	0	0	0.13	0.62	26.07
Power	100.263	11	75	100	115	580
Cylinder_capacity	1,675.461	150	1398	1598	1896	6753
Value_vehicle	19,475.03	270.46	14046	18120.51	22850	220675.8

Length	4.192	1.978	3.965	4.206	4.395	5.575
Weight	1,228.866	246	1070	1205	1357	2710
Licence_years	24.069	0	14	23	33	74
Driver_age	46.724	18	37	46	56	98
Car_age	11.676	0	8	12	15	65
N_claims_year	0.397	0	0	0	0	18

De lo que se puede observar que varias tienen colas pronunciadas y valores atípicos, por lo que se procede a analizar esta casuística, descartando las observaciones que perturban los datos. Así, en la figura 10 queda recogida la distribución final de las variables.

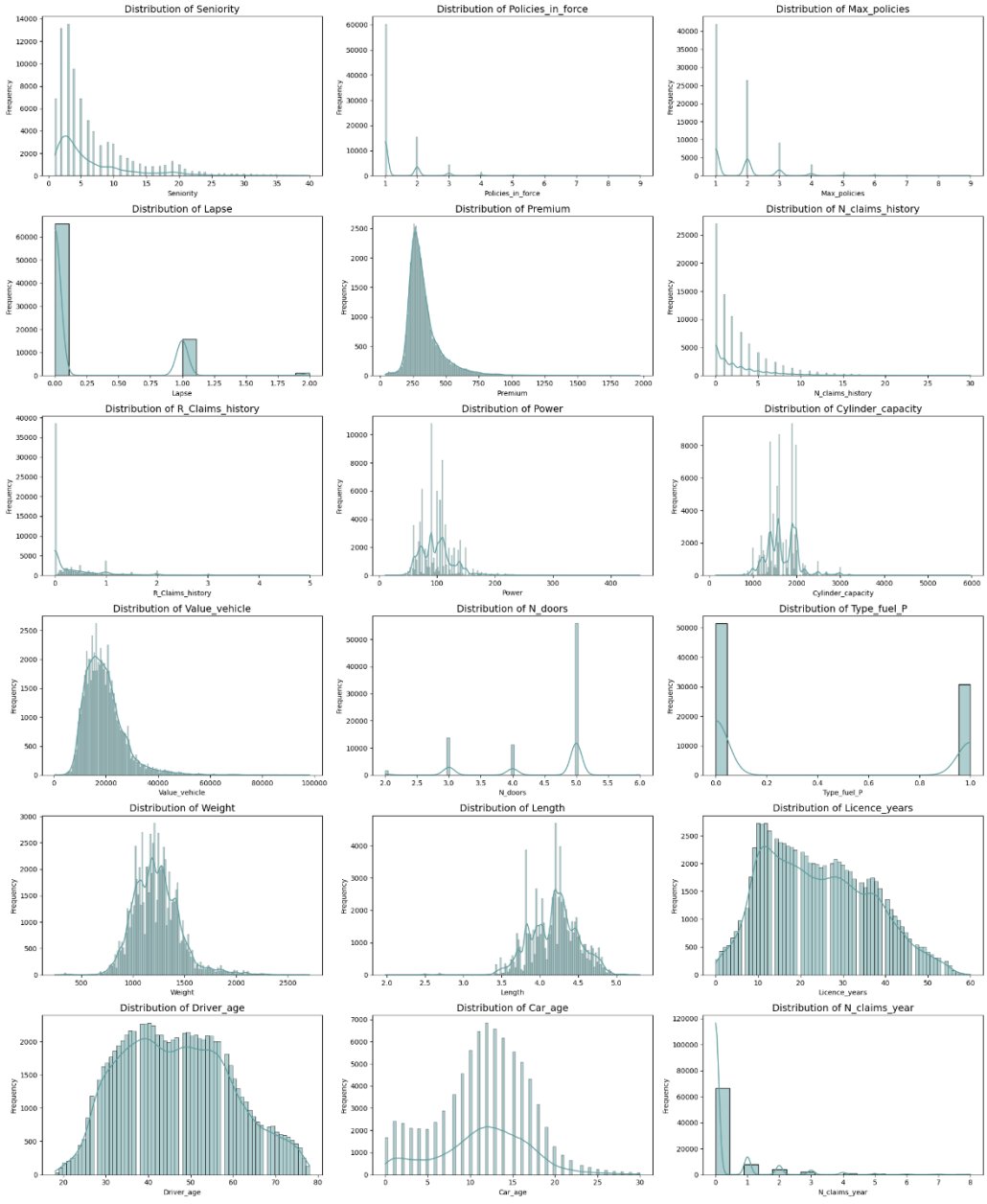

Figura 10. Distribución de las variables del *dataset*. Fuente: Elaboración propia

Tras esto, y como paso previo al entrenamiento de los modelos, se estudian los factores de inflación de la varianza (VIF, por sus siglas en inglés) para tratar de evitar la presencia de multicolinealidad en los modelos, obteniendo los siguientes indicadores:

Tabla 5. Factores de inflación de la varianza de las Variables

Variable	VIF	Variable	VIF	Variable	VIF
Broker_channel	1.12	N_doors_6	1.00	Value_vehicle	4.68
Half_yearly_payment	1.10	Seniority	1.72	Power	2.66
Urban_area	1.05	Policies_in_force	2.57	Cylinder_capacity	3.40
Second_driver	1.08	Max_policies	3.09	Length	3.72
Type_fuel_P	1.36	Lapse	1.07	Weight	5.77
More_than_one_product	1.15	N_claims_history	1.94	Licence_years	4.66
N_doors_2	1.09	R_Claims_history	1.38	Driver_age	4.58
N_doors_3	1.16	Premium	1.33	Car_age	1.56
N_doors_4	1.30				

De lo que se desprende que, efectivamente, hay variables muy relacionadas entre sí y pueden ser explicadas por otras, lo que podría generar problemas en la interpretación de los resultados. Además, como se puede observar la siguiente figura, hay variables que presentan una correlación muy elevada:

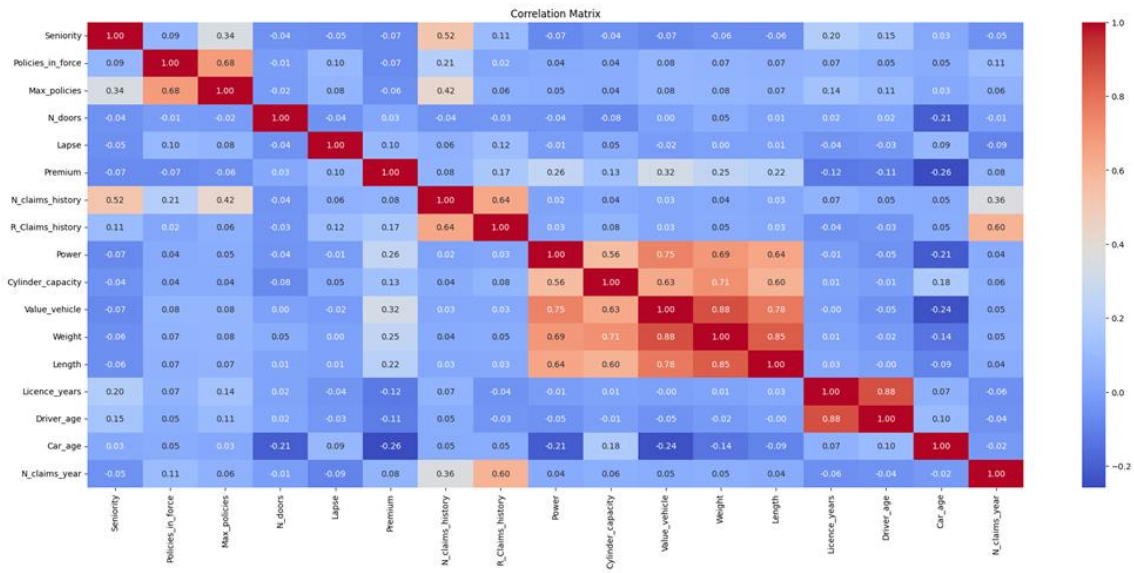

Figura 11. Matriz de correlaciones entre variables. Fuente: Elaboración propia

Por tanto, se descarta incluir en los modelos las variables de Edad del conductor, número máximo de pólizas, peso, longitud, potencia y valor del vehículo, puesto que la información que aportarían ya se encuentra debidamente recogida por otras variables más representativas.

4. RESULTADOS

En el presente apartado se procede a exponer los resultados obtenidos para la predicción del número de siniestros a través de los modelos propuestos en el apartado metodológico. Como se ha comentado en la introducción, este ejercicio se ha desarrollado en Python, encontrándose el código en el anexo.

4.1. GLM

Dado que la variable respuesta a estudiar es discreta y necesariamente no negativa (el número de siniestros incurridos durante el año por cada póliza), se entrena un modelo lineal generalizado donde la variable a explicar sigue una distribución de Poisson y la función de enlace es el logaritmo, por lo que el modelo sigue la fórmula:

$$\mu = e^{(\beta_0 + \sum_{i=1}^{n} \beta_i x_i))}$$

Utilizando como variables dependientes las determinadas tras el análisis del Factor de Inflación de la Varianza (*VIF*), obtenemos las estimaciones de los parámetros observables en la tabla 6. Las columnas de la tabla representan lo siguiente: 'coef' representa los coeficientes estimados del modelo para cada variable independiente; 'std err' es el error estándar de cada coeficiente estimado, midiendo la incertidumbre del coeficiente; 'z' representa el estadístico z para cada coeficiente, calculado como el coeficiente dividido por su error estándar; 'P>|z|' es el p valor asociado al estadístico z, reflejando si el coeficiente es estadísticamente significativo, es decir, distinto de 0; mientras que las últimas dos columnas reflejan los límites inferior y superior del intervalo de confianza al 95% para el coeficiente estimado.

Tabla 6. Resumen GLM con todas las variables

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.069	0.012	-86.586	0.000	-1.094	-1.045
Broker_channel	0.075	0.010	7.329	0.000	0.055	0.095
Half_yearly_payment	0.132	0.010	12.976	0.000	0.112	0.152
Urban_area	0.019	0.011	1.778	0.075	-0.002	0.040
Second_driver	0.087	0.013	6.751	0.000	0.061	0.112
Type_fuel_P	0.009	0.011	0.791	0.429	-0.013	0.031
More_than_one_product	-0.183	0.025	-7.429	0.000	-0.231	-0.135
N_doors_2	0.033	0.026	1.264	0.206	-0.018	0.083
N_doors_3	0.032	0.014	2.214	0.027	0.004	0.059
N_doors_4	-0.089	0.015	-5.818	0.000	-0.118	-0.059
N_doors_6	0.520	0.083	6.286	0.000	0.358	0.682
Seniority	-0.263	0.009	-28.750	0.000	-0.281	-0.245
Policies_in_force	0.092	0.003	29.067	0.000	0.086	0.098
Lapse	-0.232	0.006	-40.125	0.000	-0.243	-0.220
Premium	-0.006	0.005	-1.219	0.223	-0.015	0.003

N_claims_history	0.259	0.004	71.601	0.000	0.252	0.266
R_Claims_history	0.339	0.003	130.871	0.000	0.334	0.344
Cylinder_capacity	0.020	0.005	3.660	0.000	0.009	0.030
Licence_years	-0.038	0.005	-7.024	0.000	-0.049	-0.028
Car_age	-0.074	0.006	-13.291	0.000	-0.085	-0.063

Al existir coeficientes no significativos, estos se descartan uno a uno hasta llegar a las estimaciones finales de los parámetros, que se recogen en la tabla 7.

Tabla 7. Resumen GLM con variables significativas

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.060	0.012	-91.973	0.000	-1.083	-1.038
Broker_channel	0.077	0.010	7.613	0.000	0.057	0.097
Half_yearly_payment	0.131	0.010	12.966	0.000	0.111	0.150
Second_driver	0.085	0.013	6.681	0.000	0.060	0.109
More_than_one_product	-0.178	0.025	-7.242	0.000	-0.226	-0.130
N_doors_3	0.031	0.014	2.223	0.026	0.004	0.059
N_doors_4	-0.091	0.015	-6.049	0.000	-0.121	-0.062
N_doors_6	0.519	0.083	6.252	0.000	0.356	0.681
Seniority	-0.264	0.009	-28.807	0.000	-0.282	-0.246
Policies_in_force	0.092	0.003	29.706	0.000	0.086	0.098
Lapse	-0.232	0.006	-40.290	0.000	-0.243	-0.221
N_claims_history	0.259	0.004	72.814	0.000	0.252	0.266
R_Claims_history	0.339	0.003	131.006	0.000	0.334	0.344
Cylinder_capacity	0.018	0.005	3.543	0.000	0.008	0.028
Licence_years	-0.038	0.005	-7.036	0.000	-0.049	-0.028
Car_age	-0.072	0.005	-13.491	0.000	-0.082	-0.061

Puesto que se toma este modelo como base, estas serán las variables que se utilicen para entrenar el resto de los modelos.

4.1.1. Métricas

Como se puede observar en la tabla 8, el modelo presenta métricas similares tanto en el conjunto de entrenamiento como en el de prueba, si bien no presenta una potencia predictiva elevada.

Tabla 8. Resultados métricas GLM

Train Set		Test Set	
Métrica	Score	Métrica	Score
R^2	0.3165	R^2	0.279

Mean Absolute Error	0.5146	Mean Absolute Error	0.5249
Root Mean Squared Error	0.8043	Root Mean Squared Error	0.8282
Mean Poisson Deviance	1.7365	Mean Poisson Deviance	1.8272
Mean Deviance	0.6469	Mean Deviance	0.6859
Media Predicciones	0.4660	Media Predicciones	0.4698

4.1.2. Comportamiento global

Si bien el modelo no explica de forma óptima la varianza de la variable respuesta, como se adelantó en el apartado segundo del presente trabajo, su principal fuerte es la sencillez y facilidad interpretativa.

Al depender únicamente de los coeficientes, la contribución marginal de cada variable a las predicciones (funciones observables en la figura 12) sólo responde a estos escalares, que representan la pendiente de cada curva.

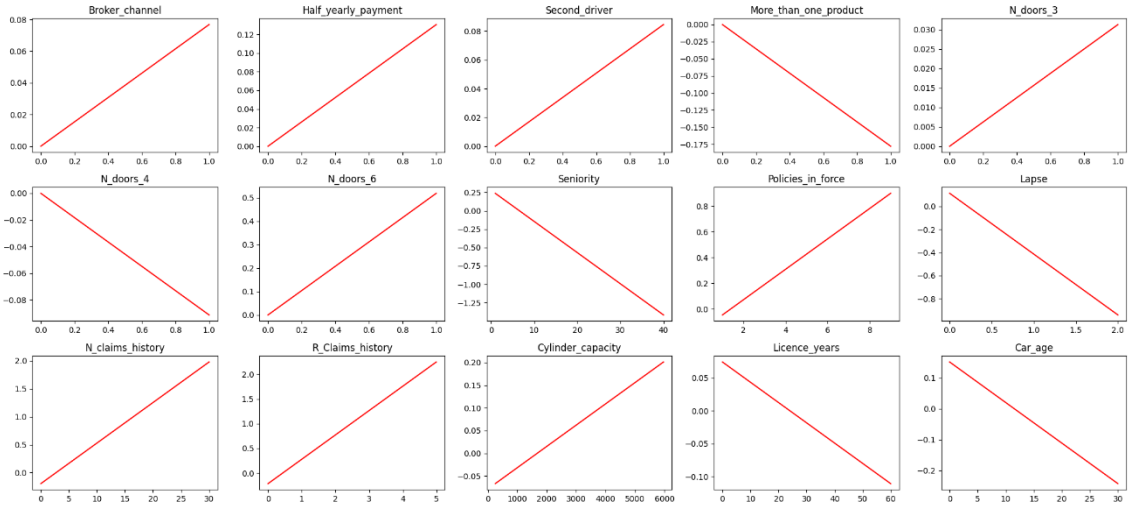

Figura 12. Contribuciones marginales de las variables en el GLM. Fuente: Elaboración propia

La importancia de cada variable se puede derivar directamente de los coeficientes de las variables a través del estadístico z (explicado previamente), siendo las 10 principales las recogidas en la tabla 9.

Tabla 9. Importancia de las variables modelo GLM

Variable	Z score	Variable	Z score
R_Claims_history	131.01	Car_age	13.49
N_claims_history	72.81	Half_yearly_payment	12.97
Lapse	40.29	Broker_channel	7.61
Policies_in_force	29.71	More_than_one_product	7.24
Seniority	28.81	Licence_years	7.04

4.1.3. Comportamiento local

Al tratarse de un modelo sencillo e interpretativo, se puede conocer de antemano tanto la predicción que resultará del modelo para una instancia concreta como la contribución que ha aportado cada variable independiente (junto con la constante) a la misma.

Para ilustrarlo, se toma como perfil de ejemplo la instancia número 5577 de los datos de prueba (que se utilizará para los sucesivos análisis de comportamiento local del resto de modelos), que toma los valores para las variables independientes (de interés) y respuesta observables en la tabla 10.

Tabla 10. Composición por variable del perfil de ejemplo

Variable	Valor	Variable	Valor
Broker_channel	0	Policies_in_force	1
Half_yearly_payment	1	Lapse	1
Second_driver	0	N_claims_history	3
More_than_one_product	0	R_Claims_history	1.63
N_doors_3	0	Cylinder_capacity	1,560
N_doors_4	0	Licence_years	7
N_doors_6	0	Car_age	6
Seniority	2	N_claims_year	2

Tomando estos valores, se obtiene la predicción del modelo, así como cada contribución marginal que la conforma, recogidos en la tabla 11.

Tabla 11. Descomposición de la predicción del GLM para el perfil de ejemplo

GLM prediction: 1.067			
Intercept	-1.060	Seniority	0.196
Broker_channel	0.000	Policies_in_force	-0.046
Half_yearly_payment	0.131	Lapse	0.113
Second_driver	0.000	N_claims_history	0.021
More_than_one_product	0.000	R_Claims_history	0.590
N_doors_3	0.000	Cylinder_capacity	-0.005
N_doors_4	0.000	Licence_years	0.052
N_doors_6	0.000	Car_age	0.073
sum of linear terms: 0.064			
exp of sum of linear terms: 1.067			

4.2. Random Forest

4.2.1. Métricas

En segundo lugar, se procede a entrenar un modelo a priori más potente que el anterior, obteniéndose tras ello las métricas de la tabla 12.

Tabla 12. Resultados métricas Random Forest

Train Set		Test Set	
Métrica	Score	Métrica	Score
R^2	0.5921	R^2	0.5100
Mean Absolute Error	0.2807	Mean Absolute Error	0.3067
Root Mean Squared Error	0.6213	Root Mean Squared Error	0.6828
Mean Poisson Deviance	0.9017	Mean Poisson Deviance	1.2218
Mean Deviance	0.3861	Mean Deviance	0.4662
Media Predicciones	0.3774	Media Predicciones	0.3782

Lo que, en efecto, confirma su mayor capacidad predictiva. No obstante, la diferencia entre el conjunto de entrenamiento y de prueba indica la presencia de sobreajuste (muy común en este tipo de modelos).

Además, como se ha comentado en el apartado metodológico, este modelo se encentra dentro de los conocidos como caja-negra, por lo que a continuación se tratará de arrojar algo de luz sobre su funcionamiento haciendo uso de las técnicas descritas previamente.

4.2.2. Comportamiento global

La implementación del algoritmo de *random forest* incluye de forma nativa un método de importancia de las variables, no obstante, es preferible el uso del método modelo-agnóstico comentado en el apartado metodológico. Esto se debe a que el método modelo-agnóstico se basa directamente en el impacto de las variables sobre el rendimiento del modelo, permitiendo para ello el uso de datos no utilizados en el entrenamiento de este como es el conjunto de prueba, al contrario que el método nativo, que presenta sensibilidad al potencial sobreajuste del modelo al basarse en la impureza de Gini que representa cada variable en el conjunto del entrenamiento del modelo.

Aplicando pues el método de permutación de las variables se obtiene el índice de importancia recogido en la figura 13.

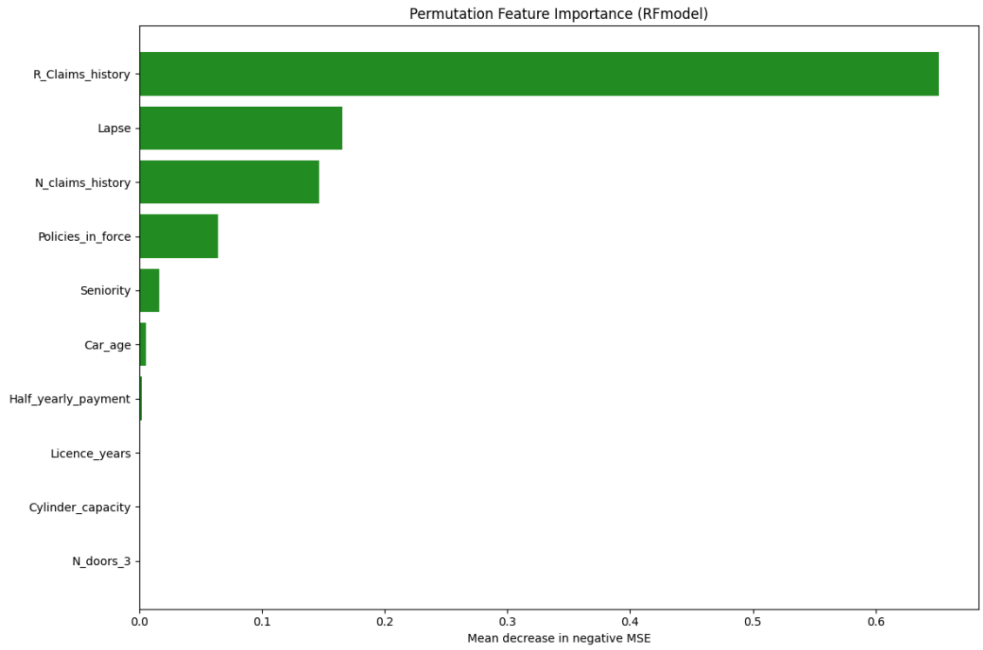

Figura 13. Importancia de las variables modelo Random Forest. Fuente: Elaboración propia

Salvo por la pérdida de importancia relativa de la variable "*N_claims_history*", este guarda fuertes similitudes con el ranking de importancia de variables obtenido por el valor z del modelo GLM base.

A diferencia de este, no es posible aislar para cada variable independiente su contribución marginal a la predicción del modelo (puesto que no es interpretativo en el sentido recogido en este trabajo). Por ello, se recurre a la aplicación de la técnica modelo-agnóstica de gráficos de dependencia parcial, vislumbrando en la figura 14 el efecto promedio que tienen las distintas variables sobre los resultados.

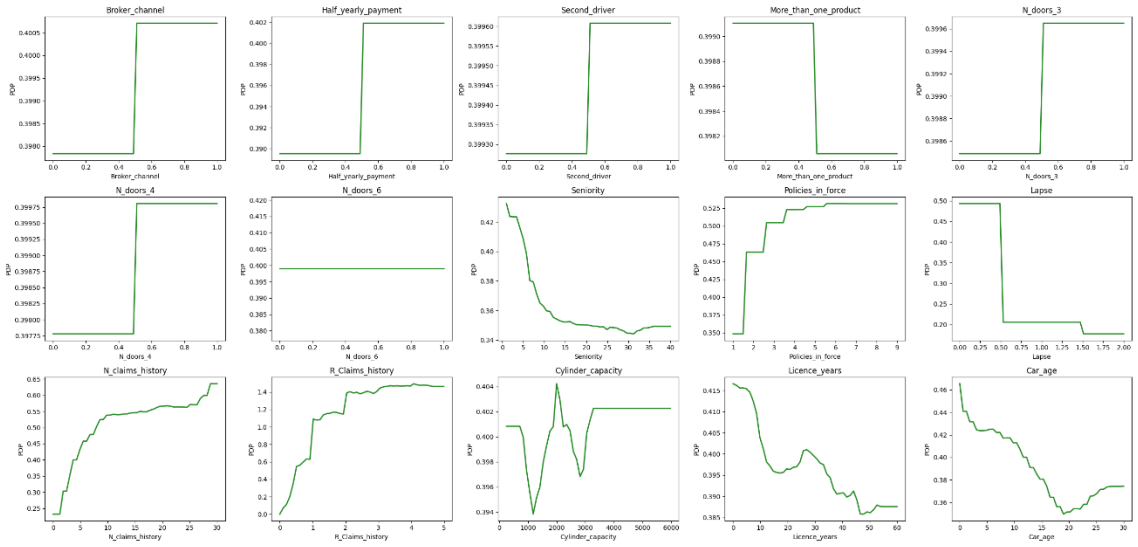

Figura 14. Gráficos de dependencia parcial del modelo Random Forest. Fuente: Elaboración propia

Si bien estas ilustraciones exponen de forma realista el efecto de las variables que recoge el modelo (como se observará en el apartado del modelo EBM posterior que sí

aísla el impacto marginal y cuyas funciones guardan una alta correlación con las expuestas), cabe remarcar que es un impacto promedio, por lo que no sirve para estimar el cambio o contribución sobre la predicción de forma independiente. Para ilustrar esta casuística, en la figura 15 se hace foco sobre el gráfico PDP de una de las variables, mostrándose también el gráfico de todas las curvas de impactos independientes que constituyen la base sobre la que se calcula este impacto promedio:

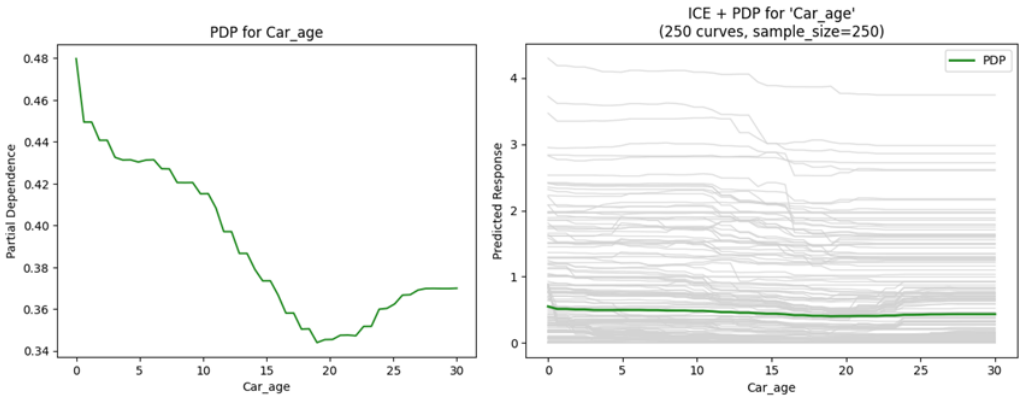

Figura 15. PDP vs curvas de dependencia individuales. Fuente: Elaboración propia

Como se puede observar, esta variación no es uniforme por la presencia de interrelaciones recogidas por el modelo, lo que implica una alta variabilidad entre instancias distintas.

4.2.3. Comportamiento local

Al tratarse de un modelo *black-box*, no es posible separar de forma nativa las contribuciones, como sí sucedía en el modelo lineal generalizado, por lo que, de nuevo, se recurre a métodos modelo-agnósticos para tratar de entender las contribuciones de las variables a las mismas.

4.2.3.1. LIME

Utilizando este método, obtenemos una aproximación del output del modelo, que, como se expone en el apartado metodológico, se puede obtener tanto por reglas como a través de una regresión.

Para el caso de aproximación local por árbol de decisión, se obtienen los resultados recogidos en la tabla 13.

Tabla 13. Aproximación LIME (por árboles de decisión) de la predicción del modelo Random Forest para el perfil de ejemplo

Regla	Contribución
Constante	-0.0966
R_Claims_history > 0.61'	1.0666
'Lapse <= 0.00'	0.2308
'Policies_in_force <= 1.00'	-0.0962
'N_doors_6 <= 0.00'	0.0778

'Car_age <= 8.00'	0.0354
'Licence_years <= 14.00'	0.0316
'Seniority <= 3.00'	0.0270
'N_doors_4 <= 0.00'	-0.0248
'1.00 < N_claims_history <= 4.00'	0.0199
'More_than_one_product <= 0.00'	-0.0162
'0.00 < Half_yearly_payment <= 1.00'	0.0140
'1398.00 < Cylinder_capacity <= 1598.00'	-0.0042
'Broker_channel <= 0.00'	-0.0007
'N_doors_3 <= 0.00'	-0.0004
'Second_driver <= 0.00'	-0.0003
Suma contribuciones	1.2637

Mientras que, en caso de la regresión, los resultados son los plasmados en la tabla 14.

Tabla 14. Aproximación LIME (por regresión lineal) de la predicción del modelo Random Forest para el perfil de ejemplo

Variable	Contribución
Constante	0.4720
R_Claims_history'	0.4989
'N_claims_history'	0.1386
'Lapse'	-0.0958
'Policies_in_force'	0.0752
'Seniority'	-0.0381
'Car_age'	-0.0360
'More_than_one_product'	-0.0097
'Half_yearly_payment'	0.0092
'Broker_channel'	0.0081
'Second_driver'	0.0058
'Licence_years'	-0.0037
'N_doors_4'	-0.0007
'N_doors_6'	0.0003
'N_doors_3'	0.0001
'Cylinder_capacity'	0.0000
Suma contribuciones	1.0241

No obstante, dado que la predicción del modelo es 1.8817, ninguno de los dos casos consigue capturar de forma fiel la predicción final a través del modelo local por vecinos estimado, por lo que en este ejercicio el método no aporta mayor visibilidad al modelo.

4.2.3.2. SHAP

Por su propia dinámica, este método es más exacto que el anterior, a pesar de ser computacionalmente más demandante. Así, para el perfil establecido como base de estudio en el apartado previo del modelo GLM, se observan las contribuciones por variable recogidas en la figura 16 que conllevan, bajo la teoría de este método, que se aleje de la media de las predicciones.

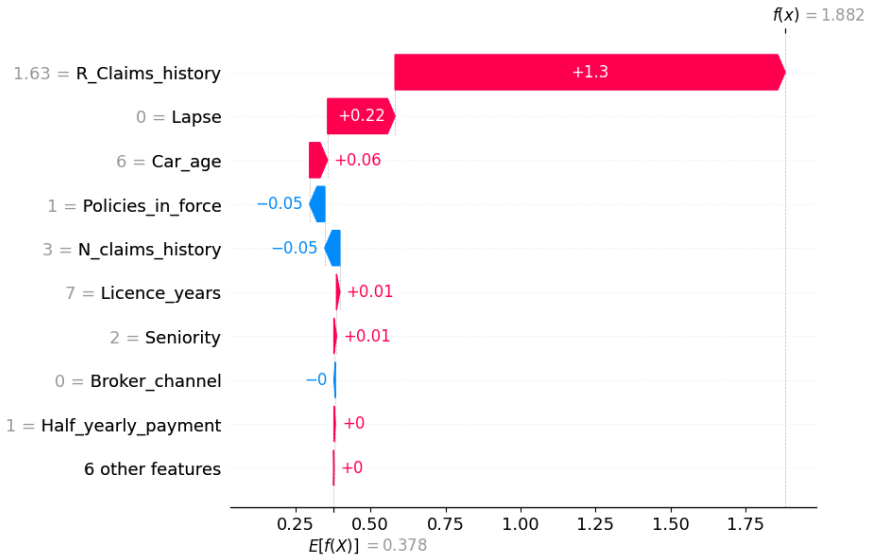

Figura 16. Aproximación SHAP de la predicción del modelo Random Forest para el perfil de ejemplo. Fuente: Elaboración propia

Además, haciendo uso de la implementación "*SHAP summary plot*", se obtiene la figura 17, que condensa la información de los valores de Shapley para diferentes instancias y su impacto en el modelo, por lo que se obtiene una vista híbrida local-global.

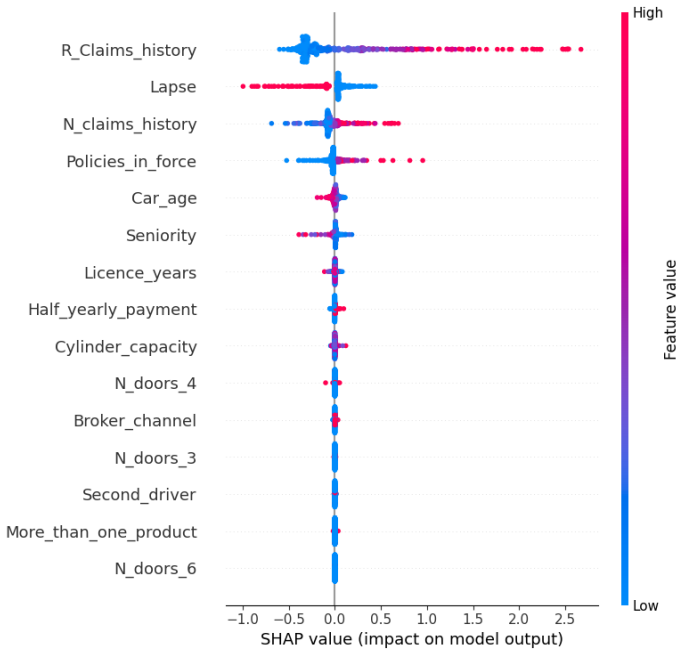

Figura 17. Gráfico resumen SHAP para el modelo Random Forest. Fuente: Elaboración propia

Así, de esta figura 17 se deriva, como ejemplo, que cuanto mayor sea el número de siniestros declarado por el asegurado previamente y más pólizas tenga el asegurado en vigor, mayor será el número de siniestros que prediga el modelo.

4.3. DNN

Se ha entrenado una red neuronal profunda, totalmente conectada, con tres capas ocultas de 64, 64 y 32 neuronas, respectivamente, y la función de activación 'ReLU (*Rectified Linear Unit*) en cada una de ellas. La capa final consta de una neura con activación exponencial, para asegurar un output positivo, consistente con el dominio de la variable a explicar.

4.3.1. Métricas

Se obtienen, para el conjunto de entrenamiento y de testeo, las métricas plasmadas en la tabla 15.

Tabla 15. Resultados métricas DNN

Train Set		Test Set	
Métrica	Score	Métrica	Score
R^2	0.5391	R^2	0.5070
Mean Absolute Error	0.2984	Mean Absolute Error	0.3078
Root Mean Squared Error	0.6604	Root Mean Squared Error	0.6848
Mean Poisson Deviance	1.1051	Mean Poisson Deviance	1.2676
Mean Deviance	0.4362	Mean Deviance	0.4690
Media Predicciones	0.3765	Media Predicciones	0.3766

Observándose unos resultados equilibrados, en línea con los observados para el modelo anterior en el caso del set de testeo. Para el conjunto de entrenamiento se obtienen peores resultados, pero, al ser el desempeño en datos no vistos similar, esto indica que este modelo no presenta los problemas de sobreajuste del *random forest* (las redes neuronales, por lo general, no sufren de este tipo de problema, generalizan bien en datos no utilizados durante el entrenamiento).

4.3.2. Comportamiento global

A través de la aplicación del método de permutación de las variables, estas se pueden clasificar en función de su importancia sobre las predicciones de la red neuronal, como se recoge en la figura 18.

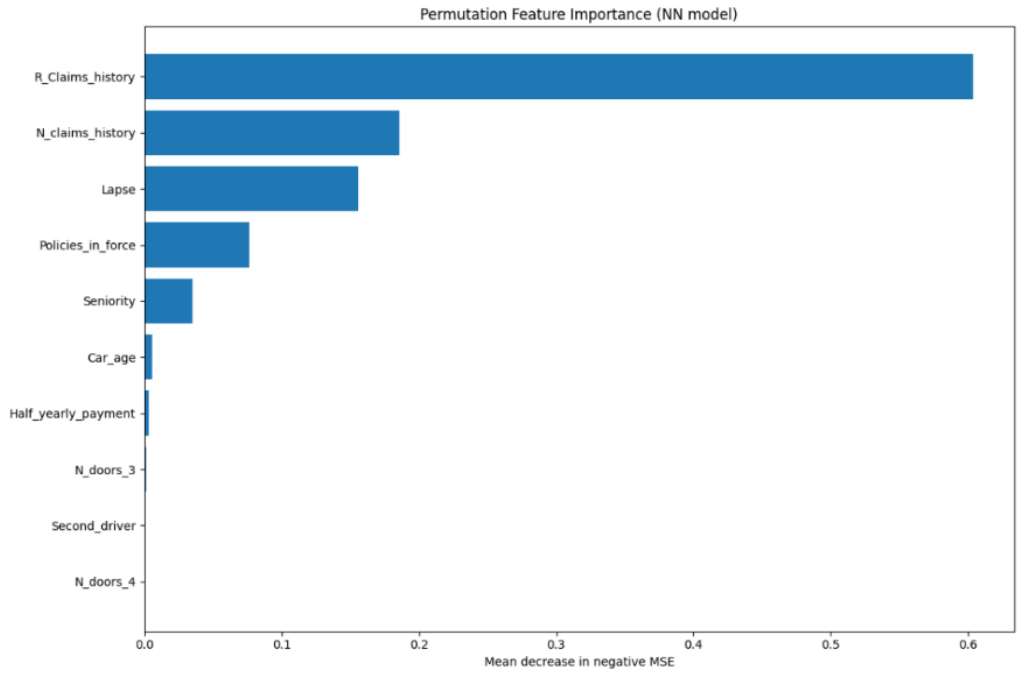

Figura 18. Importancia de las variables modelo de Red Neuronal Profunda. Fuente. Elaboración propia

En este caso, este orden se encuentra totalmente en línea con lo observado en el apartado del GLM para aquellas variables con importancia significativa sobre los resultados del modelo.

Como sucedía en el modelo Random Forest, y a diferencia del GLM, no es posible aislar para cada variable independiente su contribución marginal a la predicción del modelo. Por lo tanto, se vuelve a recurrir a la aplicación de gráficos de dependencia parcial entre la variable respuesta y las independientes, recogidos en la figura 19.

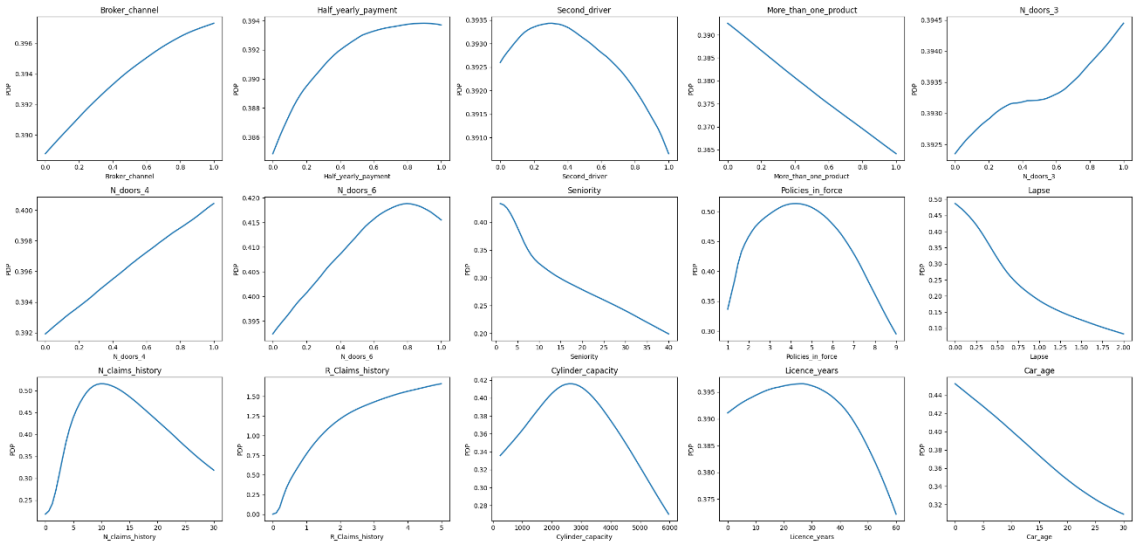

Figura 19. Gráficos de dependencia parcial del modelo Random Forest. Fuente: Elaboración propia

Como se puede observar en esta figura 19, las curvas obtenidas son mucho más uniformes, característica también típica de las redes neuronales. Esto presenta la ventaja

de que puede evitar incluir patrones derivados de los datos de entrenamiento, pero no se trasladan al conjunto de sucesos, lo que mejora su generalización. Pero, por otro lado, puede que no esté recogiendo variaciones que sí aportan valor y una mejora en las predicciones. No obstante, la flexibilidad que presenta este tipo de arquitectura de modelo permite crear, con los componentes adecuados, funciones que aprendan estos 'saltos', como se expondrá más adelante.

4.3.3. Comportamiento local – Modelo sustituto

La aplicación y resultados de LIME y SHAP no presentan grandes diferencias respecto al apartado análogo del modelo previo. Por tanto, se procederá en este caso a aplicar el método de modelo sustituto. Esta herramienta, si bien tiene un enfoque global y no únicamente local, permite realizar una inferencia realmente interpretativa de la predicción del modelo para la instancia concreta que se ha tratado en los apartados análogos previos.

En este caso, el modelo interpretativo por diseño que se ha elegido es el algoritmo *RuleFit*. Entrenado tomando como variable dependiente las predicciones de la red neuronal, se logra explicar aproximadamente el 98% de la varianza en los dos conjuntos de datos trabajados, por lo que resulta una réplica fiel (pero sencilla e interpretativa) de la dinámica del modelo caja-negra.

Con los coeficientes obtenidos de la parte del modelo de regresión, se obtienen las contribuciones marginales de las variables independientes para el perfil objeto de estudio recogidas en la tabla 16.

Tabla 16. Contribuciones de las variables a la predicción del modelo sustituto para el perfil de ejemplo

Variable	Contribución
Constante	0.4120
Half_yearly_payment	0.0035
R_Claims_history	0.4300
Cylinder_capacity	0.0436
Licence_years	-0.0005
Car_age	-0.0158

El resto de las variables presentan una contribución de 0 sobre la predicción. Por otro lado, el conjunto de reglas estimado es muy extenso, por lo que, en la tabla 17, se recogen algunos ejemplos de estas.

Tabla 17. Ejemplos contribuciones reglas a la predicción del modelo sustituto sobre el perfil de ejemplo

Regla	Contribución
R_Claims_history > 0.335	0.0298
Seniority <= 6.5 and N_claims_history > 2.5	0.0140

Lapse <= 0.5 and R_Claims_history > 0.555 and N_claims_history > 2.5	0.0256
Seniority <= 7.5 and Lapse <= 0.5 and R_Claims_history > 0.315 and N_claims_history > 2.5	0.0373

Siendo la suma de las contribuciones de las variables y las reglas cumplidas el valor exacto de la predicción del modelo sustituto para la instancia específica.

Si bien el modelo (sustituto) sigue resultando inherente y fácilmente interpretativo, el elevado número de reglas y el número de combinaciones que conllevan puede resultar tedioso de manejar. No obstante, en la construcción del modelo se pueden especificar explícitamente estos dos parámetros, por lo que se puede limitar a voluntad su complejidad. Dado el elevado porcentaje de la varianza explicada del modelo principal es viable sacrificar parte de este desempeño en pos de una menor complejidad.

4.4. EBM

La implementación en Python del algoritmo (dentro del paquete de herramientas de *InterpretML*), como se comentó en el apartado metodológico, permite incluir (o no) los pares de interacciones más relevantes al modelo. Por ello, se han entrenado dos modelos distintos, uno sólo con las variables independientes y otro teniendo en cuenta también estas interacciones.

4.4.1. Métricas

Para el modelo que sólo incluye las variables, se obtienen las métricas recogidas en la tabla 18.

Tabla 18. Resultados métricas EBM (solo variables)

Train Set		Test Set	
Métrica	Score	Métrica	Score
R^2	0.4825	R^2	0.4634
Mean Absolute Error	0.3318	Mean Absolute Error	0.3399
Root Mean Squared Error	0.6998	Root Mean Squared Error	0.7145
Mean Poisson Deviance	1.2838	Mean Poisson Deviance	1.3435
Mean Deviance	0.4898	Mean Deviance	0.5105
Media Predicciones	0.3773	Media Predicciones	0.3803

Mientras que el que incorpora las interacciones presenta los resultados observables en la tabla 19.

Tabla 19. Resultados métricas EBM (variables e interacciones)

Train Set	Test Set

Métrica	Score	Métrica	Score
R^2	0.5361	R^2	0.5035
Mean Absolute Error	0.3084	Mean Absolute Error	0.3179
Root Mean Squared Error	0.6626	Root Mean Squared Error	0.6873
Mean Poisson Deviance	1.1553	Mean Poisson Deviance	1.2531
Mean Deviance	0.439	Mean Deviance	0.4723
Media Predicciones	0.3773	Media Predicciones	0.3810

Como es lógico, la interacción de los términos entrenados sobre los residuos mejora la capacidad predictiva del modelo, situándolo en el mismo rango que los dos estudiados anteriormente, pero manteniendo la interpretabilidad total, como se expondrá en los siguientes apartados.

4.4.2. Comportamiento global

La implementación del algoritmo también incorpora por defecto el ranking de importancia de las variables, los cuales se muestran a continuación para el modelo conformado solo por las variables, en la figura 20, y para el modelo que incluye las interacciones, en la figura 21.

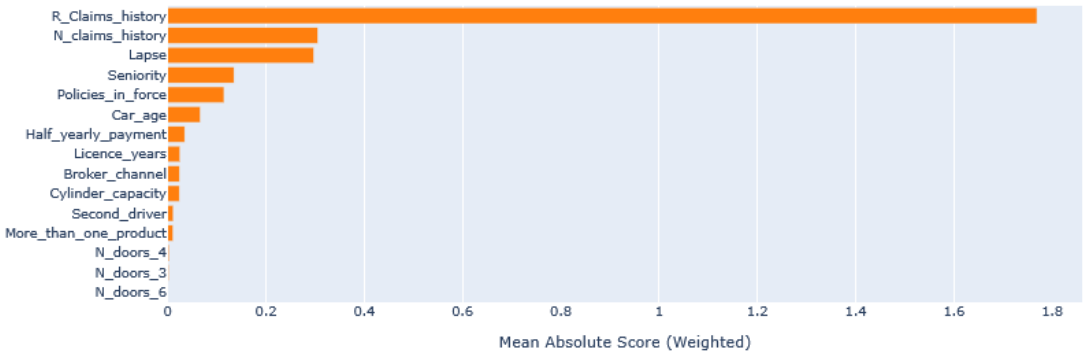

Figura 20. Importancia de las variables modelo EBM (solo variables). Fuente. Elaboración propia

Salvo por la mayor importancia relativa de la variable "*Seniority*", el ranking de esta figura 20 guarda fuertes similitudes con el ranking de importancia de variables obtenido por el valor z del modelo GLM base.

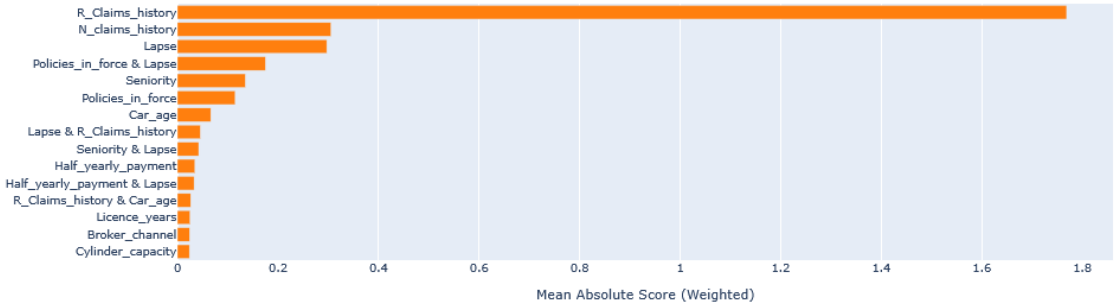

Figura 21. Importancia de las variables modelo EBM (con interacciones). Fuente. Elaboración propia

El ranking del gráfico de la figura 21 es equivalente al de la figura 20 para las variables independientes, puesto que, como se describe en el apartado metodológico, estas se entrenan primero y después se estiman las interacciones. La diferencia proviene de la inclusión del efecto de las interacciones, que en muchos casos tienen un efecto mayor al capturado por las propias variables.

De igual forma, las contribuciones marginales al output del modelo de cada variable independiente son equivalentes para ambos modelos, siendo las funciones que las recogen las expuestas en la figura 22.

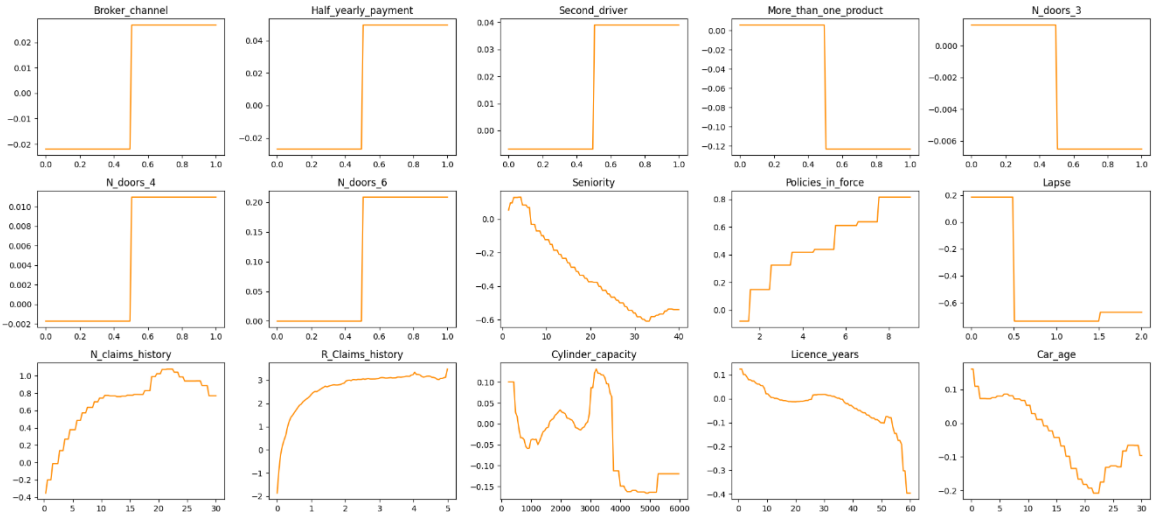

Figura 22. Contribuciones marginales de las variables en el modelo EBM. Fuente: Elaboración propia

Se puede observar que estas funciones guardan una relación fuerte con los gráficos de dependencia parcial del modelo *Random Forest*. Pero, a diferencia de estos, al tratarse de un modelo aditivo, estas sí representan la contribución exacta e independiente de cada variable en función del valor que tomen en la instancia, al igual que sucedía en el caso del GLM, simplemente cambiando el coeficiente por la $f(x)$ correspondiente.

Estas funciones discretas se han construido a través de la interpolación del resultado de la implementación del algoritmo en Python, que aporta como output una serie de valores para las variables discretizadas, por lo que también se podrían recoger en tablas de sencilla implementación en los sistemas actuariales. En la tabla 20, se muestra un ejemplo para la variable independiente antigüedad del vehículo ("*Car_age*").

Tabla 20. contribuciones marginales por rangos de la variable antigüedad del vehículo

Rango de la variable	**Contribución marginal**	**Rango de la variable**	**Contribución marginal**	**Rango de la variable**	**Contribución marginal**
0.0-0.5	0.160	10.5-11.5	0.028	20.5-21.5	-0.193
0.5-1.5	0.109	11.5-12.5	0.011	21.5-22.5	-0.208

1.5-2.5	0.073	12.5-13.5	-0.008	22.5-23.5	-0.175
2.5-3.5	0.072	13.5-14.5	-0.023	23.5-24.5	-0.131
3.5-4.5	0.076	14.5-15.5	-0.043	24.5-25.5	-0.127
4.5-5.5	0.080	15.5-16.5	-0.068	25.5-26.5	-0.130
5.5-6.5	0.086	16.5-17.5	-0.099	26.5-27.5	-0.083
6.5-7.5	0.081	17.5-18.5	-0.135	27.5-28.5	-0.066
7.5-8.5	0.072	18.5-19.5	-0.166	28.5-29.5	-0.066
8.5-9.5	0.067	19.5-20.5	-0.182	29.5-30.0	-0.096
9.5-10.5	0.053				

Otra opción, si se desea suavizar el efecto de los saltos (que, no obstante, pueden recoger efectos significativos), sería aproximar la función discreta a través de polinomios o *splines*. En la figura 23 se ilustra este ejemplo para la misma variable del ejemplo anterior, con polinomios de distintos grados.

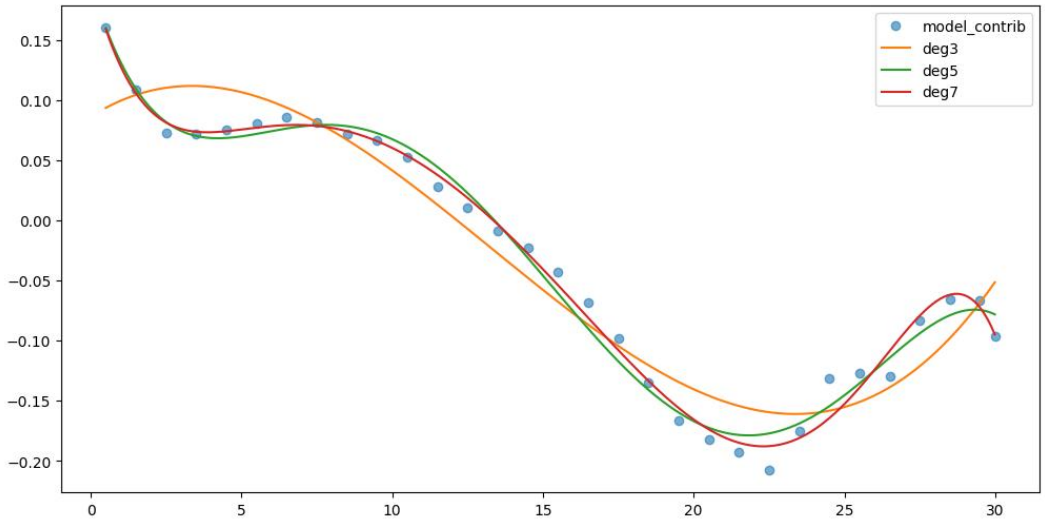

Figura 23. Suavizado de la función discreta de contribución marginal. Fuente: Elaboración propia

En el caso de las contribuciones de las interacciones entre variables, también quedan recogidas de forma independiente para cada interacción en forma de mapas de calor, como se muestra en la figura 24 para las variables Número de pólizas en vigor y registro de caídas.

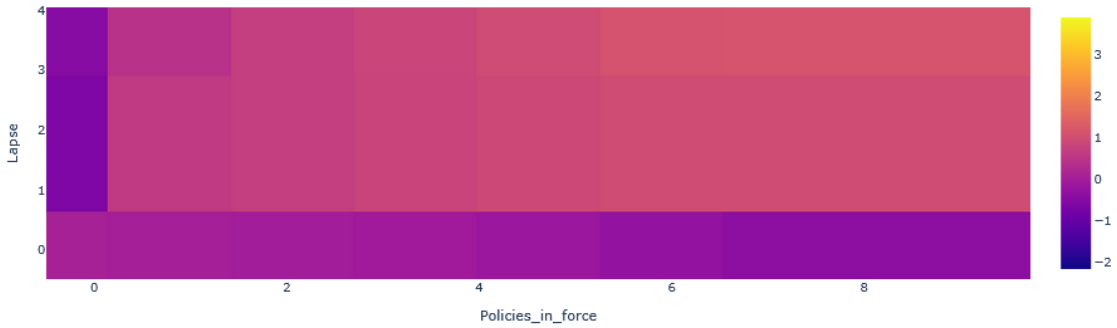

Figura 24. Ejemplo contribuciones de las interacciones en el modelo EBM. Fuente: Elaboración propia

Estas interacciones representan el valor de $f(x_i, x_j)$ correspondiente, que se añade como otro término del componente aditivo.

4.4.3. Comportamiento local

En el caso del modelo que sólo incluye las variables, en la tabla 21 se recoge la predicción sobre el perfil de ejemplo, así como cada contribución marginal que la conforma.

Tabla 21. Descomposición de la predicción del EBM (solo variables) para el perfil de ejemplo

EBM prediction: 1.7328			
Intercept	-2.770	Seniority	0.096
Broker_channel	-0.022	Policies_in_force	-0.079
Half_yearly_payment	0.050	Lapse	0.186
Second_driver	-0.007	N_claims_history	0.137
More_than_one_product	0.006	R_Claims_history	2.813
N_doors_3	0.001	Cylinder_capacity	-0.009
N_doors_4	-0.002	Licence_years	0.062
N_doors_6	0.000	Car_age	0.086
sum of linear terms: 0.5498			
exp of sum of linear terms: 1.7328			

De forma análoga al funcionamiento del modelo lineal generalizado, estas contribucciones representan, para cada variable independiente, su correspondiente $f(x)$ en cada función individual expuesta en la figura 22 del apartado previo.

Para el modelo que incluye además las interacciones, dada la operativa del algoritmo, las contribuciones marginales de las variables son idénticas a las expuestas, salvo por el valor que toma la constante. A la suma de estos términos se le añaden las contribuciones

marginales de las interacciones, obteniendo la desagregación por componente de la predicción recogida en la tabla 22.

Tabla 22. Descomposición de la predicción del EBM (con interacciones) para el perfil de ejemplo

EBM prediction: 1.8463			
Intercept	-2.968	Seniority	0.096
Broker_channel	-0.022	Policies_in_force	-0.079
Half_yearly_payment	0.050	Lapse	0.186
Second_driver	-0.007	N_claims_history	0.137
More_than_one_product	0.006	R_Claims_history	2.813
N_doors_3	0.001	Cylinder_capacity	-0.009
N_doors_4	-0.002	Licence_years	0.062
N_doors_6	0.000	Car_age	0.086
Half_yearly_payment & Lapse	-0.027	Policies_in_force & R_Claims_history	0.031
N_doors_6 & R_Claims_history	0.010	Lapse & N_claims_history	0.015
Seniority & Lapse	0.025	Lapse & R_Claims_history	0.080
Seniority & N_claims_history	0.010	N_claims_history & R_Claims_history	0.004
Seniority & R_Claims_history	0.022	N_claims_history & Cylinder_capacity	0.003
Policies_in_force & Lapse	0.068	N_claims_history & Car_age	0.008
Policies_in_force & N_claims_history	0.004	R_Claims_history & Car_age	0.011
sum of linear terms: 0.6132			
exp of sum of linear terms: 1.8463			

Así, al igual que sucedía con el modelo base GLM, en ambos casos se tiene el detalle granular de la contribución que aporta cada variable independiente. En este caso el aporte de cada variable independiente representa bien únicamente su contribución marginal o esta y el valor del efecto de sus interacciones con otras variables.

4.5. NAM

Al igual que en el apartado anterior, se han entrenado dos modelos, sobre los que se presentan sus respectivos resultados, el primero únicamente recoge las variables independientes, mientras que el segundo recoge, además de estas, las interacciones relevantes entre pares de variables estimadas por el algoritmo EBM.

Esta decisión responde a un tema de eficiencia computacional, pues ya se tienen identificados estos pares de interacciones relevantes, simplificando el proceso. No obstante, se reconoce que, debido a la flexibilidad y el mayor potencial para capturar interacciones de forma efectiva, el modelo se beneficiaría de incluir todos los posibles pares de interacciones con regularización en cada subred. Este enfoque no ha sido abordado en el presente trabajo por limitaciones técnicas en el hardware, pero sería interesante estudiarlo contando con recursos computacionales más potentes. Otra opción sería incluir interacciones de mayor orden (tres variables), lo que recogería otros efectos no valorados, aunque su interpretación, si bien posible, no es tan sencilla a juicio humano como en el caso de los pares de variables.

4.5.1. Métricas

Se obtienen las métricas recogidas en la tabla 23 para el modelo que sólo incluye las variables:

Tabla 23. Resultados métricas NAM (solo variables)

Train Set		Test Set	
Métrica	Score	Métrica	Score
R^2	0.4780	R^2	0.4626
Mean Absolute Error	0.3283	Mean Absolute Error	0.3359
Root Mean Squared Error	0.7029	Root Mean Squared Error	0.7150
Mean Poisson Deviance	1.2813	Mean Poisson Deviance	1.3436
Mean Deviance	0.4940	Mean Deviance	0.5112
Media Predicciones	0.3747	Media Predicciones	0.3775

Por otro lado, el que incorpora las interacciones presenta los resultados de la tabla 24.

Tabla 24. Resultados métricas NAM (variables e interacciones)

Train Set		Test Set	
Métrica	Score	Métrica	Score
R^2	0.5197	R^2	0.5031
Mean Absolute Error	0.3093	Mean Absolute Error	0.3158
Root Mean Squared Error	0.6742	Root Mean Squared Error	0.6875
Mean Poisson Deviance	1.1761	Mean Poisson Deviance	1.2591
Mean Deviance	0.4546	Mean Deviance	0.4727
Media Predicciones	0.3746	Media Predicciones	0.3776

De nuevo, el modelo que incluye los efectos de las interacciones presenta mejores resultados, a la par de los vistos para su homólogo previo y los dos modelos caja-negra. En este caso, como el EBM, también tiene la ventaja sobre estos últimos de ser inherentemente interpretativo.

4.5.2. Comportamiento global

A continuación, se muestran en las tablas 25 y 26 la importancia de las variables, clasificadas en función de sus aportaciones a las predicciones en el set de prueba, para el modelo que solo tienen en cuenta las contribuciones marginales de las variables independientes y para el que incluye el efecto de sus interacciones, respectivamente:

Tabla 25. Importancia de las variables modelo NAM (solo variables)

Variable	Contribución Promedio	Variable	Contribución Promedio
R_Claims_history	3.84346	Car_age	0.08112
N_claims_history	0.54987	Half_yearly_payment	0.04059
Lapse	0.19876	Broker_channel	0.01774
Seniority	0.19547	More_than_one_product	0.00407
Policies_in_force	0.09362	Licence_years	0.00004

Este ranking guarda fuertes similitudes con el orden de importancia de variables obtenido por el estadístizo *z* del modelo GLM base, exceptuando la mayor importancia relativa de la variable "*Seniority*".

Tabla 26. Importancia de las variables modelo NAM (variables e interacciones)

Variable	Contribución Promedio	Variable	Contribución Promedio
R_Claims_history	3.84346	Policies_in_force*Lapse	0.14493
N_claims_history	0.54987	Lapse*N_claims_history	0.10470
Seniority*Lapse	0.29805	N_doors_6*R_Claims_histor	0.10436
Lapse	0.19876	Policies_in_force	0.09362
Seniority	0.19547	Car_age	0.08112

El ranking de esta tabla 26 es equivalente al de la tabla 25 para las variables independientes. La diferencia proviene de la inclusión del efecto de las interacciones, que en muchos casos tienen un efecto mayor al capturado por las propias variables.

Dada la operativa seguida, imitando el proceso del EBM, las contribuciones marginales al output del modelo de cada variable independiente son equivalentes para ambos modelos (solo con las variables e incluyendo las interacciones), siendo las funciones que las recogen las observables en la figura 25.

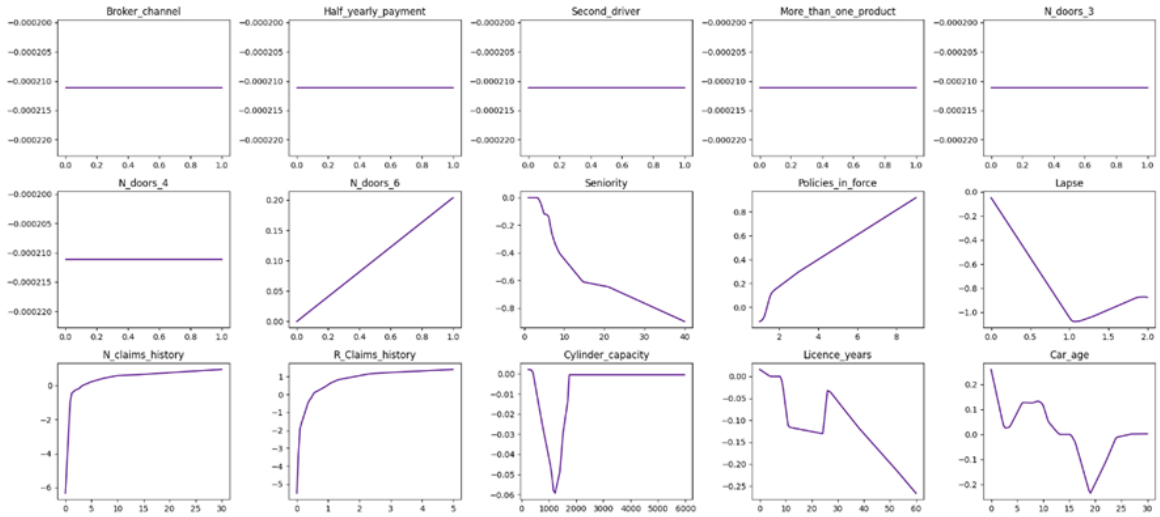

Figura 25. Contribuciones marginales de las variables en el modelo NAM. Fuente: Elaboración propia

Estas funciones conservan la misma tendencia que la observada en casos previos para las variables significativas, aunque capturan efectos distintos, lo que se estudiará más adelante al enfrentar los modelos. De cualquier forma, estas también representan la contribución exacta e independiente de cada variable en función del valor que tomen en la instancia, al igual que sucedía en el caso del GLM y el EBM.

En este caso, el output de la red neuronal ya representa una función continua (y, dada su naturaleza, más suavizada), por lo que no se exponen tablas ni aproximaciones, aunque si se desease el primer caso para su implantación en sistemas estas se pueden discretizar.

En el caso de las contribuciones de las interacciones entre variables, se ha replicado el output en forma de mapa de calor (figura 26) y una representación en 3D interactiva (figura 27), quedando igualmente recogidas de forma independiente para cada interacción:

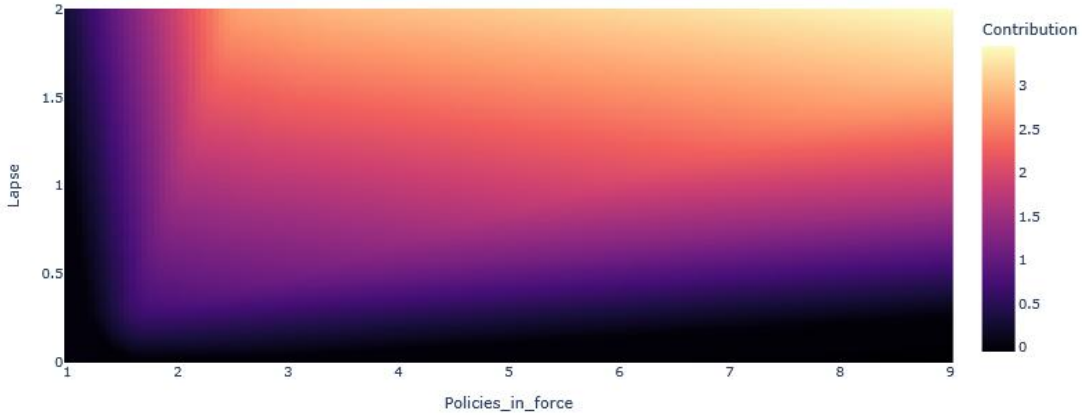

Figura 26. Ejemplo de Mapa de calor de contribuciones marginales de las interacciones en el modelo NAM. Fuente: Elaboración propia

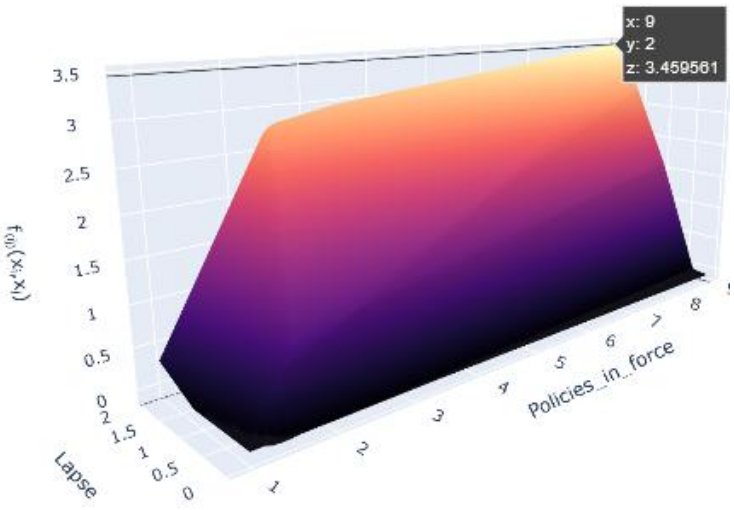

Figura 27. Ejemplo de visualización en 3D de contribuciones marginales de las interacciones en el modelo NAM. Fuente: Elaboración propia

Tanto la figura 26 como la 27 representan el valor de $f(x_i, x_j)$ correspondiente, que se añade como otro término del componente aditivo.

4.5.3. Comportamiento local

Al ser un modelo aditivo e inherentemente interpretativo, al igual que sucedía con el EBM y el GLM, la contribución granular que aporta cada variable independiente (o la interacción entre estas) a la predicción final del modelo es conocida.

Así, para el modelo que sólo incluye las variables, se recoge en la tabla 27 la predicción para la instancia utilizada y cada contribución marginal que la conforma.

Tabla 27. Descomposición de la predicción del NAM (solo variables) para el perfil de ejemplo

EBM prediction: 1.7972			
Intercept	-0.209	Seniority	0.000
Broker_channel	0.000	Policies_in_force	-0.118
Half_yearly_payment	0.000	Lapse	-0.048
Second_driver	0.000	N_claims_history	-0.068
More_than_one_product	0.000	R_Claims_history	0.929
N_doors_3	0.000	Cylinder_capacity	-0.026
N_doors_4	0.000	Licence_years	0.000
N_doors_6	0.000	Car_age	0.127
sum of linear terms: 0.5862			
exp of sum of linear terms: 1.7972			

Como ya se ha comentado previamente, estas contribucciones representan, para cada variable independiente, su correspondiente $f(x)$ en cada función individual expuesta en la figura 25 del apartado previo.

Como se ha decidido entrenar primero las redes neuronales correspondientes a las variables independientes para incluir después las interacciones, para el modelo que incluye dichas interacciones, las contribuciones marginales de las variables son idénticas a las previas, salvo por el valor que toma la constante. A la suma de estos términos se le añaden las contribuciones marginales de las interacciones, obteniendo la desagregación por componente de la predicción recogida en la tabla 28.

Tabla 28. Descomposición de la predicción del NAM (con interacciones) para el perfil de ejemplo

EBM prediction: 2.0761			
Intercept	-0.078	Seniority	0.000
Broker_channel	0.000	Policies_in_force	-0.118
Half_yearly_payment	0.000	Lapse	-0.048
Second_driver	0.000	N_claims_history	-0.068
More_than_one_product	0.000	R_Claims_history	0.929
N_doors_3	0.000	Cylinder_capacity	-0.026
N_doors_4	0.000	Licence_years	0.000
N_doors_6	0.000	Car_age	0.127
Half_yearly_payment & Lapse	-0.061	Policies_in_force & R_Claims_history	0.207
N_doors_6 & R_Claims_history	0.001	Lapse & N_claims_history	0.001
Seniority & Lapse	0.001	Lapse & R_Claims_history	0.001
Seniority & N_claims_history	0.001	N_claims_history & R_Claims_history	0.001
Seniority & R_Claims_history	-0.027	N_claims_history & Cylinder_capacity	0.001
Policies_in_force & Lapse	0.001	N_claims_history & Car_age	0.001
Policies_in_force & N_claims_history	-0.021	R_Claims_history & Car_age	-0.091
sum of linear terms: 0.7305			
exp of sum of linear terms: 2.0761			

4.6. RuleFit

En el presente apartado se exponen los resultados y características de un modelo construido utilizando el algoritmo *RuleFit* sin restricciones a la profundidad y número de reglas. No obstante, como se comentó en el apartado previo en el que se utilizaba como modelo sustituto, se pueden aplicar fácilmente restricciones a dichos parámetros para reducir la complejidad. Cabe destacar que las penalizaciones sobre el número de reglas conllevan una pérdida mayor de capacidad predictiva del modelo que aquellas aplicadas sobre la profundidad de los árboles utilizados para construir estas reglas.

4.6.1. Métricas

Para este modelo sin restricciones, se obtienen las métricas recogidas en la tabla 29.

Tabla 29. Resultados métricas RuleFit

Train Set		Test Set	
Métrica	Score	Métrica	Score
R^2	0.5317	R^2	0.5016
Mean Absolute Error	0.3082	Mean Absolute Error	0.3180
Root Mean Squared Error	0.6657	Root Mean Squared Error	0.6886
Mean Deviance	0.4432	Mean Deviance	0.4742
Media Predicciones	0.3796	Media Predicciones	0.3823

Que, como se puede observar, se asemejan a los resultados obtenidos con los modelos más potentes expuestos previamente.

4.6.2. Comportamiento global

La implementación del algoritmo en Python, al estimar los coeficientes de la regresión y de las reglas de decisión, también incorpora por defecto la importancia relativa que conlleva cada término, siendo los 10 principales los recogidos en la tabla 30 (en este caso son todo reglas, los términos representativos de los coeficientes de la regresión lineal tienen menor importancia).

Tabla 30. Importancia de las variables y reglas modelo RuleFit

Término	Tipo	Importancia	Coef.
N_claims_history <= 18.5 & Lapse > 0.5	regla	0.1224	-0.2922
R_Claims_history <= 1.985 & R_Claims_history <= 4.135 & Lapse > 0.5	regla	0.0700	0.1814
N_doors_6 <= 0.5 & Lapse <= 0.5 & R_Claims_history <= 0.995	regla	0.0648	-0.1386
Policies_in_force <= 5.5 & Lapse > 0.5	regla	0.0582	-0.1429

R_Claims_history > 1.2 & Lapse > 0.5 & R_Claims_history <= 3.025	regla	0.0550	-0.3282
N_claims_history > 3.5 & R_Claims_history > 0.465 & Policies_in_force > 1.5	regla	0.0543	0.2351
R_Claims_history <= 2.495 & Policies_in_force <= 1.5 & Lapse <= 0.5 & R_Claims_history > 0.275	regla	0.0528	-0.1315
R_Claims_history <= 2.045 & Lapse <= 0.5 & R_Claims_history > 0.255	regla	0.0455	0.0986
N_claims_history <= 14.5 & R_Claims_history <= 3.490 & Lapse <= 0.5 & R_Claims_history <= 0.965	regla	0.0438	-0.0948
R_Claims_history > 1.505 & Lapse > 0.5	regla	0.0436	0.2657

La contribución marginal de cada variable independiente a las predicciones viene determinada por dos factores, siendo el primero de ellos la parte del efecto lineal de las variables, depende únicamente de los coeficientes estimados, escalados por el valor que tome la variable, representando estos coeficientes la pendiente de cada curva de contribución marginal recogida en la figura 28.

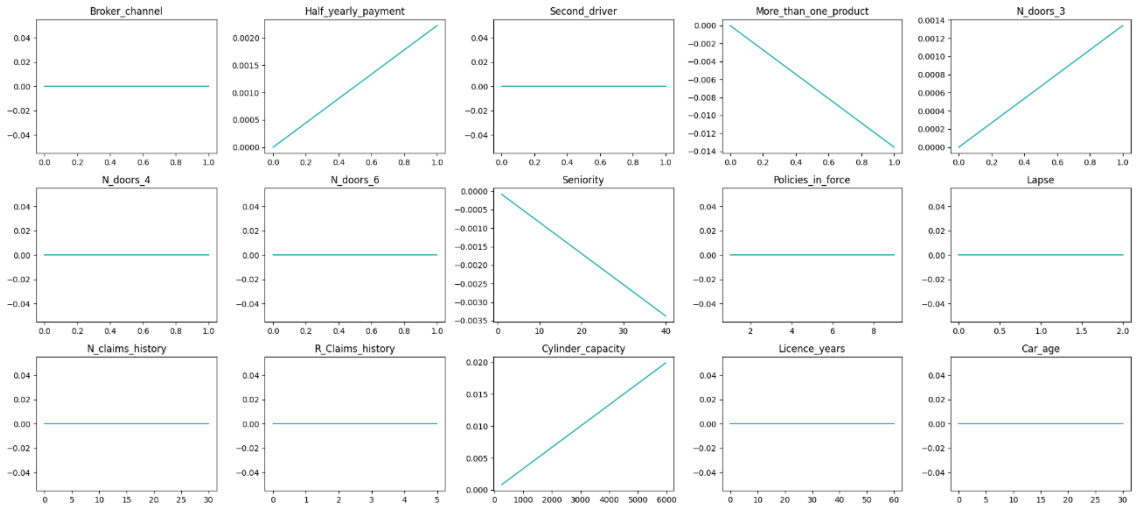

Figura 28. Contribuciones marginales de las variables en el modelo RuleFit. Fuente: Elaboración propia

Al valor correspondiente de cada $f(x)$ obtenido se le suma el de aquellas reglas que se cumplan, como ejemplo se pueden observar las presentadas en la tabla anterior. Al ser el modelo entrenado con regresión de Lasso, se descartan efectos marginales y reglas que no aportan mejoras al modelo, por lo que se observan esas rectas planas con $f(x) = 0$ para todos los valores.

4.6.3. Comportamiento local

De nuevo, al tratarse de un modelo inherentemente interpretativo, se puede conocer de antemano tanto la predicción que resultará del modelo para una instancia concreta.

Con los coeficientes obtenidos de la parte del modelo de regresión, se obtienen las contribuciones marginales de las variables independientes recogidas en la tabla 31

Tabla 31. Contribuciones de las variables a la predicción del modelo RuleFit para el perfil de ejemplo

Variable	Contribución
Constante	0.6212
Half_yearly_payment	0.0022
Seniority	-0.002
Cylinder_capacity	0.0052

El resto de las variables presentan una contribución de 0 sobre la predicción. Por otro lado, el conjunto de reglas estimado es extenso, por lo que a continuación se recogen algunos ejemplos de estas:

Tabla 32. Ejemplos contribuciones reglas a la predicción del modelo RuleFit sobre el perfil de ejemplo

Regla	Contribución
N_claims_history <= 18.5 and Lapse <= 0.5 and R_Claims_history > 0.1850	0.0254
R_Claims_history > 0.565	0.0039
Car_age <= 15.5 and R_Claims_history > 0.295	0.0173
Car_age <= 15.5 and Cylinder_capacity <= 2495.5 and N_doors_6 <= 0.5 and R_Claims_history > 0.995 and Lapse <= 0.5	0.0817

Siendo la suma de las contribuciones de las variables y las reglas cumplidas 2.076, es decir, el valor exacto de la predicción del modelo para la instancia específica.

5. COMPARATIVA MODELOS

En los capítulos previos se han descrito los diferentes modelos y expuesto sus resultados en el contexto del caso práctico presentado. Así, haciendo uso del conjunto de esta información, el presente capítulo destaca las principales similitudes y diferencias entre los modelos, para finalmente ordenar, bajo el paradigma de interpretabilidad que es objeto de estudio, aquellos modelos que presentan mayores virtudes.

5.1. Principales Métricas

Poniendo el foco en el desempeño observado sobre el conjunto de prueba, donde realmente se puede discernir qué modelos son capaces de extender los patrones aprendidos a instancias no vistas previamente, se recoge en la tabla 33 el conjunto de métricas principales.

Tabla 33. Comparativa de métricas de los modelos sobre el *test set*

Modelo/Métrica	R^2	Mean Absolute Error	Root Mean Squared Error	Media Predicciones
GLM	0.279	0.5249	0.8282	0.4698
Random Forest	0.5100	0.3067	0.6828	0.3782
DNN	0.5070	0.3078	0.6848	0.3766
EBM variables	0.4634	0.3399	0.7145	0.3803
EBM interacciones	0.5035	0.3179	0.6873	0.3810
NAM variables	0.4626	0.3359	0.7150	0.3775
NAM interacciones	0.5024	0.3156	0.6880	0.3811
RuleFit	0.5016	0.3180	0.6886	0.3823

De esta primera comparativa resaltan varios puntos:

I. Como se apuntaba, el modelo lineal generalizado (GLM) presenta un desempeño notablemente inferior al resto de modelos expuestos.
II. Además, siendo el promedio observado de número de siniestros del conjunto de prueba 0.3787, el GLM sobreestima a nivel portfolio de forma mucho más notoria que el resto de los modelos, mientras que el modelo de bosques aleatorios es el que más se ajusta a este promedio del portfolio.
III. Si bien los dos modelos caja negra, es decir, Random Forest y DNN, se sitúan a la cabeza en el porcentaje de varianza explicada (R^2), no presentan diferencias significativas respecto a los modelos inherentemente interpretativos expuestos en sus versiones más completas (EBM interacciones, NAM interacciones y RuleFit),

diferencia que no justificaría en ningún caso su elección sobre estos últimos dada la pérdida de interpretabilidad que conllevan.

5.2. Predicción promedio versus observada por variable

La dinámica de sobreestimación del promedio del número de siniestros por parte del GLM se traslada también a un nivel más granular. Esto se refleja en el estudio del promedio por variable, donde se expone la figura 29 como ejemplo para la variable antigüedad del vehículo (‘*Car_age*’).

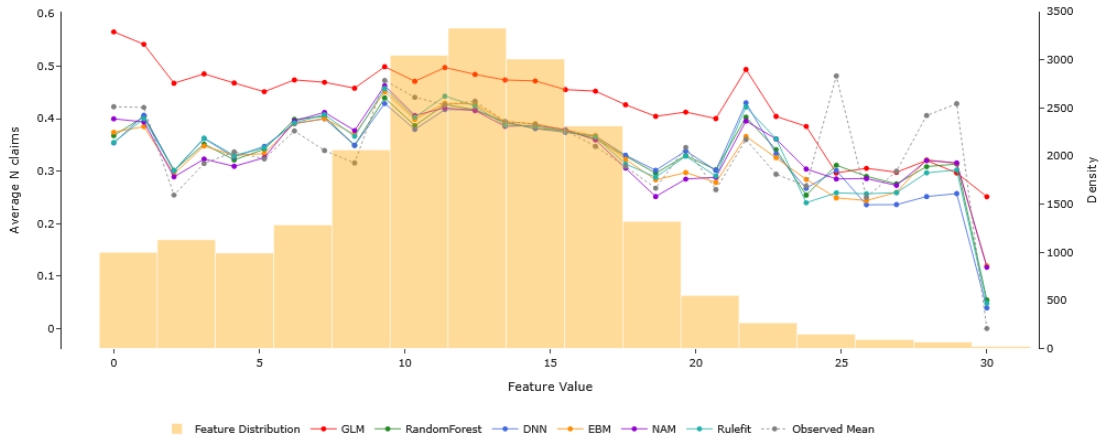

Figura 29. Comparativa de la Frecuencia promedio observada versus los distintos modelos en función del valor de la variable '*Car_age*'. Fuente: Elaboración propia

Mientras que la figura 30 representa también esta predicción promedio pero para la variable antigüedad en la compañía (‘*Seniority’*).

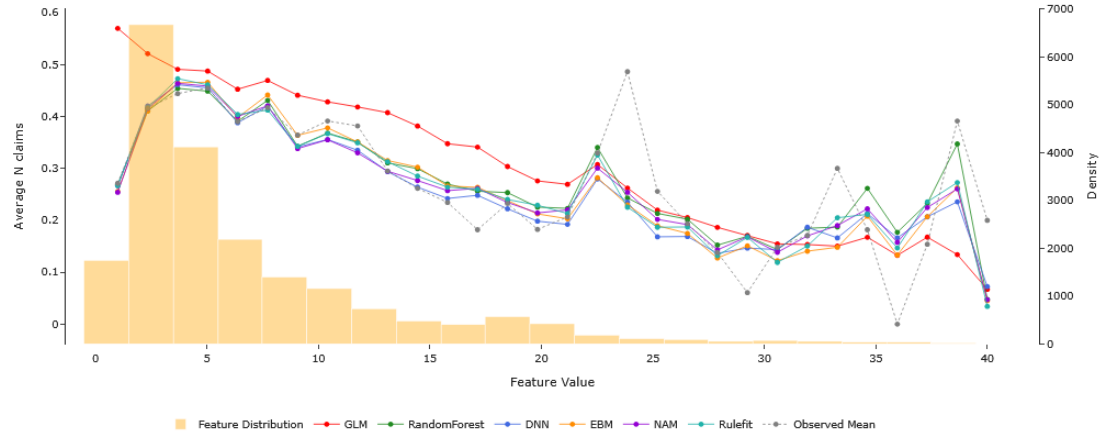

Figura 30. Comparativa de la Frecuencia promedio observada versus los distintos modelos en función del valor de la variable 'Seniority'. Fuente: Elaboración propia

Como se puede observar tanto en la figura 29 como 30, cualquiera de los modelos que no son el GLM recoge correctamente la tendencia real observada en el grueso de la variable donde se concentran los datos, solo presentando divergencias comunes en la cola de la distribución, donde las estimaciones son menos fiables.

En cualquier caso, dejando de lado el modelo base (GLM), este aspecto no aporta una distinción clara entre el resto de modelos.

5.3. Contribución por variable y estimaciones locales

No obstante, ninguna de las dos comparativas previas arroja valor a la hora de dilucidar qué tan interpretativo es un modelo, o si bien no lo es en absoluto.

Por tanto, fijando como objetivo la determinación de esta interpretabilidad, resulta vital comprender y tener certeza de los efectos que tienen las variables, en su rango completo de valores, sobre el modelo. Así, un modelo interpretativo debe permitir conocer con exactitud qué contribuciones marginales suponen estas variables para cada valor que pueden tomar y con independencia del valor que tomen el resto de las variables, lo que además permite conocer cómo varían las predicciones al modificar cualquier componente.

En este punto encontramos ya la desventaja que suponen los modelos Random Forest y DNN propuestos, puesto que, al margen de las buenas métricas que presentan, no permiten conocer con exactitud estos impactos. Si bien se ha tratado en apartados previos de mostrar el impacto de las variables, a través de los gráficos de dependencia parcial, estos sólo aportan una intuición de la tendencia promedio de subida o bajada de la predicción ante una alteración de las variables, pero sólo reflejan dicha tendencia, no representan ninguna contribución exacta.

Por tanto, ejemplificado de nuevo para la variable de antigüedad del vehículo, la siguiente figura recoge los modelos que **sí** permiten discernir dicho efecto concreto sobre el output final del modelo:

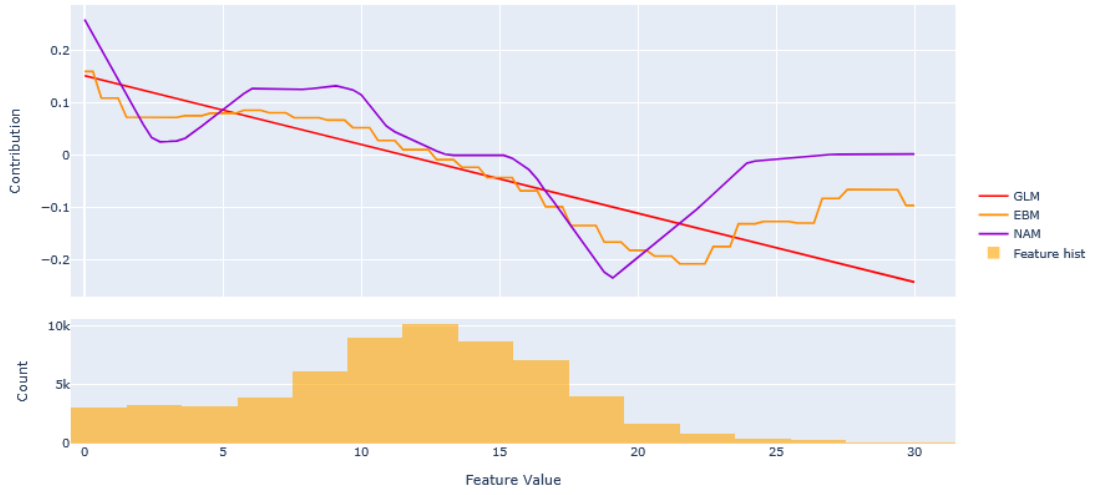

Figura 31. Comparativa de las contribuciones marginales de los modelos interpretativos propuestos en función del valor de la variable 'Car_age'. Fuente: Elaboración propia

Esto aporta una capacidad de análisis sobre el modelo notablemente superior, permitiendo conocer tanto el qué se predice, como lo que es más importante, el por qué genera dicha predicción.

Cabe resaltar que en este tipo de visualización también se incluirían los términos lineales correspondientes a las variables del modelo que sigue el algoritmo *RuleFit*. No obstante, esta no se incluye en la figura 31 puesto que, como se vio en el capítulo previo, para el caso de estudio presentado el modelo RuleFit estima prácticamente solo en base a reglas, por lo que, como se observa en la figura 28, la contribución marginal de la variable '*Car_age*' es 0 para cualquier valor.

De esta forma, con cualquiera de estos modelos (GLM, EBM, NAM y RuleFit) se conoce la predicción local exacta que hará el modelo para un conjunto determinado de valores de las variables, mientras que en aquellos no interpretativos (Random Forest y DNN) esto no es posible.

5.5. Balance de la comparativa

Se puede comprobar a través de las métricas expuestas que los modelos inherentemente interpretativos propuestos (EBM, NAM y RuleFit) efectivamente alcanzan una capacidad predictiva similar a la de otros modelos punteros caja negra, y muy superior a la presentada por el modelo lineal generalizado (GLM).

Estos tres modelos se basan fundamentalmente en la capacidad de descomposición que aporta su aditividad, que permite modelar con distintas soluciones cada componente y analizarlo de forma específica e independiente del resto.

Por ello, ponderando la capacidad de justificación ante terceros, facilidad de descubrimiento de tendencias y potencial mejora del modelo que aporta la interpretabilidad de estos, resultan victoriosos en la comparativa frente a los modelos opacos (Random Forest y DNN).

Ahora bien, ¿es alguno de los tres modelos objetivamente mejor? Como suele suceder, depende. En este caso, depende de qué aspecto resulte primordial a la entidad responsable del modelo:

I. Si se busca flexibilidad para adaptar partes del modelo a diversos requisitos y/o un potencial predictivo mayor, el modelo NAM se situaría a la cabeza.
II. Si se busca una solución más simple y cercana a modelos de sobra conocidos, el algoritmo RuleFit presenta estas características (aunque se debe valorar el impacto sobre la capacidad predictiva para la tarea que se requiera derivada de reducir la complejidad de las reglas).
III. Por último, si se busca una solución robusta, con una implementación sólida ya desarrollada, sería recomendable optar por el modelo EBM.

6. CONCLUSIONES

Al inicio del presente trabajo se ha descrito la relevancia de asegurar la interpretabilidad en la implementación de modelos de aprendizaje automático y las grandes ventajas que conlleva. Puesto que es un campo que ha cobrado mayor relevancia e interés recientemente, son numerosos los avances que continúan sucediéndose en el mismo, que sin duda se siguen produciendo durante el desarrollo de esta tesis.

Por tanto, el número de opciones disponibles a utilizar para implementar esta interpretabilidad es elevado, por lo que en este caso se ha optado por utilizar métodos más contrastados como es el caso de los métodos modelo-agnósticos tanto globales como locales, junto con otros más novedosos bajo el prisma de los modelos de aprendizaje automático inherentemente interpretativos.

Tras aplicarlos en el contexto de un caso práctico propio de la práctica actuarial, se observa que, si bien los métodos contrastados son útiles a la hora de aportar información sobre el comportamiento general y local del modelo, no alcanzan el nivel granular de interpretabilidad que se entiende requerido para cumplir el objetivo último del presente trabajo, que es contribuir al desarrollo de metodologías que aúnen el potencial predictivo del aprendizaje automático con la transparencia necesaria, pudiendo así garantizar el uso de estos modelos en el contexto del sector asegurador.

No obstante, los resultados obtenidos por los tres modelos inherentemente interpretativos (EBM, NAM y RuleFit) resultan muy esperanzadores, puesto que en todos los casos igualan en capacidad predictiva manteniendo la transparencia requerida. Así, la elección de cualquiera de estos tres modelos resulta conveniente, debiendo ponderar sus principales ventajas individuales a la hora de decantarse por uno en concreto.

El caso presentado es simple, y los resultados en cuanto a métricas de todos los modelos tienen potencial de mejora con el ajuste de hiperparámetros necesario, entre otras medidas, pero suficientemente robusto para establecer una base sólida de partida que extender a otras aplicaciones en sector seguros y similares.

Como se ha comentado, el número de soluciones de aprendizaje automático interpretativo no hace sino aumentar, como es el caso de la reciente publicación de las redes de Kolmogorov-Arnold, que sustituyen la base de funcionamiento de las redes neuronales clásicas por una estimación directa de las funciones que minimizan la pérdida. Así, se abre la puerta a la inclusión de estos nuevos desarrollos en futuras investigaciones, pudiendo demostrar que estos no sólo logren mantener la transparencia requerida, sino que también superan en prestaciones a los modelos caja-negra.

BIBLIOGRAFÍA

Biran, O., & Cotton, C. (2017). Explanation and Justification in Machine Learning: A Survey. *Proceedings of the IJCAI-17 Workshop on Explainable Artificial Intelligence (XAI)*.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5-32.

Fisher, A., Rudin, C., & Dominici, F. (2019). All Models are Wrong, but Many are Useful: Learning a Variable's Importance by studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research, 20*, 177.

Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, Vol 29, No. 5*, 1189-1232.

Friedman, J., & Popescu, B. (2008). Predictive Learning via Rule Ensembles. *The Annals of Applied Statistics, Vol. 2, No. 3*, 916-954.

Kim, B., Khanna, R., & Koyejo, O. (2016). Examples are not Enough, Learn to Criticize! Criticism for Interpretability. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, (págs. 2288–2296). Nueva York.

Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *31st Conference on Neural Information Processing Systems (NIPS).* Long Beach, CA, USA.

Molnar, C. (2025). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable (3rd ed.).* Obtenido de christophm.github.io/interpretable-ml-book/

Nelder, J., & Wedderburn, R. (1972). Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General), 135, Part 3.*, 370-384.

Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). *InterpretML: A Unified Framework for Machine Learning*. Obtenido de https://arxiv.org/abs/1909.09223

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135-1144.

Segura-Gisbert, J., Lledó, j., & Pavía, J. M. (2025). Dataset of an actual motor vehicle insurance portfolio. *European actuarial Journal, 15*, 241-253.