uc3m | Universidad Carlos III de Madrid

Máster Universitario en Ciencias Actuariales y Financieras
2019-2020

Trabajo Fin de Máster

"Modelos de predicción del coste del siniestro en Seguros de Salud"

Pablo Gil de Gómez Jiménez

Tutor/es

José Miguel Rodríguez-Pardo

Jesús Ramón Simón del Potro

Madrid, 2020

DETECCIÓN DEL PLAGIO

La Universidad utiliza el programa **Turnitin Feedback Studio** para comparar la originalidad del trabajo entregado por cada estudiante con millones de recursos electrónicos y detecta aquellas partes del texto copiadas y pegadas. Copiar o plagiar en un TFM es considerado una **Falta Grave**, y puede conllevar la expulsión definitiva de la Universidad.

ÍNDICE DE CONTENIDO

ÍNDICE DE ILUSTRACIONES

RESUMEN

Los datos siempre han estado en el núcleo de las compañías de seguros. En las últimas décadas, la disrupción de la Inteligencia Artificial coloca a las técnicas de manejo de datos en el centro de una profunda revolución. Este trabajo profundiza en el uso de técnicas de Machine Learning en los estudios de severidad de siniestros del ramo de salud, exponiendo su base teórica, funcionamiento y comparando su desempeño en un caso práctico real. La principal conclusión es que los algoritmos avanzados de Machine Learning arrojan mejoras increíbles con respecto a los métodos tradicionales de regresión.

ABSTRACT

Data has always been at the heart of insurance companies. In recent decades, the disruption of Artificial Intelligence places data management techniques at the center of a profound revolution. This paper explores the use of Machine Learning algorithms in loss severity studies of health claims, exposing their theoretical basis, operation and comparing their performance in a real case study. The main conclusion is that Machine Learning advanced algorithms show incredible improvements over traditional regression methods.

1. INTRODUCCIÓN

La asistencia sanitaria tiene una misión fundamental en la marcha de una sociedad y es que garantiza el cuidado de la salud de sus ciudadanos. Es por ello que los Gobiernos centran sus esfuerzos en desarrollar y mejorar continuamente sus sistemas de salud. Una cuestión de vital importancia es el acceso por parte de la población a las coberturas de asistencia sanitaria. En algunos países, esto no supone un problema, ya que la sanidad es universal. Es el caso de España, que presenta uno de los sistemas públicos de salud más completos y eficaces dentro de la Unión Europea, tanto por su universalización como por la calidad y seguridad de sus servicios, permitiéndole alcanzar unas tasas de mortalidad tratable muy bajas y un nivel de esperanza de vida espectacular con respecto a otros países de la Unión Europea[1].

Paralelamente a los sistemas públicos, se desarrolla un enredado esquema de seguros privados de salud, que en muchos lugares del mundo supone la principal o única vía de acceso a la asistencia sanitaria. A simple vista, el funcionamiento de un seguro de salud es sencillo, puesto que trabaja como un instrumento que facilita el acceso a los servicios sanitarios a cambio de una cantidad de dinero. Sin embargo, esconde detrás una compleja maquinaria que convierte a este tipo de seguros en un tema interesante a estudiar.

Si se centra el análisis en la arquitectura de las compañías de seguros, son de especial interés las etapas dedicadas a la obtención de información y al posterior análisis de datos. Estos datos se suelen utilizar para estudiar el perfil de los individuos que solicitan un seguro y clasificarlos en función del riesgo que presentan. Se trata de las funciones de tarificación y suscripción ("pricing and underwriting" en inglés), que basan su actividad en técnicas de inteligencia de negocio, traduciéndose esto a menudo en procesos manuales, lentos y poco eficientes.

La disrupción de disciplinas como la Inteligencia Artificial o el Machine Learning promete iniciar una profunda transformación en los métodos existentes. En palabras de Andrés González García, socio y cofundador de CleverData, "este nuevo enfoque disruptivo crea sistemas que aprenden de los datos y modelan el comportamiento y características de los clientes para establecer una prima de riesgo personalizada y adaptada a cada cliente. La principal ventaja de esta perspectiva radica en que se trata de sistemas dinámicos: el modelo se reentrena periódicamente y aprende automáticamente con el tiempo. A medida que el modelo se alimenta con nuevos datos, descubre nuevos comportamientos y patrones que permiten conocer mejor a los clientes y, en definitiva, mejorar el cálculo del riesgo asociado"[2].

Este trabajo tiene como fin estudiar alguno de los algoritmos más utilizados en la actualidad por los científicos de datos y presentar su posible aplicación en un ejemplo de predicción de coste de siniestros de salud.

[1] COMISIÓN EUROPEA: "State of Health in the EU: España, perfil sanitario del país 2017" *European Observatory on Health Systems and Policies*, 2017.

[2] TILVES M.: "A fondo: Así ayuda el Machine Learning al sector seguros". Silicon, 2017. Disponible en:
https://www.silicon.es/a-fondo-machine-learning-seguros-2335520

2. VALORACIÓN DE RIESGOS EN SEGUROS DE SALUD

2.1. Concepto y alcance del seguro de salud

En la mayoría de los países de la Comunidad Económica Europea, el **servicio de asistencia sanitaria** es ofrecido tanto por el sistema público, como por agentes privados. Por un lado, la sanidad pública es universal y está financiada a través de las contribuciones a la Seguridad Social. Por el otro, las compañías aseguradoras ofrecen seguros de salud privados que permiten el acceso a la sanidad privada, que se distingue por un mayor número de centros especializados y por una menor congestión que el sistema público.

Con respecto al marco legislativo, en España el **seguro de salud** se rige, por un lado, por el título 1º de la LCS que se aplica a todos los contratos de seguro y, por otro lado, por la sección 1º del título III que se refiere a todos los seguros de personas. Al basarse el marco legislativo en establecer mínimos y permitir flexibilidad de actuación a las partes intervinientes, adquieren gran importancia las condiciones generales y particulares del contrato de seguro, donde se recogen coberturas, riesgos excluidos o cláusulas de exoneración de responsabilidad.

Antes de comenzar el estudio, conviene distinguir entre **seguro de enfermedad y seguro de asistencia sanitaria.** El primero consiste en el pago de una indemnización cuando se produzca el hecho contingente, la enfermedad, con el fin de cubrir todas las consecuencias económicas derivadas de ella. El segundo, en cambio, asume solamente los gastos médicos, bien sea directamente a través del cuadro médico de la aseguradora, o bien mediante el reembolso de gastos al asegurado. Cuando se habla de seguro de salud, normalmente se hace referencia al seguro de asistencia sanitaria.

También conviene señalar que los seguros de salud se suelen comercializar bajo dos modalidades básicas, diferenciando entre contratos de seguro **particulares** y contratos de seguro **colectivos**. Normalmente cada contrato colectivo se diseña y negocia a medida con la empresa y a menudo bajo condiciones muy especiales, por lo que resulta complicado establecer patrones generales para los mismos. De esta manera, presenta mayor interés el estudio de los contratos particulares.

2.2. Consideraciones generales sobre tarificación en seguros de salud

En el mercado europeo de seguros de salud, se distinguen dos formas básicas de calcular una prima: el método basado en un colectivo (o grupo) y el método basado en el riesgo individual.

- Por un lado, las primas calculadas por el método de grupo son las mismas para todos los integrantes de un colectivo. En este caso, a través de un censo del grupo, con fechas de nacimiento y datos de sexo, se calcula una prima nivelada (cuantía única para todos) o una prima por tramos de edad.
- Por otro lado, las primas calculadas por el método del riesgo individual se basan en la evaluación de distintos factores, incluyendo de nuevo la edad y el género, pero también la profesión, el historial de enfermedades familiares, el histórico de siniestralidad, los hábitos de tabaco y alcohol, indicadores de obesidad, etc. No

obstante, en la práctica, las compañías aseguradoras no pueden acceder a tantos datos y se limitan a utilizar la edad, el género y las enfermedades preexistentes.

Estos métodos están relacionados con la forma de comercializar el seguro, de forma individual o a través de empresas. El primer método lo suelen seguir los contratos colectivos, mientras que el segundo es el preferido por los contratos individuales. La excepción se encuentra en Irlanda, donde los aseguradores están obligados a ofrecer primas calculadas por el método de grupo, además de garantizar la libre adhesión y coberturas vitalicias[3]. Al margen de esta excepción y, como se ha indicado en el apartado anterior, presenta mayor interés el estudio de los contratos individuales y, por lo tanto, el **método basado en el riesgo individual**.

Bajo este método y, a modo de simplificación, la prima de un seguro de salud se podría aproximar por el Principio de Prima del Valor Esperado (PPVE), representado a través de la siguiente expresión.

$$\pi(y) = (1 + k_1)E(y) + k_2\rho(-y)$$

Siendo $E(y)$ el valor esperado del coste agregado de una póliza, k_1 el recargo de seguridad, k_2 el recargo por riesgo y $\rho(-y)$ una medida de riesgo[4].

Al margen de consideraciones más avanzadas sobre la medida de riesgo, se aprecia la importancia de la formulación de un correcto modelo para el coste agregado de una póliza. De manera general, el coste agregado o siniestralidad total se puede aproximar mediante la siguiente ecuación[5]:

$$S_t = X_1 + X_2 + \cdots + X_{N_t}$$

Siendo N_t la variable aleatoria número total de siniestros en t años y $\{X_i\}_{i=1}^{N_t}$ la variable aleatoria severidad del siniestro i.

En este punto, es imprescindible establecer modelos para las variables severidad y frecuencia, fundamentales para alimentar cualquier modelo de tarificación de seguros de no vida. Este trabajo surge como respuesta a este problema y, aunque se trabaja con un ejemplo de severidad, las técnicas planteadas también serían de aplicación para modelos de frecuencia.

[3] Traducido y adaptado de EUROPEAN OBSERVATORY ON HEALTH CARE SYSTEMS SERIES: "Voluntary health insurance in the European Union." Chap. 5, In *Funding Health Care: Options for Europe*, edited by Mossialos, E., Dixon, A., Figueras, J. Y Kutzin, J.. Philadelphia (USA), Open University Press, 2001 pp. 129-142.

[4] TSE Y. K.: "Nonlife actuarial models. Theory, methods and evaluation". Cambridge University Press, 2009.

[5] KLUGMAN S. A., PANJER H. H. y WILLMOT G.E.: "Loss Models". Willey, 2029 (Fifth Edition).

2.3. Machine Learning e Inteligencia Artificial en seguros

La Inteligencia Artificial y el Maching Learning están transformando las prácticas habituales en la industria aseguradora. Los datos siempre han estado en el corazón de la industria aseguradora. Pero en los últimos años, estos se sitúan en el eje de una profunda revolución debido a cantidad de información generada diariamente y la velocidad a la que las máquinas las procesan y aprenden de esta.

La Inteligencia Artificial se basa en la filosofía de construir máquinas que puedan pensar como humanos. En el núcleo de esta disciplina, se encuentra el concepto de Machine Learning, que se refiere a la idea de enseñar a los ordenadores a aprender de la misma manera que lo hacen los humanos.

Algunas de las áreas que sufrirán cambios con la revolución de los datos son las siguientes:

- **Los servicios de atención al cliente**. Los algoritmos de Inteligencia Artificial podrían realizar un análisis del perfil de los consumidores para elaborar recomendaciones más personalizadas y mejorar la satisfacción del cliente. También el proceso de gestión de reclamaciones podría automatizarse, ahorrando tiempos e incrementando la calidad del servicio.
- **Detección del fraude**. Algunos algoritmos de Inteligencia Artificial permiten identificar eficazmente siniestros fraudulentos y subrayarlos para análisis más detenidos y elaborados por humanos. Esto podría mejorar las cuentas de pérdidas y ganancias de las compañías.
- **La valoración de riesgos en las funciones de suscripción y tarificación**. Las técnicas de Inteligencia Artificial prometen increíbles aplicaciones en estas áreas. Se abre la vía al acceso de nueva información, como los hábitos de conducción a través de dispositivos de monitorización en los automóviles, o los hábitos de salud a través de dispositivos wearables. Esto permitiría a las compañías mejorar sus modelos de valoración de riesgos y ofrecer servicios personalizados a los clientes. Los algoritmos de Machine Learning también prometen mejorar el rendimiento de los modelos predictivos de siniestralidad.

Existen tres tipos de modelos de Machine Learning: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje reforzado[6].

- En primer lugar, los modelos de aprendizaje supervisado son aquellos que se basan en el estudio del pasado, intentando reproducir la respuesta y anticipar el comportamiento. Normalmente se selecciona una muestra de entrenamiento para ajustar el modelo. Más adelante, se escoge una muestra de validación para estudiar si el modelo es válido para datos no empleados en la fase anterior. Se incluyen aquí los

[6] GUILLEN M. y PESANTEZ-NARVAEZ J.: "Machine Learning y modelización predictiva para la tarificación en el seguro de automóviles". Anales del Instituto de Actuarios Españoles, 4ª época, 24, 2018/123-147. Disponible en:
https://www.actuarios.org/wp-content/uploads/2018/11/123_147_A06.pdf

modelos clásicos de predicción, los árboles de decisión, las redes neuronales y el Support Vector Machine.

- Por otro lado, los modelos de aprendizaje no supervisado tratan de encontrar estructuras o agrupaciones en las que existen similitudes o se comparten rasgos comunes. Seguidamente se trata de forma idéntica a todas las unidades pertenecientes a esos subgrupos.
- Finalmente, los modelos de aprendizaje reforzado se basan en la idea de que una acción particular es seguida por una respuesta deseable o indeseable. Para saber qué tipo de acciones conllevan respuestas deseables, se recurre a procesos de prueba y error para elaborar nuevas estrategias. De esta manera, el objetivo es que, a base de prueba y error, los modelos puedan aprender de sí mismos.

Los modelos de Machine Learning utilizados en este estudio son de aprendizaje supervisado y son un total de seis: regresión lineal, regresión cubist, random forest, gradient boosting machine, extreme gradient boosting y support vector machine.

3. ANÁLISIS EXPLORATORIO DE LOS DATOS

Los datos utilizados se han obtenido del libro "Machine Learning with R" de Brett Lantz[7]. En esta obra el autor utiliza diferentes bases de datos, todas de dominio público, para explicar distintas técnicas de Machine Learning. Entre ellas, se encuentra una base de datos estadounidense de siniestros de salud.

3.1. Presentación de las variables

La base de datos estudiada contiene 7 campos, relacionados todos con las características personales de los contratantes de un seguro de salud. Los campos son los siguientes:

- **Edad**: este campo recoge la edad en años del asegurado.
- **Sexo**: este campo recoge el sexo del asegurado. Se trata de una variable dicotómica, que sólo puede tomar dos valores: "hombre" o "mujer".
- **BMI**: este campo corresponde con el Índice de Masa Corporal (BMI), calculado como el cociente entre peso y estatura [$BMI = \frac{peso[kg]}{estatura[m2]}$]. Esta ratio proporciona una comprensión de la obesidad de una persona, estudiando si los pesos son relativamente altos o bajos en relación con la altura. Idealmente este índice se debería situar entre 18 y 25. Se considera que una persona es obesa cuando esta medida se sitúa por encima de 30.
- **Hijos**: este campo recoge el número de hijos del asegurado.

[7] LANTZ B.: "Machine Learning with R". Packt Publishing, 2013. Disponible en:

https://edu.kpfu.ru/pluginfile.php/278552/mod_resource/content/1/MachineLearningR__Brett_Lantz.pdf

- **Fumador**: este campo recoge los hábitos de tabaco del asegurado. Se trata de una variable dicotómica, que sólo puede tomar dos valores: “sí” o “no”.
- **Región**: este campo recoge el área de residencia en EEUU del asegurado. Esta variable puede tomar los siguientes valores: “noreste”, “noroeste”, “sureste” y “suroeste”.
- **Importe**: este campo recoge la severidad o coste del siniestro de salud.

Cada registro de la base de datos corresponde a un siniestro y, en total, se dispone de 1.338 registros o siniestros.

En los siguientes cuadros se recogen las medidas básicas de estadística descriptiva para cada una de las variables (el análisis de la variable “importe” se reserva para el siguiente subapartado). Tras un análisis inicial, se pueden señalar varios aspectos:

- Se trata de una muestra joven. La media de edad se sitúa en los 39 años. Además, la edad máxima es de 65 años.
- En media los asegurados presentan obesidad. Más de la mitad de la muestra anota un BMI superior a los 30. Además, el primer cuartil es de 26,3, por encima de los valores recomendados de este índice.
- Finalmente, el 79,5% de los asegurados no fuma.

	EDAD	BMI	HIJOS
Mínimo	18	15,96	0
1º Cuartil	27	26,30	0
Mediana	39	30,4	1
Media	39,21	30,66	1,095
3º Cuartil	51	34,69	2
Máximo	64	53,13	5

Sexo	Fumador	Región
Hombre: 676	No: 1.064	Noreste: 324
Mujer: 662	Si: 274	Noroeste: 325
		Sureste: 364
		Suroeste: 325

Dadas las características de la muestra, a partir de la variable BMI, se ha creado una octava variable con el fin de obtener otro indicador de la obesidad del asegurado. De esta manera, se

define la variable dicotómica **"BMI30"**, que muestra el texto "si" para los casos en los que el índice es superior a 30 y el texto "no" si es inferior.

3.2. Estudio de la severidad del siniestro

Recuérdese del capítulo anterior que el cálculo de la Siniestralidad Total consiste en la suma de una serie de variables aleatorias[8]:

$$S_t = X_1 + X_2 + \cdots + X_{N_t}$$

Siendo N_t el número total de siniestros en t años, el objetivo de este trabajo es estudiar modelos para estudiar la severidad de los siniestros $\{X_i\}_{i=1}^{N_t}$.

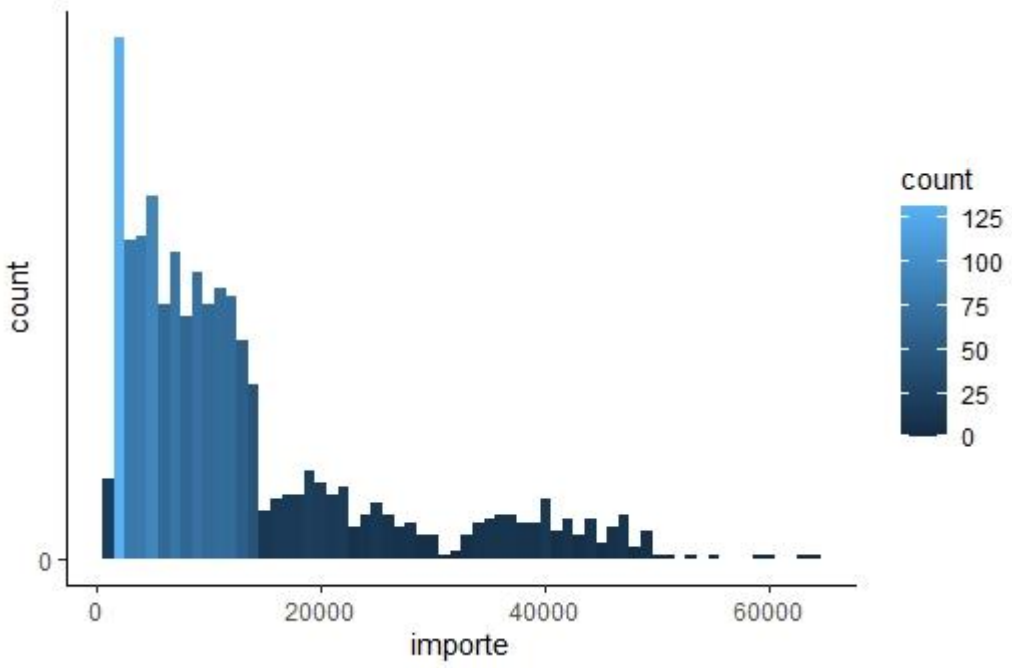
En el siguiente cuadro se recogen las principales medidas de estadística descriptiva para la variable "importe" de los siniestros. La gran diferencia entre media y mediana se debe a la existencia de 'outliers' o valores atípicos.

Mínimo	**1º Cuartil**	**Mediana**	**Media**	**3º Cuartil**	**Máximo**
1.122	4.740	9.382	13.270	16.640	63.770

En la siguiente ilustración se recoge el comportamiento de la variable

[8] KLUGMAN S. A., PANJER H. H. y WILLMOT G.E.: "Loss Models". Willey, 2029 (Fifth Edition).

Ilustración 1. Histograma del importe de los siniestros



Fuente: elaboración propia

A continuación, se muestran algunos modelos paramétricos que suelen utilizarse para modelizar el comportamiento de este tipo de variables aleatorias. Se trata de estudiar si las observaciones se ajustan a alguna de las distribuciones de probabilidad propuestas o si, por el contrario, se debe plantear otro tipo de modelos.

3.2.1. Distribución Gamma

En primer lugar, se propone la **distribución Gamma**. Esta distribución permite modelizar colas exponenciales, es decir, aquellos casos en los que la función de densidad alcanza exponencialmente cero.

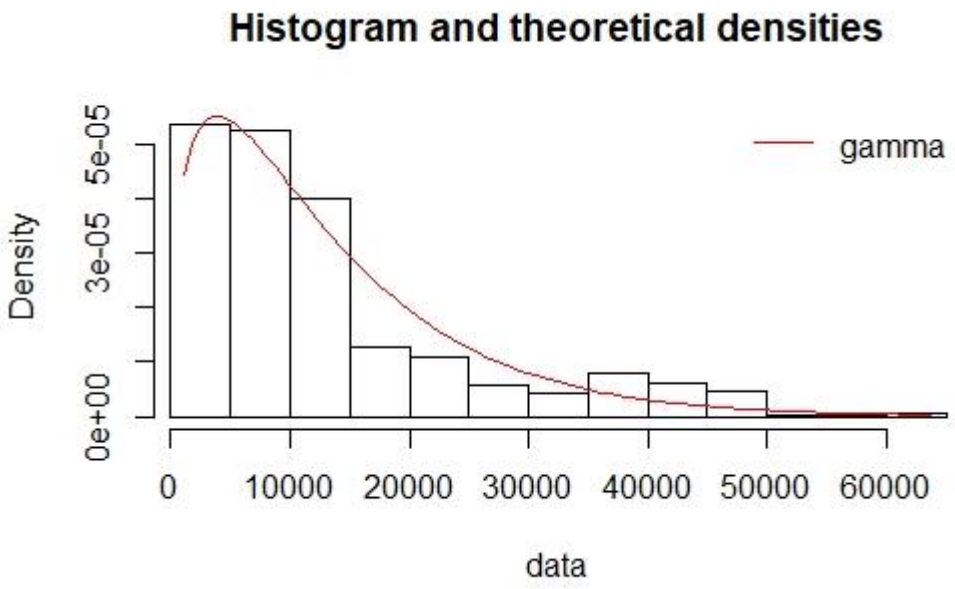
El ajuste al ejemplo ha originado una Gamma con los parámetros recogidos en el siguiente cuadro.

	Estimate	Error
Scale	9.405,928	NA
Shape	1,4112	NA

LogLikelihood: -13.996,09	AIC: 27.996,18	BIC: 28.006,58

Como se puede observar en la siguiente ilustración, la distribución Gamma no consigue capturar la cola de la distribución, por lo tanto, no parece adecuada para el ejemplo.

Ilustración 2. Ajuste a una distribución Gamma



Fuente: elaboración propia

3.2.2. Distribución Log-Normal

En segundo lugar, se propone la **distribución Log-Normal**. Este tipo de distribución es más interesante para capturar los siniestros de cola larga, es decir, los siniestros grandes o catastróficos.

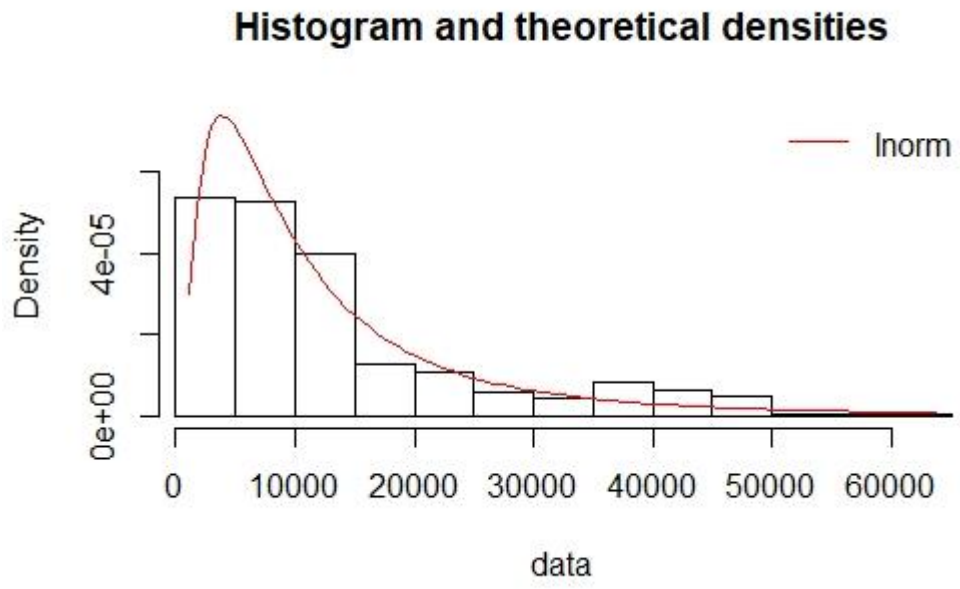
El ajuste al ejemplo ha originado una Log-Normal con los parámetros recogidos en la siguiente tabla.

	Estimate	Error
meanlog	9,0986587	0,02512894
sdlog	0,9191834	0,01776875

LogLikelihood: -13.959,79	AIC: 27.923,58	BIC: 27.933,98

Como se visualiza ilustración, la distribución Log-Normal sí recoge cierto comportamiento en la cola, pero no resulta demasiado preciso en todos los tramos.

Ilustración 3. Ajuste a una distribución Log-Normal



Fuente: elaboración propia

3.2.3. Distribución Pareto

En tercer lugar, se propone la **distribución Pareto**, una de las distribuciones más utilizadas en la modelización de riesgos catastróficos.

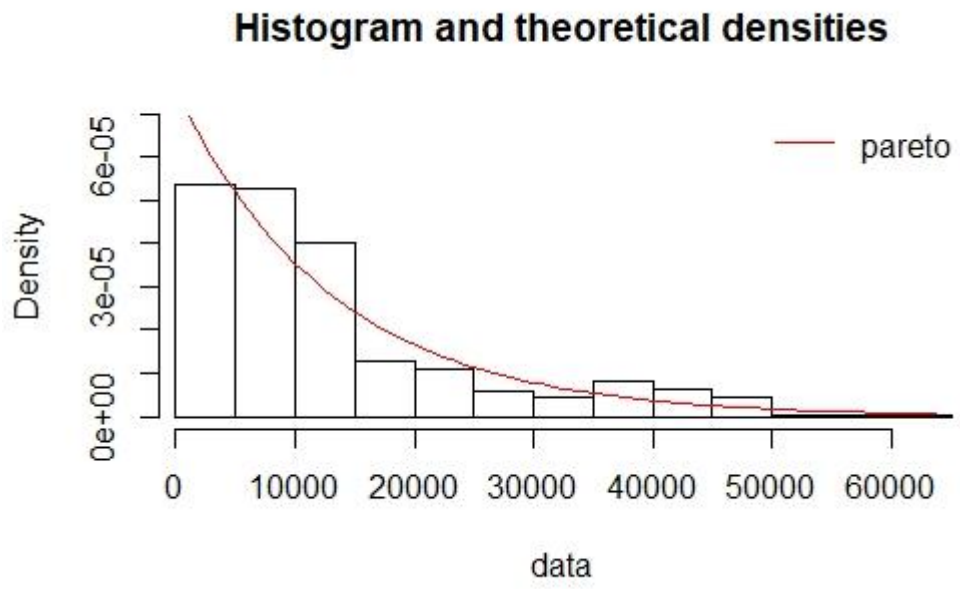
El ajuste al ejemplo ha originado una Pareto con los parámetros recogidos en la siguiente tabla.

	Estimate	Error
shape	7,928337E+05	NA
scale	1,051091E+10	NA

LogLikelihood: -14.040,03	AIC: 28.084	BIC: 28.094,45

Como se visualiza en la ilustración, la distribución Pareto, al igual que la Log-Normal, recoge cierto comportamiento de la cola, pero no parece muy preciso en todos los tramos.

Ilustración 4. Ajuste a una distribución Pareto



Fuente: elaboración propia

3.2.4. Test Kolmogorov-Smirnov

Para comprobar que la severidad se distribuye según alguna de las distribuciones planteadas, se realiza la prueba **Kolmogorov-Smirnov**[9]. Tras realizar la prueba para cada una de las distribuciones candidatas, se han obtenido los siguientes resultados:

Distribución Gamma	Distribución Log-Normal	Distribución Pareto
p-valor = 2,095E-07	p-valor = 0,05566	p-valor = 2,2E-16

Observando los resultados, se puede concluir que en todos los casos se rechaza la hipótesis nula de la prueba Kolmogorov-Smirnov y la severidad no se puede modelizar bajo ninguna de las distribuciones planteadas.

3.3. Interacción entre las variables

[9] En la prueba Kolmogorov-Smirnov se contrasta la hipótesis nula (H0) de que la variable sigue la distribución planteada.

Puede ser interesante el estudio de la relación entre la severidad y el resto de las variables que contiene la base de datos. Este estudio se realiza con la ayuda de gráficos y diagramas.

Se presentan, en primer lugar, los **diagramas de caja**. Un diagrama de caja es una forma de representar un conjunto de datos a través de sus cuartiles. Las cajas indican el cuartil inferior en su extremo inferior, el cuartil superior en su extremo superior y la mediana en su línea intermedia. Las líneas que se extienden más allá de las cajas se llaman "bigotes" e indican la variabilidad fuera de los cuartiles. Los datos fuera de los "bigotes" se conocen como valores atípicos. Para el ejemplo estudiado, el diagrama de caja puede servir para analizar la severidad de los distintos grupos definidos en el resto de las variables. En las siguientes ilustraciones se observan los diagramas de caja del importe de los siniestros para cada una de las variables de la base de datos.

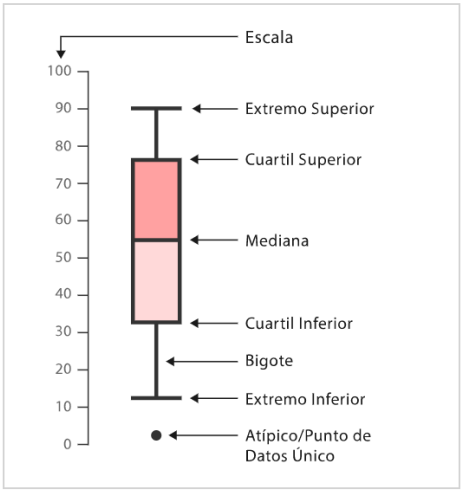
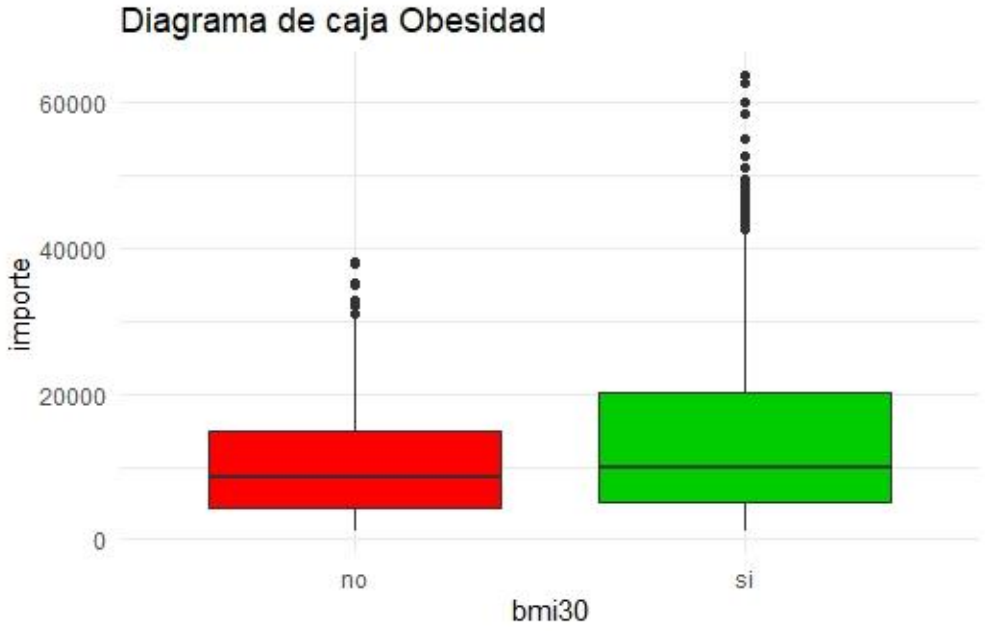


Ilustración 5. Funcionamiento de un diagrama de caja

Con respecto a las "cajas" para la obesidad, la "caja" de las personas obesas presenta una mayor amplitud, su bigote es más extenso y señala un mayor número de valores atípicos.

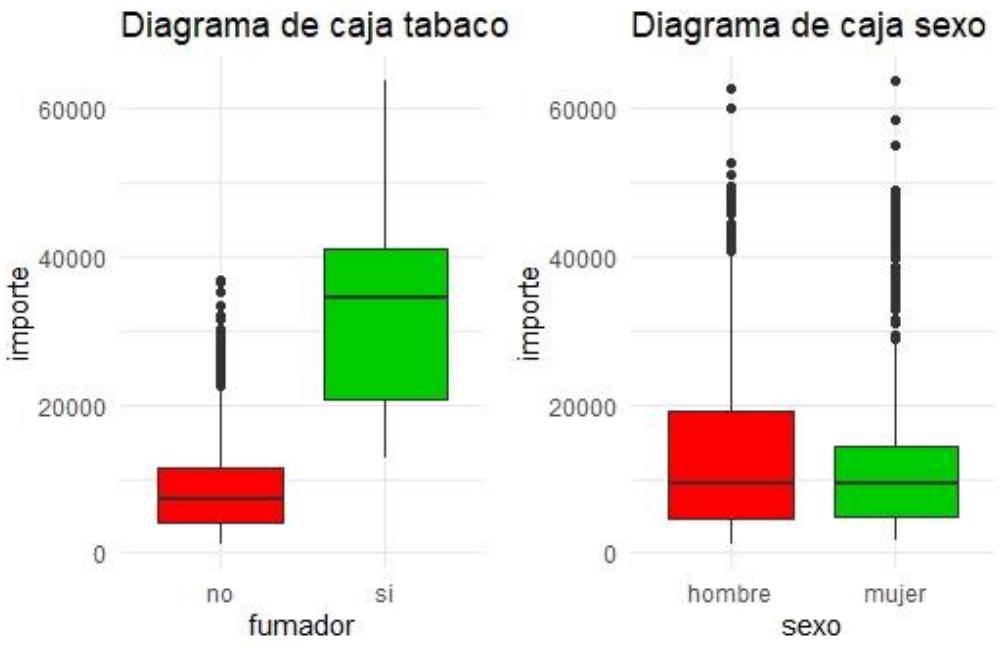
Ilustración 6. Diagrama de costes médicos por obesidad



Fuente: elaboración propia

Con respecto a las "cajas" para los hábitos de tabaco, estas indican que existe una evidente relación positiva entre el tabaco y la severidad del siniestro. No se puede decir lo mismo para "cajas" para el sexo, que parece no ser clave para el estudio de la severidad.

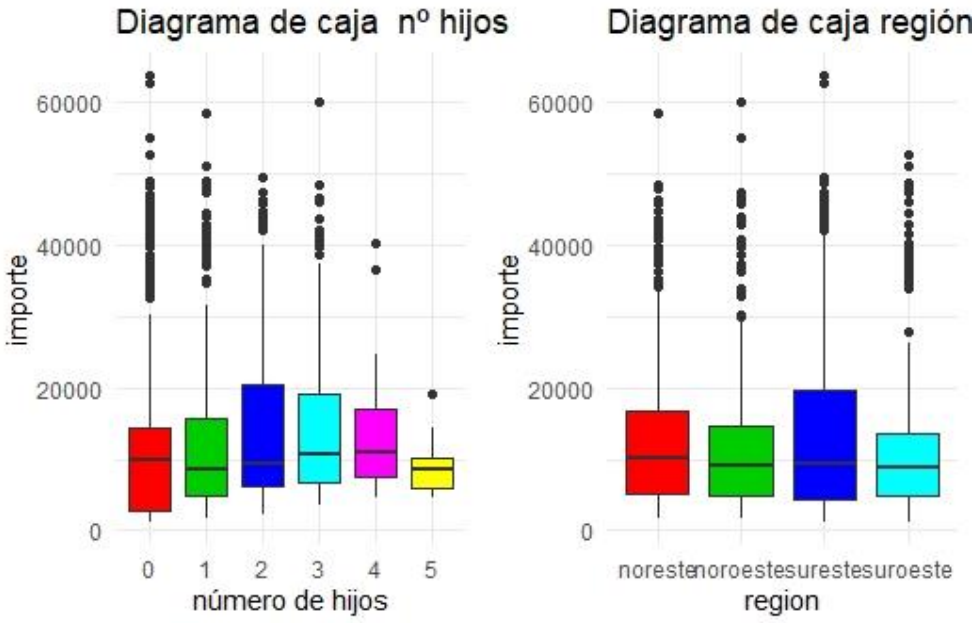
Ilustración 7. Diagrama de costes médicos por hábitos de tabaco y sexo



Fuente: elaboración propia

Con respecto a las "cajas" para el número de hijos, parece existir mayor número de datos atípicos para las personas que no tienen hijos, pero esto también podría deberse a que suponen la mayor parte de la muestra. Tampoco podrían extraerse conclusiones claras para las "cajas" de región de residencia.

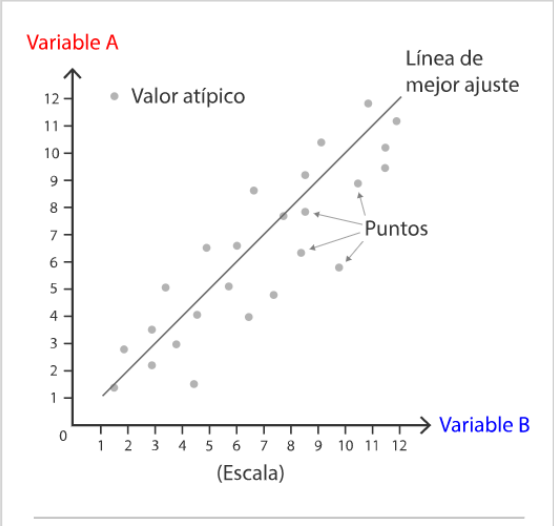
Ilustración 8. Diagrama de costes médicos por número de hijos y región de residencia



Fuente: elaboración propia

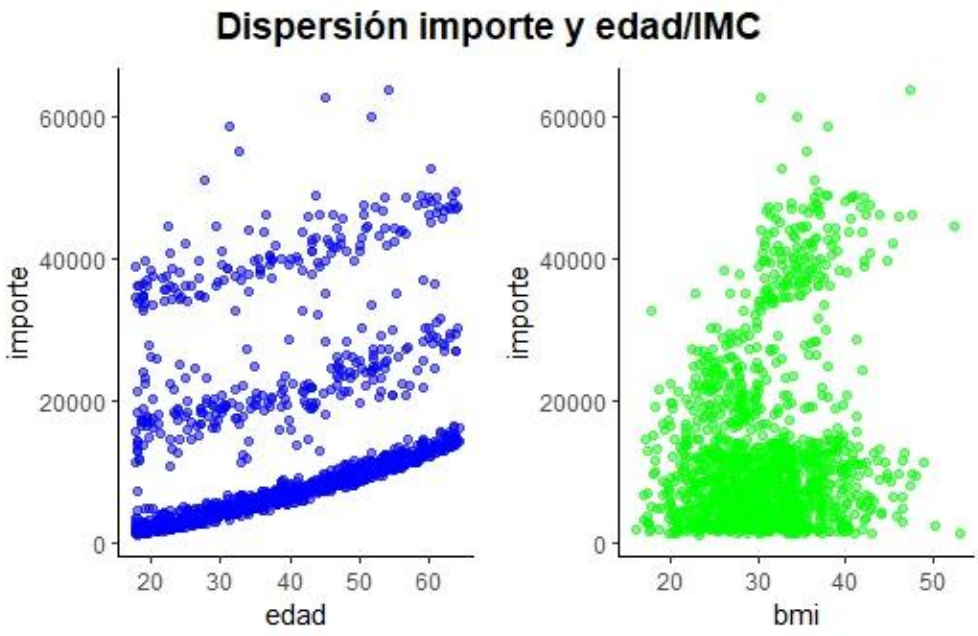
A continuación, se presentan los **gráficos de dispersión**. Un diagrama de dispersión contiene una serie de puntos fijados utilizando un sistema de coordenadas cartesianas, de tal forma que en cada eje se muestra una de las variables. La nube de puntos formada permitirá identificar la relación o correlación entre las variables. Entre los tipos de correlación que existen merece la pena destacar las siguientes: positivo, negativo, nulo, lineal, exponencial y en forma de U. En esta figura también se pueden identificar los datos atípicos como aquellos puntos que aparecen separados de la nube de puntos. En el ejemplo, el gráfico de dispersión puede servir para analizar la correlación entre la severidad y el resto de las variables. En las siguientes ilustraciones se muestran los gráficos de dispersión para el importe y el resto de las variables.

Ilustración 9. Funcionamiento de un gráfico de dispersión



En primer lugar, entre la edad y la severidad se observa una relación positiva. No es tan clara la relación entre la severidad y el Índice de Masa Corporal, aunque se puede intuir una pendiente positiva.

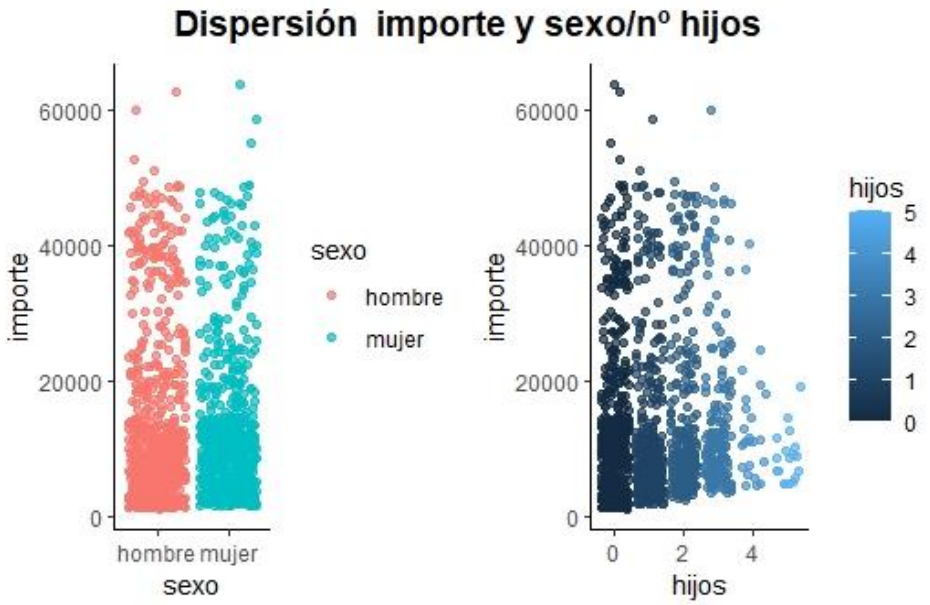
Ilustración 10. Gráfico de dispersión entre importe y edad/IMC



Fuente: elaboración propia

Con respecto a las siguientes dispersiones, no se aprecian tendencias claras para la relación entre la severidad y las variables sexo y número de hijos.

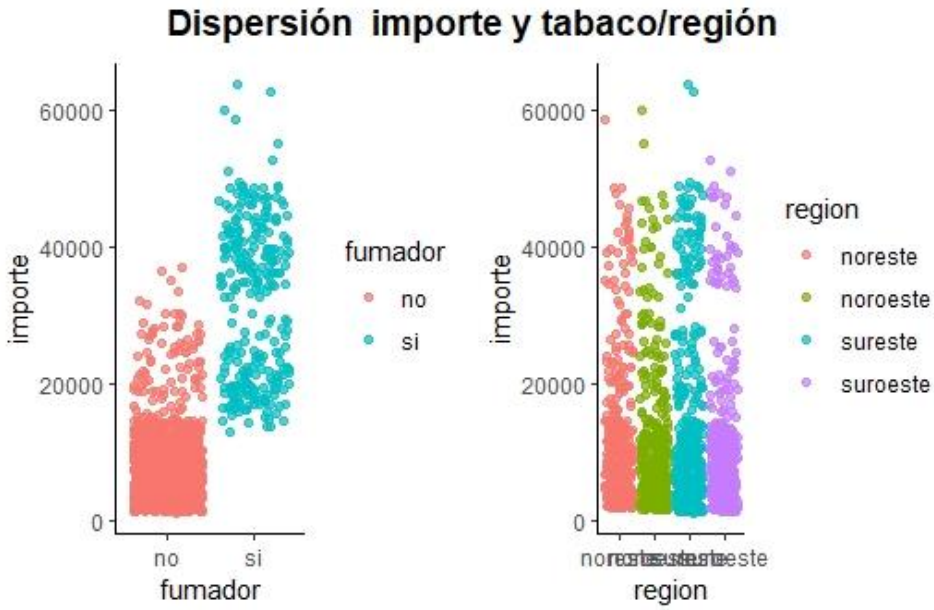
Ilustración 11. Gráfico de dispersión entre importe y sexo/nº hijos



Fuente: elaboración propia

En cuanto a la relación entre severidad y tabaco, se observa una evidente pendiente positiva. No se puede decir lo mismo para la relación entre severidad y región de residencia.

Ilustración 12. Gráfico de dispersión entre el importe y los hábitos de tabaco / región de residencia

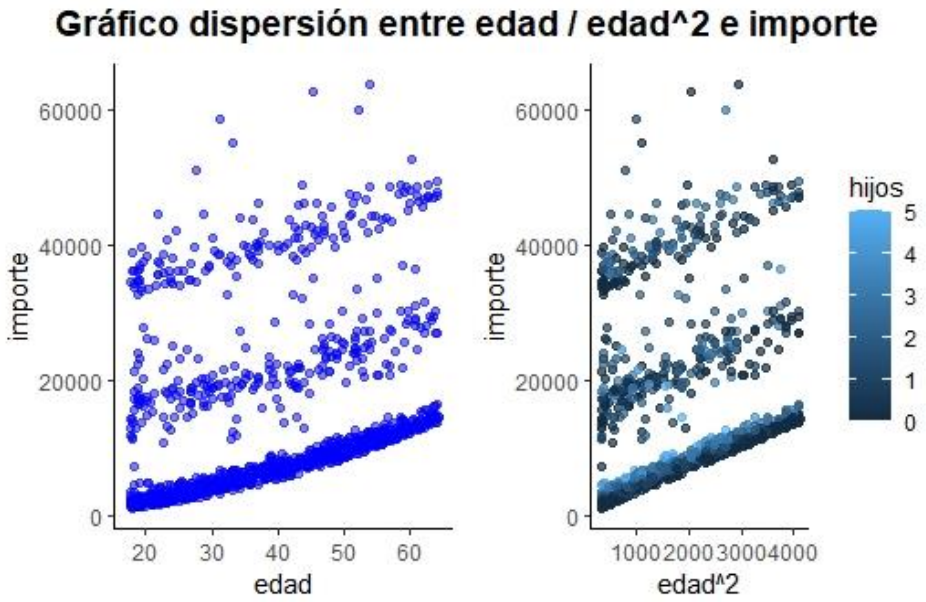


Fuente: elaboración propia

Hasta ahora se ha pensado siempre en explicar la severidad a partir de otras variables de forma lineal. No obstante, podrían existir relaciones de otro tipo, que conviene señalar.

En el caso de la relación entre edad y severidad, esta podría ser cuadrática. Podría pensarse que las edades más bajas no inciden tanto en la severidad como edades más altas. Para ello, conviene transformar dicha variable y volver a obtener el gráfico de dispersión entre la severidad y la edad al cuadrado. De esta forma, se podría realizar una comparación y estudiar qué relaciones son más dinámicas. Efectivamente, en el gráfico se observa que la relación entre severidad y edad al cuadrado es mucho más fuerte.

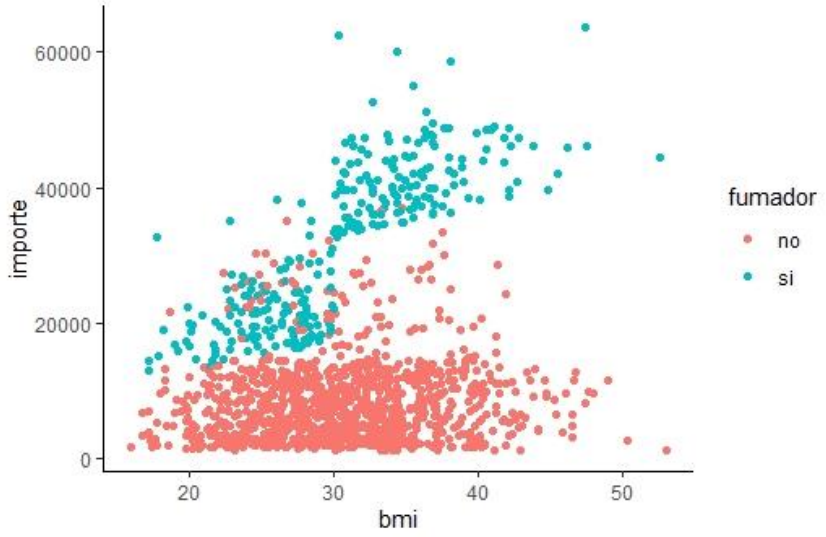
Ilustración 13. Primera transformación: edad al cuadrado



Fuente: elaboración propia

Por otro lado, cuando se interpretaba el gráfico de dispersión entre el Índice de Masa Corporal y la severidad, se intuía una ligera relación positiva, aunque no se podían realizar comentarios consistentes. Ahora bien, si se interacciona dicha variable con otra, como por ejemplo los hábitos de tabaco, puede que se obtengan conclusiones más sólidas. Efectivamente, en el gráfico se observa cómo la severidad presenta una evidente relación positiva con la interacción entre las variables Índice de Masa Corporal y fumador.

Ilustración 14. Segunda transformación: interacción entre BMI y fumador

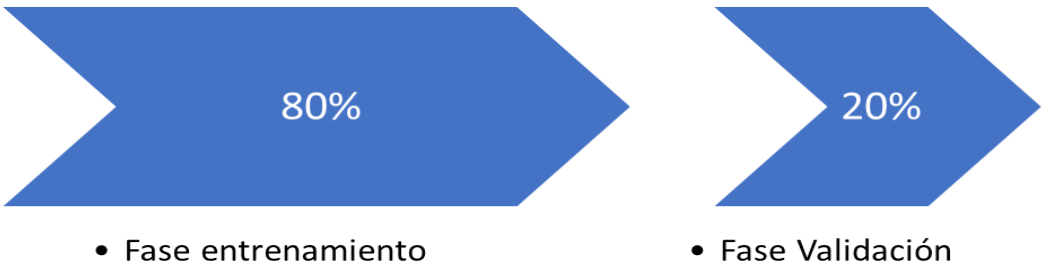


Fuente: elaboración propia

En los próximos capítulos se utilizarán todas estas relaciones para plantear distintos modelos de predicción. Se comenzará explicando los modelos más sencillos, como la regresión lineal, y se terminará abordando algoritmos más complejos. Sobre todos ellos, se debe señalar que la idea de un modelo estadístico de predicción consiste en poder entrenar y construir un modelo con datos conocidos y utilizarlo posteriormente para nuevos datos desconocidos. Por ello, se ha dividido la base de datos en dos conjuntos para entrenar, primero, el modelo y probar su validación, después.

- Entrenamiento: el 80% de los datos (1.070 registros) se utilizan para construir (o "entrenar") cada uno de los modelos.
- Validación: el 20% de los datos restantes (268 registros) se utilizan para validar y evaluar cada uno de los modelos. La eficacia del modelo se obtiene en función de los resultados obtenidos en esta fase.

Ilustración 15. Metodología modelos aprendizaje supervisado



Fuente: elaboración propia

4. MODELO DE REGRESIÓN LINEAL

Se distinguen dos tipos de modelos de regresión lineal: simple y múltiple.

4.1. Modelo de regresión lineal simple

4.1.1. Marco teórico

La **regresión lineal simple** es un método estadístico que consiste en generar la ecuación de una recta que permita explicar la relación lineal existente entre dos variables, una dependiente y otra independiente[10]. La forma analítica de la ecuación generada es la siguiente:

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

Siendo β_0 la ordenada en el origen, β_1 la pendiente, Y la variable dependiente, X_1 la variable independiente y ε el término de error de la predicción.

Los valores de β_0 y β_1 son desconocidos y deben ser estimados a partir de una muestra de datos. Normalmente se estiman bajo el *método de mínimos cuadrados ordinarios*, que minimiza la suma de cuadrados residuales, dando lugar a la recta más próxima a todos los puntos.

Una regresión lineal simple debe cumplir una serie de condiciones:

- **Linealidad**. La relación entre las variables debe ser lineal.
- **Distribución normal de los residuos**. Los residuos se deben distribuir de manera normal, con media igual a 0.
- **Varianza de los residuos constante (homocedasticidad**). La varianza de los residuos debe ser constante a lo largo del rango de observaciones.
- **Independencia y autocorrelación**. Las observaciones deben ser independientes entre sí.

También se deben definir las medidas de desempeño más importantes de una salida de regresión.

- El **Residual Estándar Error** consiste en la diferencia promedio que existe entre la línea de regresión y las observaciones. Cuanto menor sea este valor, mayor desempeño se asociará a este modelo. Esta medida puede calcularse bajo la siguiente expresión:

[10] LAGUNA C.: "Correlación y regresión lineal". Instituto Aragonés de Ciencias de la Salud. Disponible en:
http://www.ics-aragon.com/cursos/salud-publica/2014/pdf/M2T04.pdf

$$Residual\ Standar\ Error = \sqrt{\frac{1}{n-p-1}RSS} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n-p-1}}$$

Siendo p el número de predictores del modelo, n el número de observaciones, y_i las observaciones y $\hat{y}_i$ las predicciones.

- El **coeficiente de determinación (R^2)** indica la proporción de la variabilidad de la variable dependiente explicada por el modelo. Su valor está comprendido entre 0 y 1. Serán preferibles los valores próximos a 1. Esta medida puede obtenerse a partir de la siguiente expresión:

$$R^2 = 1 - \frac{\sum(\hat{y}_i - y_i)^2}{\sum(y_i - \hat{y}_i)^2}$$

Siendo y_i las observaciones y $\hat{y}_i$ las predicciones.

4.1.2. Aplicación del ejemplo

Para el ejemplo se presenta el siguiente modelo de regresión lineal simple:

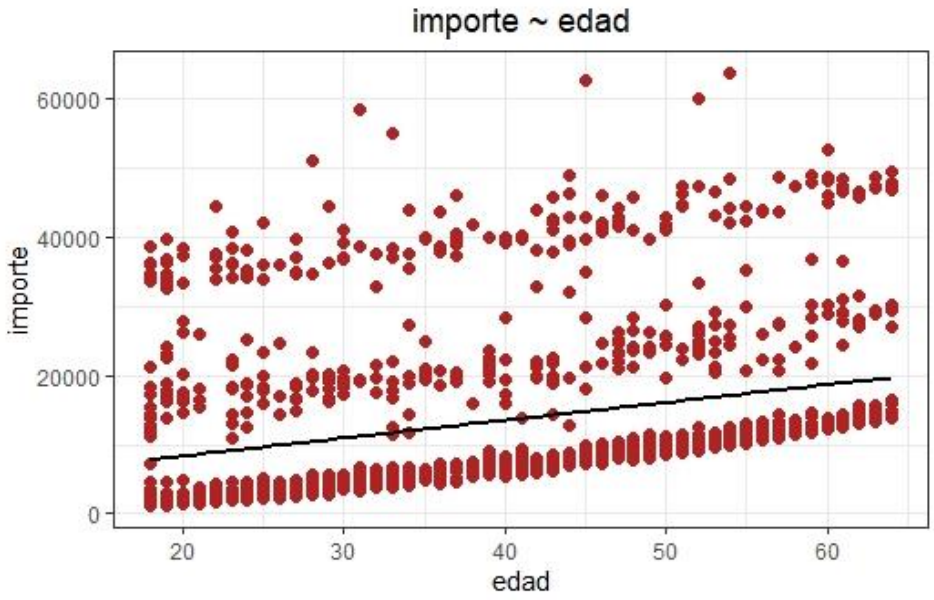
$$importe_i = \beta_0 + \beta_1 edad_i + \varepsilon_i$$

Para el cual se obtienen los siguientes coeficientes:

	Estimación	Error estándar	Valor t	Prob(>|t|)
(Intercepto)	2.984,11	1.029,39	2,899	0,00382
edad	260,14	24,62	10,565	2E-16

Al situarse el p-valor asociado al coeficiente de la edad cerca de 0, se concluye que se trata de una variable significativa. En el siguiente gráfico se muestra la recta del modelo. Se puede intuir que la recta no recoge con precisión el comportamiento de la nube de puntos.

Ilustración 16. Recta de regresión entre edad e importe



Fuente: elaboración propia

Efectivamente, al calcular las medidas de desempeño, se obtiene que el modelo no es adecuado por los siguientes motivos. Por un lado, el RSE es muy elevado, lo cual indica que la predicción presenta una gran desviación. Por el otro, el R2 muestra que sólo una pequeña proporción de la variabilidad de la severidad (el 9,4%) es explicada por el modelo.

Residual Standard Error	11.350 on 1.068 degrees of freedom
Multiple R-squared	0,09462
Adjusted R-squared	0,09377
F-stadistic	111,6 on 1 and 1.068 DF
p-value	2,2E-16

4.2. Modelo de regresión lineal múltiple

4.2.1. Marco teórico

La **regresión lineal múltiple** es un método estadístico que consiste en generar la ecuación que permita explicar la relación lineal existente entre una variable dependiente y un conjunto de variables independientes[11]. Este modelo puede formularse de la siguiente manera:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_n X_n + \varepsilon$$

En este caso, las β_i se conocen como coeficientes parciales de regresión.

Una regresión lineal múltiple debe cumplir una serie de condiciones:

- **Ausencia de multicolinealidad**. Los predictores deben ser independientes, no deben presentar relación entre sí mismos. Es importante que no exista multicolinealidad, debido a que, si existiera, no se podría identificar el efecto individual de cada predictor sobre la variable dependiente.
- **Relación lineal entre los predictores numéricos y la variable dependiente**. Cada predictor debe estar relacionado linealmente con la variable dependiente cuando el resto de predictores se mantienen constantes.
- **Distribución normal de los residuos**. Los residuos deben distribuirse bajo una normal con media cero.
- **Variabilidad constante de los residuos (homocedasticidad)**. La varianza de los residuos debe ser constante a lo largo del rango de observaciones.
- **No autocorrelación (independencia)**. Las observaciones deben ser independientes entre sí.

Para comprobar estas condiciones, puede resultar de utilidad el análisis gráfico de los residuos, que normalmente se compone de 4 gráficos:

- En primer lugar, el gráfico "Residuals vs Fitted" sirve para analizar la homocedasticidad, identificar los datos atípicos e intuir la falta de términos de mayor orden.
- En segundo lugar, el gráfico "Normal Q-Q" ayuda a identificar la normalidad de los residuos.
- Por otro lado, el gráfico "Scale-Location" sirve para estudiar la homocedasticidad.
- Finalmente, el gráfico "Residuals vs Leverage" permite señalar datos atípicos influyentes. No todos los datos atípicos alteran los resultados obtenidos en el análisis de regresión, sólo los que superan la distancia de Cook.

[11] PÉRTEGA DÍAZ S. y PITA FERNÁNDEZ S.: "Técnicas de Regresión: Regresión lineal múltiple" Unidad de Epidemiología Clínica y Bioestadística, 2000. Disponible en: https://www.fisterra.com/gestor/upload/guias/regre_lineal_multi2.pdf

4.2.2. Resultados

Se han presentado dos modelos, a la vista de los resultados del análisis realizado en el capítulo anterior. Mientras que el primer modelo contempla todas las variables originales (sin transformar), el segundo elimina las variables menos relevantes e incluye las dos transformaciones comentadas. A continuación, se exponen las principales conclusiones para cada uno de los modelos.

Modelo 1

$$importe = \beta_0 + \beta_1 edad + \beta_2 sexo + \beta_3 bmi + \beta_4 hijos + \beta_5 fumador + \beta_6 regionnoroeste + \beta_6 regionsureste + \beta_6 regionsuroeste + \beta_6 bmi30 + \varepsilon$$

Conviene señalar que las variables texto (sexo, fumador, region y bmi30) se han modelizado como variables dicotómicas, es decir, toman únicamente dos valores: 0 o 1.

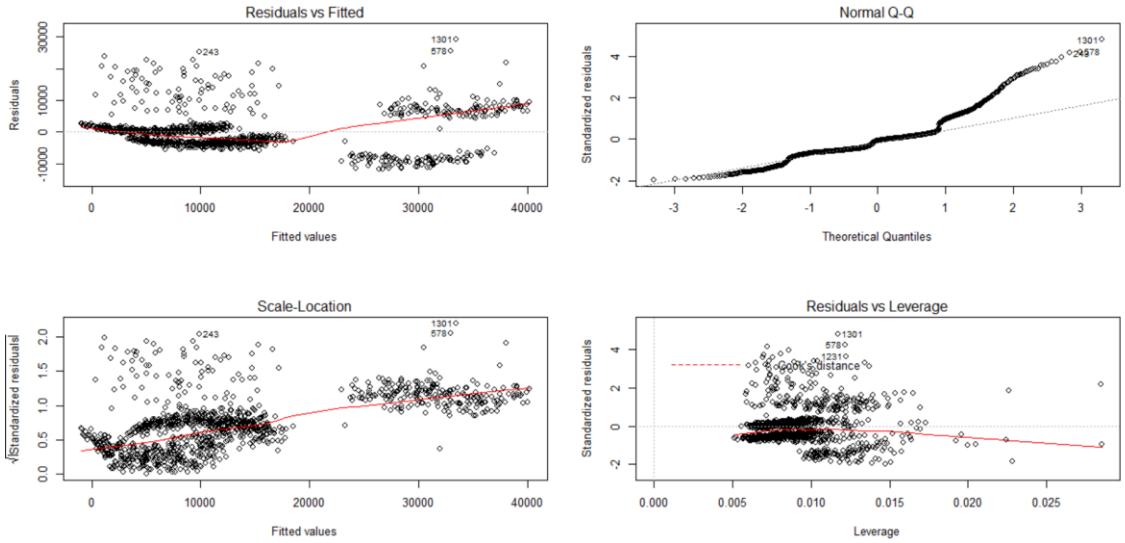
Para este modelo se han obtenido los siguientes coeficientes parciales de la regresión. Se puede observar que las variables sexo y región no superan las pruebas individuales de significación y podría plantearse su exclusión del modelo.

	Estimación	Error estándar	Valor t	Prob(>|t|)
(Intercepto)	-7.205,31	1.434,22	-5,024	5,94E-07
edad	250,88	13,33	18,815	2E-16
sexo	97,58	374,66	0,260	0,79458
bmi	151,54	51,84	2,923	0,00354
hijos	495,98	153,95	3,222	0,00131
fumador	23.359,12	466,73	50,048	2E-16
regionnoroeste	-504,34	536,01	-0,941	0,34696
regionsureste	-1.255,97	535,37	-2,346	0,01916
regionsuroeste	-1.333,72	533,07	-2,502	0,01250
bmi30	2.654,27	624,88	4,248	2,35E-05

A continuación, se muestra el análisis gráfico de los residuos del modelo, del cual pueden extraerse las siguientes conclusiones:

- Gráfico “Residuals vs Fitted”. En primer lugar, al distinguirse varios grupos de puntos se puede intuir heterocedasticidad. Por otro lado, los puntos muy alejados de 0 informan de la existencia de “outliers” o datos atípicos. Finalmente, la forma curvilínea de la recta advierte de la falta en el modelo de algún término de mayor orden (por ejemplo, cuadrático).
- Gráfico “Normal Q-Q”. Al no formarse una línea recta, los datos no son normales.
- El gráfico “Scale-Location”. En este caso, al encontrar una curva no horizontal se puede concluir que los datos son heterocedásticos.
- El gráfico “Residuals vs Leverage”. En este caso, se observa un número importante de “outliers” o datos atípicos influyentes.

Ilustración 17. Análisis gráfico de los residuos del modelo



Fuente: elaboración propia

Finalmente, se presentan las medidas de desempeño. Se observa un gran avance con respecto a los modelos de regresión lineal simple, tanto por la disminución del RSE como por el aumento del R2.

Residual Standard Error	6.078 on 1.060 degrees of freedom
Multiple R-squared	0,7422
Adjusted R-squared	0,74
F-stadistic	339,1 on 9 and 1.060 DF
p-value	2,2E-16

Modelo 2

$$importe = \beta_0 + \beta_1 hijos + \beta_2 edad^2 + \beta_3 bmi + \beta_4 fumador + \beta_5 bmi30 + \beta_6 fumador * bmi + \varepsilon$$

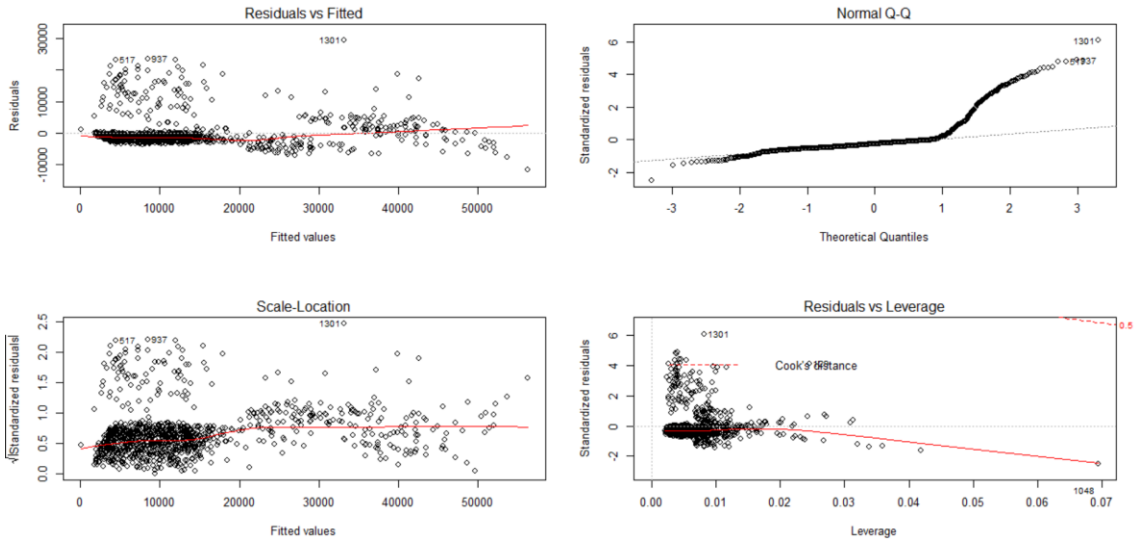
Para este modelo se han obtenido los siguientes coeficientes parciales de la regresión.

	Estimación	Error estándar	Valor t	Prob(>|t|)
(Intercepto)	6,314E+03	1,130E+03	5,585	2,96E-08
hijos	7,001E+02	1,228E+02	5,702	1,53 E-08
edad^2	3,309	1,323E-01	25,018	2 E-16
bmi	-1,897E+02	4,228E+01	-4,487	8.01E -06
fumador	-2,020E+04	1,819E+03	-11,105	2 E-16
bmi30	2,803E+03	4,976E+02	5,632	2,28 E-08
fumador*bmi	1,426E+03	5,832E+01	24,449	2 E-16

También se ha realizado el análisis gráfico de los residuos, que arroja las siguientes conclusiones:

- El gráfico "Residuals vs Fitted" muestra que, aunque siguen existiendo datos atípicos, se ha mejorado el problema de la heterocedasticidad y se ha incluido en el modelo el término de mayor orden (la edad).
- El gráfico "Normal Q-Q" indica que, al no formarse una línea recta, los datos siguen sin ser normales.
- El gráfico "Scale-Location" muestra que la curva ha pasado a ser más horizontal, aunque todavía no se logra para todas las observaciones, por lo que sigue existiendo cierta heterocedasticidad.
- El gráfico "Residuals vs Leverage" indica que se ha reducido el número de datos atípicos influyentes.

Ilustración 18. Análisis gráfico de los residuos del modelo



Fuente: elaboración propia

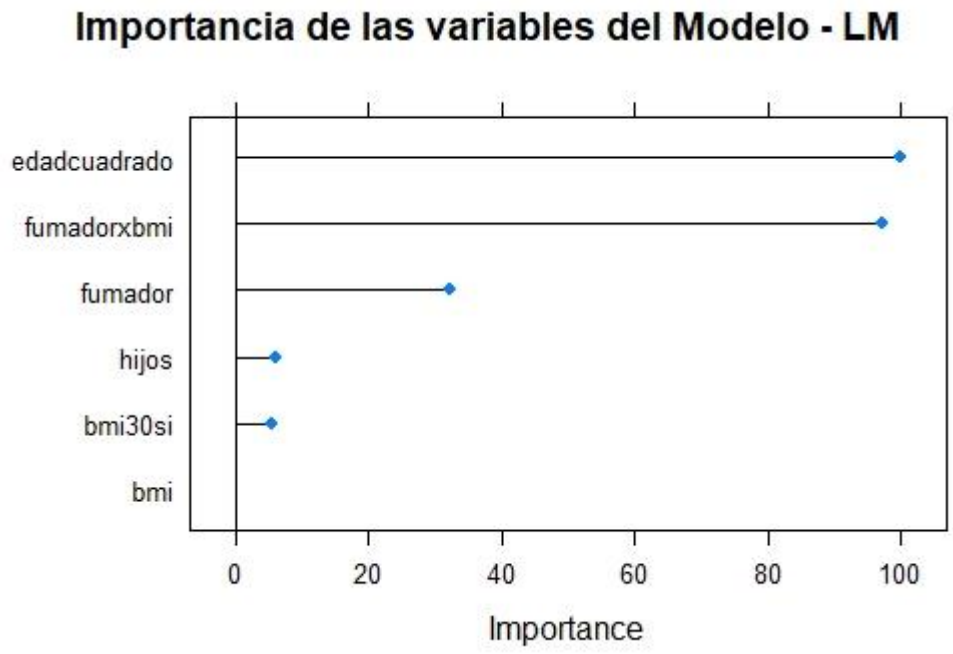
Seguidamente, se anotan las medidas de desempeño:

Residual Standard Error	4.859 on 1.063 degrees of freedom
Multiple R-squared	0,8348
Adjusted R-squared	0,8338
F-stadistic	895,1 on 6 and 1.063 DF
p-value	2,2E-16

Tras el análisis gráfico de los residuos y la comparación de las medidas de desempeño, se puede concluir que el modelo 2 es más preciso y eficiente en la predicción de la severidad.

Por otro lado, se puede extraer la importancia relativa de las variables utilizadas en el modelo. La siguiente ilustración indica que las variables más influyentes son las dos transformaciones, así como la variable “fumador”.

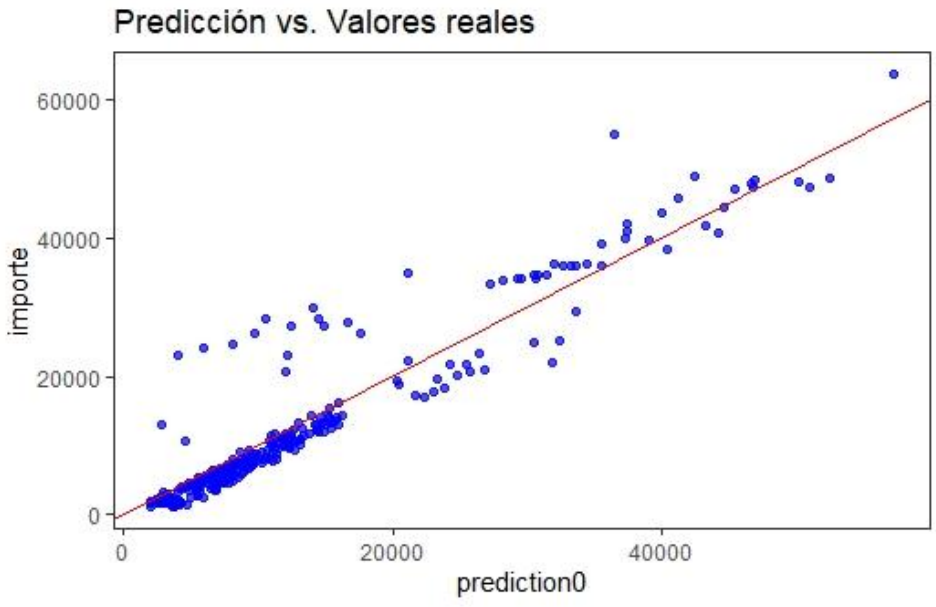
Ilustración 19. Importancia de las variables en el modelo de regresión lineal



Fuente: elaboración propia

Una vez utilizado el modelo 2 para el conjunto de datos de la etapa de validación, podría ser interesante visualizar gráficamente el rendimiento del modelo. Para ello se ha dibujado un gráfico que muestra en el eje de las X el valor calculado por el modelo y en el eje de las Y el valor real. Si la predicción hubiera sido exacta, los valores situarían sobre la línea roja. Cuanto más próximos a la línea estén los datos, mejor será el modelo. El modelo de regresión lineal parece ser más preciso para aquellos importes más bajos de siniestros.

Ilustración 20. Gráfico predicción vs. valores reales



Fuente: elaboración propia

Además, para la comparación entre todos los modelos estudiados se presentan dos medidas estadísticas de desempeño fundamentales:

- Por un lado, el **Relative Squared Error (RSE)** mide el error de la predicción en relación con el que se hubiera obtenido si se hubiera utilizado un predictor más sencillo (por ejemplo, si se hubiera tomado como predictor la media aritmética de los valores). Normalmente este valor es inferior a la unidad, ya que se están analizando siempre modelos predictivos más avanzados que la media aritmética. Serán preferibles valores inferiores, que indicarán menor error en las predicciones. El RSE se calcula de la siguiente manera:

$$RSE_i = \frac{\sum_{j=1}^{n}(P_{i,j} - T_j)^2}{\sum_{j=1}^{n}(T_j - \bar{T})^2}$$

  Donde $P_{i,j}$ es la predicción del modelo i para el registro j (de un total de n registros), T_j es el valor observado del registro j, $\bar{T}$ es la media aritmética de los n registros.

- Por el otro, el **Root Mean Squared Error (RMSE)** mide la distancia entre las predicciones y las observaciones. Para su cálculo se utiliza la siguiente fórmula:

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

  Donde $\hat{y}_i$ son las predicciones, y_i los valores observados y n el número de observaciones.

Para el modelo seleccionado de regresión lineal múltiple se han obtenido las siguientes medidas de desempeño:

	Regresión Lineal
RSE	0,112376460851679
RMSE	4.303,78574065245

Al ser el primer modelo estudiado, estos valores servirán como referencia para comparar el desempeño del resto de modelos.

5. ÁRBOLES DE DECISIÓN Y REGRESIÓN CUBIST

5.1. Marco teórico

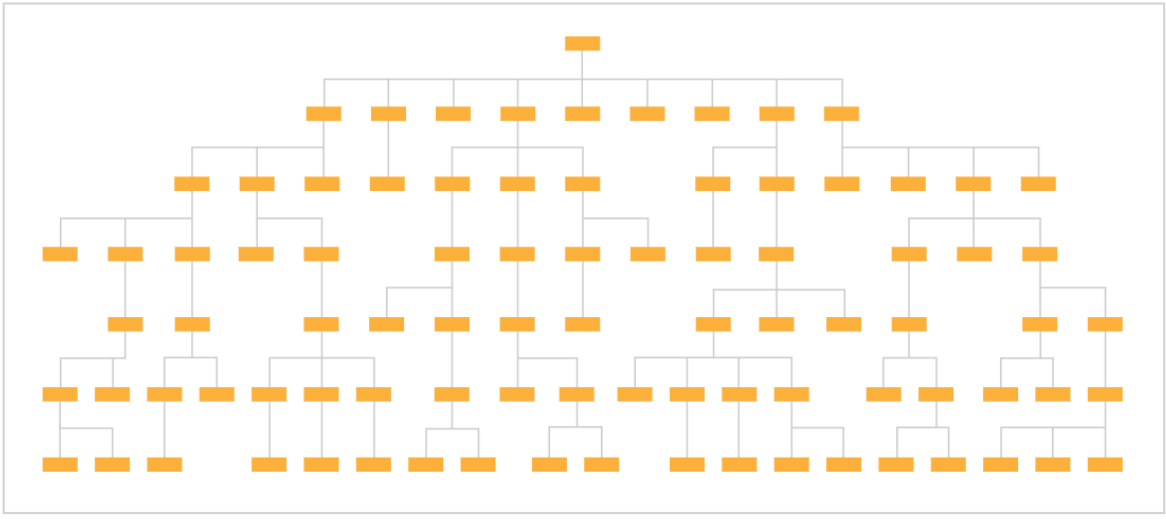
La regresión Cubist es una técnica de regresión basada en ciertas condiciones que fue desarrollada con base en las ideas de Quinlan. Quinlan es un investigador de minería de datos que ha contribuido enormemente en el campo de los algoritmos de árboles de decisión.

5.1.1. Introducción a los árboles de decisión

Los modelos predictivos como la regresión lineal o polinómica se basan explicar todo el espacio muestral a partir de una única ecuación. Sin embargo, cuando existen múltiples predictores y estos interaccionan entre ellos de forma compleja y no lineal, los modelos pierden capacidad predictiva y se plantea necesario buscar otros métodos de ajuste no lineal. El problema de estos es que suelen ser difíciles de interpretar. Los modelos basados en árboles consiguen segmentar el espacio de los predictores en regiones simples, lo cual hace más sencilla su interpretación.

El árbol de decisión es una de las técnicas más utilizadas en Machine Learning y consiste en aportar una solución gráfica que incluya todas las posibles respuestas a un problema. Un árbol de decisión se compone de un primer nodo, llamado raíz, a partir del cual nacen varias ramas. Cada una de las ramas se bifurcará en nuevas subramas, y así sucesivamente hasta los nodos finales, que contienen la solución al problema. El funcionamiento de un árbol de decisión se puede observar en la siguiente ilustración.

Ilustración 21. Ejemplo de árbol de decisión



Fuente: https://datavizcatalogue.com/

Un algoritmo de árbol de decisión detecta qué combinación es preferible para tomar las decisiones. Esto se puede comprender mejor a través de un ejemplo. Supóngase que para predecir el costo del siniestro se quiere utilizar un árbol de decisión utilizando las variables "sexo" y "fumador". Podría crearse un árbol en el que primero se dividiera por "sexo" y luego se dividiera por "fumador". O podría ser al revés: primero por "fumador" y luego por "sexo". El algoritmo será el que decida la combinación óptima para tomar las decisiones.

Los árboles de decisión presentan una serie de ventajas y desventajas, que se recogen en el siguiente cuadro[12]:

Ventajas	Desventajas
Son fáciles de interpretar, de hecho, no requieren conocimientos avanzados de estadística para su comprensión.	Los algoritmos avanzados basados en múltiples árboles de decisión no son fáciles de interpretar ni se pueden representar gráficamente.
Los modelos que utilizan un solo árbol se pueden representar gráficamente.	Los árboles basados en un único árbol suelen presentar una capacidad más baja que otros modelos.
Admiten tanto predictores cuantitativos, como cualitativos.	En el tratamiento de variables continuas, al categorizarlas, puede ocasionar una fuga de información.
Generalmente no están alterados por "outliers", por lo que no requieren de tanta limpieza o procesado de datos como otros modelos.	

Un árbol de decisión se construye siguiendo dos pasos. Primero, se realiza una división del espacio de los predictores en regiones que no se solapen, construyendo los nodos terminales: $R_1, R_2, R_3, \dots, R_j$. Después, se predice la variable respuesta en cada una de las regiones anteriores.

Aunque a simple vista puede parecer una tarea sencilla, la complejidad radica en la metodología seguida para dividir el espacio en regiones. Esta metodología es distinta en función del objetivo del árbol de decisión. Los árboles de decisión pueden ser de clasificación o regresión:

- Los árboles de clasificación son aquellos cuya variable de salida es una clase discreta.
- Los árboles de regresión son aquellos cuya variable de salida es continua. Es el caso del ejemplo estudiado en este trabajo, por ello se describe únicamente el proceso de división de esta clase de árboles.

[12] CARDONA HERNANDEZ P. A.: "Aplicación de árboles de decisión en modelos de riesgo crediticio" Revista Colombiana de Estadística, Volumen 27 Nº2, 2004.

El objetivo de la división es encontrar el número "J" de regiones que minimizan el Residual Sum of Squares (RSS), una medida del error de la predicción, definido como:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2$$

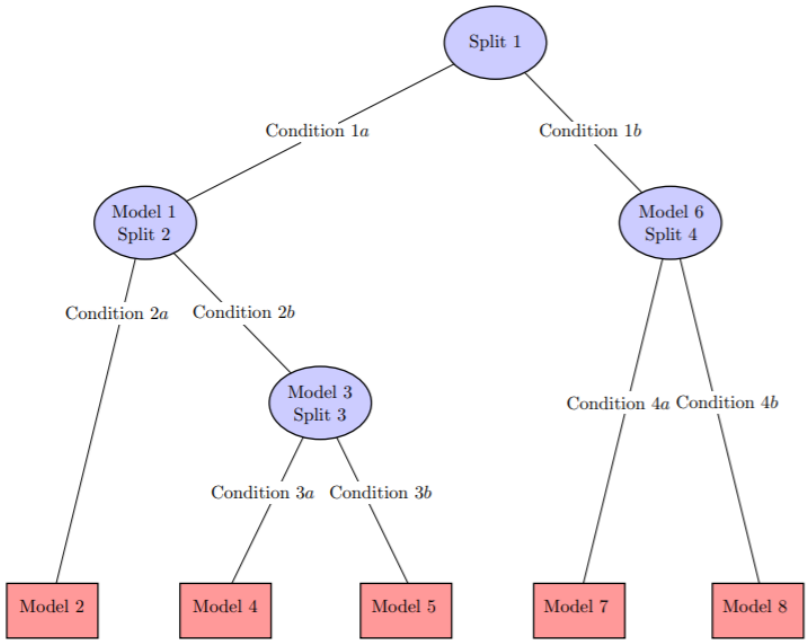
Siendo $\hat{y}_{R_j}$ la media de la variable salida en la región R_j

Al no ser posible estudiar todas las posibles divisiones del espacio de predictores, se utiliza la técnica *recursive binary splitting* (o *división binaria recursiva*), cuyo funcionamiento se resume en 5 pasos:

1. Se comienza en lo más alto del árbol, donde no existen particiones y todas las observaciones se encuentran en la misma región.
2. Se anotan todos los posibles puntos de corte para cada predictor. Para los predictores cualitativos, los puntos de corte son cada uno de los niveles. Para los predictores cuantitativos, en cada par de valores se toma el valor medio y este es un punto de corte.
3. Se calcula el Residual Sum of Squares (RSS) para cada una de las divisiones realizadas en el punto anterior. A continuación, se agrega para obtener el resultado total para cada uno de los predictores.
4. Se escoge el predictor que genera un RSS total menor.
5. Se repiten los pasos 1-4 hasta que sólo quede un predictor.

Una vez creado el árbol, se utilizarán diferentes modelos de regresión lineal en cada uno de los nodos finales. Estas regresiones se basarán en los predictores seguidos en los nodos intermedios. Este tipo de árboles se conocen como "grown". También pueden utilizarse modelos de regresión lineal en los nodos intermedios.

Ilustración 22. Esquema de árbol de decisión "grown"



Fuente: "Rules Rules Rules! Cubist Regression Models" KUHN M.

Conviene explicar el **proceso de "pruning" o "podado"**. Al construir árboles de gran tamaño, se tiende a ajustar muy bien a las observaciones empleadas en la fase de entrenamiento. Esto puede provocar que la capacidad explicativa/predictiva del modelo se reduzca al utilizarse con nuevos datos, lo que se conoce como "overfitting". Una de las estrategias para abordar este problema es el proceso de "pruning" o "podado", que consiste en eliminar las partes del árbol que presenten menor robustez. Para ello se puede utilizar la técnica "cost complexity pruning" para encontrar el número óptimo de nodos finales del árbol. Esta técnica penaliza aquellos modelos con mayor número de nodos finales. Se busca minimizar la siguiente ecuación:

$$\sum_{j=1}^{/T/} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2 + \alpha /T/$$

Siendo $/T/$ el número de nodos terminales del árbol.

El primer término de la ecuación corresponde al RSS que, como se ha explicado, es una medida del error de una predicción. Por lo tanto, se trata de añadir al RSS la restricción comentada anteriormente, que penaliza el mayor número de nodos finales[13].

[13] KUHN M., WESTON S., KEEFER C y COULTER N.: "Cubist Models for Regression". 11 de mayo de 2012. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.398.3360&rep=rep1&type=pdf

5.1.2. Algoritmo Cubist

El algoritmo Cubist es una de las técnicas basadas en árboles de decisión menos documentadas, en comparación con otras como "gradient boosting" o "random forest". Sin embargo, se ha convertido en los últimos años en un método muy popular gracias a su introducción en R por *Kuhn y otros* en 2013[14] a través del paquete "Cubist". Además, ha mostrado increíbles resultados en recientes investigaciones, como la de *Meyer y otros*[15] y *Zhang y otros*[16].

El algoritmo Cubist presenta varios rasgos que lo diferencian con respecto a otros árboles:

- La estrategia de "pruning" o "podado" es diferente. La ecuación para minimizar presenta algunos cambios con respecto a la planteada bajo el método "cost complexity pruning".[17]
- El modelo Cubist puede usar un esquema del tipo "Boosting" llamado "commitees". A diferencia del "Boosting" tradicional, los pesos para cada committee no se usan para ponderar las predicciones de cada árbol, sino que la predicción final es simplemente la media aritmética de todas las predicciones. El mecanismo de "Boosting" se aborda con más detalle en los próximos capítulos.

Para este ejemplo, no se ha utilizado la estrategia "commitees", ya que se trata de estudiar el árbol de decisión más simple posible, antes de pasar a analizar técnicas árboles de decisión más complejas.

5.2. Resultados

5.2.1. Etapa de entrenamiento

Aplicando el modelo de regresión Cubist al conjunto de datos de entrenamiento se ha obtenido el siguiente árbol de decisión:

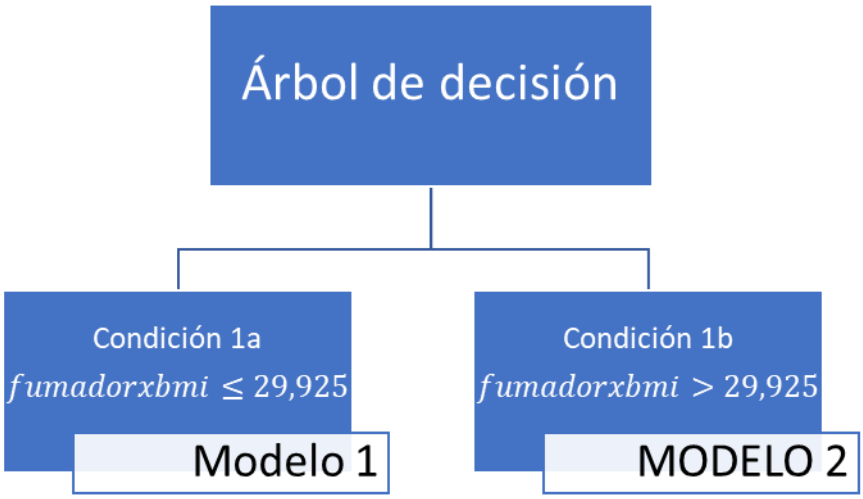
[14] KUHN M., WESTON S., KEEFER C y COULTER N.: "Cubist Models for Regression". 11 de mayo de 2012. Disponible en:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.398.3360&rep=rep1&type=pdf

[15] MEYER H., KNUDBY A., APPELHANS T., MÜLLER M.U. y otros: "Mapping daily air temperatura for Antarctica base don MODIS LST." Remote Sens, 2016.

[16] ZHANG W., HUANG Y., YU Y.Q. y SUN W.J: "Empirical models for estimating daily máximum, mínimum and mean air temperaturas with MODIS land surface temperaturas." Int. J. Remote Sens, 2011.

[17] Para más detalles consultar:
KUHN M.: "Rules Rules Rules! Cubist Regression Models" Boston R User Group. Disponible en:
https://static1.squarespace.com/static/51156277e4b0b8b2ffe11c00/t/56e3056a3c44d8779a61988a/1457718645593/cubist_BRUG.pdf

Ilustración 23. Árbol de decisión generado por la regresión Cubist



Fuente: elaboración propia

Se trata de un árbol de decisión que incluye sólo dos condiciones: la *condición 1a* es cumplida por 961 casos, mientras que la *condición 1b* por los 109 casos restantes.

Cada una de las condiciones lleva asociado un modelo de regresión lineal.

MODELO 1:

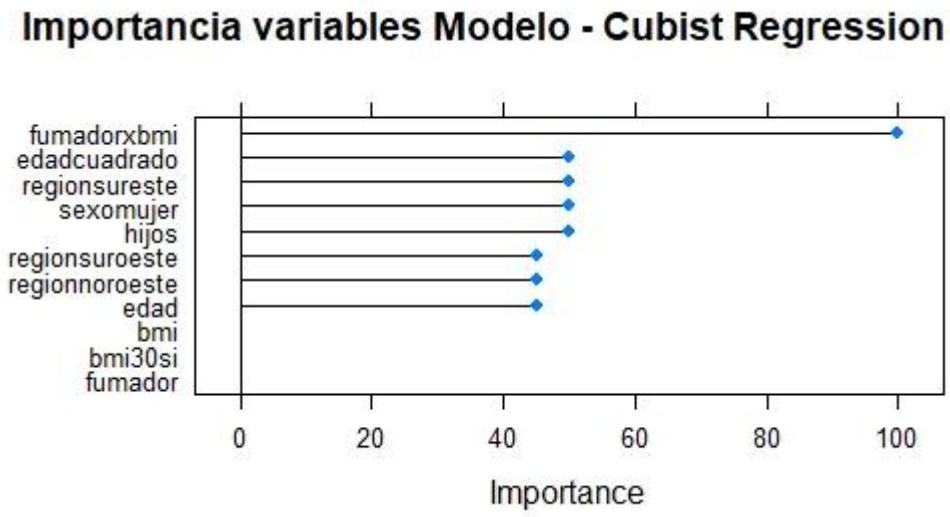
$$importe_i = 1.388{,}131 + 535\, fumadorxbmi_i + 3{,}7\, edadcuadrado_i + 682\, hijos_i - 32\, edad_i - 937\, regionsuroeste_i + 491\, sexomujer_i - 452\, regionsureste_i - 456\, regionnoroeste_i$$

MODELO 2:

$$importe = 17.749{,}696 + 469\, fumadorxbmi_i + 3{,}4\, edadcuadrado_i + 549\, hijos_i - 582\, regionsureste_i + 504\, sexomujer_i$$

También se puede extraer la importancia relativa de las variables utilizadas en el modelo. En este caso la variable más influyente es la transformación "fumadorxbmi", aunque también se deben señalar la importancia de otras variables: "edadcuadrado", "regionsureste", "sexomujer", "hijos", "regionsuroeste", "regionnoroeste" y "edad". A diferencia del modelo de regresión lineal, el sexo y la región de procedencia sí parecen ser importantes en el modelo.

Ilustración 24. Importancia de las variables del modelo - Regresión Cubist

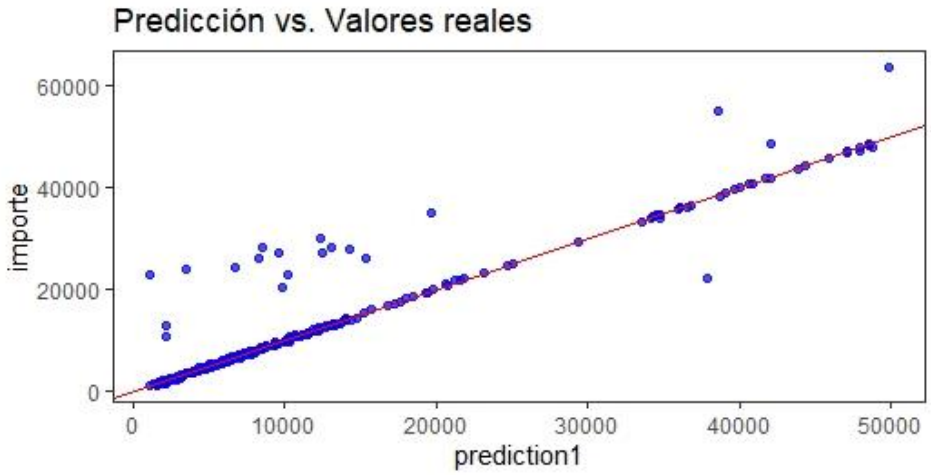


Fuente: elaboración propia

5.2.2. Etapa de validación: análisis gráfico y medidas de desempeño

Una vez utilizado el modelo para el conjunto de datos de la etapa de validación, se ha dibujado el mismo gráfico que para el modelo de regresión lineal múltiple. Se puede observar una evidente mejora del rendimiento, debido a que hay un mayor número de datos situados sobre la línea roja.

Ilustración 25. Gráfico predicción vs. valores reales



Fuente: elaboración propia

Para la comparación de los modelos se presentan las dos medidas de desempeño de este modelo:

	Regresión Lineal	Regresión Cubist
RSE	0,112376460851679	0,108597029070604
RMSE	4.303,78574065245	4.230,79457908986

Al observarse una mejora de ambas medidas, se puede concluir que este modelo tiene una mayor capacidad predictiva que el modelo de regresión lineal.

6. ESTRATEGIA DE BAGGING: RANDOM FOREST

6.1. Marco teórico

Cualquier modelo predictivo se enfrenta al problema del equilibrio "bias-varianza". Para explicar en qué consiste este equilibrio, se debe definir sus dos componentes:

- El error de "bias" se calcula como la diferencia media entre las predicciones de un modelo y los valores observados. Hace referencia entonces al error del modelo para los datos de entrenamiento.
- El error de "varianza" se refiere a los cambios producidos en el modelo al modificar los datos de entrenamiento. Se refiere, por tanto, al error del modelo al utilizarse nuevos datos distintos a los de entrenamiento.

La precisión de los árboles de decisión, descritos en el capítulo anterior, se mide bajo este criterio. Aquellos árboles que tienen pocas ramificaciones suelen presentar alto error de "bias" y bajo de "varianza". Aquellos con muchas ramificaciones, por el contrario, presentan bajo error de "bias" y alto de "varianza". Para lograr el equilibrio entre ambos errores surgen las estrategias de ensemble, entre las cuales las más utilizadas son:

- **Bagging**. Se ajustan múltiples árboles a la vez formando un "bosque". Este método se aborda con más detalle en este capítulo.
- **Boosting**. Se ajustan secuencialmente modelos de tal forma cada uno va aprendiendo sobre los errores del anterior. Este método se aborda con más detalle en el siguiente capítulo.

Aunque ambos métodos comparten el mismo fin, lograr un equilibrio bias-varianza, la estrategia seguida en cada uno de ellos es muy distinta. En el bagging se utilizan muchos árboles grandes, con poco "bias" pero mucha "varianza", y se trata de mantener el mismo nivel de "bias" reduciendo el nivel de varianza a través de la agregación. Por su parte, en el boosting se utilizan muchos árboles pequeños, con mucho "bias" y poca "varianza", y se trata de reducir secuencialmente el "bias". Además, también difieren en la relación entre los

modelos utilizados: mientras que en el bagging cada modelo es distinto de los demás; en el boosting cada modelo es muy parecido al anterior.

6.1.1. Estrategia de bagging

La palabra "bagging" viene de la expresión **"Bootstrap aggregation"** y consiste en aplicar el muestreo repetido con el objetivo de minimizar el error de "varianza" de algunos modelos de Machine Learning, como los árboles de predicción.

El objetivo es que, considerando un conjunto de variables independientes $Y_1, \dots, Y_n$, y teniendo cada una misma varianza σ^2, la varianza de la media de las observaciones $\bar{Y}$ será $\frac{\sigma^2}{n}$ y, por lo tanto, se consigue reducir la varianza y mejorar la precisión del modelo predictivo.

La estrategia de "bagging" consiste en obtener varias muestras de una población y ajustar un modelo distinto para cada una de ellas. La predicción final se calculará como la media aritmética de las predicciones anteriores. La dificultad se encuentra en la consecución de múltiples muestras, para lo cual se recurre al bootstrapping, que permite generar pseudo-muestras.

La forma de implementar "bagging" en un árbol de decisión es la siguiente:

1. Se utiliza la técnica bootstrapping para generar pseudo-muestras a partir de una muestra de entrenamiento original.
2. Se entrena un árbol con cada una de las pseudo-muestras. Se obtiene tantos árboles como pseudo-muestras se hayan generado.
3. Para las nuevas observaciones, se recoge cada una de las predicciones de los árboles anteriores. La predicción final se calculará como la media aritmética de todas ellas.

Cuando se está generando muestras por bootstrapping, estas sólo utilizan aproximadamente dos tercios de las observaciones originales. A la parte restante se le conoce como "out-of-bag". Para una observación incluida en una de las divisiones de la muestra, se podría predecir la respuesta utilizando los árboles que se hubieran entrenado sin esta observación, promediando finalmente las predicciones obtenidas. Si esta acción se repitiera, se podría calcular el "out-of-bag" error, que sirve como estimación del error de la predicción y simplifica la comparación entre modelos.

Aunque el proceso de "bagging" consigue mejorar la capacidad predictiva de los modelos, la interpretación de estos se vuelve más compleja. La representación gráfica pasa a ser menos intuitiva e identificar la importancia de los predictores ya no es un proceso tan directo. No obstante, al incrementar el número de árboles, también se abre la puerta a nuevas herramientas para medir la importancia de los predictores:

- Se podría utilizar el out-of-bag error de cada uno de los predictores para medir la influencia de cada uno de los predictores sobre el error total del modelo.
- Al contar con tantos árboles, se podría sumar el número de divisiones en las que participa el predictor. No obstante, habría que tener cuidado sobre su interpretación,

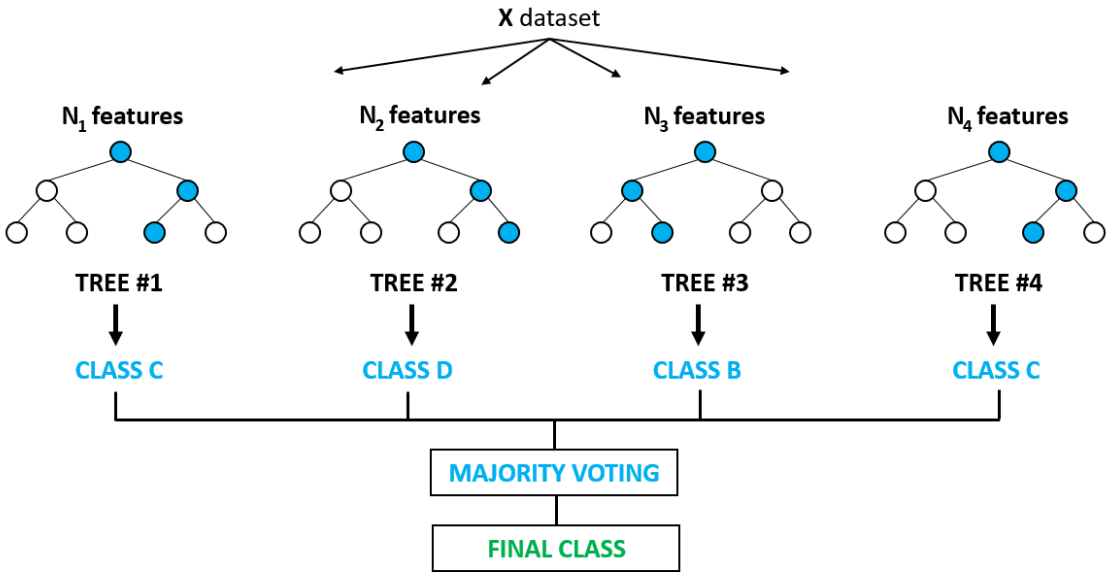
ya que se estaría cuantificando la influencia de los predictores sobre el modelo, no sobre la variable respuesta.

6.1.2. Random Forest

El algoritmo Random Forest se basa en la estrategia de "bagging", pero introduce algunas modificaciones sobre esta. El objetivo es mejorar los resultados obtenidos en el proceso de "bagging" decorrelacionando los árboles obtenidos. La idea es que, si los árboles obtenidos están correlacionados, la reducción de la varianza que se pretende podría ser menor de la esperada.

Supóngase que, en el modelo estudiado, existe un predictor muy influyente que domina sobre los demás. Si se utilizara una estrategia de "bagging", es posible que todos los árboles creados estuvieran dominados por dicho predictor y, no sólo serían parecidos, sino que además presentarían una gran correlación entre ellos. En este escenario, apenas se podría disminuir la varianza y, por tanto, mejorar el modelo.

Ilustración 26. Funcionamiento del modelo Random Forest



Fuente: https://rpubs.com/Avalos42/randomforest

El método Random Forest realiza una selección aleatoria de "m" predictores antes de realizar cada división. De esta manera, una media de $\frac{p-m}{p}$ divisiones no contemplarán el predictor influyente, permitiendo a otros predictores ser seleccionados. Con esto se logra decorrelacionar los árboles, lo cual permite una mayor disminución de la varianza y, por tanto, una mejora del modelo.

La diferencia entre Random Forest y "bagging" radicará en el valor del parámetro "m" escogido. Si "m" es igual al valor total de predictores, entonces los resultados de Random Forest y "bagging" serán iguales. El valor recomendado para el parámetro "m" es el siguiente:

$$m = \sqrt{p}$$

Siendo p el número total de predictores.

No obstante, la mejor forma de escoger el valor óptimo de "m" es comparar el valor del "out-of-bag" error para distintos valores de "m". Normalmente, si los predictores presentan alta correlación, valores inferiores a $\sqrt{p}$ consiguen mejores resultados[18].

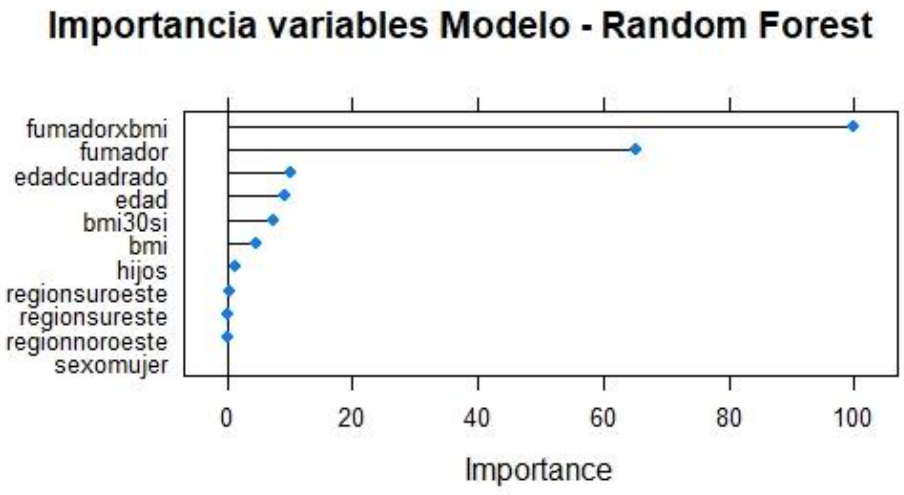
6.2. Resultados

6.2.1. Etapa de entrenamiento

En el proceso de muestreo, se han obtenido 10 particiones o muestras, cada una de ellas aproximadamente de tamaño 962. Además, el modelo obtenido se compone de 6 árboles, número que se ha seleccionado en función del valor mínimo de RMSE.

Con respecto a la importancia relativa de las variables utilizadas en el modelo, se puede señalar que las dos variables más influyentes son "fumadorxbmi" y "fumador".

Ilustración 27. Importancia de las variables del modelo – Random Forest



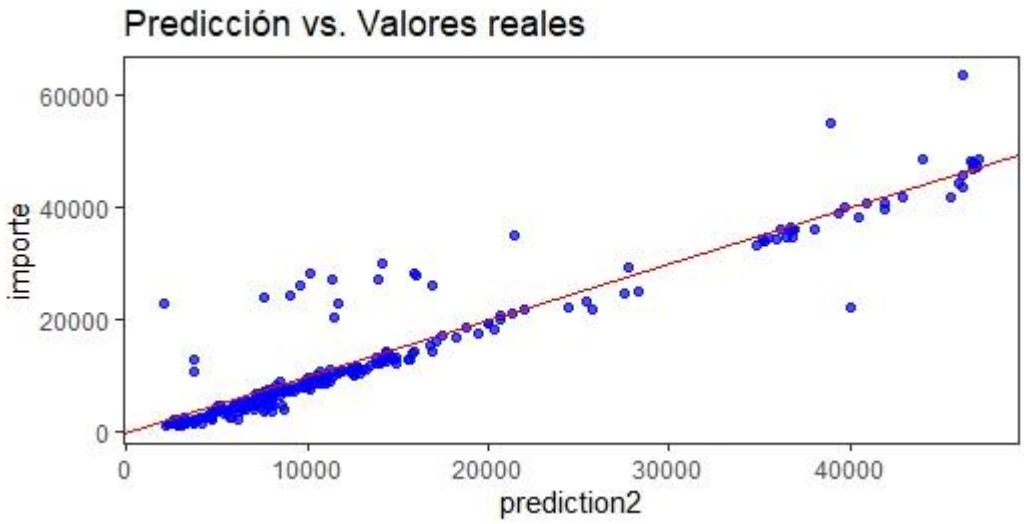
Fuente: elaboración propia

[18] LOUPPE G.: "Understanding Random Forests" University of Liège, 2015. Disponible en: https://arxiv.org/pdf/1407.7502.pdf

6.2.2. Etapa de validación: análisis gráfico y medidas de desempeño

Una vez utilizado el modelo para el conjunto de datos de la etapa de validación, se ha dibujado el mismo gráfico que para los anteriores modelos. Se puede observar una evidente mejora de rendimiento, con respecto al modelo de regresión lineal, debido a que hay un mayor de datos situados sobre la línea roja. Sin embargo, no se intuyen mejoras con respecto a la regresión Cubist. Se debe acudir a las medidas de desempeño para realizar comentarios más precisos.

Ilustración 28. Gráfico predicción vs. valores reales



Fuente: elaboración propia

Para la comparación definitiva, se presentan las dos medidas de desempeño de este modelo:

	Regresión Lineal	Regresión Cubist	Random Forest
RSE	0,112376460851679	0,108597029070604	0,108061551949828
RMSE	4.303,78574065245	4.230,79457908986	4.220,35095399366

Al observarse una mejora de ambas medidas con respecto a los modelos de regresión lineal y regresión cubist, se puede concluir que este modelo tiene una mayor capacidad predictiva.

7. ESTRATEGIA DE BOOSTING: GRADIENT BOOSTING MODEL (GBM) Y EXTREME GRADIENT BOOSTING (XGB)

7.1. Marco teórico

En el capítulo anterior, se presentaba el Boosting como una estrategia para lograr el equilibrio "bias-varianza", objetivo de todo modelo predictivo en Machine Learning. En este capítulo se expone con más detalle dicha estrategia y se estudian los modelos más utilizados en la actualidad.

7.1.1. Estrategia de boosting y algoritmo AdaBoost

La **estrategia de "boosting"** consiste en entrenar, de manera iterativa, varios modelos sencillos con poco valor predictivo, de tal manera que cada nuevo modelo utiliza información del anterior para adaptarse y aprender de sus fallos y aciertos. En el caso de árboles de decisión, estos modelos sencillos serán árboles con una o pocas ramificaciones. A diferencia de la estrategia de "bagging", bajo el "boosting" se pretende conseguir árboles que estén muy correlacionados entre sí.

Otra diferencia, con respecto a los algoritmos de "bagging", es el gran número de hiperparámetros que utilizan. Los principales son los siguientes:

- **Número de modelos** sencillos o **número de iteraciones (M)**. Si este número es demasiado alto, los algoritmos podrían sufrir overfitting y perder capacidad predictiva.
- **Learning rate** (λ). Es un término de regularización que se utiliza para evitar el overfitting en los algoritmos de "boosting". Muestra el ritmo al que aprenden los modelos. Es recomendable que se sitúe entre 0,01 y 0,001.
- **Número de divisiones (d)** de cada árbol. Es conveniente utilizar valores pequeños (entre 1 y 10).

El **algoritmo AdaBoost** es un metaclasificador, es decir, obtiene un nuevo modelo clasificador a partir de un conjunto de clasificadores, creados mediante la estrategia de Boosting, consiguiendo así un modelo clasificador más eficiente. Se habla de modelo clasificador porque el objetivo de este algoritmo es clasificar las observaciones en dos únicas salidas.

Para implementar este algoritmo es necesario establecer una serie de premisas:

- Se debe definir un modelo sencillo, llamado base learner, que sea capaz de predecir la variable respuesta de forma ligeramente superior a cómo lo haría un modelo basado en el azar. Para los árboles de decisión, este modelo sería un árbol con pocas ramificaciones.
- La variable respuesta debe codificarse de tal manera que sólo admita dos únicas salidas: +1 y -1.
- Se debe fijar un peso inicial único para todo el conjunto de observaciones de entrenamiento.

$$w_i = \frac{1}{N}, \qquad i = 1, 2, \dots, N$$

Siendo w_i el peso de la observación i y N el número de observaciones de entrenamiento.

Una vez fijadas las anteriores premisas, se comienza un proceso iterativo basado en los siguientes pasos:

1. Se ajusta el modelo utilizando el conjunto de observaciones de entrenamiento y los pesos iniciales.
2. Se predicen las observaciones de entrenamiento y se señalan los aciertos y errores. En función de esto, se puede calcular el error del modelo:

$$err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$$

Siendo $G_m(x_i)$ la predicción del modelo número m.

3. Se asigna un peso total al modelo en función de su número de aciertos. Cuantos más aciertos logre, mayor será el peso que consiga para influir sobre el modelo final.

$$\alpha_m = \log\left(\frac{1 - err_m}{err_m}\right)$$

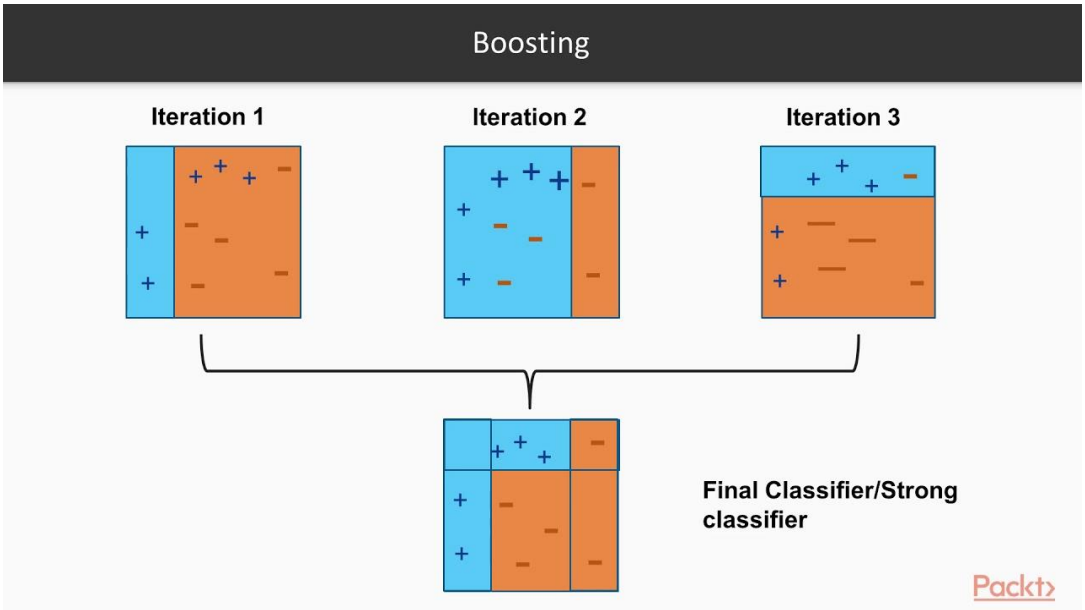
Siendo α_m el peso total asignado al modelo

4. Se modifican los pesos de las observaciones, reduciéndose el de aquellas observaciones acertadas y aumentándose el de las falladas. Se trata de que cada nuevo modelo se centre en predecir correctamente las observaciones que los anteriores no han conseguido predecir.

$$w_i = w_i \exp\left[\alpha_m I\left(y_i \neq G_m(x_i)\right)\right], \qquad i = 1, 2, \dots, N$$

Este proceso se repetirá M veces y generará M modelos sencillos. El modelo o clasificador final tomará las predicciones de cada uno de los M modelos, las ponderará en función de los pesos de cada modelo y obtendrá una clasificación final.

Ilustración 29. Funcionamiento del algoritmo AdaBoost



Fuente: https://thatware.co/adaboost/

7.1.2. Gradient Boosting Machine (GBM)

El algoritmo Gradient Boosting Machine (GBM) es una generalización del algoritmo AdaBoost para otro tipo de funciones de coste del modelo[19], debiendo ser en este caso diferenciables. Esto permite ampliar el campo de estudio a otro tipo de problemas, como la regresión o la clasificación con más de dos clases.

El funcionamiento de este algoritmo es análogo al anterior, pero con algunas modificaciones:

1. Se establecen las premisas del algoritmo. En este punto debe escogerse el modelo sencillo (por ejemplo, un árbol con pocas ramificaciones que incluya regresiones lineales en sus nodos finales), la variable de respuesta (por ejemplo, el coste del siniestro) y el peso único para las observaciones.
2. Se ajusta el modelo utilizando el conjunto de observaciones de entrenamiento.
3. Se predicen las observaciones de entrenamiento y se calcula la función escogida de coste del modelo. Puede ser, por ejemplo, residuos cuadrados para la regresión.

$$obj = \sum_{k=1}^{n} I(y_i, \hat{y}_i)$$

Siendo $I(y_i, \hat{y}_i)$ una función de coste entre el predictor $\hat{y}_i$ y la observación y_i

4. Se inicia una nueva iteración y, a partir de los residuos calculados del modelo anterior, se trata de ajustar un nuevo modelo que intente minimizar la función de error.

[19] En Machine Learning una función de coste o de pérdida es aquella que mide el error del modelo con respecto a la respuesta correcta.

5. Este proceso se repite M veces, de manera que cada modelo intenta minimizar los residuos del modelo anterior.

Este algoritmo puede ocasionar overfitting, por lo que se introduce un valor de regularización, learning rate (λ), que limita la influencia de cada modelo en el conjunto final.

7.1.3. Extreme Gradient Boosting (XGB)

El Extreme Gradient Boosting (XGB) es uno de los algoritmos predictivos más utilizados en la actualidad debido a los increíbles resultados que ofrece. El algoritmo se basa en la misma idea que el GBM y sigue la idea de llevar al límite los recursos computacionales utilizados en dicho modelo para obtener mejores resultados[20], procurando crear un sistema escalable de este mismo.

El algoritmo XGB presenta tres diferencias fundamentales con respecto a los algoritmos tradicionales de Gradient Boosting

- Lo que hace a al algoritmo XGB único es el uso de una forma de regularización mucho más avanzada para controlar el overfitting.
- Además, permite el procesamiento en paralelo, lo cual lo convierte en mucho más rápido y eficiente.
- Finalmente, se debe señalar que, mientras los algoritmos Gradient Boosting tradicionales trabajan con la función de error para la minimización del error total del modelo, XGB trabaja con la derivada de segundo orden como aproximación.

Estas modificaciones han permitido grandes avances en el rendimiento de los modelos y por ello es el algoritmo de Gradient Boosting más utilizado en la actualidad. [21]

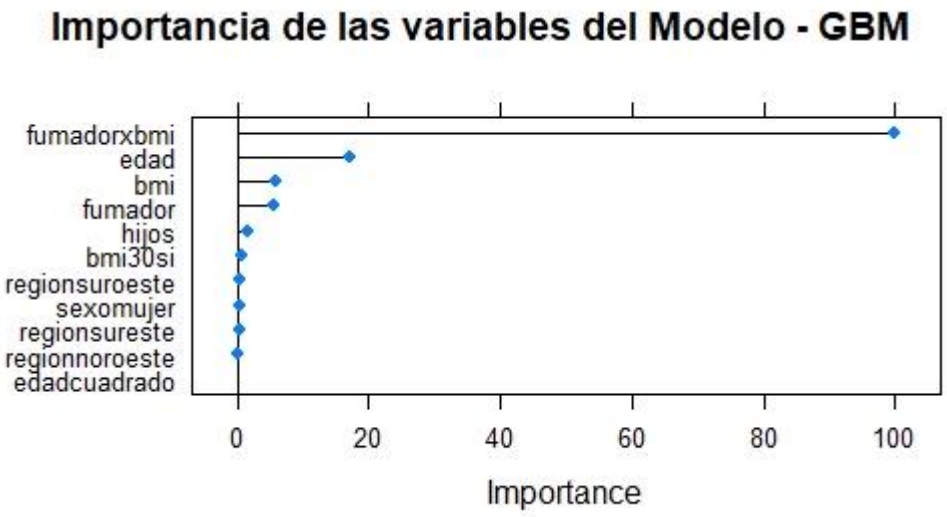
7.2. Resultados

7.1.1. Etapa de entrenamiento

Tras entrenar ambos modelos, se puede extraer la importancia relativa de las variables en cada uno de ellos. En ambos casos las variables más influyentes son “fumadorxbmi” y “edad”.

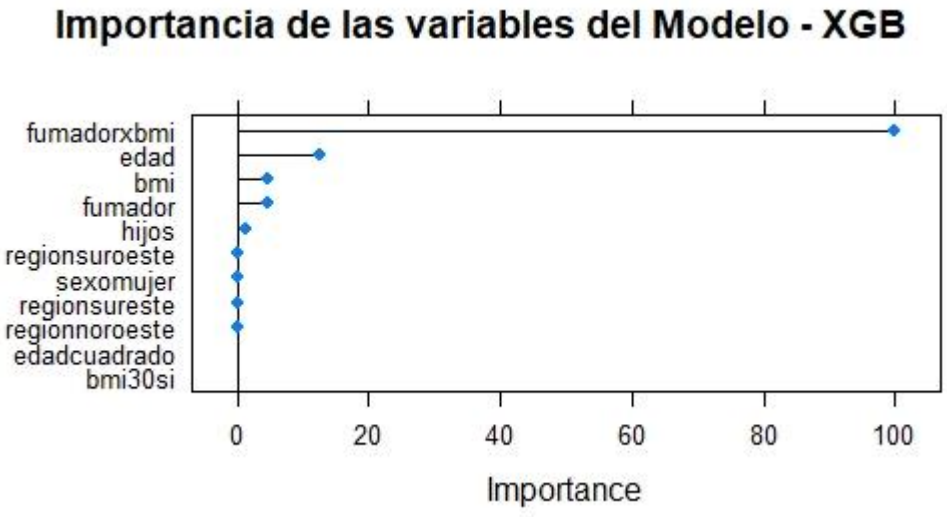
[20] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
[21] https://blog.exploratory.io/introduction-to-extreme-gradient-boosting-in-exploratory-7bbec554ac7

Ilustración 30. Importancia de las variables del modelo – GBM



Fuente: elaboración propia

Ilustración 31. Importancia de las variables del modelo – XGB

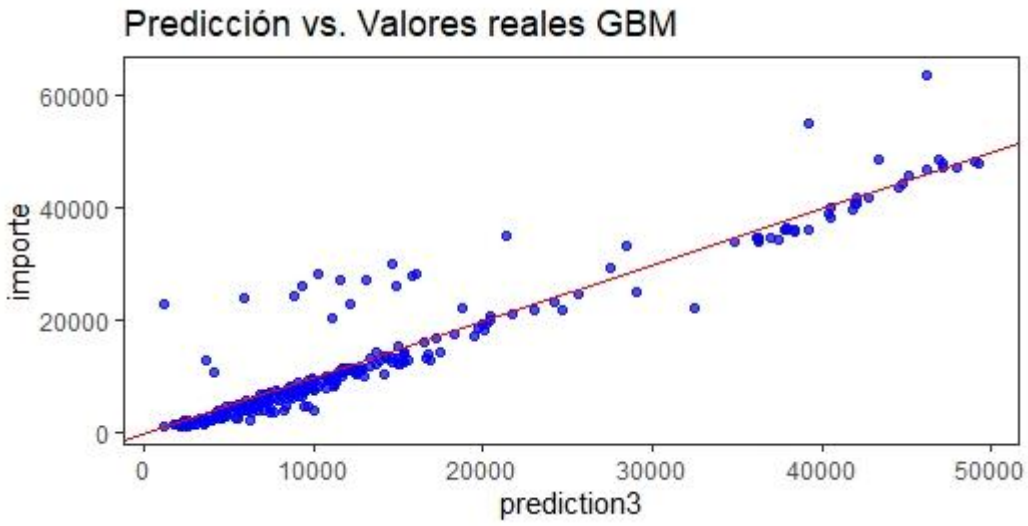


Fuente: elaboración propia

7.2.2. Etapa de validación: análisis gráfico y medidas de desempeño

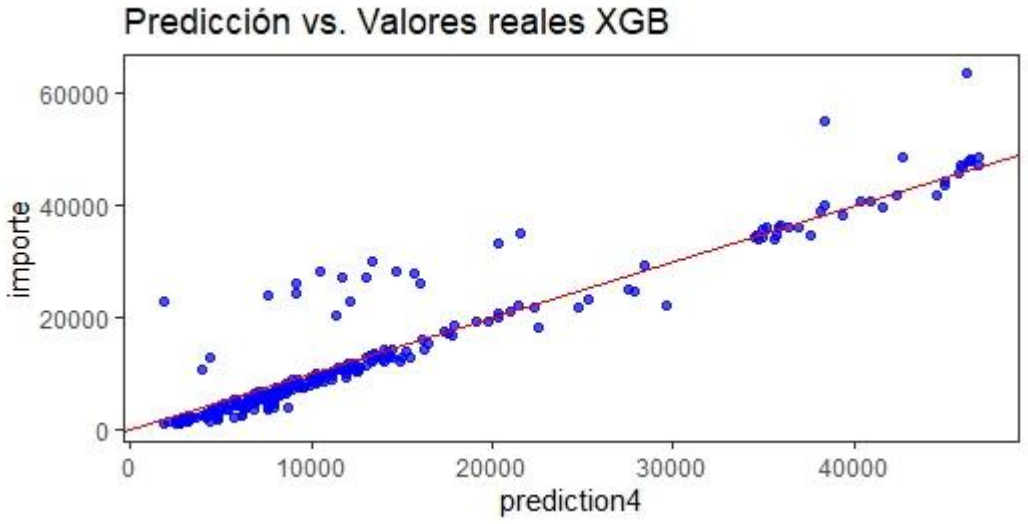
Una vez modelado el conjunto de datos de la etapa de validación, se ha dibujado el mismo gráfico que para los modelos anteriores. Se puede observar una evidente mejora en el rendimiento, con respecto al modelo de regresión lineal, debido a que hay un mayor de datos situados sobre la línea roja. Sin embargo, no se intuyen mejoras con respecto a los otros métodos.

Ilustración 32. Gráfico predicción vs. valores reales GBM



Fuente: elaboración propia

Ilustración 33. Gráfico predicción vs. valores reales XGB



Fuente: elaboración propia

Para la comparación definitiva se presentan las dos medidas de desempeño de ambos modelos.

	Regresión Lineal	Regresión Cubist	Random Forest
RSE	0,112376460851679	0,108597029070604	0,108061551949828
RMSE	4.303,78574065245	4.230,79457908986	4.220,35095399366

	GBM	XGB
RSE	0,108705886305833	0,106589624650418
RMSE	4.232,91451390034	4.191,50929448767

Se puede concluir que, aunque el modelo GBM haya arrojado un peor rendimiento que los modelos “cubist” y “random forest”, el modelo XGB ofrece las mejores medidas de rendimiento hasta el momento.

8. SUPPORT VECTOR MACHINE (SVM)

8.1. Marco teórico

Support Vector Machine (SVM) es otra de las técnicas de Machine Learning más extendidas en la actualidad. Para entender su funcionamiento se deben presentar primero los conceptos de hiperplano y Maximal Margin Classifier.

8.1.1. Hiperplano y algoritmo Support Vector Classifier

Supóngase un espacio p-dimensional, es decir, que tiene p dimensiones. Un hiperplano es un subespacio comprendido en este plano y afín (no tiene que pasar por el origen) con dimensiones p – 1. Bajo esta definición, se exponen algunos ejemplos:

- En un espacio de dos dimensiones, un hiperplano sería un subespacio de una dimensión, es decir, una recta.
- En un espacio de tres dimensiones, un hiperplano sería un subespacio de dos dimensiones, es decir, un plano tradicional.
- En un espacio de dimensión superior a tres, no sería tan intuitiva la representación de un hiperplano, pero el concepto se mantiene.

Un hiperplano de dimensión p tendría la siguiente forma matemática:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = 0$$

Dados los parámetros $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ todos los puntos definidos por el vector $x = (x_1, x_2, \dots, x_p)$ que satisfacen la ecuación forman y pertenecen al hiperplano. Si x no cumple la ecuación, el punto está a un lado u otro del hiperplano. Por lo tanto, un hiperplano

divide un espacio en dos partes y se sabrá a cuál cae cada punto en función del signo de la ecuación.

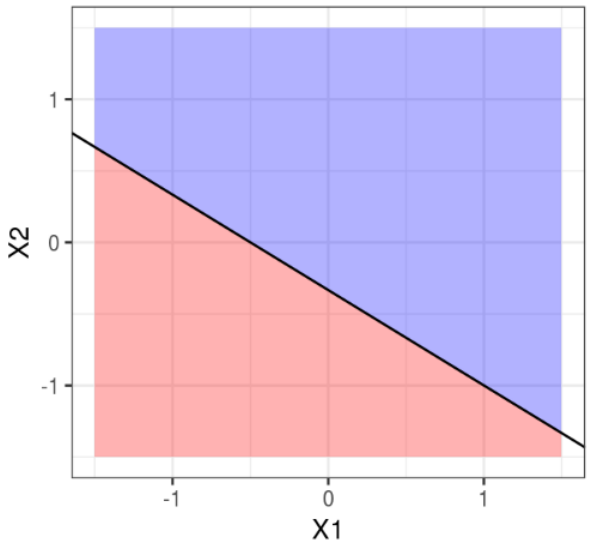
De esta manera, se podría utilizar el concepto de hiperplano para construir un modelo clasificador que dividiera las observaciones en dos grupos o clases. Si se pudiera separar de forma perfecta en dos clases (+1 y -1), un hiperplano de separación cumpliría lo siguiente:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p > 0 \rightarrow y_i = +1$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p < 0 \rightarrow y_i = -1$$

Se trata del clasificador más sencillo que existe debido a que únicamente con el signo de la función se obtiene la variable de salida. Además, la magnitud de la función indica cómo de lejos está la observación del hiperplano y, con ello, se obtiene la confianza del modelo.

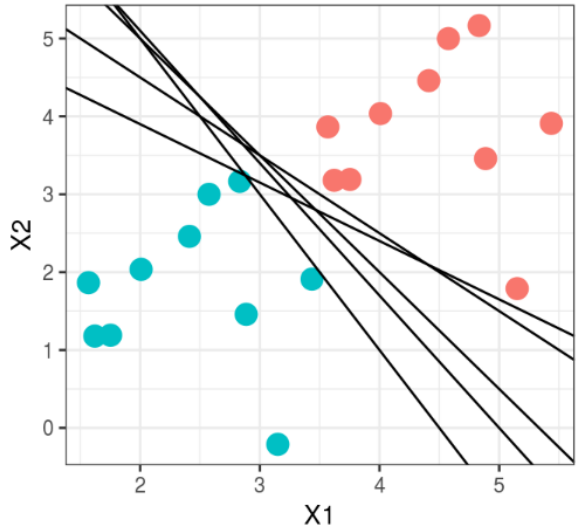
Ilustración 34. Concepto de hiperplano de clasificación



Fuente: cienciadedatos.net

Se podrían obtener múltiples funciones para abordar un problema de separación lineal de casos perfectamente separables. La cuestión sería encontrar un método que seleccionara la función óptima (el hiperplano óptimo de separación). Para ello, se debe calcular la distancia de cada observación al hiperplano. La menor distancia definirá el hiperplano óptimo de separación. El problema se encuentra en que el número de hiperplanos posibles a comparar sería infinito, por lo que se recurre a métodos de optimización.

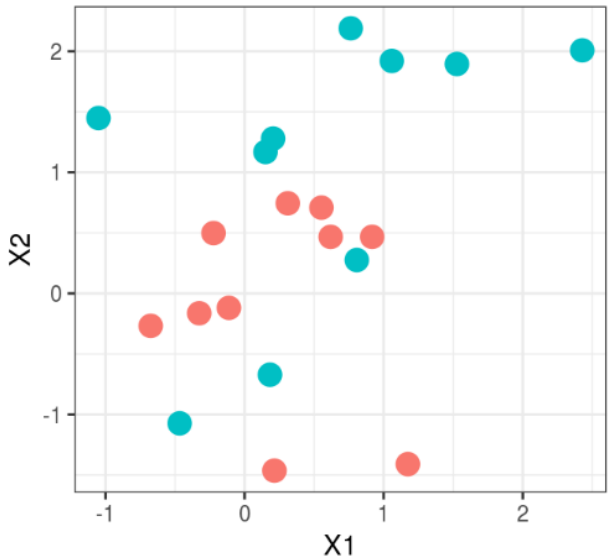
Ilustración 35. Hiperplano para casos perfectamente separables



Fuente: cienciadedatos.net

Aunque la idea de hiperplano es útil en aquellos casos en los que los datos son perfectamente separables, en la mayoría de las ocasiones, los datos no se pueden separar linealmente de forma perfecta. Incluso si los datos de entrenamiento se pudieran separar linealmente, puede que nuevos datos estudiados no siguieran el mismo patrón, lo que se traduciría en problemas de overfitting. Para abordar este tipo de problemas surge un nuevo tipo de hiperplano conocido como Support Vector Machine.

Ilustración 36. Casos no separables linealmente



Fuente: cienciadedatos.net

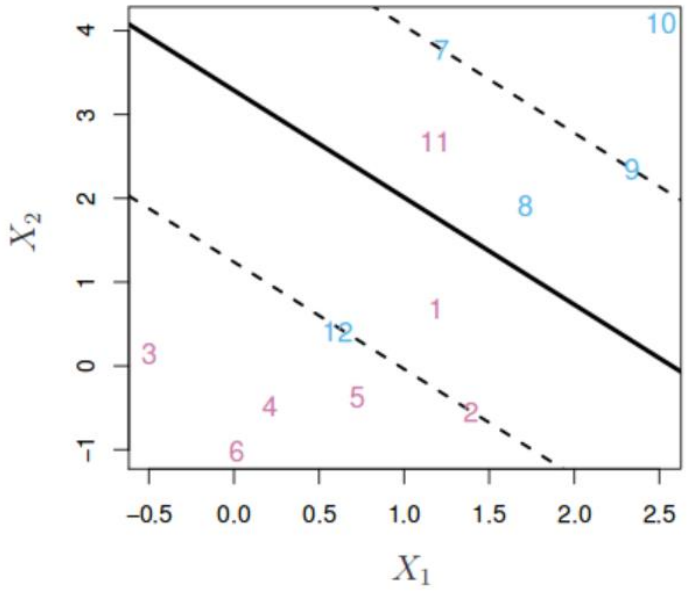
El Support Vector Classifier aporta más robustez y capacidad a los modelos predictivos de hiperplanos permitiendo que algunas observaciones se encuentren en el lado del hiperplano opuesto al que les corresponde.

Como indica la ilustración, a ambos lados del hiperplano se define un margen. Las observaciones que se encuentran sobre las líneas de este margen (tanto los aciertos como los fallos) se conocen como vectores soporte y son las únicas que influyen sobre el modelo obtenido. Las observaciones que quedan fuera del margen no presentan influencia sobre este.

No se va a profundizar en demostraciones matemáticas, pero se debe señalar que en el proceso se introduce un hiperparámetro C que mide el número y violaciones del margen que se toleran. Cuanto más se acerque a 0, mayor número de penalizaciones se permitirán. Si tiende a infinito, no se permitirá ninguna violación. El valor escogido para C será aquel que logre un mejor equilibrio bias-varianza:

- Si este es pequeño, el margen es más amplio y más observaciones forman los vectores soporte. Esto aumenta el "bias" porque hay más predicciones erróneas, pero reduce la "varianza".
- Si este es grande, el margen será más pequeño y menos observaciones formarán los vectores soporte. Esto reduce el "bias" porque hay menos predicciones erróneas, pero aumenta la "varianza".

Ilustración 37. Funcionamiento del Supporting Vector Classifier

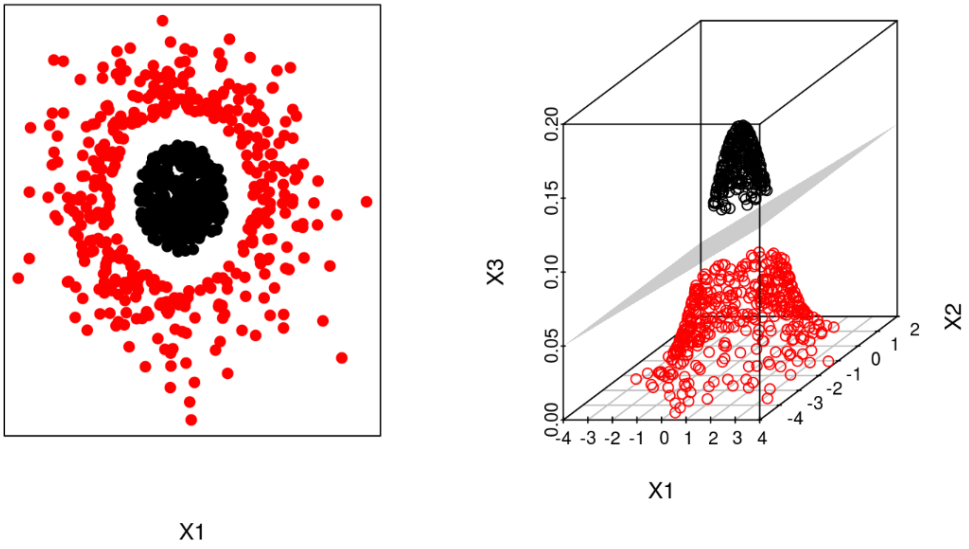


Fuente: "An Introduction to Statistical Learning with applications in R" por James G. y otros.

8.1.2. Support Vector Machine (SVM)

El algoritmo anterior presenta buenos resultados si el hiperplano es lineal, pero, si no lo es, su robustez disminuye. Esto se debe a que puede que un conjunto de datos no sea linealmente separable pero sí sea separable en un espacio de dimensiones mayores. El algoritmo Support Vector Machine (SVM) extiende el Support Vector Classifier a dimensiones superiores.

Ilustración 38. Funcionamiento Support Vector Machine (SVM)



Fuente: cienciadedatos.net

Para aumentar la dimensión se utilizan las funciones kernel, que transforman el espacio de dos dimensiones en otro de dimensiones superiores. Algunos ejemplos de kernels son los siguientes:

Kernel lineal:

$$K(x, x') = x * x'$$

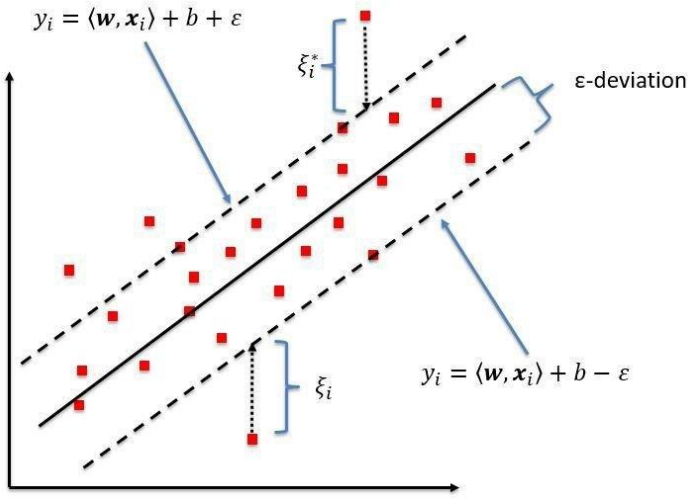
Kernel polinómico:

$$K(x, x') = (x * x' + c)^d$$

Existen múltiples estrategias para extender esta idea al estudio de situaciones con más de 2 clases, así como a problemas de regresión en los que la variable respuesta es continua. Gracias a estas estrategias, este algoritmo se ha convertido en uno de los más populares en cualquier tipo de estudio de Machine Learning.

Para un problema lineal de regresión, como el ejemplo de este trabajo, el algoritmo tratará de buscar la curva que mejor modele la tendencia de los datos. Se deberá definir un margen de tamaño ε. Ahora la idea será agrupar a todas las observaciones en este margen en torno a la función de predicción. Sólo las observaciones que disten más de ε del hiperplano serán consideradas vectores soporte.

Ilustración 39. Funcionamiento del SVM para un problema de regresión lineal



Fuente: researchgate.net

Para problemas de regresión no lineales se deberá utilizar una función del tipo kernel para transformar dicha curva[22].

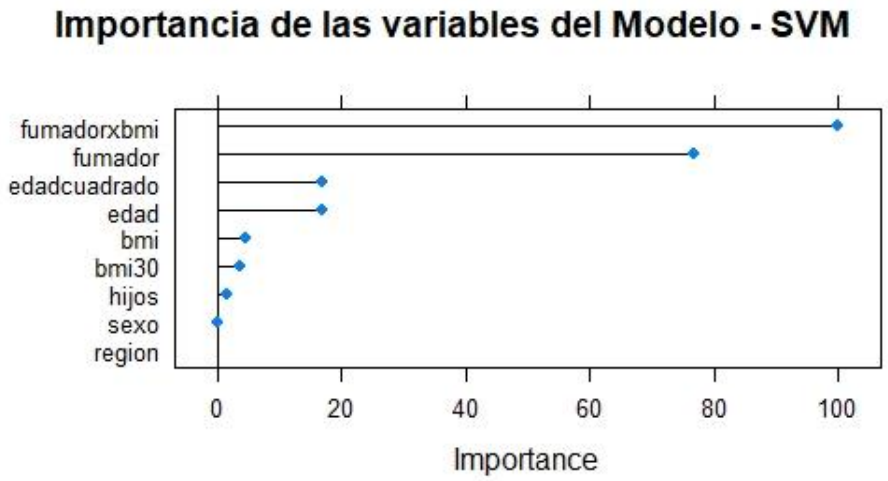
8.2. Resultados

8.1.1. Etapa de entrenamiento

Se puede extraer la importancia relativa de las variables utilizadas en el modelo, obteniéndose que las variables “fumadorxbmi” y “fumador” son las más influyentes.

[22] CORTES, C. y VAPNIK V.: “Support-vector networks” Machine Learning, 20(3), 1995

Ilustración 40. Importancia de las variables del modelo – SVM

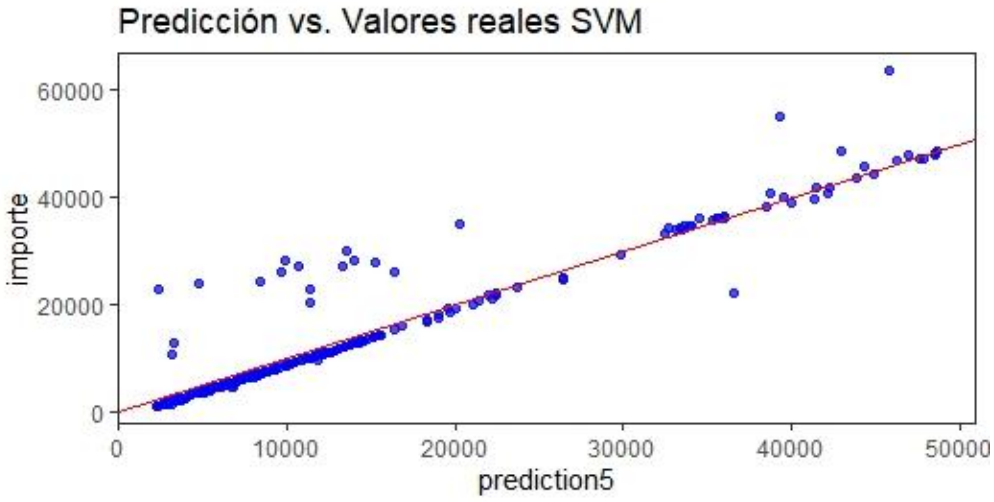


Fuente: elaboración propia

8.2.2. Etapa de validación: análisis gráfico y medidas de desempeño

Una vez utilizado el modelo para el conjunto de datos de validación, se ha dibujado el mismo gráfico que para el resto de los modelos. Se puede observar que este modelo se ajusta particularmente bien al conjunto de observaciones de validación. No obstante, se debe acudir a las medidas de desempeño para extraer conclusiones definitivas.

Ilustración 41. Gráfico predicción vs. valores reales



Fuente: elaboración propia

Para la comparación definitiva se presentan las dos medidas de desempeño de este modelo, que se añaden a las anteriormente obtenidas.

	Regresión Lineal	Regresión Cubist	Random Forest
RSE	0,112376460851679	0,108597029070604	0,108061551949828
RMSE	4.303,78574065245	4.230,79457908986	4.220,35095399366

	GBM	XGB	SVM
RSE	0,108705886305833	0,106589624650418	0,104070185501751
RMSE	4.232,91451390034	4.191,50929448767	4.141,67609587954

A la vista de los resultados, se concluye que el algoritmo Support Vector Machine muestra las menores medidas de error de todo el estudio.

9. CONCLUSIONES

La intención de este estudio era recorrer la base teórica y el funcionamiento de alguno de los algoritmos de Machine Learning más utilizados en la actualidad siguiendo un ejemplo de siniestros de seguros de salud. Tras observar la variedad de técnicas existentes y analizar las fortalezas y debilidades de cada una de ellas, se subraya la necesidad de examinar y comparar distintas técnicas en cada caso de estudio.

Para el caso de estudio se ha realizado, en primer lugar, un análisis exploratorio de los datos que ha permitido, por un lado, identificar aquellas variables más influyentes sobre la variable respuesta, los hábitos de tabaco y la edad; y, por el otro, agregar al estudio transformaciones de las variables, la edad cuadrática y el producto entre BMI y hábitos de tabaco, que mejoran la capacidad predictiva de los modelos. Seguidamente, se ha entrenado varios modelos, obteniéndose que dichas variables son las más influyentes en cada uno de ellos. Además, las técnicas más sencillas, como la regresión lineal, han anotado mayores errores que aquellas basadas en algoritmos más avanzados, como “random forest” o “gradient boosting”.

Como principal conclusión de este trabajo, con base en los análisis desarrollados en los capítulos anteriores y a la vista del cuadro de comparación de las medidas de desempeño, el modelo que registra el mejor rendimiento es el basado en el algoritmo Support Vector Machine (SVM).

Paralelamente a las cuestiones técnicas, conviene señalar que la valoración de riesgos supone actualmente un reto en seguros de salud, donde entran en juego múltiples factores que cuestionan y desafían a los modelos y técnicas propuestas en este trabajo. Convendrá, por ende, trabajar en las próximas décadas en el acceso por parte de las compañías a nuevas fuentes de datos; sin perder de vista todos los posibles cambios o novedades que introduzca la normativa.

BIBLIOGRAFÍA

CARDONA HERNANDEZ P. A.: "Aplicación de árboles de decisión en modelos de riesgo crediticio" Revista Colombiana de Estadística, Volumen 27 Nº2, 2004.

COMISIÓN EUROPEA: "State of Health in the EU: España, perfil sanitario del país 2017" *European Observatory on Health Systems and Policies*, 2017.

CORTES, C. y VAPNIK V.: "Support-vector networks" Machine Learning, 20(3), 1995

EUROPEAN OBSERVATORY ON HEALTH CARE SYSTEMS SERIES: "Voluntary health insurance in the European Union." Chap. 5, In *Funding Health Care: Options for Europe*, edited by Mossialos, E., Dixon, A., Figueras, J. Y Kutzin, J.. Philadelphia (USA), Open University Press, 2001 pp. 129-142.

GUILLEN, M. y PESANTEZ-NARVAEZ, J.: "Machine Learning y modelización predictiva para la tarificación en el seguro de automóviles". Anales del Instituto de Actuarios Españoles, 4ª época, 24, 2018/123-147. Disponible en: https://www.actuarios.org/wp-content/uploads/2018/11/123_147_A06.pdf

KLUGMAN S. A., PANJER H. H. y WILLMOT G.E.: "Loss Models". Willey, 2029 (Fifth Edition).

KUHN M., WESTON S., KEEFER C y COULTER N.: "Cubist Models for Regression". 11 de mayo de 2012. Disponible en:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.398.3360&rep=rep1&type=pdf

KUHN M.: "Rules Rules Rules! Cubist Regression Models" Boston R User Group. Disponible en:
https://static1.squarespace.com/static/51156277e4b0b8b2ffe11c00/t/56e3056a3c44d8779a61988a/1457718645593/cubist_BRUG.pdf

LAGUNA C.: "Correlación y regresión lineal". Instituto Aragonés de Ciencias de la Salud. Disponible en:
http://www.ics-aragon.com/cursos/salud-publica/2014/pdf/M2T04.pdf

LANTZ B.: "Machine Learning with R". Packt Publishing, 2013. Disponible en: https://edu.kpfu.ru/pluginfile.php/278552/mod_resource/content/1/MachineLearningR__Brett_Lantz.pdf

LOUPPE G.: "Understanding Random Forests" University of Liège, 2015. Disponible en:
https://arxiv.org/pdf/1407.7502.pdf

MEYER H., KNUDBY A., APPELHANS T., MÜLLER M.U. y otros: "Mapping daily air temperature for Antarctica base don MODIS LST." Remote Sens, 2016.

PÉRTEGA DÍAZ S. y PITA FERNÁNDEZ S.: Técnicas de Regresión: Regresión lineal múltiple" Unidad de Epidemiología Clínica y Bioestadística, 2000. Disponible en:
https://www.fisterra.com/gestor/upload/guias/regre_lineal_multi2.pdf

TILVES M.: "A fondo: Así ayuda el Machine Learning al sector seguros". Silicon, 2017. Disponible en: https://www.silicon.es/a-fondo-machine-learning-seguros-2335520

TSE Y. K.: "Nonlife actuarial models. Theory, methods and evaluation". Cambridge University Press, 2009.

ZHANG W., HUANG Y., YU Y.Q. y SUN W.J: “Empirical models for estimating daily máximum, mínimum and mean air temperaturas with MODIS land surface temperaturas.” Int. J. Remote Sens, 2011.

ANEXO 1. FORMULARIO SHINY

Se ha construido un formulario para mostrar los resultados del trabajo con la ayuda del paquete Shiny de R. Shiny permite elaborar aplicaciones web interactivas a partir de código R.

Por un lado, los inputs de la aplicación son los siguientes:

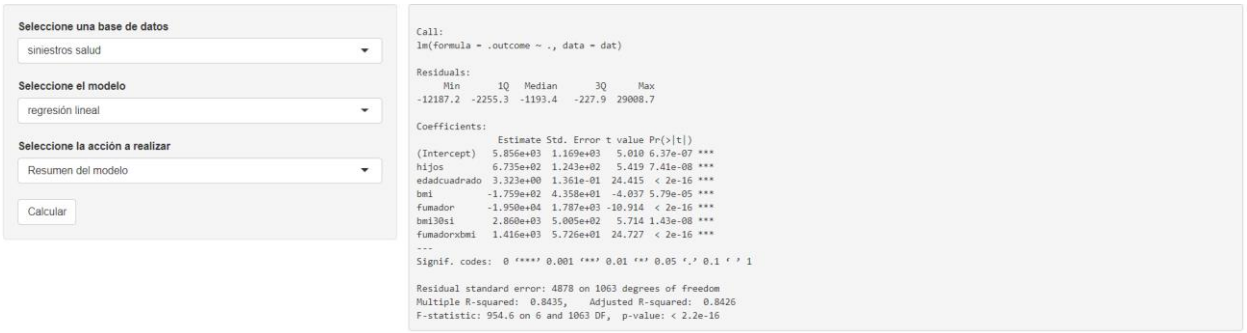
- Seleccione una base de datos. La aplicación sólo permite seleccionar la base de datos de siniestros de salud.
- Seleccione el modelo. La aplicación permite seleccionar los 6 modelos estudiados: regresión lineal, regresión cubist, random forest, gbm, xgb y svm.
- Seleccione la acción a realizar. Este elemento permite seleccionar el tipo de resultado que debe mostrar la aplicación.

Por otro lado, las salidas o resultados de la aplicación son los siguientes:

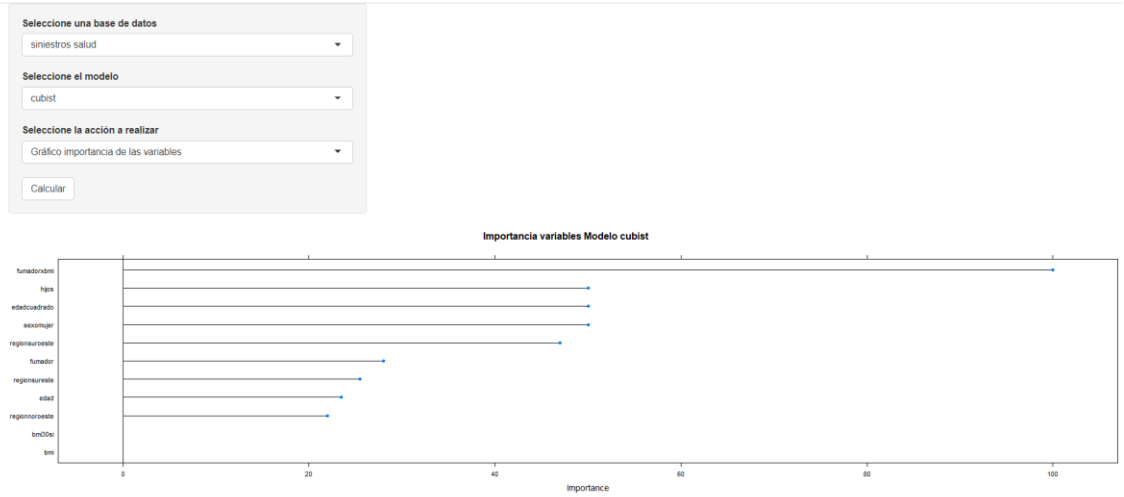
- Resumen del modelo. Esta salida consiste en un cuadro resumen del modelo seleccionado.
- Gráfico de la importancia de las variables. Esta salida consiste en un gráfico que incluye la importancia relativa de las variables empleadas en el modelo seleccionado.
- Gráfico de la calidad de la predicción. Esta salida consiste en un gráfico que muestra las predicciones obtenidas frente a la observación.
- Medidas de desempeño. Esta salida incluye las dos medidas de desempeño estudiadas en el trabajo.

Finalmente, se muestran algunas capturas de la aplicación desarrollada.

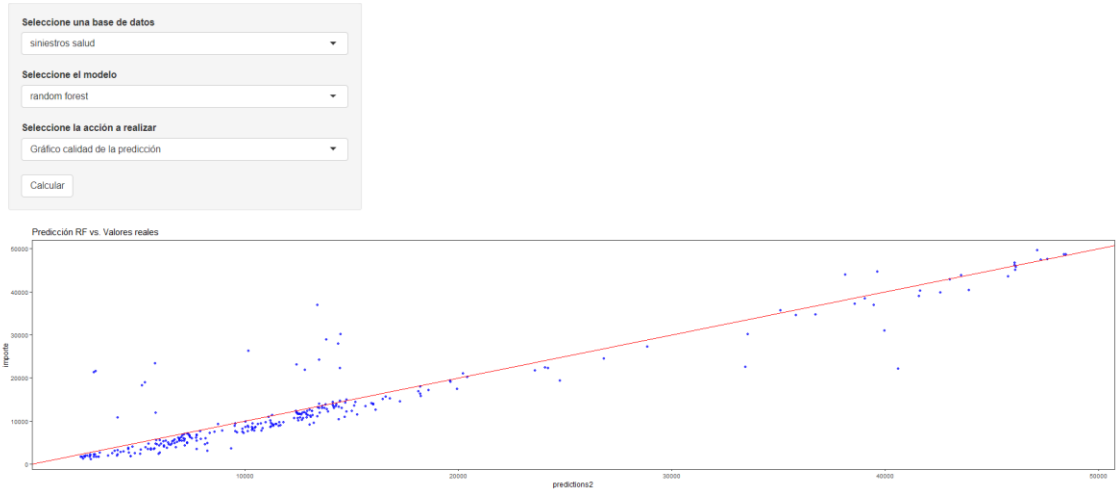
Captura 1. Resumen del modelo de regresión lineal



Captura 2. Gráfico importancia de las variables del modelo Cubist

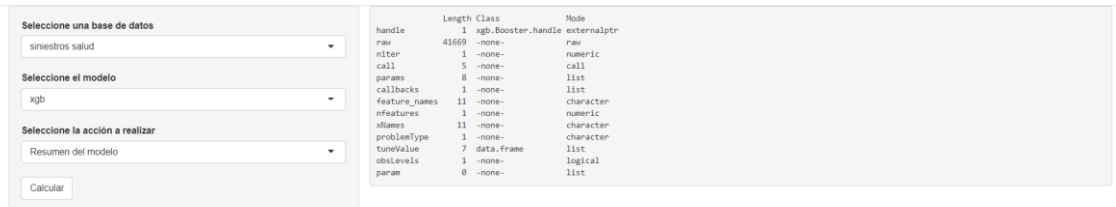


Captura 3. Gráfico calidad de la predicción del modelo random forest

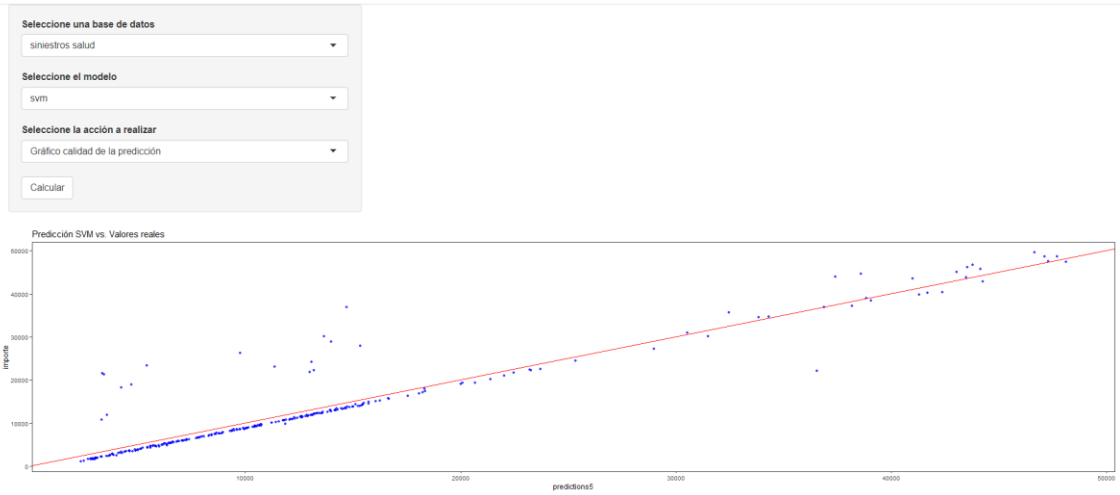


Captura 4. Medidas de desempeño del modelo gbm

Captura 5. Resumen del modelo xgb



Captura 6. Gráfico calidad de la predicción del modelo svm



ANEXO 2. CÓDIGO

```
#######################################
########ACTIVACIÓN DE LIBRERÍAS#######
#######################################
library(ggplot2)
library(ggthemes)
library(psych)
library(relaimpo)
library(GGally)

library(dplyr)
library(Hmisc)
library(cowplot)
library(WVPlots)

library(gridExtra)

library(pscl)
library(dominanceanalysis)
library(reshape2)

library(randomForest)
library(tidyverse)
library(caret)
library(MASS)
library(gbm)
library(Metrics)
library(xgboost)
library(neuralnet)
library(kernlab)

library(brnn)
library(arm)
library(survival)
library(fitdistrplus)
library(qcc)
library(actuar)

library(gridExtra)
```

```
library(e1071)
library(Cubist)
library(party)

#######################################
############CARGA DE DATOS#############
#######################################
insurance <- read.csv("C:/…/ TFM/insurance.csv")
insurance$bmi30 <- ifelse(insurance$bmi>=30,"si","no")
insurance$fumador <- as.numeric(insurance$fumador)
insurance$fumador <- insurance$fumador -1
insurance$edadcuadrado <- insurance$edad^2
insurance$fumadorxbmi <-insurance$fumador*insurance$bmi

#######################################
#########VISTAZO AL DATA FRAME########
#######################################
str(insurance)
summary(insurance)

#######################################
###########DIAGRAMAS DE CAJA##########
#######################################
dcaja_region <-ggplot(data = insurance,aes(region,importe)) + geom_boxplot(fill = c(2:5)) +
  theme_minimal() + ggtitle("Diagrama de caja región")

dcaja_fumador <- ggplot(data = insurance,aes(fumador,importe)) + geom_boxplot(fill = c(2:3)) +
  theme_minimal() + ggtitle("Diagrama de caja tabaco")

dcaja_sexo <- ggplot(data = insurance,aes(sexo,importe)) + geom_boxplot(fill = c(2:3)) +
  theme_minimal() + ggtitle("Diagrama de caja sexo")

dcaja_hijos <- ggplot(data = insurance,aes(as.factor(hijos),importe)) + geom_boxplot(fill = c(2:7)) +
  theme_minimal() + xlab("número de hijos") + ggtitle("Diagrama de caja  nº hijos")

dcaja_bmi30 <-ggplot(data = insurance,aes(bmi30,importe)) + geom_boxplot(fill = c(2:3)) +
  theme_minimal() + ggtitle("Diagrama de caja Obesidad")

grid.arrange(dcaja_fumador, dcaja_sexo, ncol=2, nrow=1)
grid.arrange(dcaja_hijos, dcaja_region, ncol=2, nrow=1)
```

```
grid.arrange(dcaja_fumador, dcaja_sexo, dcaja_hijos, dcaja_bmi30, ncol=2, nrow=2)

#######################################
###############DISPERSIÓN#############
#######################################
#gráfico dispersión por edad y obesidad

x <- ggplot(insurance, aes(edad, importe)) +
  geom_jitter(color = "blue", alpha = 0.5) +theme_classic()

y <- ggplot(insurance, aes(bmi, importe)) +
  geom_jitter(color = "green", alpha = 0.5) +theme_classic()

p <- plot_grid(x, y)
title <- ggdraw() + draw_label("Dispersión importe y edad/IMC",
                               fontface='bold')
plot_grid(title, p, ncol=1, rel_heights=c(0.1, 1))

#gráfico correlación por sexo y número de hijos
x <- ggplot(insurance, aes(sexo, importe)) +
  geom_jitter(aes(color = sexo), alpha = 0.7) +theme_classic()

y <- ggplot(insurance, aes(hijos, importe)) +
  geom_jitter(aes(color = hijos), alpha = 0.7) +theme_classic()

p <- plot_grid(x, y)
title <- ggdraw() + draw_label("Dispersión  importe y sexo/nº hijos", fontface='bold')
plot_grid(title, p, ncol=1, rel_heights=c(0.1, 1))

#gráfico correlación por hábitos de tabaco y región
x <- ggplot(insurance, aes(fumador, importe)) +
  geom_jitter(aes(color = fumador), alpha = 0.7) +theme_classic()

y <- ggplot(insurance, aes(region, importe)) +
  geom_jitter(aes(color = region), alpha = 0.7) +theme_classic()

p <- plot_grid(x, y)
title <- ggdraw() + draw_label("Dispersión  importe y tabaco/región", fontface='bold')
plot_grid(title, p, ncol=1, rel_heights=c(0.1, 1))
```

```
#se pinta un cuadro con las correlaciones
ggcorr(insurance[-8], name = "corr", label = TRUE) +  theme(legend.position="none")

#######################################
###########VARIABLE IMPORTE###########
#######################################
ggplot(data=insurance, aes(x=importe, fill = ..count..)) +
  geom_histogram(binwidth=1000)+
  scale_y_continuous(breaks = seq(0, 65000, by = 5000))+theme_classic()

fg<-fitdist(insurance$importe, "gamma", method = "mle",lower = c(0, 0), start = list(scale = 1, shape = 1))
summary(fg)
denscomp(fg)
ks.test(insurance$importe, "pgamma", scale = 9405.928057, shape =1.411199)

fln<-fitdist(insurance$importe, "lnorm")
summary(fln)
denscomp(fln)
ks.test(insurance$importe, "plnorm", meanlog=9.0986587, sdlog=0.9191834)


fp<-fitdist(insurance$importe, "pareto",method = "mle", start = list(shape = 1, scale = 500))
summary(fp)
denscomp(fp)
ks.test(insurance$importe, "ppareto", scale = 7.928337e+05, shape =1.051091e+10)


#######################################
#ITERACIONES Y RELACIONES CUADRÁTICAS#
#######################################
#explicación de la relación cuadrática con la edad
x <- ggplot(insurance, aes(edad, importe)) +
  geom_jitter(color = "blue", alpha = 0.5) +theme_classic()

y <- ggplot(insurance, aes(edad^2, importe)) +
  geom_jitter(aes(color = hijos), alpha = 0.7)+theme_classic()

p <- plot_grid(x, y)
title <- ggdraw() + draw_label("Gráfico dispersión entre edad / edad^2 e importe", fontface='bold')
```

```
plot_grid(title, p, ncol=1, rel_heights=c(0.1, 1))

#explicación de la iteración entre bmi y fumador
ggplot(insurance, aes(x = bmi, y = importe, col = fumador)) +
  geom_point()+theme_classic()

########################################
#######DIVISION DE LOS DATOS##########
########################################
# Dividir la base de datos entre fase de entrenamiento y fase de validación
set.seed(100)
train <- sample(nrow(insurance), 0.8*nrow(insurance), replace = FALSE)
TrainSet <- insurance[train,]
ValidSet <- insurance[-train,]
summary(TrainSet)
summary(ValidSet)
########################################
#############MODELO 0: LM#############
########################################
#modelo regresión lineal simple
lmsimple <- lm(importe~edad,data=TrainSet)
summary(lmsimple)
ggplot(data = insurance, mapping = aes(x = edad, y = importe)) +
  geom_point(color = "firebrick", size = 2) +
  labs(title  =  'importe ~ edad', x  =  'edad') +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

#Selección del modelo regresión lineal múltiple
lm0 <- lm(importe~edad+sexo+bmi+hijos+fumador+region+bmi30,data=TrainSet)
summary(lm0)
par(mfrow = c(2, 2))
plot(lm0)
shapiro.test(lm0$residuals)

lm1 <- lm(importe~hijos+edadcuadrado+bmi+fumador+bmi30+fumadorxbmi,data=TrainSet)
summary(lm1)

par(mfrow = c(2, 2))
```

```
plot(lm1)

#formulación del modelo 0 definitivo
modelo0 <- train(importe ~ hijos+edadcuadrado+bmi+fumador+bmi30+fumadorxbmi,
                 data = TrainSet, method = "lm", trControl=trainControl(method = "cv", number = 5))
summary(modelo0)
print(modelo0)
#Validación del modelo
predictions0 <- predict(modelo0, ValidSet)
#medidas de la performance del modelo en la fase de validación
rmse0<-rmse(ValidSet$importe,predictions0)
rse0<-rse(ValidSet$importe,predictions0)
paste0("RSE is ",rse0)
paste0("RMSE is ",rmse0)

#gráfico de importancia de las variables del modelo
plot(varImp(modelo0),main="Importancia de las variables del Modelo - LM")



########################################
############MODELO 1: CUBIST############
########################################

modelo1 <- train(importe ~.,data = TrainSet, method = "cubist",
                 trControl=trainControl(method = "cv", number = 5), grid =
                   expand.grid(committees = c(1, 10, 50, 100),neighbors = c(0, 1, 5, 9)))
summary(modelo1)
print(modelo1)
ggplot(modelo1)
#Validación del modelo
predictions1 <- predict(modelo1, ValidSet)
#medidas de la performance del modelo en la fase de validación
rmse1<-rmse(ValidSet$importe,predictions1)
rse1<-rse(ValidSet$importe,predictions1)
paste0("RSE is ",rse1)
paste0("RMSE is ",rmse1)

#gráfico de importancia de las variables del modelo
plot(varImp(modelo1),main="Importancia variables Modelo - Cubist Regression")
```

```
#######################################
########MODELO 2: RANDOM FOREST#######
#######################################
# FASE ENTRENAMIENTO
#definición del modelo
modelo2<- train(importe~., data =TrainSet,
          method = "cforest", trControl = trainControl("cv", number = 10))
modelo2
#gráfico de importancia de las variables del modelo
plot(varImp(modelo2),main="Importancia variables Modelo - Random Forest")
#medidas de la performance del modelo en la fase de entrenamiento
getTrainPerf(modelo2)
#FASE VALIDACIÓN
#Puesta en marcha del modelo
predictions2 <- predict(modelo2, ValidSet)
#medidas de la performance del modelo en la fase de validación
rmse2<-rmse(ValidSet$importe,predictions2)
rse2<-rse(ValidSet$importe,predictions2)
paste0("RSE is ",rse2)
paste0("RMSE is ",rmse2)

#######################################
#####MODELO 3: Gradient Boosting######
#######################################
#FASE ENTRENAMIENTO
#definición del modelo
set.seed(12345)
modelo3<- train(importe ~., data =TrainSet, method = "gbm", trControl =
          trainControl("cv", number = 10),tuneGrid =
          expand.grid(interaction.depth = c(4,6,8),n.trees=c(1,3,5,7,9)*100,
               shrinkage = c(0.1) , n.minobsinnode = 20),metric = "RMSE" )
modelo3
plot(modelo3)
#gráfico de importancia de las variables del modelo
plot(varImp(modelo3),main="Importancia de las variables del Modelo - GBM")
#medidas de la performance del modelo en la fase de entrenamiento
getTrainPerf(modelo3)
#FASE VALIDACIÓN
#Puesta en marcha del modelo
```

```
predictions3 <- predict(modelo3, ValidSet)
#medidas de la performance del modelo en la fase de validación
rmse3<-rmse(ValidSet$importe,predictions3)
rse3<-rse(ValidSet$importe,predictions3)
paste0("RSE is ",rse3)
paste0("RMSE is ",rmse3)

#####################################
#MODELO 4: Extreme Gradient Boosting##
#####################################
#FASE ENTRENAMIENTO
#definición del modelo
set.seed(12345)
modelo4<- train(importe ~., data =TrainSet, method = "xgbTree", trControl =
                trainControl("cv", number = 10),tuneGrid =
                expand.grid(max_depth = c(4,6,8), nrounds = c(30,40,60,80,100,200,300),
                        eta = c(0.1, 0.12, 0.14, 0.16, 0.18, 0.2) , gamma=0,colsample_bytree= 1,
                        min_child_weight= 1, subsample= 1),metric = "RMSE" )
modelo4
plot(modelo4)
#gráfico de importancia de las variables del modelo
plot(varImp(modelo4),main="Importancia de las variables del Modelo - XGB")
#medidas de la performance del modelo en la fase de entrenamiento
getTrainPerf(modelo4)
#FASE VALIDACIÓN
#Puesta en marcha del modelo
predictions4 <- predict(modelo4, ValidSet)
#medidas de la performance del modelo en la fase de validación
rmse4<-rmse(ValidSet$importe,predictions4)
rse4<-rse(ValidSet$importe,predictions4)
paste0("RSE is ",rse4)
paste0("RMSE is ",rmse4)


#########################################
##############MODELO 5: SVM#############
#########################################
#FASE ENTRENAMIENTO
#definición del modelo
set.seed(12345)
```

```
modelo5<- train(importe ~., data =TrainSet, method = "svmRadial", trControl =
                trainControl("cv", number = 10), metric = "RMSE" )
modelo5
plot(modelo5)
#gráfico de importancia de las variables del modelo
plot(varImp(modelo5),main="Importancia de las variables del Modelo - SVM")
#medidas de la performance del modelo en la fase de entrenamiento
getTrainPerf(modelo5)
#FASE VALIDACIÓN
#Puesta en marcha del modelo
predictions5 <- predict(modelo5, ValidSet)
#medidas de la performance del modelo en la fase de validación
rmse5<-rmse(ValidSet$importe,predictions5)
rse5<-rse(ValidSet$importe,predictions5)
paste0("RSE is ",rse5)
paste0("RMSE is ",rmse5)

########################################
##########DIBUJOS PREDICCIONES#########
########################################

#Modelo 0
ValidSet$prediction0 <- predictions0

ggplot(ValidSet, aes(x = prediction0, y = importe)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") + theme_test() +
  ggtitle("Predicción vs. Valores reales")

#Modelo 1
ValidSet$prediction1 <- predictions1

ggplot(ValidSet, aes(x = prediction1, y = importe)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") + theme_test() +
  ggtitle("Predicción vs. Valores reales")

#Modelo 2
ValidSet$prediction2 <- predictions2
```

```
ggplot(ValidSet, aes(x = prediction2, y = importe)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") + theme_test() +
  ggtitle("Predicción vs. Valores reales")

#Modelo 3
ValidSet$prediction3 <- predictions3

ggplot(ValidSet, aes(x = prediction3, y = importe)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") + theme_test() +
  ggtitle("Predicción vs. Valores reales GBM")

#Modelo 4
ValidSet$prediction4 <- predictions4

ggplot(ValidSet, aes(x = prediction4, y = importe)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") + theme_test() +
  ggtitle("Predicción vs. Valores reales XGB")


#Modelo 5
ValidSet$prediction5 <- predictions5

ggplot(ValidSet, aes(x = prediction5, y = importe)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") + theme_test() +
  ggtitle("Predicción vs. Valores reales SVM")
####################################
#########FORMULARIO SHINY#########
####################################
ui <- fluidPage(
 # * Input() functions

   sidebarPanel(selectInput(inputId = "bbdd", label = "Seleccione una base de datos", multiple = FALSE,
           choices = "siniestros salud"),
  selectInput(inputId = "model", label = "Seleccione el modelo", multiple = FALSE,
         choices =  c("regresión lineal", "cubist", "random forest", "gbm", "xgb", "svm")),
  selectInput(inputId = "accion", label = "Seleccione la acción a realizar", multiple = FALSE,
```

```
          choices =  c("Resumen del modelo", "Gráfico importancia de las variables", "Gráfico calidad de
la predicción", "Medidas de desempeño")),
  actionButton(inputId = "click", label = "Calcular")),


  # * Output() functions

    verbatimTextOutput("tabla"),
    plotOutput("importanciavariables"),


          )


server <- function(input, output) {

  #importar datos
  insurance <- read.csv("C:/…/TFM/insurance.csv")
  insurance$bmi30 <- ifelse(insurance$bmi>=30,"si","no")
  insurance$fumador <- as.numeric(insurance$fumador)
  insurance$fumador <- insurance$fumador -1
  insurance$edadcuadrado <- insurance$edad^2
  insurance$fumadorxbmi <-insurance$fumador*insurance$bmi

  #división de la muestra en Train y Valid

  train <- sample(nrow(insurance), 0.8*nrow(insurance), replace = FALSE)
  TrainSet <- insurance[train,]
  ValidSet <- insurance[-train,]
  summary(TrainSet)
  summary(ValidSet)


  modelo0 <- train(importe ~ hijos+edadcuadrado+bmi+fumador+bmi30+fumadorxbmi,
           data = TrainSet, method = "lm", trControl=trainControl(method = "cv", number = 5))
  modelo1 <- train(importe ~.,data = TrainSet, method = "cubist",
            trControl=trainControl(method = "cv", number = 5), grid =
              expand.grid(committees = c(1, 10, 50, 100),neighbors = c(0, 1, 5, 9)))
  set.seed(100)
  modelo2<- train(importe~., data =TrainSet,
           method = "cforest", trControl = trainControl("cv", number = 10))
```

```
set.seed(12345)
modelo3<- train(importe ~., data =TrainSet, method = "gbm", trControl =
            trainControl("cv", number = 10),tuneGrid =
            expand.grid(interaction.depth = c(4,6,8),n.trees=c(1,3,5,7,9)*100,
                  shrinkage = c(0.1) , n.minobsinnode = 20),metric = "RMSE" )
set.seed(12345)
modelo4<- train(importe ~., data =TrainSet, method = "xgbTree", trControl =
            trainControl("cv", number = 10),tuneGrid =
            expand.grid(max_depth = c(4,6,8), nrounds = c(30,40,60,80,100,200,300),
            eta = c(0.1, 0.12, 0.14, 0.16, 0.18, 0.2) , gamma=0,colsample_bytree= 1,
            min_child_weight= 1, subsample= 1),metric = "RMSE" )
set.seed(12345)
modelo5<- train(importe ~., data =TrainSet, method = "svmRadial", trControl =
            trainControl("cv", number = 10), metric = "RMSE" )
predictions0 <- predict(modelo0, ValidSet)
predictions1 <- predict(modelo1, ValidSet)
predictions2 <- predict(modelo2, ValidSet)
predictions3 <- predict(modelo3, ValidSet)
predictions4 <- predict(modelo4, ValidSet)
predictions5 <- predict(modelo5, ValidSet)

rmse0<-rmse(ValidSet$importe,predictions0)
rse0<-rse(ValidSet$importe,predictions0)
rmse1<-rmse(ValidSet$importe,predictions1)
rse1<-rse(ValidSet$importe,predictions1)
rmse2<-rmse(ValidSet$importe,predictions2)
rse2<-rse(ValidSet$importe,predictions2)
rmse3<-rmse(ValidSet$importe,predictions3)
rse3<-rse(ValidSet$importe,predictions3)
rmse4<-rmse(ValidSet$importe,predictions4)
rse4<-rse(ValidSet$importe,predictions4)
rmse5<-rmse(ValidSet$importe,predictions5)
rse5<-rse(ValidSet$importe,predictions5)


predictmatrix <- as.data.frame(ValidSet$importe)
predictmatrix$importe <- ValidSet$importe
predictmatrix$predictions1 <-predictions1
predictmatrix$predictions2 <-predictions2
predictmatrix$predictions3 <-predictions3
```

```
 predictmatrix$predictions4 <-predictions4
 predictmatrix$predictions5 <-predictions5

 observeEvent(input$click, {
  output$tabla <-renderPrint({
   if(input$accion == "Resumen del modelo"){
    if(input$model == "regresión lineal" ){summary(modelo0)}
    else if(input$model == "cubist" ){summary(modelo1)}
    else if(input$model == "random forest" ){summary(modelo2)}
    else if(input$model == "gbm" ){summary(modelo3)}
    else if(input$model == "xgb" ){summary(modelo4)}
    else if(input$model == "svm" ){summary(modelo5)}
   }
   else if (input$accion == "Medidas de desempeño"){
    if(input$model == "regresión lineal" ){paste0("RSE is ", rse0, " and RMSE is ", rmse0)}
    else if(input$model == "cubist" ){paste0("RSE is ", rse1, " and RMSE is ", rmse1)}
    else if(input$model == "random forest" ){paste0("RSE is ", rse2, " and RMSE is ", rmse2)}
    else if(input$model == "gbm" ){paste0("RSE is ", rse3, " and RMSE is ", rmse3)}
    else if(input$model == "xgb" ){paste0("RSE is ", rse4, " and RMSE is ", rmse4)}
    else if(input$model == "svm" ){paste0("RSE is ", rse5, " and RMSE is ", rmse5)}
   }
  })
 output$importanciavariables <- renderPlot({
  if(input$accion == "Gráfico importancia de las variables"){
  if(input$model == "regresión lineal" ){plot(varImp(modelo0), main = "Importancia variables Modelo
LM")}
  else if(input$model == "cubist" ){plot(varImp(modelo1), main = "Importancia variables Modelo
cubist")}
  else if(input$model == "random forest" ){plot(varImp(modelo2), main = "Importancia variables
Modelo RF")}
  else if(input$model == "gbm" ){plot(varImp(modelo3), main = "Importancia variables Modelo GBM")}
  else if(input$model == "xgb" ){plot(varImp(modelo4), main = "Importancia variables Modelo XGB")}
  else if(input$model == "svm" ){plot(varImp(modelo5), main = "Importancia variables Modelo SVM")}
  }
  else if(input$accion == "Gráfico calidad de la predicción"){
  if(input$model == "regresión lineal" ){ggplot(predictmatrix, aes(x = predictions0, y = importe)) +
     geom_point(color = "blue", alpha = 0.7) +
     geom_abline(color = "red") + theme_test() +
     ggtitle("Predicción LM vs. Valores reales")}
  else if(input$model == "cubist" ){ggplot(predictmatrix, aes(x = predictions1, y = importe)) +
     geom_point(color = "blue", alpha = 0.7) +
```

```
    geom_abline(color = "red") + theme_test() +
    ggtitle("Predicción Cubist vs. Valores reales")}
  else if(input$model == "random forest" ){ggplot(predictmatrix, aes(x = predictions2, y = importe)) +
    geom_point(color = "blue", alpha = 0.7) +
    geom_abline(color = "red") + theme_test() +
    ggtitle("Predicción RF vs. Valores reales")}
  else if(input$model == "gbm" ){ggplot(predictmatrix, aes(x = predictions3, y = importe)) +
    geom_point(color = "blue", alpha = 0.7) +
    geom_abline(color = "red") + theme_test() +
    ggtitle("Predicción GBM vs. Valores reales ")}
  else if(input$model == "xgb" ){ggplot(predictmatrix, aes(x = predictions4, y = importe)) +
    geom_point(color = "blue", alpha = 0.7) +
    geom_abline(color = "red") + theme_test() +
    ggtitle("Predicción XGB vs. Valores reales ")}
  else if(input$model == "svm" ){ggplot(predictmatrix, aes(x = predictions5, y = importe)) +
    geom_point(color = "blue", alpha = 0.7) +
    geom_abline(color = "red") + theme_test() +
    ggtitle("Predicción SVM vs. Valores reales ")}
  }
 })

 })

}

shinyApp(ui = ui, server = server)
```