uc3m | Universidad Carlos III de Madrid

Máster Universitario en Ciencias Actuariales y Financieras
2022-2023

Trabajo Fin de Máster

"Análisis de Técnicas Avanzadas de Pricing Actuarial en una Cartera de Autos: Integración de Variables Telemáticas a Modelos Clásicos"

Miguel Bueno Roda

Tutores:

José Miguel Rodríguez-Pardo

Jesús Ramón Simón del Potro

Universidad Carlos III de Madrid

Madrid, 2023

RESUMEN

La era del *big data* viene a cambiar las metodologías que se vienen aplicando en la tarificación de los seguros de autos. No solo en la modelización y cálculo de las primas, sino en la manera de trabajar y en las competencias que el mercado requiere de los actuarios. La inclusión de avanzadas metodologías de modelización estadística mediante técnicas de *machine learning* ya es un hecho y a esto se le une la aparición de la telemática mediante nuevos tipos de seguros como el caso de los seguros basados en el uso (UBI, por sus siglas en inglés).

Por ello, a partir de factores de riesgo tradicionales, se van a utilizar métodos de modelaje avanzados junto a técnicas de *machine learning* para luego adaptarlo a los modelos lineales generalizados que son los estándares en la industria. Después, se va a realizar un análisis del valor añadido que pueden aportar las variables telemáticas en el ajuste y la predicción frente al uso de factores clásicos a través de modelos *Gradient Boosting Machine*, así como el impacto que pueden tener en la mejora de los hábitos de conducción y el uso de los automóviles a partir de las primas.

Palabras Clave: Tarificación, Telemática, m*achine learning*, Modelización, Seguros.

ABSTRACT

The era of big data is changing the methodologies employed in auto insurance pricing. This transformation extends beyond the modeling and calculation of premiums, impacting the way actuaries work and the competencies demanded by the market. The inclusion of advanced statistical modeling methodologies using machine learning techniques is already a reality, complemented by the emergence of telematics through innovative insurance products such as Usage-Based Insurance (UBI).

Consequently, by leveraging traditional risk factors, advanced modeling methods and machine learning techniques will be employed, followed by adaptation to industry-standard generalized linear models. An analysis will then be conducted to assess the added value that telematics variables can contribute to fitting and prediction, compared to the use of classical factors, through Gradient Boosting Machine models. Additionally, the potential impact on improving driving habits and car usage, based on the influence of premiums, will be examined.

Keywords: Pricing: Telematics, Machine Learning, Modeling, Insurance.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN 1

1.1. Motivación. 1

1.2. Objetivos. 2

1.3. Resumen de resultados. 3

2. REVISIÓN DE LA LITERATURA 4

2.1. Transferencia de riesgos y los nuevos retos en el mercado asegurador. 4

2.2. Situación actual del mercado de autos en España. 5

2.3. Los datos y el uso de la telemática. 6

2.4. Adopción de los seguros UBI en Europa. 8

2.5. El proceso de integración y situación actual en el mercado nacional. 9

3. METODOLOGÍA 11

3.1. Métodos clásicos de tarificación. 11

3.1.1. Modelos Lineales Generalizados. (GLMs) 12

3.1.2. Modelos Aditivos Generalizados. (GAMs) 14

3.1.3. Criterios de información 15

3.2. Métodos de Machine Learning 16

3.2.1. Aprendizaje Supervisado 17

3.2.2. Aprendizaje No Supervisado 20

3.3. Interpretación en el aprendizaje supervisado. 21

3.3.1. Evaluación del rendimiento de los modelos. 22

3.3.2. Double lift charts. 24

4. CASO DE ESTUDIO. 25

4.1. Análisis estadístico descriptivo de las variables clásicas. 27

4.1.1. Variables Categóricas. 28

4.1.2. Variables Continuas. 28

4.1.3. Variables de respuesta. 31

4.2. Modelización flexible a partir de métodos GAMs. 39

4.2.1. Modelo de Frecuencia GAM. 41

4.2.2. Modelo de Severidad GAM 46

4.3. Traspaso del modelo GAM a uno GLM. 50

5. INCORPORACIÓN DE VARIABLES TELEMÁTICAS. 53

5.1. Análisis estadístico de las variables telemáticas. 53
5.1.1. Variables Pay-how-you-drive (PHYD). *53*
5.1.2. Variables Pay-as-you-drive (PAYD). *56*
5.2. Herramienta H2O en Rstudio 60
5.3. Modelos GLM Regularizados. 60
5.3.1. Modelo de Frecuencia. *61*
5.3.2. Modelo de Severidad *63*
5.4. Modelos Gradient Boosting Machine (GBM). 65
5.5. Partial dependence plots (pdp). 67
5.6. Valor añadido de las variables telemáticas. 68
5.6.1. Double lift charts. *69*
5.6.2. Diferencias relativas entre las primas según los grupos de riesgo. *70*
5.7. Comparativa de la deviance entre los modelos de frecuencia. 71
6. CONCLUSIONES. 73

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 - TIPOS DE VARIABLES CLÁSICAS Y TELEMÁTICAS 7
ILUSTRACIÓN 2 - ASEGURADORAS LIDERES DEL MERCADO EN TELEMÁTICA. 9
ILUSTRACIÓN 3 - COMPARACIÓN ENTRE MODELOS LINEALES SIMPLES Y MODELOS LINEALES GENERALIZADOS 13
ILUSTRACIÓN 4 - TIPOS DE MACHINE LEARNING 16
ILUSTRACIÓN 5 - DESCRIPCIÓN DE LA VARIABLE CREDIT SCORE 30
ILUSTRACIÓN 6 - ALGORITMO PARA AJUSTAR MODELO GAM DE FRECUENCIA 42
ILUSTRACIÓN 7 - SALIDA DEL MODELO GAM DE FRECUENCIA 43
ILUSTRACIÓN 8 - SALIDA DEL MODELO GAM DE SEVERIDAD 48

ÍNDICE DE GRÁFICOS

GRÁFICO 1 - EVALUACIÓN EN LA PRECISIÓN DE LA SIMULACIÓN DEL IMPORTE AGREGADO DE LOS SINIESTROS. 26
GRÁFICO 2 - VARIABLES CLÁSICAS CATEGÓRICAS 28
GRÁFICO 3 - VARIABLES CLÁSICAS CONTINUAS 29
GRÁFICO 4 - VARIABLES DE RESPUESTA 32
GRÁFICO 5 - ROOTOGRAMS DE POISSON Y BINOMIAL NEGATIVA PARA LA FRECUENCIA. 34
GRÁFICO 6 - VARIABLE VALOR DEL SINIESTRO ANTES (IZQUIERDA) Y DESPUÉS DEL ELIMINAR EL VALOR ATÍPICO (DERECHA). 36
GRÁFICO 7 - EVALUACIÓN DE LA CONSISTENCIA SEGÚN AUMENTAN EL NÚMERO DE REPETICIONES. 38
GRÁFICO 8 - CORRELOGRAMA VARIABLES CLÁSICAS CONTINUAS 40
GRÁFICO 9 - EFECTO DE LAS VARIABLES CONTINUAS EN EL MODELO DE FRECUENCIA GAM 44
GRÁFICO 10 - ÁRBOL DE REGRESIÓN DE LA VARIABLE EDAD DEL ASEGURADO PARA REALIZAR AGRUPACIONES EN FRECUENCIA. 45
GRÁFICO 11 - AGRUPACIONES DE LA EDAD DEL ASEGURADO EN FRECUENCIA. 46
GRÁFICO 12 - EFECTO DE LAS VARIABLES CONTINUAS EN EL MODELO DE SEVERIDAD GAM. 48
GRÁFICO 13 - ÁRBOL DE REGRESIÓN DE LA VARIABLE EDAD DEL ASEGURADO PARA REALIZAR AGRUPACIONES EN SEVERIDAD. 49
GRÁFICO 14 - AGRUPACIONES DE LA EDAD DEL ASEGURADO EN SEVERIDAD. 50
GRÁFICO 15 - COMPARATIVA DE LA PRIMA PURA PARA EL MODELO GLM Y GAM. 52
GRÁFICO 16 - NÚMERO DE ACELERONES SEGÚN SU INTENSIDAD. 54
GRÁFICO 17 - NÚMERO DE FRENAZOS SEGÚN SU INTENSIDAD. 54
GRÁFICO 18 - GIROS A DERECHAS E IZQUIERDAS SEGÚN SU INTENSIDAD. 55
GRÁFICO 19 - DÍAS QUE CONDUCE POR SEMANA, TIEMPO AL AÑO EN CARRETERA Y DISTANCIA RECORRIDA. 56
GRÁFICO 20 - TIEMPO CONDUCIENDO POR CADA DÍA DE LA SEMANA. 57
GRÁFICO 21 - TIEMPO CONDUCIENDO ENTRE FINDES DE SEMANA/ENTRE SEMANA. 58
GRÁFICO 22 - TIEMPO CONDUCIENDO ENTRE AM/PM. 59
GRÁFICO 23 - HORAS POR TRAYECTO. 59
GRÁFICO 24 - COMPARATIVA DE MODELOS GLM REGULARIZADOS DE FRECUENCIA TRAS VALIDACIÓN CRUZADA 62
GRÁFICO 25 - IMPORTANCIA DE PREDICTORES EN FRECUENCIA. 63
GRÁFICO 26 - COMPARATIVA DE MODELOS GLM REGULARIZADOS DE SEVERIDAD TRAS VALIDACIÓN CRUZADA. 64
GRÁFICO 27 - IMPORTANCIA DE PREDICTORES EN SEVERIDAD. 64
GRÁFICO 28 - IMPORTANCIA DE LOS PREDICTORES EN EL GBM DE FRECUENCIA. 66
GRÁFICO 29 - IMPORTANCIA DE LOS PREDICTORES EN EL GBM DE SEVERIDAD. 67
GRÁFICO 30 - PDP DE VARIABLES EN FRECUENCIA. 67

GRÁFICO 31 - PDP DE VARIABLES EN SEVERIDAD. 68
GRÁFICO 32 - DOUBLE LIFT CHART ENTRE LOS MODELOS GBM. 70
GRÁFICO 33 - COMPARATIVA DE LA DIFERENCIAS EN LA PRIMA POR GRUPOS DE RIESGO. 71
GRÁFICO 34 - COMPARATIVA DE LA DEVIANCE EN LOS MODELOS DE FRECUENCIA 72

ÍNDICE DE TABLAS

TABLA 1 – TOTAL DE VARIABLES DE LA BASE DE DATOS. 27
TABLA 2 – TEST DE DISPERSIÓN DE POISSON. 33
TABLA 3 – COMPARATIVA DE MÉTRICAS DE LOS MODELOS DE FRECUENCIA. 34
TABLA 4 – RESULTADO DE LA PRUEBA DE AJUSTE A UNA DISTRIBUCIÓN BINOMIAL NEGATIVA. 35
TABLA 5 – MÉTRICAS DE UN GLM BASE PARA DISTRIBUCIÓN LOG-NORMAL Y GAMMA. 37
TABLA 6 – SESGO Y MSE DE LOS MÉTODOS DE ESTIMACIÓN. 37
TABLA 7 – INTERVALOS DE CONFIANZA DE LA DISTRIBUCIÓN DE LOS PARÁMETROS ESTIMADOS. 39
TABLA 8 – MÉTRICAS DE LOS CINCO MEJORES MODELOS GAM DE FRECUENCIA ESTIMADOS CON LAS VARIABLES CLÁSICAS. 42
TABLA 9 – MÉTRICAS DE LOS CINCO MEJORES MODELOS GAM DE SEVERIDAD ESTIMADOS CON LAS VARIABLES CLÁSICAS. 47
TABLA 10 - MÉTRICAS AIC Y BIC DE LOS MODELOS GAM Y GLM. 62
TABLA 11 - MÉTRICAS DE LOS MODELOS DE FRECUENCIA PARA CADA ALPHA. 65
TABLA 12 - MODELOS OBTENIDOS GBM DE FRECUENCIA. 65
TABLA 13 - MODELOS GBM TRAS EL TUNNING DE HIPERPARÁMETROS. 69
TABLA 14 - COMPARATIVA DE LAS MÉTRICAS EN LOS DATOS DE ENTRENAMIENTO. 69

1. INTRODUCCIÓN

1.1. Motivación.

Trabajar en un departamento de tarificación o *pricing* dentro de una aseguradora que opera con productos de autos significa tener un día a día inmerso en la estadística, el análisis de datos y el modelaje predictivo. A pesar de que pasen los años, los estándares para estimar la frecuencia y la severidad se mantienen con las mismas técnicas y procesos de modelización tradicionales.

La motivación radica en la necesidad de adaptarse y comprender estos estándares en el ámbito de la tarificación de seguros de autos, así como investigar nuevas metodologías que con el paso del tiempo se incorporan en el mercado. A medida que la industria se moderniza, es fundamental estar al tanto de los avances y enfoques que surgen para manejar y comprender estos nuevos métodos.

Además, en la era del dato y el creciente acceso a grandes cantidades de información, se abre un mundo de oportunidades para añadir y mejorar a las actuales técnicas de tarificación. La incorporación de variables telemáticas, como pueden ser los datos recopilados a través de dispositivos de seguimiento instalados en los vehículos, ofrece una nueva fuente de información alternativa. La adopción de la telemática enfocada a este ámbito dentro del sistema europeo ha entrado con mayor relevancia en Italia, donde los seguros de autos que se basan en estas nuevas variables están en continuo crecimiento. Sin embargo, tras las últimas encuestas realizadas por Mordor Intelligence (s.f), dentro de los países con mayor interés para implementar a sus pólizas factores telemáticos estaría España con un 70% de los conductores afirmando estar definitiva o probablemente interesados.

Por último, en la actualidad, la figura del actuario se está fusionando cada vez más con la del científico de datos. Es crucial adquirir habilidades en el manejo de herramientas técnicas de codificación como pueden ser *RStudio o Python*, que permiten realizar análisis avanzados de datos y modelización estadística. Esta capacidad de combinar el conocimiento actuarial con las técnicas y herramientas de ciencia de datos se ha vuelto fundamental para abordar los actuales desafíos y oportunidades que surgen en la industria aseguradora.

1.2. Objetivos.

El objetivo de este trabajo es aplicar bajo la herramienta de programación *Rstudio*, metodologías clásicas de tarificación en el contexto de seguros de autos y aprender nuevos enfoques más avanzados que permitan mejorar la precisión y la predicción de los modelos actuariales de no vida. También, se van a utilizar metodologías avanzadas de *machine learning* para estudiar el posible impacto de la telemática en la modelización aplicada a los seguros de autos junto al valor añadido que pueden aportar tanto para el asegurado como para la compañía.

En primer lugar, se cambiará el enfoque estándar empezando con una tarificación basada en el uso de los Modelos Aditivos Generalizados (GAM) junto a técnicas de árboles de regresión para obtener las agrupaciones óptimas en las variables continuas. Estos modelos GAM ofrecen una mayor flexibilidad en la modelización al permitir la inclusión de términos no lineales y suavizados en la relación entre las variables predictoras y la variable respuesta.

Después, se traspasan los resultados con las agrupaciones a los clásicos Modelos Lineales Generalizados (GLM), que son los más utilizados en seguros gracias a la interpretabilidad y menor complejidad respecto a otros métodos, para modelar y analizar las variables tradicionales usadas en la tarificación. El objetivo es evaluar la capacidad predictiva de ambos modelos y comprender cómo influyen estas variables en la determinación de las primas de seguros.

Por otro lado, se emplearán técnicas más avanzadas de *machine learning*, específicamente los *Gradient Boosting Machines* (GBM), para desarrollar modelos más complejos y precisos. Estas técnicas se utilizarán para el estudio y comparativa sobre la incorporación de variables telemáticas a los tradicionales factores de riesgo que han surgido con la era del *big data* y la digitalización, proporcionando información detallada sobre los hábitos y estilos de conducción. Por último, se investigará cómo la inclusión de estas variables telemáticas puede mejorar la predicción de la frecuencia de los siniestros, y cómo pueden utilizarse para incentivar cambios en los comportamientos de los conductores a partir de descuentos sobre las primas.

1.3. Resumen de resultados.

Durante la primera parte del proyecto donde se ha llevado a cabo la modelización de los factores clásicos a partir de los GAM junto a las agrupaciones mediante árboles de regresión para el posterior traspaso a los estándares en la tarificación, se ha visto como la aplicación de modelos GAM permiten capturar relaciones no lineales entre las variables predictoras y las variables de respuesta. Tras el traspaso de los resultados obtenidos en los GAM junto a un proceso de agrupación de las variables continuas mediante árboles de regresión a modelos GLM, se han obtenido resultados similares en el cálculo de la prima pura siendo la resultante por el primer método algo mayor que por los modelos lineales. Con los resultados de los criterios de información, se puede intuir la comparativa entre flexibilidad y simplicidad en el proceso de modelado. Los GAM son una posible mejor opción para lograr flexibilidad, ya que penalizan menos la complejidad como resultado de un AIC más bajo en frecuencia y cercano en Severidad. Por otro lado, los GLMs se elegirían para obtener resultados más simples, dado que penalizan más la complejidad obteniendo valores del BIC mejores.

Después, durante la comparativa a través de la inclusión de variables telemáticas en el proceso de tarificación, los resultados han mostrado que las variables telemáticas pueden influir tanto en un mejor ajuste, como en las diferencias existentes entre grupos de riesgo de los asegurados. Mediante el uso del gráfico *double lift chart* se observa como la inclusión de estos factores mejora el ajuste a la hora de calcular la prima pura, acercándose más a los valores observados de la cartera.

En cuanto a los grupos de riesgo, al darles un valor según la frecuencia que se ha modelizado en la muestra *out of sample*, se saca la conclusión de que los grupos de riesgo mas bajos tienen unas primas menores que los estimados solo con el uso de regresores clásicos, mientras que, en los conductores con niveles más altos de riesgo, son mayores. Estos resultados pueden ayudar a atraer a los perfiles de riesgo más bajos ya que garantizas menores pagos por sus coberturas, y que a los peores perfiles se les dé un incentivo a mejorar sus hábitos para obtener descuentos en sus primas.

Por último, se ha realizado una comparativa con los resultados de la *deviance* que indica que los modelos que la minimizan son los obtenidos a partir de los GBM siendo el que incluye los factores telemáticos el que mejor ajusta. En cuanto a los estimados son con los regresores clásicos, tanto el GAM como el GLM dan resultados similares pero peores que el GBM.

2. REVISIÓN DE LA LITERATURA

2.1. Transferencia de riesgos y los nuevos retos en el mercado asegurador.

Numerosos estudios introducen el funcionamiento del mercado en el sector de los seguros, así como la transferencia de riesgos que de manera característica se produce en este ámbito. Esto implica que la aseguradora recibe una prima fija y se compromete a pagar futuras pérdidas en forma de siniestros a los asegurados. Los pagos futuros son inciertos, por lo que la compañía vende un producto cuyo coste es desconocido en el momento de la venta. Esto se conoce como el ciclo de producción inversa y hace que sea de vital importancia para la empresa, evaluar adecuadamente el riesgo de las posibles cuantías de los siniestros de los asegurados en función de la información disponible.

La gestión de los riesgos a los que puede estar expuesta esta industria puede variar en función de cómo cambien los patrones y comportamientos tanto de la sociedad, como de los mercados tecnológicos, financieros o regulatorios.

El análisis continuado de los anteriores mercados junto a la cantidad de datos con los que trabajan las compañías desempeña un papel esencial para manejar de una forma más precisa y competitiva los riesgos a los que están expuestos. Los últimos retos que se afrontan en este sector se enmarcan dentro de los cambios medioambientales y sanitarios, así como de la convulsa situación económica y social debido a puntuales acontecimientos tales como la guerra de Ucrania o la reciente pandemia.

En cuanto al mercado español, se ha visto afectado por unas fuertes subidas de tipos de interés tras muchos años a unos niveles muy bajos y a lo que se le ha añadido un periodo de alta inflación. Ambas situaciones provocan aumentos en los gastos generales de las compañías y en los costes medios que generan los siniestros.

La repercusión de tales subidas junto a la alta competitividad que existe en este mercado afecta de forma directa a las primas de los asegurados, que se ven incrementadas para poder mantener la estabilidad económico-financiera y el posicionamiento de las aseguradoras. Si a esto se le incluyen situaciones nuevas que se han vivido en los últimos años como importantes ciberataques, riesgos relacionados con el cambio climático o riesgos pandémicos, se produce un aumento sustancial en la asignación presupuestaria para minimizar este tipo de eventos, así como en la implementación de programas de concienciación y formación a nivel individual y sectorial.

Además, hay una mayor consciencia entre los consumidores respecto a todos los temas relacionados con la sostenibilidad. Esta sensibilidad, sumada a los requisitos regulatorios, está obligando a todas las entidades a comenzar una planificación entorno al diseño de su negocio incluyendo por ejemplo variables que recojan estos factores. Sería el caso de los criterios ESG (*Environmental, Social and Governance*) que se refieren a índices ambientales, sociales y de gobierno corporativo que se tienen en cuenta a la hora de invertir en una empresa.

A pesar de los retos actuales a los que se enfrentan las compañías aseguradoras, el sector de los seguros creció, a finales del año 2022, un 4,65% (Cinco Días, 2023).

2.2. Situación actual del mercado de autos en España.

Según la Asociación Europea de Fabricantes de Automóviles (ACEA, por sus siglas en inglés), el número de matriculaciones de turismos en la Unión Europea disminuyó un 20,5% en marzo de 2022, con 844.187 unidades vendidas. La producción de automóviles se ha visto perjudicada por las persistentes interrupciones en la cadena de suministros, motivada por la guerra entre Rusia y Ucrania, la cual supuso que la mayoría de los países de la región experimentaran descensos de hasta dos dígitos en sus ventas. Algunos ejemplos de tales caídas serían los siguientes: España (-30,2%), Italia (-29,7%), Francia (-19,5%) y Alemania (-17,5%) (Mordor Intelligence, s.f).

Como se ha comentado anteriormente, las primas se han visto afectadas de manera directa con grandes subidas y por la tendencia que se espera en cuanto a la inflación e inestabilidad global, se puede anticipar una continuidad en el aumento de los precios.

Otro de los hechos que ha afectado a las empresas que trabajan con los seguros de autos es el incremento de la siniestralidad por al aumento de la movilidad y el envejecimiento del parque móvil. Esto se observa en cifras como el incremento en un 3% de la movilidad en 2022 respecto a 2019 o la cifra de 1,4 coches vendidos con más de 10 años por cada coche nuevo matriculado (Cinco Días, 2023). Además, nuevas tendencias que modifican los hábitos en el transporte como el caso del *carsharing* o el impulso de la movilidad eléctrica mediante patinetes o bicis en ámbitos sobre todo urbanos ya generaron una fuga del 10% en clientes.

El resultado de todos estos factores da lugar a que los costes de los seguros de automóviles aumenten significativamente en solo un año, con un aumento total de 45,7 euros o un

7,86%. Según el comparador Kelisto, asegurar un vehículo ahora cuesta 627,2 euros en comparación con los 581,5 euros del año anterior. En particular, los seguros a todo riesgo han experimentado un aumento cercano al 11%, lo que se traduce en un coste adicional de 163 euros. Por otro lado, en el caso del seguro a terceros ampliado - la opción más popular en España - el aumento fue del 7,33%, lo que equivale a un coste adicional de 29,10 euros. Finalmente, el seguro a terceros también ha experimentado un aumento del 7,66%, lo que supone un coste adicional de 24,4 euros (La Razón, 2023).

2.3. Los datos y el uso de la telemática.

No todo son malas noticias para los seguros de autos, el constante desarrollo económico junto a la era del dato en la que nos encontramos vienen a proporcionar nuevos retos a los que adaptarse a la hora de tarificar y medir los riesgos de las pólizas.

Una de estas tecnologías que viene a revolucionar la manera de hacer *pricing* es la telemática. De forma genérica este término indica la unión de dos ciencias diferentes, las telecomunicaciones y la informática. En la actualidad uno de sus principales usos se da en los vehículos de flotas comerciales con el que se es capaz de tener una visión conjunta del estado, la rentabilidad y la productividad de toda la flota.

La telemática se utiliza en el seguro de autos para rastrear, almacenar y enviar datos relacionados con la conducción. Esta información ayuda a determinar los hábitos de conducción y los seguros de vehículos adecuados. Funciona de la siguiente manera: se incorpora a los coches un dispositivo que envía, recibe y almacena datos de telemetría, el dispositivo recopila junto con los datos GPS, información detallada del vehículo y envía todo a un servidor centralizado mediante GPRS (servicio de paquetes de radio), 4G u otra tecnología de comunicación móvil o satelital. La principal forma de acceder y utilizar esta tecnología es basándose en el OBD-II, en teléfonos inteligentes, híbridos y en cajas negras instaladas en el interior del vehículo. Esta cantidad de información que se obtiene se puede usar para ofrecer descuentos a los conductores que tengan un mejor comportamiento a la hora de conducir, para penalizar a los peores o para ofrecer servicios de asesoramiento de acuerdo a la manera de conducir.

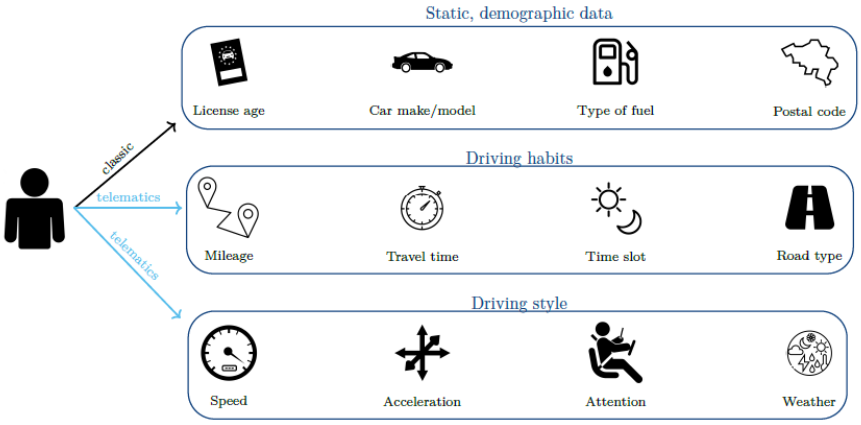
También ha permitido introducir nuevas modalidades de seguros como los UBI (*Usage-based Insurance*). Es el principal seguro que surge tras la implementación de la telemática, se basa en los datos que recibe de un dispositivo en forma de caja negra situado

en el coche que monitoriza y reporta los hábitos y comportamientos de conducción, así como cuando y cuanto conducen sus asegurados. Analizando los datos recibidos, la aseguradora puede ajustar de una manera más apropiada la prima para cada cliente de manera individualizada.

Actualmente, se están implementando en países europeos como es el caso de Italia principalmente, o Estados Unidos. Los dos principales tipos de factores que se relacionan con los seguros UBI son: P*ay-as-you-drive (PAYD)* y P*ay-how-you-drive (PHYD)*.

- P*ay-as-you-drive (PAYD)*. Mide los hábitos de conducción del asegurado. Estos incluyen por ejemplo el total de kilómetros, las horas de conducción, el momento del día en el que cogen el coche o los tipos de carreteras por los que transitan.
- P*ay-how-you-drive (PHYD)*. Mide el estilo de conducción del asegurado. Ejemplos de variables PHYD podrían ser la velocidad a la que conduces, las aceleraciones que haces, la intensidad en los giros de las curvas a izquierdas o a derechas e incluso la intensidad de los frenazos.

ILUSTRACIÓN 1 - TIPOS DE VARIABLES CLÁSICAS Y TELEMÁTICAS



Fuente: (Henckaerts, R, 2021).

Los modelos actuariales para tarificar se encuentran ya muy ajustados y con poco margen, por lo que añadir a las clásicas bases de datos nuevas variables telemáticas puede ofrecer beneficios significativos para las compañías de seguros, los clientes y la sociedad en general. La cuestión es la monitorización del comportamiento sobre la conducción permitiendo reducir la información asimétrica entre la aseguradora y sus asegurados, mitigando así los problemas de riesgo moral y selección adversa.

En cuanto a la parte negativa, afecta a algunos aspectos relacionados con la privacidad. En el caso de clientes individuales, deben de aceptar las condiciones de privacidad que se incluyen en la póliza para registrar los datos sobre su forma de conducir, pero en el caso de flotas, los trabajadores que utilicen esos vehículos podrían están monitorizados y vigilados sin un consentimiento expreso fuera del alcance contractual. A través de un método de codificación y anonimato se podría proteger más la privacidad y aprovechar las ventajas de esta tecnología garantizando una conducción más segura.

2.4. Adopción de los seguros UBI en Europa.

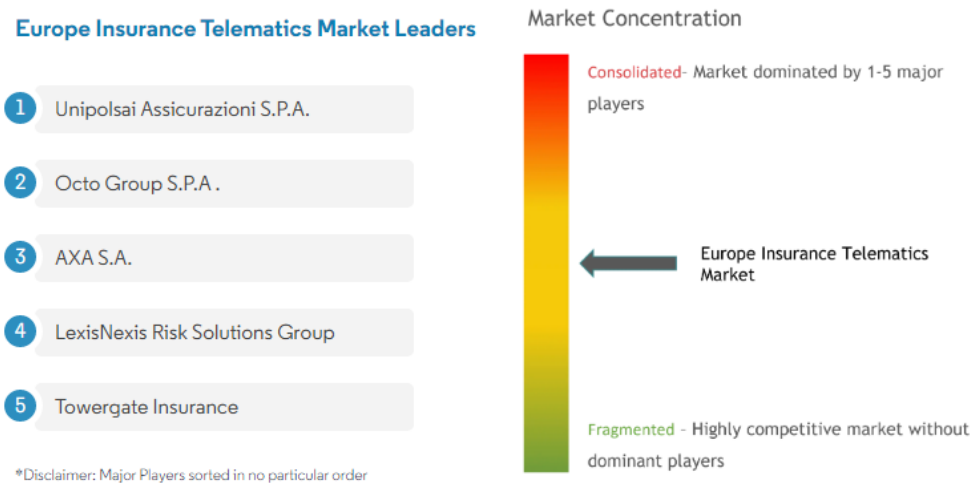
Según Mordor Intelligence (s.f), la integración del seguro UBI es la última innovación en cuanto a los seguros de autos, ha ayudado a fomentar estilos de conducción más seguros entre los conductores, permitiéndoles ahorrar en las primas de seguros al mismo tiempo. El dispositivo de monitoreo constante disponible en la actualidad, gracias a los avances en telemática ayuda a analizar con precisión a los conductores y a proceder a asegurar sus vehículos. Además, según una investigación realizada por Towers Watson, la mayoría de los conductores en los seis mayores mercados de seguros de automóviles de Europa están ansiosos por adoptar soluciones basadas en la telemática. Los países con mayor interés son Italia y España, donde casi el 70% de los conductores afirmaron estar definitiva o probablemente interesados en obtener una póliza con factores telemáticos. En todos los seis países europeos, el 55% de los conductores estaban interesados en el seguro de telemática. En los Estados Unidos, donde la telemática ya es un producto de mercado masivo, la proporción correspondiente fue del 50%.

Como también informa Mordor Intelligence (s.f), durante marzo de 2022, Ford Motor Company y Verisk se unieron en Europa y el Reino Unido para facilitar datos de vehículos conectados “listos para seguros”, que ayudarán a los aseguradores a crear programas basados en el uso. Este servicio pronto estará disponible en Alemania, Francia, Italia, España y el Reino Unido. Según Verisk, este acuerdo proporcionará información más detallada a las compañías de seguros sobre los riesgos y forma de conducción de cada usuario, lo que permitirá una personalización de las pólizas.

Una vez que un conductor lo autorice, los aseguradores podrán acceder y analizar las métricas estandarizadas que Verisk obtiene de los vehículos conectados a través de su plataforma *Data Insight Hub*. Ya hay compañías extranjeras de vehículos comerciales

que buscan expandirse en Europa y ofrecen descuentos de hasta un 20% en las primas a los conductores que tienen un comportamiento más 'seguro'.

ILUSTRACIÓN 2 - ASEGURADORAS LIDERES DEL MERCADO EN TELEMÁTICA.



Fuente: Mordor Intelligence (s.f).

La figura anterior indica que la adopción de la tecnología se encuentra en proceso de consolidación, pero en la actualidad no ha llegado aún a ese paso. Italia es donde se encuentran las empresas líderes de este mercado, pero también multinacionales como AXA o LexixNexis lideran este ranking. En el análisis de mercado que realiza Mordor Intelligence (s.f), también se muestra como la tasa de crecimiento anual compuesta "CAGR" que expresa el crecimiento de la telemática en este caso, en el mercado europeo, estaría entorno a un crecimiento del 19% para el 2028.

Otro estudio realizado por la consultora independiente Berg Insight (2022) afirma que el mercado telemático de los seguros se encuentra actualmente en una fase de fuerte crecimiento que se espera se acelere en los próximos años. Estima que el número de pólizas de seguros telemáticos en vigor en Europa alcance los 35,1 millones en 2025.

2.5. El proceso de integración y situación actual en el mercado nacional.

En el caso del mercado español, a pesar de los resultados obtenidos en la investigación por Towers Warson, la adopción está siendo lenta. La principal causa son las características del mercado nacional con primas muy ajustadas y con el riesgo

mutualizado donde los costes que se tendrían que añadir para la instalación y mantenimiento del dispositivo dificultan la obtención de un margen de beneficios que incite su implantación.

Este producto innovador parece resultar atractivo para el mercado español como se ha comentado en el apartado anterior, siendo uno de los países con mayor interés dentro de Europa. El motivo de este interés podría ser debido a que se pueden ofrecer primas más bajas a aquellos que realizan una conducción de bajo riesgo y, para los de más riesgo (con primas más altas), les permite reducir el coste de su seguro mejorando sus "malos" hábitos de conducción.

En cuanto a la situación actual, en cambio, sigue la adopción lenta y todavía no está al nivel de algunos países vecinos. Si se compara el número de pólizas registradas, España se encuentra junto a Austria y Francia, en torno a las 50.000 y 100.000, mientras que Italia cuenta con unas 4.3 millones, seguida de Reino Unido con 540.000 (Mazorco, 2022).

3. METODOLOGÍA

3.1. Métodos clásicos de tarificación.

Una cartera de cualquier empresa que ofrezca un seguro de autos engloba miles de pólizas a las que ofrecer protección para el riesgo al que están suscritos. La cuestión es que hay heterogeneidad dentro de cada cartera y, por ello, hay que tarificar teniendo en cuenta los diferentes perfiles de riesgo para establecer las primas adecuadas. Si la prima fuese igual para todos, ocurriría que los buenos perfiles se irían a otra compañía que les ofreciese pagar una prima menor, quedándose solo con los malos perfiles que pagarían una prima mucho más baja en relación con su alto riesgo.

Para eludir ese problema, las bases de datos contienen factores de riesgo que permiten agrupar a los asegurados por clases según perfiles de riesgo similares. Estos factores de riesgo suelen incluir variables categóricas, continuas o espaciales. Los dos últimos tipos se suelen transformar en variables categóricas agrupándolas en varios niveles para facilitar la modelización.

Dentro del ramo de autos, la metodología más extendida en las compañías son los modelos lineales generalizados o GLMs (*Generalized Linear Models*). Se trata de modelos predictivos que se desarrollan a partir de datos históricos para predecir el futuro número de siniestros conocido como frecuencia, y la cantidad a pagar por los siniestros que se denomina severidad.

La frecuencia se mide por el número de siniestros por unidad de exposición al riesgo del asegurado que, en no-vida, normalmente es de máximo 1 año (Lo que dura la póliza).

$$Frecuencia = numero\ de\ siniestro\ /\ exposicion$$

Por lo contrario, la severidad engloba el coste total de los siniestros de un asegurado entre el total de siniestros que ha tenido en el periodo de exposición.

$$Severidad = Coste\ total\ /\ numero\ de\ siniestro$$

Se suele asumir independencia entre ambos componentes y, a partir del producto de ellos, se obtiene la prima pura, que cubre la parte relacionada puramente con el riesgo al que se está expuesto, en este caso, el de sufrir un accidente.

$$Prima\ pura = severidad * frecuencia$$

El uso de esta técnica predictiva de modelaje proporciona una serie de ventajas que han facilitado su implementación, estableciéndose como el método estándar desde hace ya muchos años y, a pesar de la aparición de nuevas metodologías más avanzadas como algoritmos de *machine learning* (ML) o de redes neuronales, se mantiene como la más usada por los departamentos de tarificación en las principales compañías.

Las ventajas que tiene, van desde la mayor facilidad de interpretación e implementación dando unos buenos resultados predictivos, como la transparencia a la hora de proporcionar al regulador los modelos para su comprobación y análisis.

3.1.1. Modelos Lineales Generalizados. (GLMs)

El método de modelaje estándar en la industria para predecir frecuencia y severidad son los modelos lineales generalizados. Estos son una extensión del modelo de regresión lineal general donde, a partir de una combinación lineal de variables independientes, se pretende predecir el resultado de la variable dependiente.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_p x_p + \epsilon$$

$$y = \beta_0 + \sum_{i=1} \beta_i x_i + \epsilon_i$$

Las principales limitaciones que establecen algunos de sus supuestos son:

- La variable dependiente sigue una distribución Normal, $Yi \sim N\ (\mu, \sigma 2)$
- Los regresores no estén altamente correlacionados entre sí.
- La relación entre los factores y la variable dependiente debe ser lineal.
- Los residuos del modelo deben seguir una distribución $\varepsilon_i \sim N\ (0, \sigma^2)$ y la varianza debe ser constante.

Sin embargo, los supuestos que se comentan anteriormente no se cumplen en muchas ocasiones por la naturaleza de los datos e información que se tiene. La solución a esto es usar la extensión GLM que, gracias a una serie de condiciones, permite ajustarse mejor a los datos para el modelaje. Sus principales partes son:

- El componente aleatorio que se encuentra en la variable dependiente Y, y que sigue una distribución de la familia exponencial. $\mu_i = E(Y_i)$

- El vector con las variables dependientes o también llamado predictor lineal

$$n_i = \sum j * \beta_j * x_j$$

- La función de enlace o *link function* relaciona la esperanza matemática de Y con el predictor lineal $g(\mu_i) = n_i$

$$g(E(Y|X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots + \beta_p X_p$$

En los GLMs la variable respuesta permite que siga diferentes distribuciones, comúnmente la severidad suele seguir una Gamma o Log-normal, mientras que la frecuencia sigue una Poisson o una Binomial Negativa. Además, acepta que la varianza no sea constante y que se realicen interacciones entre los factores.

ILUSTRACIÓN 3 - COMPARACIÓN ENTRE MODELOS LINEALES SIMPLES Y MODELOS LINEALES GENERALIZADOS

Modelo Lineal (ML)	Modelo Lineal Generalizado (MLG)
$\mathbf{y}_i = \sum_j \boldsymbol{\beta}_j \mathbf{X}_{ij} + \varepsilon_i$	$\mathbf{y}_i = \sum_j \boldsymbol{\beta}_j \mathbf{X}_{ij} + \varepsilon_i$
$\mu_i = E(Y_i)$	$\mu_i = E(Y_i)$
$\boldsymbol{\eta}_i = \sum_j \boldsymbol{\beta}_j \mathbf{X}_{ij}$	$\boldsymbol{\eta}_i = \sum_j \boldsymbol{\beta}_j \mathbf{X}_{ij}$
$\eta_i = \mu_i$	$\eta_i = g(\mu_i)$
$\mathbf{y}_i$: vector de la variable respuesta,	
$\mathbf{X}_{ij}$: matriz de variables predictoras y covariables	
$\boldsymbol{\beta}_j$: vector de parámetros	
$\boldsymbol{\eta}_i$: vector del predictor lineal	

Fuente: López-González y cols., 2002[a]

En cuanto a la estimación de los parámetros del vector de variables dependientes, se utiliza el método de máxima verosimilitud. Por otro lado, si se pretende analizar el ajuste que el modelo ha obtenido, el estadístico Chi-cuadrado es un buen método comúnmente utilizado.

Una peculiaridad de estos modelos es la inclusión de un término adicional llamado offset. Se utiliza para equilibrar el valor esperado de la variable respuesta en función de una variable conocida y no aleatoria que no se considera como una variable explicativa dentro del modelo. En el caso de la tarificación en autos, el offset facilita que el valor de la variable respuesta se ajuste a la exposición de riesgo de los asegurados, lo que ayuda a reflejar mejor el riesgo real que representa cada asegurado en la compañía. Este término se añade a la fórmula del predictor lineal con un coeficiente fijo de 1 haciendo que el efecto se modele no de forma aditiva, sino proporcional.

3.1.2. Modelos Aditivos Generalizados. (GAMs)

Los modelos GAMs son una herramienta de predicción que se basan en una extensión de los GLMs introducida por Trevor Hastie y Robert Tibshirani en 1986 (Hastie y Tibshirani, 1986).

Puede ocurrir que el efecto de la variable explicativa sobre la dependiente tenga una forma desconocida y, por tanto, mediante el uso del Modelo Aditivo Generalizado, se pueden usar funciones de cualquier forma dando mayor flexibilidad a la hora del modelaje. Así, permiten generar regresiones incorporando formas no lineales, al contrario de lo que ocurría con los modelos GLM.

Para las variables que se incluyen en los modelos de autos, puede resultar interesante incluir funciones que suavicen los factores de riesgo continuos que se relacionan con la variable respuesta de forma no lineal. En la práctica, sin embargo, los actuarios tienden a preferir la simplicidad de los GLMs con factores de riesgo categóricos sobre GAMs con efectos suaves, porque los modelos de tarificación han de ser interpretables, intuitivos y fáciles de explicar, tanto a clientes como a reguladores.

La forma matemática que tienen los GAM es la siguiente.

$$n_i = g(\mu_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}^d + \sum_{j=1}^{q} f_j(x_{ij}^c) + \sum_{j=1}^{r} f_j(x_{ij}^s, y_{ij}^s)$$

Donde μ_i es la media de una variable de respuesta con una distribución de la familia exponencial y $g(.)$ es la función de enlace. Las variables x_{ij}^d representarían los factores de riesgo categóricos como se representarían en el marco de los GLMs, esto es, seleccionando un nivel base para modelar las diferencias entre los demás niveles y este

nivel de referencia. Por otro lado, el coeficiente β_j capta el efecto de la variable x_j^d. La extensión del modelo GAM se observa en la inclusión de las funciones de suavizado para las variables continuas x_{ij}^c y las posibles interacciones de estas $f_j(x_{ij}^s, y_{ij}^s)$.

La estimación de un modelo GAM se realiza mediante un procedimiento iterativo llamado "método de suavizado penalizado". Este método consiste en la optimización de una función de pérdida que combina la verosimilitud del modelo con un término de penalización que controla la complejidad del modelo. La optimización se realiza mediante un algoritmo de optimización numérica, como el algoritmo de optimización de descenso gradiente.

En la estimación de un modelo GAM, se utiliza una función de suavizado para cada variable predictiva no lineal. Estas funciones de suavizado pueden ser, por ejemplo, funciones polinómicas o funciones spline.

3.1.3. Criterios de información

Cuando se están realizando labores de tarificación, lo normal es trabajar con diferentes modelos para luego compararlos y obtener el que mejor se ajusta a los datos. Para ver la capacidad predictiva y poder seleccionar algún modelo hay diferentes métodos de valoración. Los que más se utilizan en el mercado actual son el AIC (*Akaike Information Criterium*) y el BIC (*Bayesian Information Criterium*). Ambos tienen en cuenta la bondad del ajuste y la complejidad del modelo.

- Akaike (AIC)

El AIC fue propuesto por Akaike (1974) como un estimador insesgado asintótico de la información de Kullback-Leibler esperada, entre un modelo candidato ajustado y el verdadero modelo.

$$AIC = -2 * \log \mathcal{L} + 2 * EDF$$

Donde $log\mathcal{L}$ la función de máxima verosimilitud del modelo, n es el número de observaciones y EDF son los grados de libertad que corresponden al número de parámetros incluidos en el modelo.

Proporciona una medida de selección entre modelos, donde el mejor modelo será el que maximiza la verosimilitud esperada. Como el criterio de Akaike establece la función de

verosimilitud en negativo, el mejor modelo será aquel que tenga un criterio de Akaike menor.

- BIC

El criterio BIC es propuesto por Schwarz. Al igual que con el AIC, el mejor modelo será aquel que tenga un BIC menor ya que será el que maximice la función de verosimilitud.

$$BIC = -2 * \log \mathcal{L} + \log(n) * EDF$$

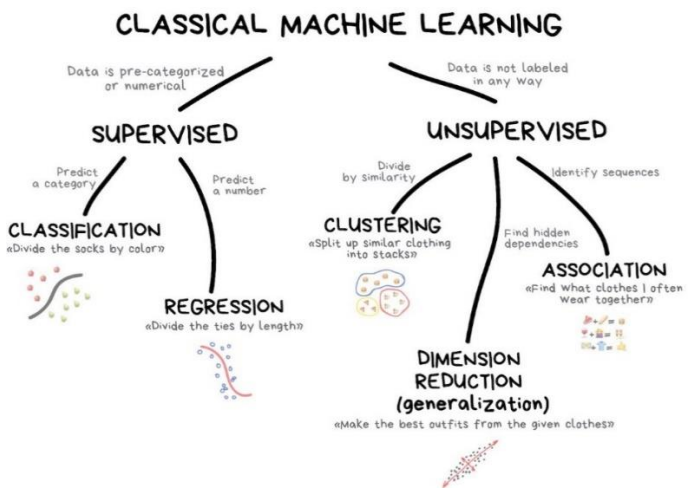
Este tiene una penalización más severa por lo que favorece a los modelos que son menos complejos lo cual va de la mano con el principio de parsimonia donde buenos modelos cuanto más simples mejor.

3.2. Métodos de Machine Learning.

El *machine learning* o aprendizaje automático es una rama perteneciente a la inteligencia artificial que se basa en la generación de aprendizaje, tomando como base la información que se extrae de los datos analizados por los ordenadores para crear modelos predictivos eficientes.

Se pueden clasificar en tres grupos diferentes: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje reforzado. El estudio se centra en la primera categoría de las que se muestran en la siguiente ilustración.

ILUSTRACIÓN 4 - TIPOS DE MACHINE LEARNING



Fuente: (Katrien, A 2022)

Una parte importante dentro de los métodos de ML es la división de las observaciones entre la parte de entrenamiento, la de validación y la de test. Estas divisiones dependen del total de datos que se obtengan, por lo que no siempre se pueden hacer tres divisiones ya que puede ocurrir el problema de quedarse con una muestra muy pequeña en algún segmento. Cuando ocurre eso y el tamaño de los datos no es lo suficientemente grande, se puede dividir solo entre entrenamiento y test.

De forma genérica, se utiliza el conjunto de entrenamiento para ajustar el modelo, el de validación para encontrar los mejores hiperparámetros (*model tuning*) y el de test para estimar el error que comete el modelo al predecir nuevos datos. En el caso de solo tener training y test, se utilizará el método de validación cruzada (*cross validation*) donde la parte de entrenamiento se utiliza para establecer el *model tuning* junto a la estimación de los modelos, y se deja la parte de test para el análisis de los errores. Una división estándar del conjunto de datos es de 70%-15%-15% o 70%-30% en caso de muestras menores, pero siempre como parte principal la de entrenamiento del modelo.

El problema que surge en la validación cruzada es que requiere ajustar el modelo repetidas veces, suponiendo un coste computacional muy alto. Cuando se trabaja con *big data* no suele ser factible aplicar validación cruzada, además, al disponer de tantos datos, un conjunto de validación único suele ser suficiente para obtener buenas estimaciones del del modelo.

3.2.1. Aprendizaje Supervisado

El primer método de aprendizaje supervisado incluye desde modelos predictivos clásicos, pasando por árboles de decisión, hasta las redes neuronales. El aprendizaje que utilizan se basa en el análisis del pasado para poder anticipar lo sucedido o reproducir la respuesta (Guillen, M., & Pesantez-Narvaez, J., 2018).

Los usos más frecuentes son los problemas de clasificación, como por ejemplo detección de fraude, o problemas de regresión, como sería el caso de predicciones en tarificación. Dependiendo de cuál sea el factor objetivo se usará un tipo u otro. En el caso que se va a presentar, se va a trabajar con los métodos de árboles de regresión.

- Árboles de regresión.

Este método de *machine learning* trata de predecir variables continuas o discretas a partir de un conjunto de características como input. Por ejemplo, en un problema actuarial de tarificación en no vida, se desarrolla un modelo predictivo f que relaciona los factores de riesgo x con el coste previsto de los siniestros $\hat{y}$, estableciendo lo siguiente $\hat{y} = f(x)$.

Esto se realiza a través de algoritmos, siendo uno de los más usados el algoritmo de árbol de clasificación y regresión (CART), introducido por Breiman et al. (1984). En este algoritmo, se establece un espacio muestral R que son los posibles valores para las p variables $x_1, \ldots, x_p$,. Un árbol divide el espacio predictor R en j regiones distintas y no superpuestas $R_1, \ldots, R_j$. En la j^{a} región, la respuesta ajustada $\hat{y}R_j$ se calcula como una media (ponderada) de las observaciones de entrenamiento que caen en esa región (Henckaerts, 2021).

$$f_{árbol}(x) = \sum_{j=1}^{J} \hat{y}\, R_j \;\; \Psi(\mathrm{x} \in R_j\,)$$

El indicador $\Psi(\mathrm{A})$ es igual a uno si se produce el suceso A e igual a cero en caso contrario. Como las regiones J no se solapan, la función indicadora difiere de cero para exactamente una región para cada x. Por lo tanto, un árbol realiza la misma predicción constante $\hat{y}R_j$ para toda la región R_j.

Sin embargo, tienen un inconveniente y es que un árbol grande es probable que sobreajuste los datos y no generalice bien cuando se introduzca en nuevos datos, mientras que un árbol pequeño puede no ajustarse bien a los datos y no captar las tendencias generales. Esto está relacionado con el equilibrio sesgo-varianza (Friedman et al., 2001), lo que significa que un árbol grande tiene un sesgo bajo y una varianza alta, mientras que un árbol pequeño tiene un sesgo alto pero una varianza baja.

- Métodos combinados de árboles

Los métodos que también se utilizan dentro del aprendizaje supervisado son los conocidos como métodos de *ensemble* o métodos combinados de árboles donde, a partir de múltiples algoritmos de aprendizaje, se posibilita una mejora de las predicciones sobre otros métodos basados en algoritmos individuales, como son los árboles de decisión o regresión, en este caso.

Estos últimos tienen como una desventaja su alta sensibilidad a la variabilidad propia de los grupos de entrenamiento, pudiendo dar resultados muy diferentes en función de las características de las muestras. Esa desventaja se puede reducir usando los métodos combinados a costa de perder interpretabilidad.

- Random Forest (RF).

Consiste en una técnica que permite construir una multitud de árboles de decisión de la parte establecida en el set de datos para el entrenamiento, lo que permite corregir aspectos como el sobreajuste (sesgo-varianza) y, por tanto, mejora el resultado final.

La forma en que funciona el algoritmo es la siguiente: primero establece árboles de decisión individuales que crecen hasta su máxima extensión posible sin ningún proceso de poda. Para ello, se ajustan con los datos de entrenamiento, pero a partir de métodos de remuestreo hay datos ligeramente distintos en cada árbol. La predicción final, mediante un algoritmo de RF, es la media de las predicciones de todos los árboles que lo forman.

Los principales hiperparámetros a tener en cuenta son:

  - ntree: número de árboles a establecer.
  - mtry: número de variables predictoras como candidatas en cada ramificación. Para el caso de regresión es el número de variables dividido entre 3.
  - sampsize: el número de muestras sobre las cuales entrenar.
  - nodesize: mínimo número de muestras dentro de los nodos terminales.
  - maxnodes: máximo número de nodos terminales. Por defecto, no se planifica un proceso de poda, dejando crecer los árboles hasta su límite máximo.

- Gradient Boosting Machine (GBM)

Otro de los principales modelos dentro de la categoría *ensembling* que combina múltiples modelos sencillos (*weak learners*). El funcionamiento que tiene, a diferencia de en los RF donde se construyen árboles de decisión independientes, aquí se realizan de manera secuencial los nuevos modelos que se incorporan al conjunto, intentando corregir los errores de los anteriores. El resultado es una combinación de numerosos modelos de la familia *boosting* de los que consigue aprender relaciones no lineales entre la variable respuesta y los predictores.

Se trata de una generalización del método *boosting machine* pero con el descenso de gradiente para optimizar cualquier función de coste durante el ajuste del modelo. Al final,

el valor que predice es la agregación que se ha ido formando sobre las predicciones de los modelos individuales que se han ido creando.

Un aspecto a tener en cuenta es el sobreajuste sobre la parte de datos de entrenamiento por el ajuste perfecto que puede tener. Para evitar esto, se incluye un hiperparámetro que se denomina *early stopping* o detección temprana.

Uno de los aspectos más atractivos del GBM es su alta flexibilidad para ajustar (*tunning*) los parámetros. Estos son:

- Número de árboles óptimo para el mejor ajuste del modelo.
- Profundidad de los árboles.
- Proporción de observaciones a tener en cuenta en cada árbol.
- Número mínimo de observaciones en cada nodo terminal.
- Nodos máximos por árbol.
- El ratio de aprendizaje que controla la velocidad del procesamiento del algoritmo.

3.2.2. Aprendizaje No Supervisado

El siguiente método se basa en algoritmos de *machine learning* para analizar y agrupar en clústeres conjuntos de datos sin etiquetar. Aquí, se desconocen las estructuras intrínsecas del conjunto de datos debido a la ausencia de un atributo que, de alguna manera, guíe o supervise la formación de dichas estructuras. Permiten, de esta esta forma, que se encuentren agrupaciones de datos o patrones sin que sea requerida la intervención humana.

El principal método que hay es el *clustering* pero también existen otros tipos como las reglas de asociación y reducción de dimensiones.

- Clustering

Se encarga de encontrar patrones o grupos dentro de un conjunto de datos. Trabaja a partir de particiones que se establecen tal que las observaciones que están dentro de un mismo grupo son similares entre ellas y distintas a las observaciones de otros grupos.

No se trata de una técnica supervisada porque no usan una parte de entrenamiento donde se conoce la verdadera clasificación, sino que el proceso ignora la variable respuesta que indica a qué grupo pertenece realmente cada observación. Dentro del *clustering* hay diferentes tipos:

- *Partitioning Clustering*: Requiere que el individuo especifique de antemano el número de *clusters* que se van a crear (*K-means*, *K-medoids*, *CLARA*).
- *Hierarchical Clustering*: No requiere que el usuario especifique de antemano el número de *clusters*. (*agglomerative Clustering*, *divisive Clusterig*).

Métodos que combinan o modifican los anteriores (*hierarchical K-means, fuzzy Clustering, model based Clustering y density based Clustering*).

3.3. Interpretación en el aprendizaje supervisado.

Las combinaciones de múltiples árboles, como ocurre en los GBM, no son tan simples de interpretar como lo son los árboles de decisión individuales y dan lugar a lo denominado como caja negra. Para poder abrir esta caja negra y comprender cómo funciona un modelo como el GBM, existen varias herramientas y métodos disponibles.

- Importancia de las variables.

Introducido por Breiman, Friedman, Stone, & Olshen (1984), mide cómo de importantes son las variables explicativas que entran en el modelo a la hora de predecir la variable respuesta. Para una variable explicativa específica, xℓ, ℓ ∈ {1, ..., p}, en un árbol de decisión m, la importancia se calcula como:

$$\mathcal{I}_l(m) = \sum_{j=1}^{J-1} I(v(j) = \ell)(\Delta L)_j$$

Es decir, se realiza la suma de las mejoras obtenidas en la función de pérdida L sobre todos los nodos internos J-1 en los que la variable xℓ se utiliza como variable de división. Cuanto mayor es la mejora en la función de pérdidas, mayor es la importancia de la variable. Para poder entender la contribución relativa de cada variable, se normalizan los resultados de la importancia para que sumen 100%. Si se aplica en los GBM, la clasificación resulta de un promedio de la importancia de la variable xℓ sobre los diferentes árboles incluidos en el proceso de ensemble.

$$\mathcal{I}_l = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_l(m)$$

- Partial dependence plots (pdp).

Son gráficos que sirven para entender el efecto de los factores respecto a la variable respuesta. Muestran el efecto marginal de una variable en las predicciones obtenidas de un modelo (Hastie, Tibshirani, & Friedman, 2009). Se realiza calculando las predicciones para una variable específica xℓ mientras se hace el promedio de los valores de las otras variables $x_{i,C}$:

$$\bar{f}_\ell(\mathrm{x}\ell) = \frac{1}{n}\sum_{i=1}^{n} f_{model}(\mathrm{x}\ell, x_{i,C})$$

Donde C es el conjunto complementario de ℓ, tal que $\ell \cup C = \{1, \ldots, p\}$; $x_{i,C}$ son los valores de las otras variables para la observación i, y n es el número de observaciones en los datos de entrenamiento.

Es importante tener en cuenta que las gráficas de dependencia parcial miden el efecto de xℓ en $f(x)$ después de tener en cuenta los efectos promedio de las otras variables x_C en $f(x)$. Por esta razón, los posibles efectos de interacción entre xℓ y otra variable en x_C pueden oscurecer el efecto.

3.3.1. Evaluación del rendimiento de los modelos.

Para poder comparar el rendimiento del modelo de los GBMs, necesitamos introducir medidas de rendimiento relevantes donde uno de los usos más comunes es la técnica seleccionada de validación cruzada anidada para el entrenamiento y evaluación del modelo.

Este método puede lidiar tanto con la selección del mejor conjunto de hiperparámetros, como con la estimación del error. La forma de realizarse es a partir de una subdivisión de los datos en los k conjuntos de igual tamaño. Después, a partir de un bucle interno se realiza una validación cruzada de $k - 1$ para cada combinación de hiperparámetros y se calcula el error de validación cruzada, aplicando la función de pérdida L, calculada promediando el error en los conjuntos de datos de validación. Los hiperparámetros óptimos son aquellos que minimizan el error de validación cruzada.

En cuanto a la comparación y evaluación de la efectividad de los modelos actuariales, se utilizan diversas métricas de validación. Estas métricas proporcionan una medida

cuantitativa del desempeño del modelo y permiten identificar fortalezas, debilidades y áreas de mejora.

Entre las más utilizadas se encuentran el error cuadrático medio (MSE), la raíz cuadrada del ECM (RMSE), el error medio absoluto (MAE) y el coeficiente de determinación (R^2), entre otras. Cada una de estas métricas ofrece una perspectiva diferente sobre el desempeño del modelo y proporciona información valiosa para respaldar la toma de decisiones.

- MSE

Mide el error cuadrado promedio de nuestras predicciones. Para cada punto, calcula la diferencia cuadrada entre las predicciones y el objetivo y, tras esto, promedia esos valores. Cuanto mayor sea este valor, peor será el modelo y sería cero para un modelo perfecto.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Donde, y_i es el resultado real esperado y $\hat{y}_i$ es la predicción del modelo.

- RMSE

RMSE es la raíz cuadrada del MSE y se introduce para hacer que la escala de los errores sea igual a la escala de los objetivos.

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

- MAE

Calcula el error como un promedio de diferencias absolutas entre los valores objetivo y las predicciones. El MAE es una puntuación lineal, lo que significa que todas las diferencias individuales se ponderan por igual en el promedio.

$$MAE = \frac{1}{N}\sum_{I=1}^{N}|y_i - \hat{y}_i|$$

- Coeficiente de determinación R^2

Está estrechamente relacionado con el MSE, pero tiene la ventaja de estar libre de escala, no importa si los valores de salida son muy grandes o pequeños, el R^2 siempre estará entre -∞ y 1.

$$R^2 = \frac{\sum_{t=1}^{T}(\hat{Y}_t - \bar{Y})^2}{\sum_{t=1}^{T}(Y_t - \bar{Y})^2}$$

Cuando R^2 es negativo, significa que el modelo es peor que predecir la media.

3.3.2. Double lift charts.

Una vez se ha modelizado, esta es otra manera más visual de comparar directamente dos modelos además de las métricas comentadas. Los *double lift charts*, a partir de los resultados de la variable respuesta que han predicho, analizan cuál de los modelos que se comparan se ajusta mejor a los datos observados. Para ello se realizan los siguientes pasos.

1. Una vez que se tienen las predicciones de ambos modelos, se calcula el ratio entre las predicciones de uno respecto al otro.

2. Se ordena ese ratio de menor a mayor y se establecen grupos que tengan la misma exposición.

3. Se calcula el promedio de la variable respuesta en cada uno de los grupos y se plasman los resultados en un gráfico.

De los modelos que se han comparado, el que más se ajuste a las observaciones empíricas será el que mejor predice y además se pueden ver los grupos donde hay más diferencias entre las predicciones de los modelos que se comparan.

4. CASO DE ESTUDIO.

La privacidad que mantienen las compañías de seguros a la hora de facilitar muestras de sus carteras es muy elevada, trabajan con muchos datos sensibles que deben proteger para preservar la privacidad del cliente. Eso dificulta en gran medida la obtención de datos para poder realizar labores de investigación y, aún más, en entornos tan nuevos como el caso de los seguros UBI y la telemática.

En este estudio se ha obtenido una base de datos sintética elaborada por el departamento de matemáticas de la Universidad de Connecticut So et al. (2021), donde trabajan a partir de una cartera con datos reales de variables tanto clásicas como telemáticas. Los datos reales conforman un portfolio de 100.000 asegurados en el país de Canadá que se emulan en la sintética para poder proporcionar una base de datos de acceso libre para la modelización de riesgos en este entorno.

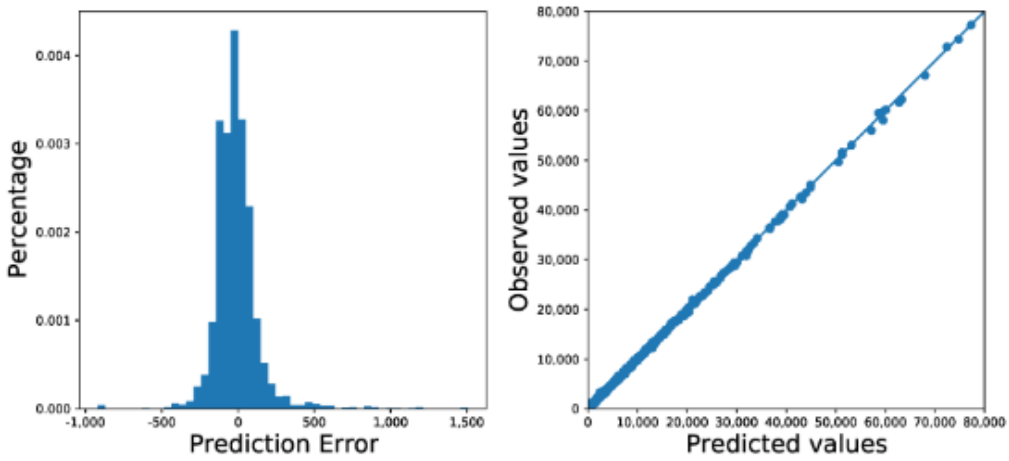
De manera resumida, el proceso que han llevado a cabo para generar dicha base de datos se basa en tres etapas:

- En la primera etapa, a partir de un algoritmo SMOTE extendido, se genera un conjunto de datos sintéticos del mismo tamaño y variables que el original.
- En segundo lugar, se simulan resultados de número de siniestros como múltiples clasificaciones binarias aplicando técnicas de redes neuronales *feedforward*.
- Por último, se simulan valores agregados para el valor de los siniestros como una regresión usando redes neuronales *feedforward* incluyendo el número de siniestros como variable.

El resultado final de la nueva cartera sintética se crea combinando los tres pasos, el portfolio sintético, el número de siniestros sintético y la cuantía agregada de los mismos. El resultado es evaluado mediante un proceso de comparación con el conjunto de datos original, ajustando modelos de regresión utilizando las funciones de Poisson y Gamma.

Las variables de respuesta han sido generadas a través de un procedimiento muy complejo y no paramétrico y pueden no reflejar de manera precisa la naturaleza de la generación de datos buenos resultado de la comparación entre ambas bases de datos (So et al., 2021). Por tanto, se incluye también en la fase de evaluación una comparación estadística de las variables de ambas carteras.

GRÁFICO 1 - EVALUACIÓN EN LA PRECISIÓN DE LA SIMULACIÓN DEL IMPORTE AGREGADO DE LOS SINIESTROS.



Fuente: So, B., Boucher, J. P., & Valdez, E. A. (2021).

En la anterior figura, So et al. (2021) muestran el resultado de los errores en la base de datos simulada donde los errores se encuentran bastante centrados en el 0 y los datos predichos se sitúan dentro de la regresión en el *quantile-quantile (QQ) plot*.

Para finalizar con la validación de la nueva cartera sintética se comparan los gráficos de la frecuencia media para alguna de las variables (Anexo A). Y de forma similar con la severidad, pero con dos variables diferentes se sigue observando la similitud en ambos sets de datos (Anexo B). Se muestra en ambos gráficos de los anexos que tanto en severidad como en frecuencia no parecen variar los valores esperados, así como los patrones de las distribuciones.

El resultado final es de un total de 52 factores que conforman la base de datos y se describen a continuación diferenciando entre los tipos de variables que son.

TABLA 1 – TOTAL DE VARIABLES DE LA BASE DE DATOS.

Tipo	Variable	Descripción
Clásica	Duration	Duración de la cobertura de la póliza, en días.
	Insured.age	Edad del asegurado, en años.
	Insured.sex	Sexo del asegurado (Hombre/Mujer).
	Car.age	Edad del vehículo, en años.
	Marital	Estado civil (Soltero/Casado).
	Car.use	Uso del vehículo: Privado, Desplazamiento, Granjero, Comercial.
	Credit.score	Credit score del asegurado
	Region	Zona en la que vive el asegurado (Rural/Urbana)
	Annual.miles.drive	Millas anuales conducidas esperadas declaradas por el conductor
	Years.noclaims	Número de años sin siniestros
	Territory	Localización territorial del vehículo
Telemática	Annual.pct.driven	Porcentaje anualizado de tiempo sobe la carretera
	Total.miles.driven	Total de distancia conducida en millas
	Pct.drive.xxx	Porcentaje del dia de la semana que ha conducido (lunes/martes/…/domingo)
	Pct.drive.xhrs	Porcentaje de tiempo conducido en las siguientes horas (2hrs/3hrs/4hrs)
	Pct.drive.xxx	Porcentaje de tiempo que ha conducido entre semana o el finde de semana (wkday/wkend)
	Pct.drive.rush.xx	Porcentaje de tiempo que ha conducido en el dia (am/pm)
	Avgdays.week	Número meido de días utilizado por semana
	Accel.xxmiles	Número de acelerones 6/8/9/…/14 mph/s cada 1000 millas
	Brake.xxmiles	Número de frenazos 6/8/9/…/14 mph/s cada 1000 millas
	Left.turn.intensityxx	Número de giros a izquierda cada 1000 millas con intensidad 08/09/10/11/12
	Right.turn.intensityxx	Número de giros a derecha cada 1000 millas con intensidad 08/09/10/11/12
Respuesta	NB_Claim	Número de siniestros
	AMT_Claim	Valor agregado de los siniestros

Fuente: Elaboración Propia

Cada línea dentro de la base de datos conforma un asegurado único con registros para cada una de sus variables. En el caso de la variable *Insured.sex* (Sexo del asegurado), la Unión Europea prohibió su uso para evitar la discriminación entre hombres y mujeres a la hora de la tarificación en empresas de seguros. Sin embargo, a nivel de modelización para medir los riesgos y analizar tarifas técnicas internas como se realiza en las compañías del mercado actual, si se utiliza. Después, quedaría excluida a nivel de tarifas comerciales.

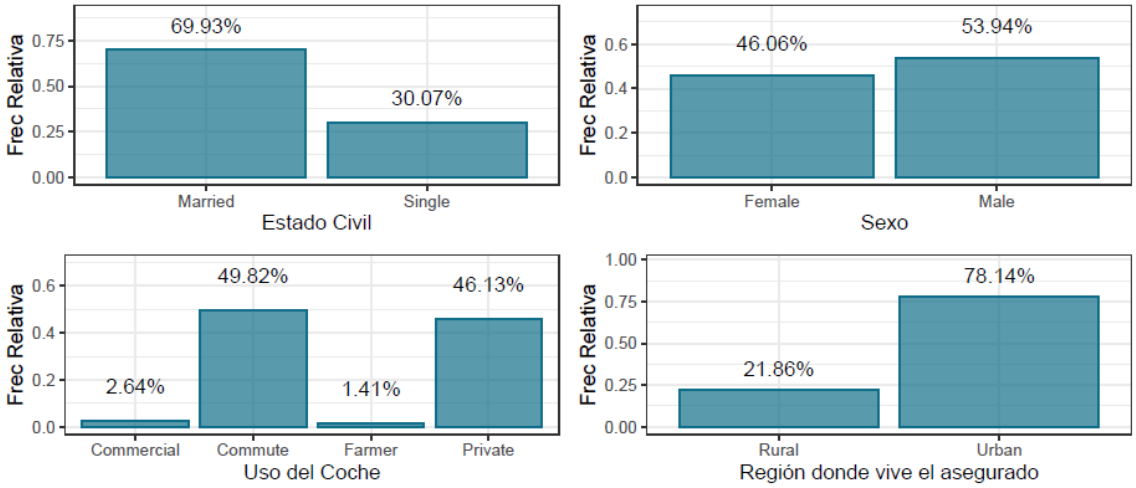
4.1. Análisis estadístico descriptivo de las variables clásicas.

A continuación, se realizará un análisis descriptivo de las variables clásicas que se encuentran en el set de datos. Dentro del amplio abanico de factores de riesgos con las que trabajan las compañías, estas suelen ser las principales para autos y se pueden dividir en este caso entre categóricas y continuas.

4.1.1. Variables Categóricas.

Dentro de la primera categoría están el estado civil, el sexo del asegurado, el uso del vehículo y la región en la que vive el asegurado.

GRÁFICO 2 - VARIABLES CLÁSICAS CATEGÓRICAS



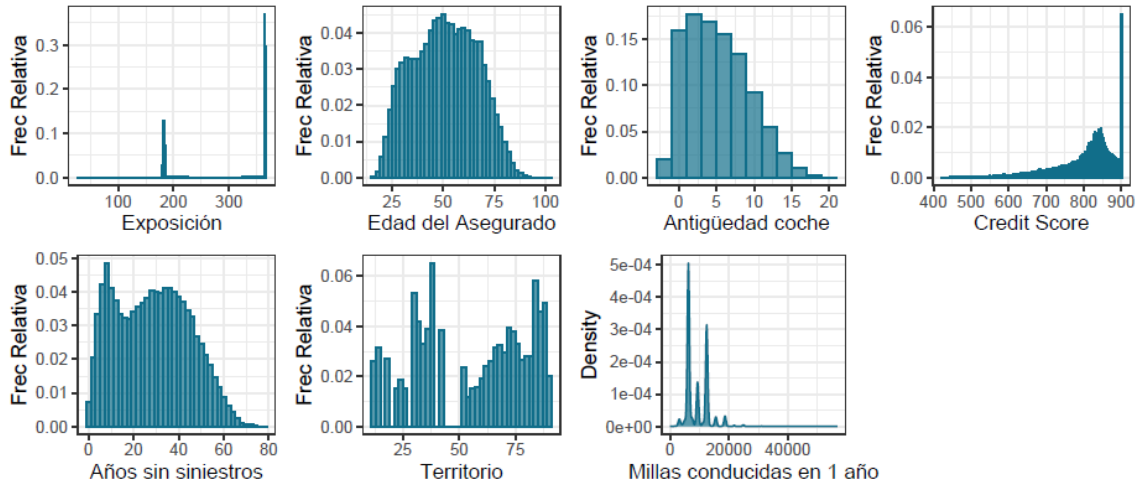
Fuente: Elaboración Propia

La mayoría de los asegurados están casados y prácticamente hay una igualdad entre hombres y mujeres. Además, como se ve en el gráfico de la parte inferior derecha, casi el 80% de la región donde habitan es urbana, lo que responde a los usos del coche que tienen más cantidad de frecuencia. Estos son coches utilizados diariamente para ir al trabajo (*Commute*) con prácticamente la mitad de la cartera, siendo la otra mitad un uso privado del mismo. El resto serían usos de tipo granjero y comercial con apenas una frecuencia del 5%.

4.1.2. Variables Continuas.

En cuanto a las variables continuas, se encuentran la exposición, edad del asegurado, la antigüedad del coche, el *credit score*, los años sin siniestros de los asegurados y el total de millas conducidas en un año.

GRÁFICO 3 - VARIABLES CLÁSICAS CONTINUAS.



Fuente: Elaboración Propia

La exposición tiene mucha importancia a la hora de modelar tanto frecuencia como severidad. La forma en la que viene expresada es en días con una duración máxima de la cartera de un año. Sin embargo, el valor máximo que toman los datos es de 366 aunque la mayoría se encuentra en 365. Para simplificar y trabajar mejor la base de datos, se unifican siendo el valor máximo que puede tomar de 365 y por tanto se modifica esta variable dividiendo entre 365 para trabajar con términos de exposición como se hace en el mercado. Se puede ver como la mayoría de los asegurados han tenido exposición 1, es decir, 365 días mientras que hay un grupo intermedio que tuvo alrededor de medio año.

Otra de las modificaciones que se van a realizar en el conjunto de datos se encuentra en la variable antigüedad del coche. Hay algo de frecuencia negativa, ya que es posible que se realice la compra de un coche, pero no sea entregado hasta dentro de uno e incluso dos años. Como no es objetivo del estudio estos coches que aún no han sido entregados se proceden a eliminar los casos donde la antigüedad es menor que 0 ya que se pretende estudiar para vehículos que han sido entregados.

Respecto al total de millas conducidas anualmente, el 99% de la distribución se encuentra entre las primeras 18641.13 millas que serían equivalentes a 30.000Km. Como se ve en el gráfico hay una cola muy larga ya que hay algún dato atípico que se procede a eliminar para que no distorsione los resultados.

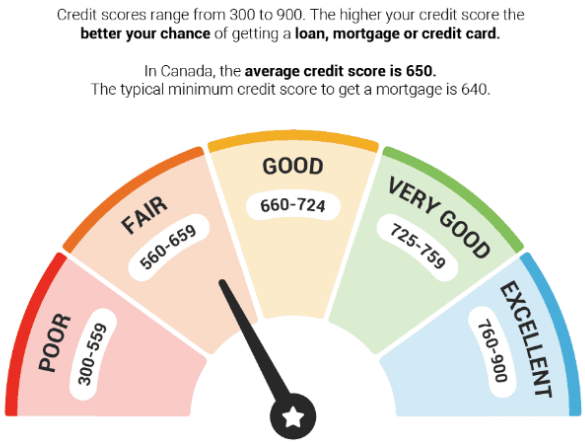
Al tratarse de una base de datos que proviene de Canadá, la edad mínima de obtención del carné es de los 16 años. Esto implica que las edades de la variable *Insured.age* vayan

desde los 16 hasta los 103 que sería el caso más extremo, pero la mayoría de la frecuencia de la cartera se encuentra entre los 20 y los 80 años. A esta le siguen los años sin siniestros que pueden ir de 0 años hasta 80, donde los años que más acumulan están entre los 4 y los 10 y luego disminuye hasta los 14 donde vuelve a incrementar ligeramente hasta que llega a un punto donde vuelve a bajar de manera fuerte hasta los 70 donde ya no hay casi valores.

La variable territorio está codificada entre los valores 11 y 91, teniendo entre medias valores sin frecuencia. Sin embargo, no se ha podido dar sentido a esta variable buscando una relación con la geografía canadiense, pero se mantiene en la base de datos.

El último factor continuo que queda por explicar es el *credit score*. Esto es una manera de puntuar de forma crediticia a las personas que residen en Canadá para los prestamistas. Esta puntuación suele oscilar entre los 300 y 900 puntos donde a mayor puntuación, mejor. Las puntuaciones suelen cuantificar características financieras que pueden tener como puede ser el riego de cumplir con los pagos de un préstamo recibido.

ILUSTRACIÓN 5 - DESCRIPCIÓN DE LA VARIABLE CREDIT SCORE.



Fuente: King, R. 2022

Según el artículo (Sterling Homes Edmonton, Head Office, 2023), estos son los indicadores que influyen a la hora de obtener este *scoring*.

- Con qué frecuencia paga las facturas a tiempo (35 por ciento)
- Cuánto debe y qué porcentaje del crédito disponible está usando (30 por ciento)
- Cuánto tiempo ha tenido las cuentas abiertas (15 por ciento)

- Si ha estado solicitando o no una gran cantidad de crédito nuevo (10 por ciento)
- Si tiene una combinación de crédito fijo y renovable (10 por ciento)

En el conjunto de nuestros aseguradores, la media es más alta y se encuentra en torno a 800 puntos siendo el mínimo 422 donde solo el 25% es menor que 766 por lo que se podría decir que se trata de una cartera formada con un muy buen *credit score*.

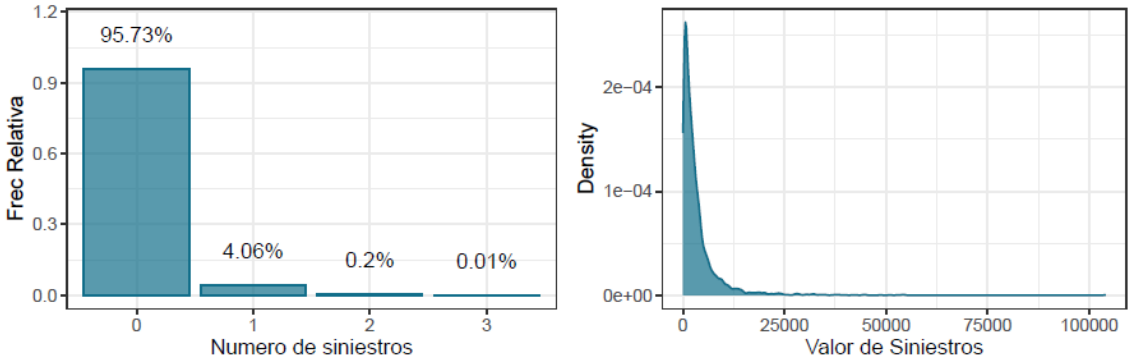
Tras todo este análisis univariante del set de datos y la eliminación de aquellos registros comentados, se pasa a tener un total de 97.175 asegurados.

4.1.3. Variables de respuesta.

El gráfico 4 muestra las distribuciones de las variables número de siniestros (*NB_Claim*) y la cuantía de estos (*AMT_Claim*) en la cartera. La mayoría de los asegurados no han sufrido siniestros durante el periodo de exposición y, en este caso, conforman el 95,73%, siendo solo un 4,05% los que tuvieron un solo siniestro y entre dos y tres siniestros conforman el 0,21% restante. El estudio del que provienen los datos no indica que tipo de coberturas son las que cubren a estos aseguradores, pero por la cantidad de ceros que hay se podría intuir que se trata de una cartera a terceros conocida como MTPL (*Motor Third Party Liability*).

Durante el análisis exploratorio de los datos, se ha encontrado varias pólizas sonde el número de siniestros era mayor que cero, pero su cuantía no. Estas situaciones podrían darse al ser siniestros que se hayan reportado pero no se sepa aun su valor ya que hay trámites de por medio que requieren de tiempo. Estos casos han sido eliminados ya que no son objeto del estudio y podrían distorsionar la modelización quedando un total de 96,782 asegurados.

GRÁFICO 4 - VARIABLES DE RESPUESTA.



Fuente: Elaboración propia

En cuanto al valor de los siniestros, para poder analizar mejor las cuantías y la distribución de esta variable, se eliminan todos los escenarios donde no hay siniestros. Este nuevo conjunto de datos se utilizará para modelizar la severidad. Se puede ver en su gráfico de densidad que se trata de una distribución muy asimétrica a la derecha con una cola muy larga, lo que es común en el sector asegurador para este tipo de carteras.

Las gráficas comentadas nos van a dar un primer paso a la hora de trabajar con lo que serán las variables dependientes de nuestros modelos de frecuencia y severidad junto a la exposición que tendrá un papel muy importante. Este será el de poder hacer comparables los datos teniendo en cuenta el tiempo al que ha estado expuesto durante el año, para lo cual se introduce en el modelo un término *offset* que no compromete la distribución natural de los datos y que permite desplazar la variable de exposición al lado derecho de la ecuación de regresión y tomar el logaritmo de esta variable en el modelo con un coeficiente limitado a uno.

- Ajuste de Frecuencia

Una vez analizado, es necesario saber a qué distribución se ajustan ambas variables, ya que de esto dependerá de cómo se va a confeccionar la modelización. La Frecuencia de cada asegurado i, se denota Fi.

Como se ha comentado, la distribución más utilizada para modelizar la variable que mide el número de siniestros es la Poisson, pero es necesario hacer un análisis previo porque puede ajustarse a otras como la Binomial Negativa. Un buen signo de que puede ser una Poisson es que la media y varianza sea iguales, en este caso son muy similares, la frecuencia media que hay en toda la cartera es de un 5,19% y la varianza de un 5,48% por

lo que un modelo lineal generalizado con distribución de Poisson y función de enlace logarítmica es una opción natural.

Un problema que puede aparecer en estos conjuntos de datos a la hora de trabajar con la frecuencia es la alta cantidad de registros sin ningún siniestro ya que, aparte de que normalmente no se tienen accidentes, los pequeños coches no suelen notificar los pequeños golpes que reciben. Además, la exposición dentro de la cartera puede ser muy variada haciendo que no sean tan comparables todas las observaciones. Por último, puede aparecer la sobredispersión característica de la distribución de la Poisson, que provoca que la varianza de los recuentos observados sea mayor a la esperada, indicando que el modelo no es el correcto.

En el proceso de modelización, para abordar el posible caso de sobredispersión, se va a elaborar un modelo de Poisson preliminar para realizar un test de hipótesis con hipótesis nula de no-dispersión y de alternativa sobredispersión o subdispersión dependiendo del valor que tome α. Si $\alpha > 0$, hay sobredispersión y si $\alpha < 0$, su subdispersión. El resultado es un p-valor de prácticamente 0 por lo que se puede decir que, de manera significativa, se rechaza la hipótesis nula de no dispersión. Al tener un alpha de $\alpha = 0.0627$ la frecuencia es sobredispersa haciendo que la varianza de los datos este inflada y sea superior a la media. Como es el caso donde la media es un 4,74% y la varianza de un 5,06%.

TABLA 2 – TEST DE DISPERSIÓN DE POISSON.

Overdispersion test

data: freq_classic_poiss
z = 8,5176, p-value = 2.2e-16
alternative hypothesis: true alpha is greater than 0
sample estimates:

alpha
0.06274455

Fuente: Elaboración propia

Tras este resultado, se procede a realizar un análisis con otras distribuciones que sean capaces de ajustarse mejor a los datos. Se utiliza la binomial negativa, y se realiza una comparativa de dos modelos lineales generalizados de frecuencia teniendo en cuenta la exposición para ver si hay algún cambio en sus métricas.

TABLA 3 – COMPARATIVA DE MÉTRICAS DE LOS MODELOS DE FRECUENCIA.

Modelo	AIC	BIC
Poisson	32703,59	32713,07
Binomial Negativa	32569,93	32588,89

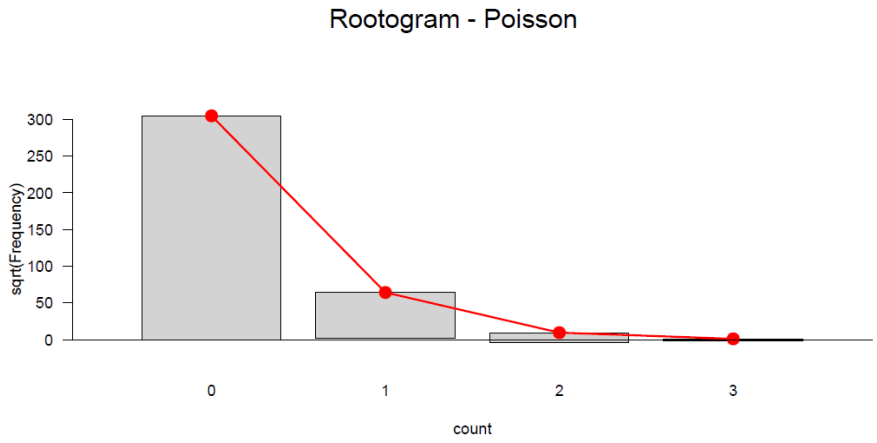
Fuente: Elaboración propia

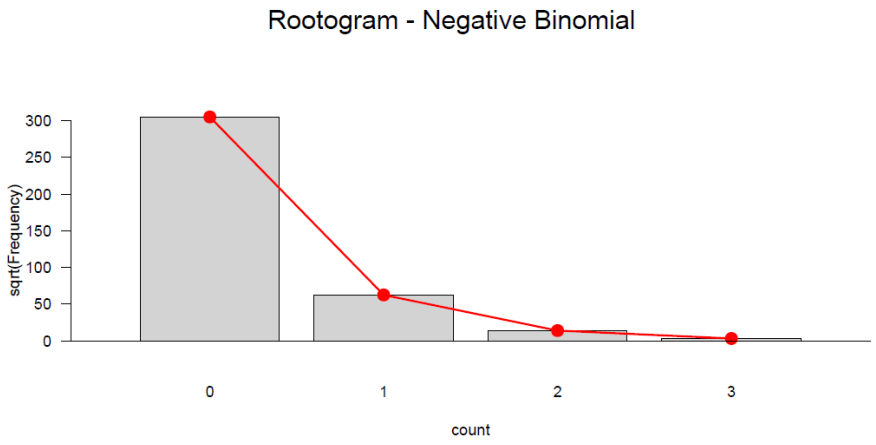
Tanto el AIC y BIC disminuyen y mejoran al modelo con la Poisson lo que ya indica que la Binomial Negativa puede ser un mejor modelo. A continuación, se va a hacer una comparativa de lo que se denomina *rootogram*.

Un *rootogram* es un procedimiento donde se comparan de manera gráfica las frecuencias de las distribuciones empíricas y los modelos ajustados. Las frecuencias observadas se muestran como barras y las frecuencias ajustadas como una línea utilizando una escala para el histograma que es la raíz cuadrada de la frecuencia observada. Para este caso, se ajusta a través del método *maximum likelihood* utilizando la librería de R MASS. Este método ajusta maximizando la probabilidad de obtener el parámetro desconocido a través del vector(θ), y en la práctica se utiliza minimizando la función *log-likelihood* negativa.

- $L(\theta) = \prod i P_r(N_i = n_i|\theta)$
- $L(\theta) = \sum i \log\left(P_r(N_i = n_i|\theta)\right)$

GRÁFICO 5 - ROOTOGRAMS DE POISSON Y BINOMIAL NEGATIVA PARA LA FRECUENCIA.





Fuente: Elaboración propia

Podemos deducir del gráfico anterior que la regresión de Poisson no es la opción ideal para este conjunto de datos, ya que gráficamente se observa cómo no se ajusta bien a los datos. El gráfico cuelga alrededor del eje x, lo que explica que el modelo para 1 siniestro sobreajuste. Por encima de 1, el histograma cae de la línea del 0 indicando un infraajuste en la frecuencia. En cambio, para la Binomial Negativa, los resultados parecen mejores que los de Poisson ya que no parece que haya ninguna situación mala de ajuste.

El último paso que se completa es un *test* de validación donde la hipótesis nula es que los datos siguen una distribución Binomial Negativa. El P-Valor no es lo suficientemente bajo para rechazar bajo la hipótesis nula que los datos siguen una Binomial Negativa, por lo que se concluye que esta distribución es adecuada para modelizar la frecuencia ya que se ajusta bien a la variable de número de siniestros.

TABLA 4 – RESULTADO DE LA PRUEBA DE AJUSTE A UNA DISTRIBUCIÓN BINOMIAL NEGATIVA.

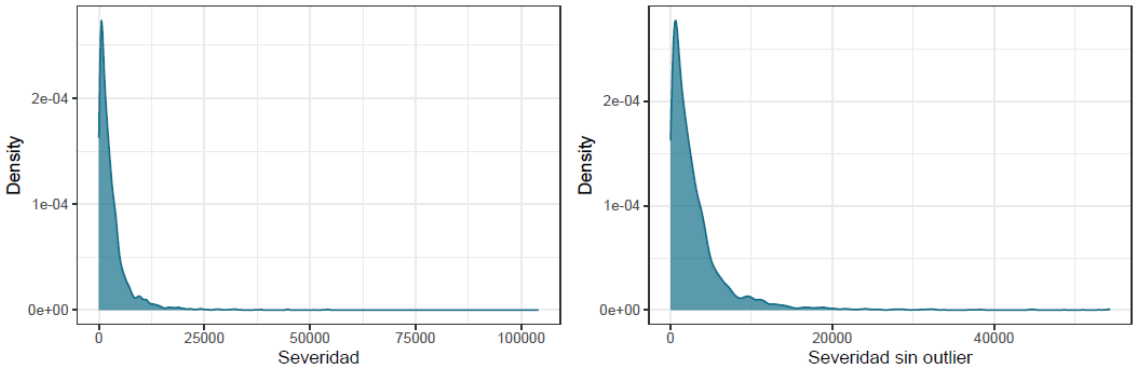
Goodness-of-fit test for nbinomial distribution			
	X^2	df	P(> X^2)
Likelihood Ratio	1,5729	1,00	0,2098

Fuente: Elaboración propia

- Ajuste de Severidad

Ahora se procede a realizar un análisis similar, pero para la variable de cuantía media del siniestro (*AMT_Claim/NB_Claim*), que es la que se tiene en cuenta para modelizar la severidad. Se denota severidad como *Si* para cada asegurado *i*.

GRÁFICO 6 - VARIABLE VALOR DEL SINIESTRO ANTES (IZQUIERDA) Y DESPUÉS DEL ELIMINAR EL VALOR ATÍPICO (DERECHA).



Fuente: Elaboración propia

Primero se filtra quitando los siniestros que sean 0 para poder analizar cuando hay pagos positivos y, tras analizarlos, se elimina el único dato atípico que superan los 100.000$ ya que en este caso afectaría distorsionando la modelización. El total de la población pasa a ser de 96,781 asegurados.

Ambos gráficos reflejan la severidad, que viene a ser el total incurrido de la póliza entre el número de siniestros que se han ocasionado. Tras eliminar el *outlier*, la cantidad de pólizas que conforman los asegurados con algún siniestro es de 3.719, siendo la media de 2955.06$ y el valor máximo de 25495.55$

Se van a comparar variables que son asimétricas a la derecha y de forma acampanada como se observa en la gráfica. El estándar en el mercado de autos para la severidad es utilizar la Gamma, pero también podrían ser otras como la log-normal.

Lo primero que se compara son dos modelos GLM base sin incluir ninguna variable, al igual que para la frecuencia. Se establecen las familias comentadas en la distribución y, como función enlace, se incluye el logaritmo de la exposición y el número de siniestros como *offset*.

TABLA 5 – MÉTRICAS DE UN GLM BASE PARA DISTRIBUCIÓN LOG-NORMAL Y GAMMA.

Modelo	AIC	BIC
Log-Normal	73436,67	73449,12
Gamma	67605,08	67617,52

Fuente: Elaboración propia

En el caso de esta variable, en lugar de utilizar un *rootogram* ya que su uso es para variables discretas, se va a realizar un análisis para estimar los parámetros de la Gamma de la que vendrían nuestros datos y después realizar una validación de hipótesis para comprobar que se ajustan a esa distribución.

El primer paso sería generar una población controlada que viniese sobre una distribución Gamma y con unos parámetros que dibujen la forma de la distribución del fenómeno aleatorio severidad de los siniestros.

Esto se ha realizado a través de un DGP (*Data Generating Process*), generando un fenómeno aleatorio de 1.000.000 de datos utilizando el método de la inversa de Montecarlo con la función en R; *rgamma* y con los parámetros que se indican en el título del gráfico del Anexo C. A partir de aquí, se estiman los parámetros en esta situación controlada utilizando el método de los momentos, *percentile matching* y *máximum likelihood* (ML) para luego realizar una comparativa y escoger el que menor error tiene.

TABLA 6 – SESGO Y MSE DE LOS MÉTODOS DE ESTIMACIÓN.

Método	Sesgo	MSE
Momentos	-0.775275	6.01E-01
PM	-0.002179	1.79E-04
ML	0.0011991	2.99E-05

Fuente: Elaboración propia

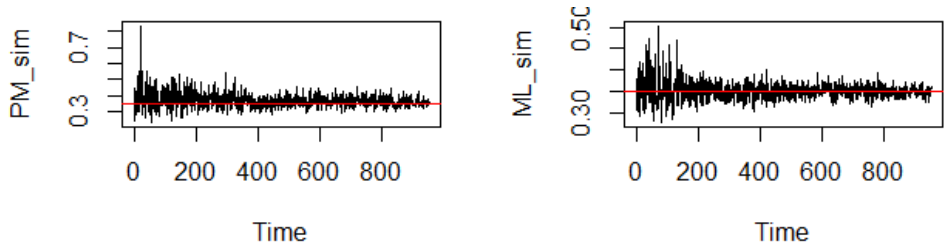
El sesgo y error cuadrático medio miden la calidad del estimador y se utilizan como método comparativo para escoger la mejor forma de estimación.

- $Sesgo = E[\hat{\varphi}] - \varphi = 0$
- $MSE(\hat{\varphi}) = E[(\hat{\varphi} - \varphi)^2] = Var(\hat{\varphi}) + Sesgo(\hat{\varphi})^2$

El objetivo final es estimar el parámetro que más se parezca al verdadero valor de la distribución que, en este caso, al ser controlada es conocido. Por tanto, con estas dos medidas de calidad lo que se quiere es que el sesgo del estimador sea 0. Es el método ML (máxima verosimilitud) el que menor sesgo y mse tiene y, por tanto, es de una mayor calidad a pesar de que todos parezcan insesgados y que el *percentile matching* proporcione unos resultados muy parecidos.

Estos métodos descritos, simplemente comparan estimadores, pero la condición necesaria para poder utilizar un método de estimación es la consistencia. Si es inconsistente, el método se rechaza y habría que buscar otra manera de estimar los parámetros. La consistencia estudia cómo, a medida que se va aumentando la muestra, el parámetro estimado tiende al parámetro real. Es por ello que la diferencia de ambos a medida que aumente la muestra tienda a 0 es decir, $\lim_{n\to\infty} \Pr\left(|\hat{\varphi}n - \varphi| > \varepsilon\right) = 0$

GRÁFICO 7 - EVALUACIÓN DE LA CONSISTENCIA SEGÚN AUMENTAN EL NÚMERO DE REPETICIONES.



Fuente: Elaboración propia

De forma gráfica, se observa que ambos métodos (PM a la izquierda y ML a la derecha) son consistentes, ya que, a medida que aumenta la muestra, los parámetros estimados tienden a los reales.

Por último, mediante la técnica de remuestreo *bootstrap* se estudia la distribución de los parámetros estimados por los tres métodos. Esto se hace mediante la generación de muchas muestras (muestras *Bootstrap*) a partir de la población original controlada y ver qué valores toman los parámetros en todas esas realizaciones para sacar la distribución de cada uno.

TABLA 7 – INTERVALOS DE CONFIANZA DE LA DISTRIBUCIÓN DE LOS PARÁMETROS ESTIMADOS.

Método	IC 2.5%	IC 97.5%
Momentos	1.1252744	1.1252763
PM	0.3325945	0.3725606
ML	0.339374	0.3589601

Fuente: Elaboración propia

Al final, se mide cómo se distribuye el parámetro en los diferentes métodos de estimación, es decir, todos los posibles valores que va a tomar el estimador. Se concluye que el ML está más acotado y, por tanto, el parámetro se encontrará entre [0.33974 – 0.3589601] (Anexo D). En el 95% de las realizaciones, nuestro estimador se encontrará entre el anterior intervalo teniendo una distribución gaussiana debido al Teorema Central del Límite.

Para finalizar, el último paso que queda es realizar un *test* de ajuste para ver si la severidad de nuestros datos, estimada por *maximumm likelihood*, procede de una distribución Gamma. Tras la estimación se obtiene que los parámetros son p = 0.6642468 y λ = 0.0001479.

Se utiliza el *test* de validación de diseñado por Anderson-Darling. Un método que pondera las diferencias entre todos los puntos de la distribución. Este método proporciona más peso a las colas de la distribución y, por tanto, es lo que más interesa para mediciones de *VaR, TVaR* o *Best-Estimate* dentro del marco actuarial.

La salida que se ve en el Anexo E es de un p-valor igual a 0,376. Al ser tan alto, no se puede rechazar la hipótesis nula de que la verdadera distribución proviene de una Gamma ya que es muy probable obtener los valores que hemos obtenido bajo $H0$ cierta (Que la severidad proviene de una distribución Gamma).

4.2. Modelización flexible a partir de métodos GAMs.

El procedimiento es el de estimar un modelo flexible tanto para la frecuencia como la severidad, denominadas *Fi* y *Si* respectivamente. A la hora de la modelización, se tiene en cuenta también la exposición que tienen los asegurados en la cartera.

Una vez tenemos ambos modelos establecidos, se calcula la prima pura denotada como π_i, de forma que $\pi_i = E[F_i] * E[S_i]$, asumiendo independencia entre ambos.

Antes de realizar ningún tipo de modelo, se divide el conjunto de datos tras el análisis descriptivo del apartado 3.1 en dos conjuntos de datos aleatorios. El primero está formado por una muestra aleatoria del 70% y se usará para entrenar a los modelos y el 30% restante para testear los resultados obtenidos en esa muestra y poder evaluar la modelización.

Una vez dividido el conjunto de datos y antes de comenzar con la modelización, se realiza un análisis de la correlación entre las variables para ver si están altamente relacionadas, lo que implicaría que se podría prescindir de alguna variable ya que no añadiría valor porque el efecto sería similar a la otra con la que estuviese correlacionada.

Para las variables continuas se utiliza la correlación de Pearson, que es una medida de la fuerza y dirección de la relación lineal entre dos variables.

GRÁFICO 8 – CORRELOGRAMA DE LAS VARIABLES CLÁSICAS CONTINUAS.

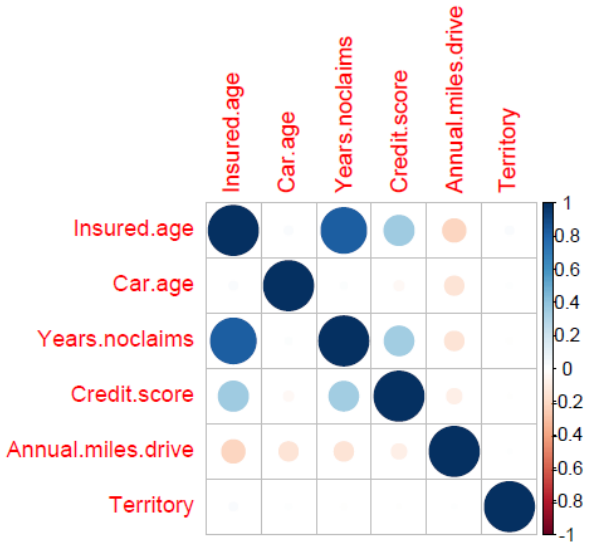


Figura. Elaboración propia

Entre la variable *Insured.age* y *Years.noclaims*, hay una fuerte relación positiva con una correlación de 0.829 por lo que se decide eliminar el factor años sin siniestros ya que la edad del asegurado es de las más importantes en los seguros de autos.

Para las categóricas se realiza el análisis mediante el coeficiente V de Cramer que está diseñado para detectar relaciones no lineales entre variables categóricas. Este, oscila entre 0 y 1 y mide de manera simétrica la relación entre variables de la siguiente manera:

$$V = \frac{\sqrt{X^2}}{\sqrt{n(min[r,c]-1)}}$$

Cuanto más cercano a 1, mayor es la relación y, a partir de 0.6, se podría decir que la relación es lo suficientemente importante como para poder suprimir una variable, ya que ambas explicarían lo mismo. Como ninguna relación está por encima del 0.3, no se procede a eliminar ninguna categoría de estos predictores.

4.2.1. Modelo de Frecuencia GAM.

La frecuencia se modela a partir de todos los datos que se tienen de la cartera, teniendo en cuenta la distribución que sigue, una Binomial Negativa.

Mediante el paquete de R *mgcv* se realiza un bucle utilizando todas las posibles combinaciones de las variables clásicas, guardando cada uno de los resultados de las métricas AIC y BIC para poder seleccionar el mejor modelo. En esta primera etapa no se buscan interacciones y se aplica un suavizado en los factores de riesgo que son continuos.

Antes de comenzar con la búsqueda del mejor modelo GAM, se establece para las variables categóricas lo que será el nivel de referencia o nivel base. Para evitar problemas de multicolinealidad, a la hora de tarificar, se establece este nivel de referencia que suele ser la categoría con más exposición en la cartera y los resultados de los coeficientes están referenciados a esta.

A continuación, se realiza el proceso de búsqueda del mejor modelo GAM a partir del siguiente bucle.

ILUSTRACIÓN 6 - ALGORITMO PARA AJUSTAR MODELO GAM DE FRECUENCIA

Algorithm 1: Selección del mejor modelo GAM entre las posibles combinaciones de las variables clásicas

Data: Datos de entrenamiento *train_classic*
Entrada: Combinaciones de las variables clásicas explicativas (511)

```
metrics ← crearTablaVacia con las métricas de los modelos();
for i in 511 do
    formula ← ();
    fit GAM model ← ajustarModeloGAM(formula, train_classic);
    variables ← obtenerVariables(combos[[i]]);
    AIC ← calcularAIC(model);
    BIC ← calcularBIC(model);
    guardarMétricasEnTabla(metrics, variables, AIC, BIC);
end
```

Result: Tabla de los mejores modelos GAM con sus métricas AIC y BIC

Fuente: Elaboración propia

Una vez se ejecuta el algoritmo donde se realiza un bucle con todas las posibles combinaciones de las variables clásicas, se crea una tabla con los factores que han entrado en el modelo junto a los valores de sus criterios de información.

TABLA 8 – MÉTRICAS DE LOS CINCO MEJORES MODELOS GAM DE FRECUENCIA ESTIMADOS CON LAS VARIABLES CLÁSICAS.

Modelo	variables	AIC	BIC
1	s(Insured.age) + s(Car.age) + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	21794,38	22071,99
2	s(Insured.age) + s(Car.age) + Marital + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	21795,66	22082,28
3	s(Insured.age) + Insured.sex + s(Car.age) + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	21795,83	22082,68
4	s(Insured.age) + Insured.sex + s(Car.age) + Marital + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	21797,22	22093,09
5	s(Insured.age) + s(Car.age) + Car.use + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	21797,51	22102,44
6	s(Insured.age) + s(Car.age) + Marital + Car.use + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	21798,69	22112,64

Fuente: Elaboración propia.

Están ordenados por AIC de menor a mayor. El modelo que se elige es el número 1, ya que minimiza ambos valores. El modelo mantiene las variables continuas y la variable categórica de la región.

$$\log\left(E(NB_{Claim})\right) \sim \log(Exposure) + \beta_0 + f_1(Insured.age) + f_2(Car.age) + \beta_1 Region_{Rural} + f_3(Credit.score) + f_4(Annual.miles.drive) + f_5(Territory)$$

ILUSTRACIÓN 7 - SALIDA DEL MODELO GAM DE FRECUENCIA.

```
Family: Negative Binomial(1.272)
Link function: log

Formula:
NB_Claim ~ s(Insured.age) + s(Car.age) + Region + s(Credit.score) +
    s(Annual.miles.drive) + s(Territory)

Parametric coefficients:
            Estimate Std. Error  z value Pr(>|z|)
(Intercept) -3.21692    0.02560 -125.668  < 2e-16 ***
RegionRural -0.17927    0.05447   -3.291 0.000998 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                       edf Ref.df Chi.sq  p-value
s(Insured.age)        5.670  6.618  66.01  < 2e-16 ***
s(Car.age)            2.125  2.679 179.63  < 2e-16 ***
s(Credit.score)       4.783  5.810 268.77  < 2e-16 ***
s(Annual.miles.drive) 6.329  7.156  45.70  < 2e-16 ***
s(Territory)          4.395  5.381  27.71 7.05e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.0189   Deviance explained = 4.97%
-REML =  10922  Scale est. = 1         n = 67746
```

Fuente: Elaboración propia.

El intercepto tiene un valor estimado de -3.2169 con un error estándar de 0.0256. Es altamente significativo, lo que indica una fuerte influencia en la frecuencia de los siniestros.

Sobre los términos que han sido suavizados, muestran una relación no lineal con la frecuencia como indican los grados de libertad efectivos (edf) ya que, un edf mayor que 1 sugiere que la relación no es lineal y permite una mayor flexibilidad en la forma de la curva al ajustar el modelo. Además, los efectos de estas son significativos tal y como refleja el alto nivel del p-valor ($p < 2e\text{-}16$).

GRÁFICO 9 - EFECTO DE LAS VARIABLES CONTINUAS EN EL MODELO DE FRECUENCIA GAM.

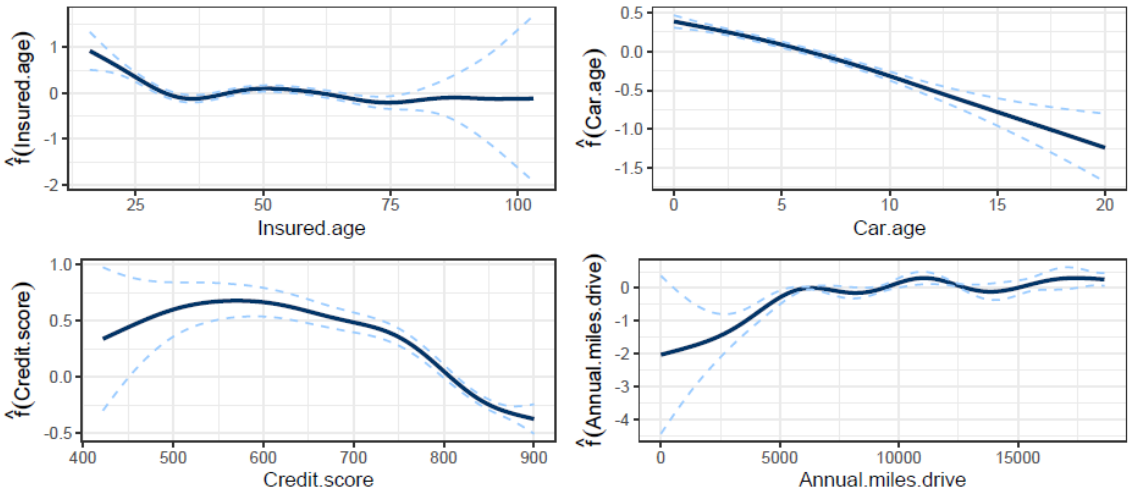


Figura. Elaboración propia

Si se analizan gráficamente los efectos de las variables suavizadas, primero se puede apreciar como los asegurados con más frecuencia son los más jóvenes de la cartera, con un descenso continuo hasta los 30 donde se encontraría el mínimo. Después se mantiene bastante constante con alguna subida hasta los 50 y un leve descenso hasta los 75 acabando con un ligero incremento final donde la exposición ya es muy baja. En cuanto a la edad del coche, tras haber eliminado aquellos que tenían una edad negativa, se observa como a mayor antigüedad, menor es el riesgo de sufrir un accidente y por tanto son los coches más nuevos los que tienen más riesgo.

La variable *credit score* comienza con un crecimiento durante las zonas de menor exposición donde por el valor del *scoring* se les podría establecer como personas con mayor riesgo financiero, mientras que luego la curva al llegar a un valor de 600 empieza a decrecer de manera persistente. Se indica que, a mayor responsabilidad financiera, menor frecuencia de siniestros. Sobre el total de millas conducidas durante el año, como era de esperar, aumenta de manera constante al tiempo que aumenta la acumulación de millas. Este incremento es más fuerte hasta que llega a las 6.000 donde hay tramos sin subidas, pero con una tendencia general siempre creciente.

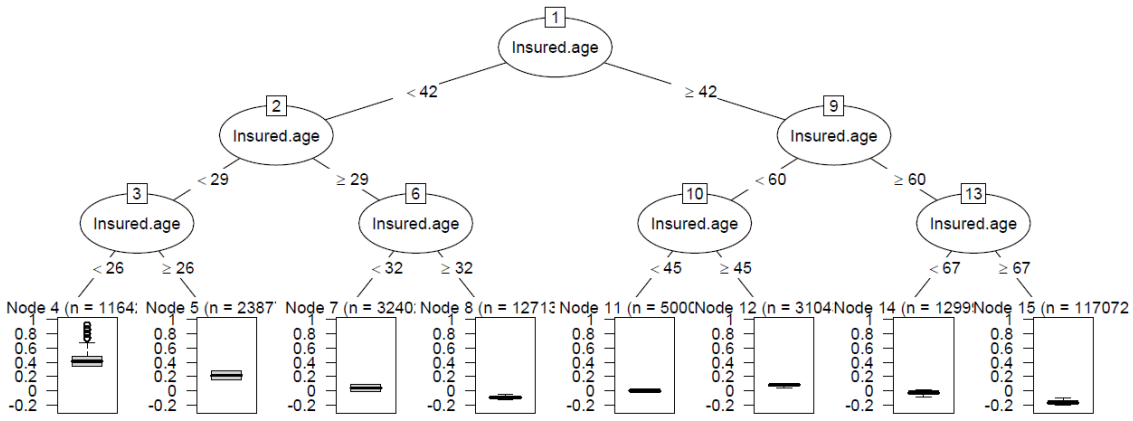
- Agrupaciones de las variables

Una vez esto, se procede a realizar agrupaciones consecutivas de las variables continuas mediante el uso de árboles de regresión. Se usa un método clásico de árboles de regresión, Árboles de Clasificación y Regresión, en inglés *Classification And Regression Trees* (CART), propuestos por Breiman et al. (1984). Este método realiza particiones recursivas que ajustan un modelo mediante una búsqueda paso a paso. En cada paso, se eligen divisiones para maximizar la homogeneidad de estas particiones y estas divisiones consecutivas se mantienen fijas en todos los siguientes pasos.

Se va a utilizar el paquete de R *evtree* desarrollado por Grubinger et al. (2014). Incorpora árboles evolutivos que combinan el marco de los árboles de regresión con algoritmos que cambian la estructura del árbol en cualquier nodo posible hasta que se alcanza la convergencia hacia una solución óptima, resultando en una partición más robusta de nuestros factores de riesgo continuos.

Para este proceso, la variable respuesta que se usa es el factor continuo estimado en el mejor modelo GAM y el regresor incluido es el mismo factor de riesgo pero con sus datos observados. La profundidad máxima de los árboles se establece en 3 nodos y se indica que, mínimo, cada uno tenga el 10% del conjunto de datos con un valor alpha intermedio. Para el ejemplo de la edad del asegurado, el resultado obtenido es el siguiente.

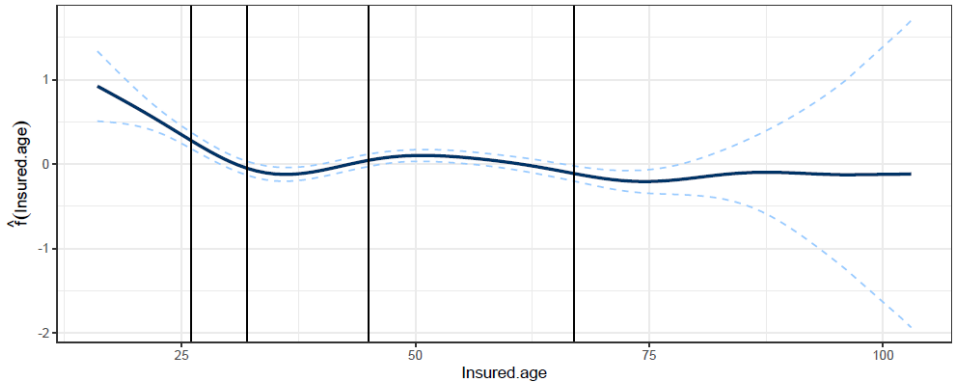
GRÁFICO 10 - ÁRBOL DE REGRESIÓN DE LA EDAD DEL ASEGURADO PARA REALIZAR AGRUPACIONES EN FRECUENCIA.



Fuente: Elaboración propia.

Por lo que la variable se agruparía en 5: De 16 a 26 años, de 26 a 32, de 32 a 45, de 45 a 67 y mayores de 67. De manera gráfica queda de la siguiente manera.

GRÁFICO 11 - AGRUPACIONES DE LA EDAD DEL ASEGURADO EN FRECUENCIA.



Fuente: Elaboración propia.

Para el resto de las variables, los gráficos se pueden ver en los anexos donde el resultado de los grupos finales son los siguientes.

- Car.age: [0 - 2), [2 - 6), [6 - 9), [9 - 13) y [13 - 20]
- Credit Score: [422 - 730), [730 - 802), [802 - 838), [838 - 883) y [884 - 900]
- Annual.miles.drive: [0 - 3169), [3169 - 5530), [5530 - 12241) y [12241 - 18642]

La variable territorio no necesita unirse de manera continua y no está bien identificada en la base de datos. Se podría tratar como una variable espacial, pero en este caso no se le aplica ninguna agrupación.

4.2.2. Modelo de Severidad GAM

Al igual que con el modelo de Frecuencia, se estima un modelo GAM que suaviza las relaciones no lineales entre las variables continuas. En este caso, al tratarse de una distribución gamma que sigue la variable dependiente valor medio de los costes del siniestro, se filtra por las pólizas que han tenido, al menos, un siniestro y por tanto, la cuantía es mayor que cero. Además, como ya se ha comentado durante el análisis estadístico descriptivo, los datos atípicos fueron eliminados, quedando un total de 2.571 observaciones en el conjunto de entrenamiento.

Respecto a la cartera de asegurados, los datos de las cuantías son agregados por póliza, no individuales, por lo que, para obtener el valor medio de los siniestros, se utiliza el número total de siniestros.

A continuación, como se hizo anteriormente en la ilustración 7, se busca el modelo GAM óptimo para la severidad distribuido como una Gamma. Se utilizan todas las combinaciones de las variables clásicas y se obtiene un ranking de los mejores modelos según los criterios AIC y BIC.

TABLA 9 – MÉTRICAS DE LOS CINCO MEJORES MODELOS GAM DE SEVERIDAD ESTIMADOS CON LAS VARIABLES CLÁSICAS.

Modelo	variables	AIC	BIC
1	s(Insured.age) + Insured.sex + s(Car.age) + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	46590,33	46733,27
2	s(Insured.age) + s(Car.age) + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	46591,27	46727,91
3	s(Insured.age) + Insured.sex + s(Car.age) + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	46592,00	46740,01
4	s(Insured.age) + s(Car.age) + Region + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	46592,72	46718,31
5	s(Insured.age) + Insured.sex + s(Car.age) + Car.use + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	46594,03	46755,04
6	s(Insured.age) + Insured.age + s(Car.age) + Marital + s(Credit.score) + s(Annual.miles.drive) + s(Territory)	46594,25	46742,94

Fuente: Elaboración propia.

Al tener una muestra mucho más pequeña, ya que solo se seleccionan las pólizas que han tenido siniestros, los modelos ajustan peor y pueden provocar que menos variables sean significativas. Se sigue el criterio anterior, pero en este caso debido a que el valor del AIC es similar en los dos primeros casos, se escoge el segundo donde solo entran las variables continuas ya que tiene un BIC menor y, además, se satisface el principio de parsimonia.

$$\log\left(E(Claim_{AMT}/NB_{Claim})\right) \sim \beta_0 + f_1(Insured.age) + f_2(Car.age) + f_3(Credit.score) + f_4(Annual.miles.drive) + f_5(Territory)$$

En la salida del modelo se ve que son todos los regresores significativos al igual que el intercepto y los valores edf mayores que 1.

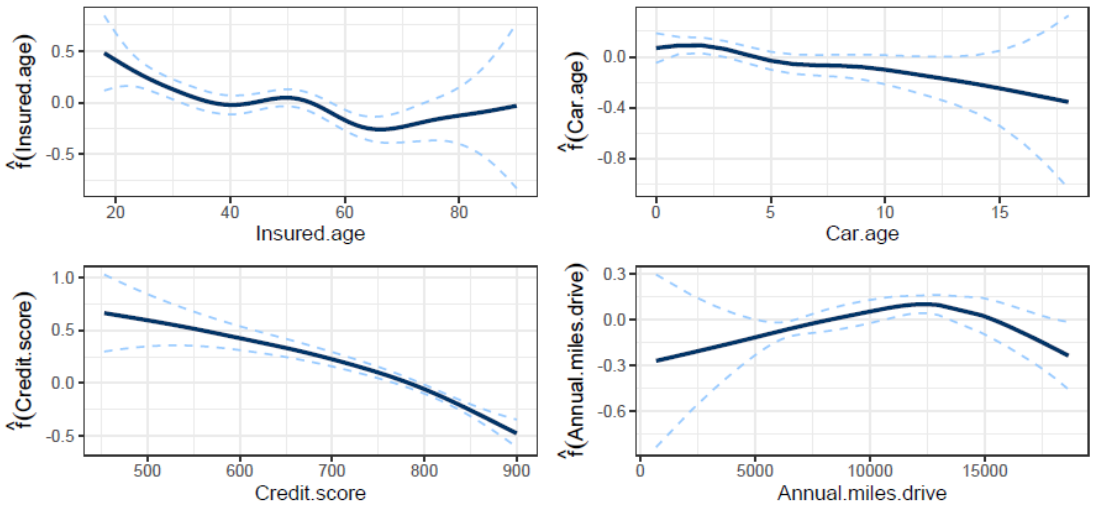
ILUSTRACIÓN 8 - SALIDA DEL MODELO GAM DE SEVERIDAD.

```
Family: Gamma
Link function: log

Formula:
AMT_avg ~ s(Insured.age) + s(Car.age) + s(Credit.score) + s(Annual.miles.drive) +
    s(Territory)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.00150    0.02486   321.8   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                       edf Ref.df      F  p-value
s(Insured.age)       5.377  6.468  5.104 2.57e-05 ***
s(Car.age)           3.763  4.653  2.585  0.03640 *
s(Credit.score)      2.407  3.039 31.686  < 2e-16 ***
s(Annual.miles.drive) 2.622  3.228  4.508  0.00293 **
s(Territory)         7.181  8.211  5.170 1.76e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0626   Deviance explained = 11.9%
GCV = 1.2359  Scale est. = 1.5895    n = 2571
```

Fuente: Elaboración propia.

Los efectos de las variables continuas se pueden ver en el siguiente gráfico.

GRÁFICO 12 - EFECTO DE LAS VARIABLES CONTINUAS EN EL MODELO DE SEVERIDAD GAM.



Fuente: Elaboración propia.

El factor que conforma la edad del asegurado tiene una pendiente ligeramente descendiente, los conductores más jóvenes son los que provocan siniestros de mayor

cuantía reduciéndose hasta los 40 donde hay un incremento hasta los 45 donde vuelve a descender la severidad. Sin embargo, crece de nuevo tras los 65 años sin llegar a los niveles de los conductores más amateurs, coincidiendo con la zona de menor exposición.

Sobre la edad del coche ocurre lo que se esperaba, donde a mayor edad del coche menor es la cuantía del siniestro, pero a pendiente es más suave que en la frecuencia. Los coches nuevos tienen más valor que los antiguos y eso provoca este efecto. Solo podría ocurrir una subida de la severidad si fuesen coches clásicos, pero esos son catalogados así a partir de los 30 años normalmente y no hay edades que lleguen a esa cifra en la cartera.

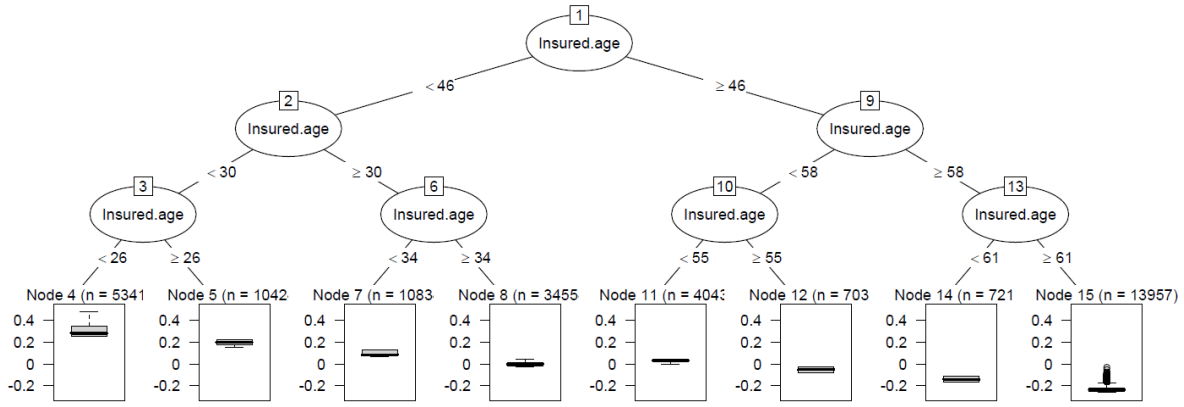
El factor de *credit score* empieza con los valores de severidad más bajos en la zona de peor calidad crediticia. Debido a que, al ser usuarios que tienen menor *credit score*, su nivel económico es más bajo y podría ocurrir que sus vehículos fuesen menos valiosos.

Sobre el número de millas conducidas en un año, tiene un crecimiento lineal de la severidad durante las primeras 12.000 millas, y es a partir de ese punto donde decrece hasta el total de millas máximo. Este resultado es interesante ya que se muestra que, a partir de un número de millas recorridas, la severidad disminuye de forma clara.

- Agrupaciones de las variables

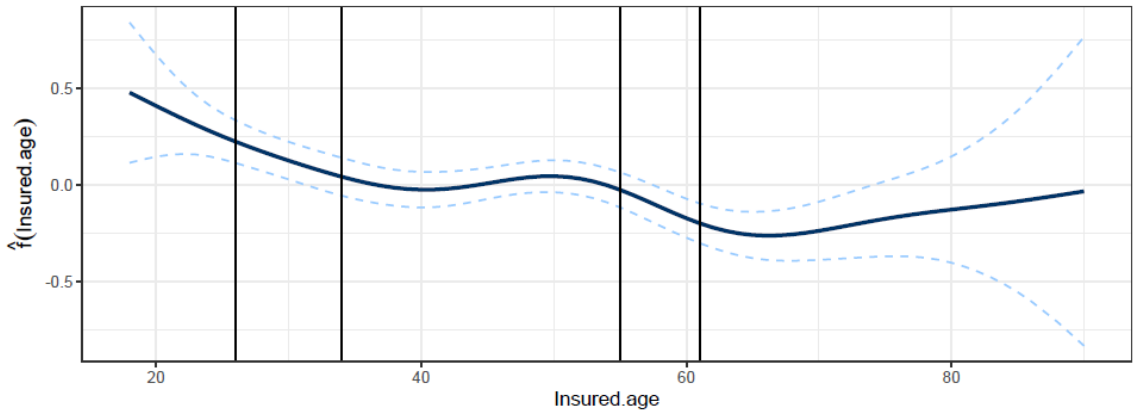
El mismo proceso para generar grupos usando arboles de regresión se utiliza para los factores continuos que han entrado en el modelo. Los grupos que se generan cambian de forma ligera respecto a los establecidos en los modelos de frecuencia. Para el factor edad del asegurado quedarían de la manera que se ve a continuación.

GRÁFICO 13 – ÁRBOL DE REGRESIÓN PARA LAS AGRUPACIONES



Fuente: Elaboración propia.

GRÁFICO 14 – AGRUPACIONES DE LA EDAD DEL ASEGURADO EN SEVERIDAD.



Fuente: Elaboración propia.

Para el resto de las variables que entran dentro del modelo de severidad, las agrupaciones que quedan finalmente son:

- Car.age: [0 - 2), [2 - 5), [5 - 8) y [8 - 12] y [12 - 20]
- Credit Score: [422 - 807), [807 - 870), [870 - 900]
- Annual.Miles.drive: [0 - 5717), [5717 - 9258), [9258 - 12489], [12489 - 18642]

El factor territorio se vuelve a deja sin agrupar, pero se mantiene en el modelo ya que es significativo.

4.3. Traspaso del modelo GAM a uno GLM.

Tras haber trabajado en el modelo GAM y usar la técnica árboles de regresión para las agrupaciones, se procede a trasladar todo a un modelo GLM. Esto se realiza ya que es el modelo estándar en la tarificación por sus ventajas ya comentadas.

Si comparamos los modelos GLM en los datos de entrenamiento tras las agrupaciones, con los modelos GAM tanto de frecuencia como se severidad, se ve que el GLM mejora en ambos casos en cuanto al BIC se refiere. Por otro lado, el AIC no mejora para la frecuencia, pero si lo hace de manera leve en severidad.

TABLA 10 – MÉTRICAS AIC Y BIC DE LOS MODELOS GAM Y GLM.

	Frecuencia		Severidad	
Tipo	AIC	BIC	AIC	BIC
GAM	21794,38	22071,99	46591,27	46727,91
GLM	21850,64	22060,48	46466,12	46559,75

Fuente: Elaboración propia.

Esta comparación muestra el equilibrio entre flexibilidad y simplicidad en el proceso de modelado. Los GAM son una posible mejor opción para lograr flexibilidad, dado que penalizan menos la complejidad según el AIC. Por otro lado, los GLMs son la opción preferida para lograr simplicidad, dado que penalizan más la complejidad según el BIC.

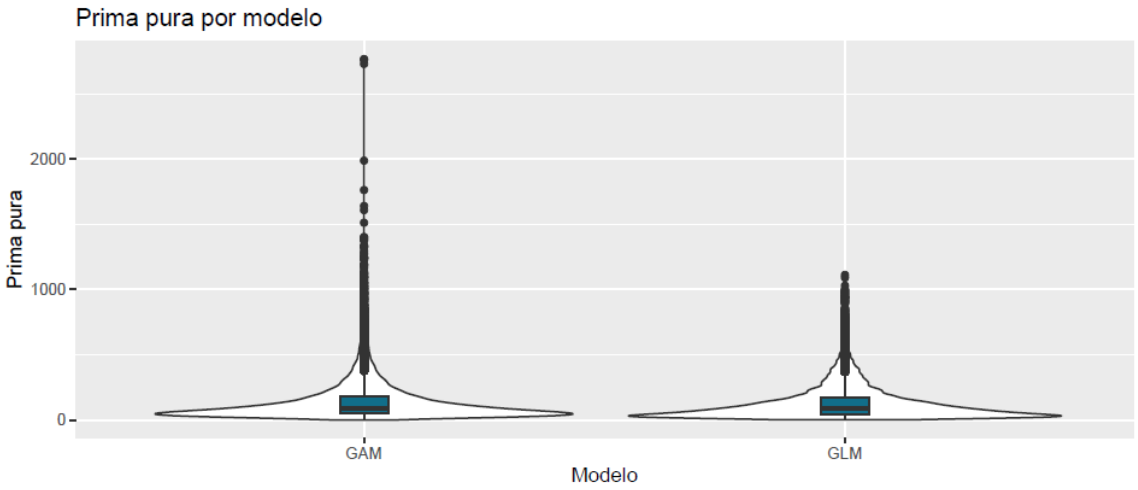
Si se analiza primero el modelo GLM de frecuencia con agrupaciones que se encuentra en el Anexo L, para la edad del asegurado se ve que son significativos todos los grupos menos el que está entre 26 y 32 años. El valor del grupo con edades más jóvenes tiene un coeficiente positivo lo que indica que es de mayor riesgo comparado con el nivel base, mientras que los otros dos darían lugar a una menor frecuencia.

Respecto a la edad del coche, todos los grupos son significativos, siendo el formado entre los dos primeros años del vehículo el que tiene un coeficiente positivo por lo que el riesgo en ese grupo es mayor respecto al base. Si se analizan el resto de los regresores continuos, los efectos son similares a los que visualmente se han visto tras la modelización GAM. En cuanto a la región rural, la frecuencia es menor que en la urbana con una significatividad alta.

En severidad los valores de los coeficientes respecto a los niveles base siguen la misma tendencia que los gráficos del modelo GAM. Sin embargo, hay algunos segmentos dentro de los factores que no serían significativos como por ejemplo la edad del asegurado entre los 55 y 61 años o la edad del coche durante su primer año.

A partir de la modelización en el conjunto de datos separado para el entrenamiento y calibración, se ajustan los datos en los datos que se han dejado para hacer el *testing* y se calcula la prima para cada asegurado según sus factores de riesgo.

GRÁFICO 15 - COMPARATIVA DE LA PRIMA PURA PARA EL MODELO GLM Y GAM.



Fuente: Elaboración propia.

La suma de la prima total para los 29.035 asegurados que se han dejado en el 30% de *testing* en el modelo GLM con agrupaciones sería de 3.759.335$ mientras que para el modelo GAM de 4.139.894$ Dando lugar a una prima media de ambas carteras respectivamente en 129.48$ y 142.58$ y como muestra el anterior gráfico, en el cuerpo de la distribución no hay diferencias entre ambos modelos.

5. INCORPORACIÓN DE VARIABLES TELEMÁTICAS.

En este apartado, se va a realizar una comparativa añadiendo las variables telemáticas de las que se disponen para poder estudiar sus resultados comparando con modelos que solo incluyen variables clásicas. La modelización se hará a partir de técnicas de aprendizaje supervisado de *machine learning* y mediante modelos GBM.

5.1. Análisis estadístico de las variables telemáticas.

En primer lugar, se va a realizar un análisis estadístico descriptico de estas variables para poder entender mejor la forma y manera en la que se presentan. El conjunto de la base de datos con la que se trabaja contiene un total de 39 variables telemáticas. De estas, muchas están expandidas por ejemplo en días de la semana u horas de conducción como se indica en las 11 variables principales que se muestran en la Tabla 1.

5.1.1. Variables Pay-how-you-drive (PHYD).

Se va a comenzar con las variables que entran dentro de la categoría PHYD, estas incluyen los estilos de conducción de los asegurados.

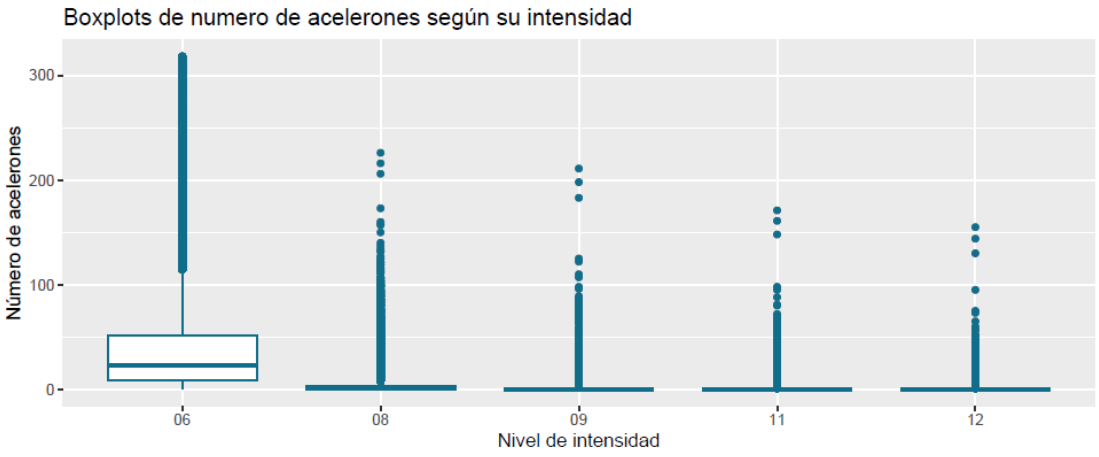
Los factores de riesgo disponibles en el conjunto de datos referidos a la forma de conducir se dividen en tres grupos y, dentro de cada uno, se subdividen en los niveles de intensidad con los que realizan esas acciones. Estas categorías son: Cantidad de acelerones y frenazos dados durante 1.000 millas, y la intensidad con la que se han generado los giros tanto a izquierdas como a derechas medido en mph/s.

- Número de aceleraciones cada 1000 millas.

Cuenta las veces que se han producido los siguientes números de acelerones de 6, 8 9, 11, 12 y 14 mph/s durante 1.000 millas recorridas (1610 km). El equivalente de mph/s en km/h serian 9,66; 12,87; 14,48; 17,70; 19,31 y 22,53 respectivamente.

Hay varias pólizas que pueden afectar a la distribución de estas variables ya que cuentan con algunos *outliers*. Para eliminarlos, se ha realizado una extracción de las observaciones que superaban el cuantil 99% (Un total de 973) de la variable acelerones de intensidad 06 y se ha quedado el siguiente gráfico donde el total de pólizas es 95.808.

GRÁFICO 16 - NÚMERO DE ACELERONES SEGÚN SU INTENSIDAD.



Fuente: Elaboración Propia

La media de número de acelerones de intensidad 06 mph es de 39,36 y el máximo es de 318. Para el resto, la media va decayendo a 3,41; 1,135; 0,546; 0,279 y 0,174 en el caso de intensidades 08, 09, 11, 12 y 14.

- Número de frenazos cada 1.000 millas.

Cuenta las veces que se han producido fuertes frenazos de 6, 8 9, 11, 12 y 14 mph/s durante 1.000 millas recorridas.

Se vuelven a encontrar valores atípicos por lo que se sigue el mismo procedimiento y se eliminan los valores que superan el percentil 99,9% que forman un total de 96 registros.

GRÁFICO 17 – NÚMERO DE FRENAZOS SEGÚN SU INTENSIDAD.

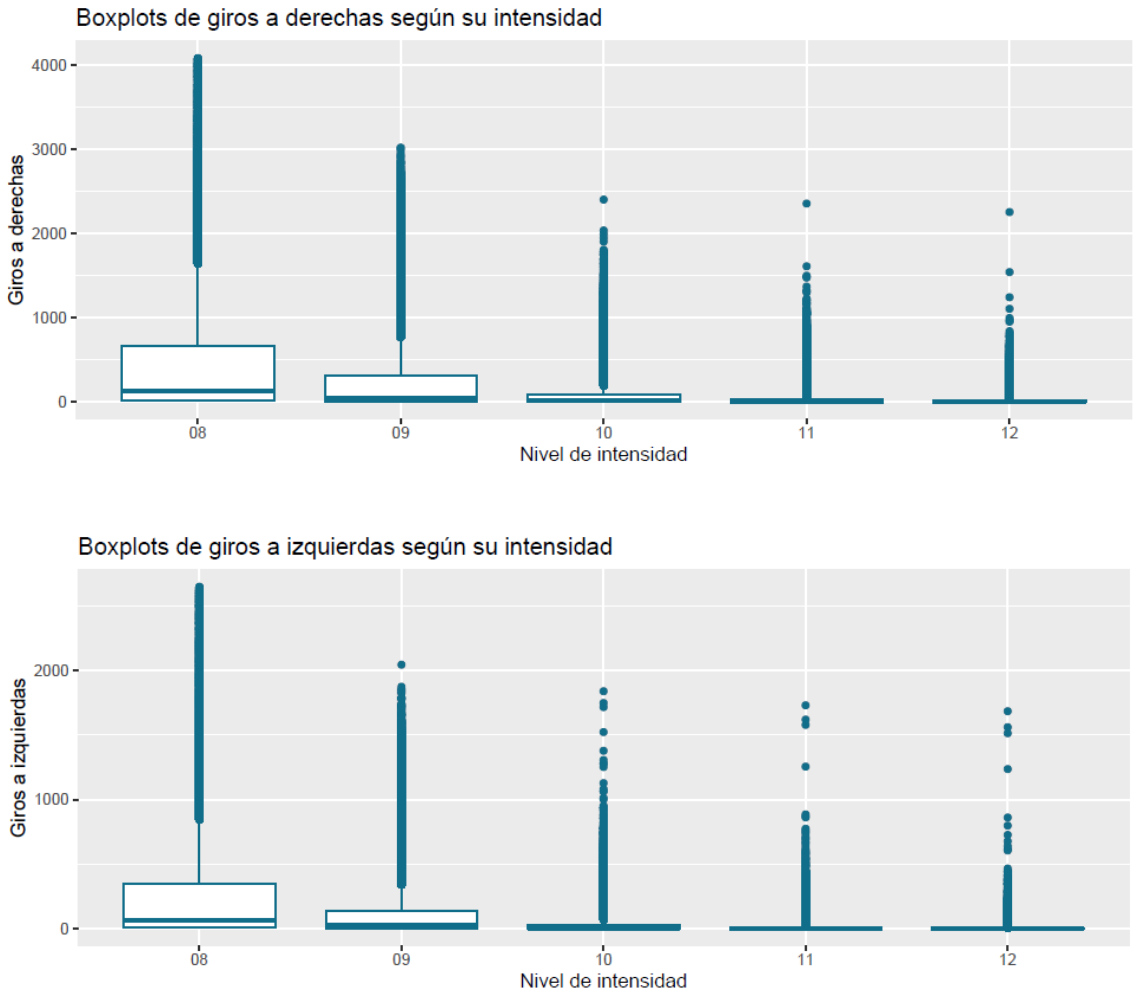
Fuente: Elaboración Propia

Como es de esperar los frenazos de intensidad 06 son los que tienen mayores registros con una media de frenazos de 80,98, muy superior a la de los acelerones. Para el resto de los niveles pasa lo mismo que en la anterior categoría, donde se sitúa en torno al 0 salvo algunos casos donde es mayor.

- Intensidad de los giros.

El último factor de este apartado se divide en la intensidad con la que se producen los giros a la izquierda y giros a la derecha. Se categorizan en niveles del más bajo que es intensidad 8 al más alto que es 12.

GRÁFICO 18 – GIROS A DERECHAS E IZQUIERDAS SEGÚN SU INTENSIDAD.



Fuente: Elaboración Propia

Ambos resultados son similares a la dinámica de reducción de frecuencia según aumenta el nivel de la intensidad. Sin embargo, de forma genérica, el número de giros a derechas se puede observar que ocurre más veces que a izquierdas.

Se ha repetido el proceso de eliminación de datos atípicos ya que desvirtúan de manera constante las distribuciones de las variables. Puede ser que en ocasiones los resultados que provienen de estas *black-box* instaladas en los autos donde se recogen los datos, por problemas de señal, no trasmitan correctamente los valores. Tras finalizar con el análisis de las variables PHYD la cartera cuenta con 93.806 pólizas.

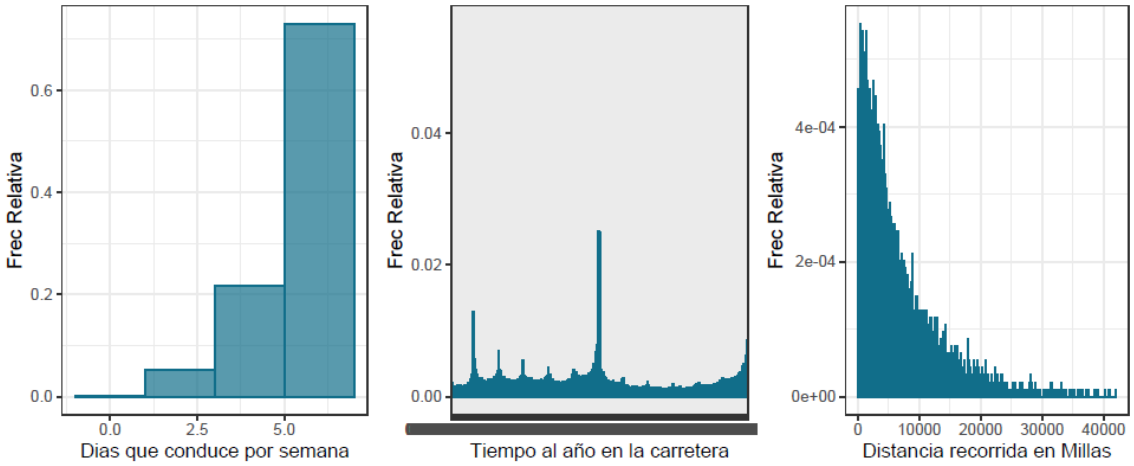
5.1.2. Variables Pay-as-you-drive (PAYD).

Una vez analizados cómo son los estilos de conducción en la cartera de asegurados, se procede a estudiar el resto de las variables telemáticas. Estas, están relacionadas con los hábitos de conducción.

- Distancia recorrida, tiempo al año en carretera y días que conduce por semana.

En estos tres primeros factores de riesgo, se puede analizar a mayores rasgos cómo es el comportamiento de los conductores de esta base de datos durante el año de exposición.

GRÁFICO 19 – DÍAS QUE CONDUCE POR SEMANA, TIEMPO AL AÑO EN CARRETERA Y DISTANCIA RECORRIDA.



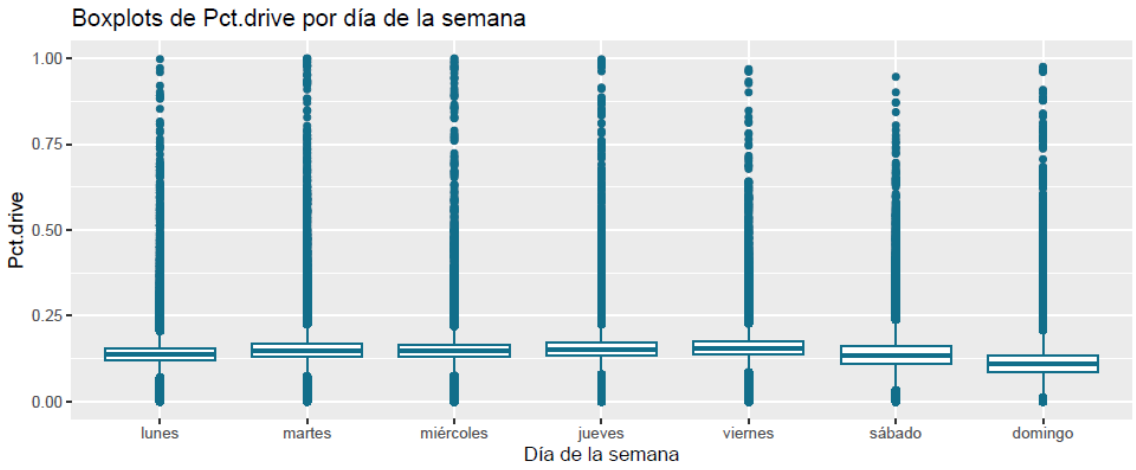
Fuente: Elaboración Propia

El gráfico de la izquierda muestra la media de días que pasan conduciendo, donde algo más del 70% conduce entre 6 y 7 días semanales, en torno al 20% de 3 a 5 días a la semana y, el resto, menos de 3 días por semana, siendo el porcentaje de días que no conducen de 0%. En relación con esto, el tiempo durante el año que se ha pasado en carretera muestra que la frecuencia siempre es mayor que 0 ya que como se ha comentado, ese porcentaje era de 0. Esta variable va de 0 a 1 y el significado es equivalente al del término de exposición ya que el tiempo que ha pasado en la carretera es el tiempo que ha estado expuesto a tener algún siniestro. Es por ello por lo que, aunque esté a diferente escala que en el gráfico 5, la distribución de los datos es la misma y la mayoría de los asegurados han pasado todos los días en la carretera y si no, al menos la mitad de año.

La distancia anual recorrida en millas muestra una distribución asimétrica a la derecha, donde la mayor parte de los conductores han conducido hasta 6.735,7 millas que es donde se encuentra el tercer cuantil que deja atrás el 75% de la distribución. La media es de 4.787,2 millas (7.705 km) y el máximo ha sido de 42.000 millas (67.600km).

- Predicción de conducción por días de la semana.

GRÁFICO 20 – TIEMPO CONDUCIENDO POR CADA DÍA DE LA SEMANA.



Fuente: Elaboración Propia

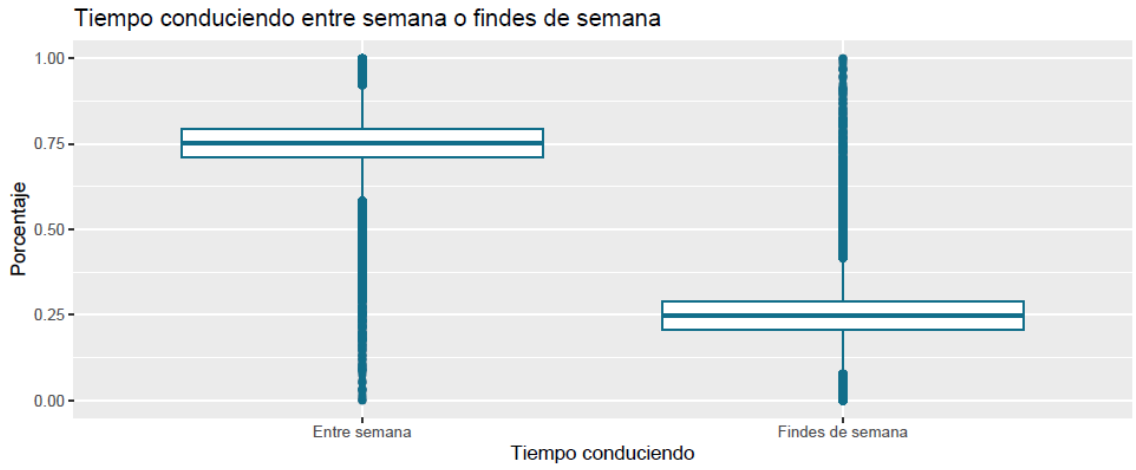
Como se puede ver, la distribución de los días de conducción en la cartera es bastante similar en todos los casos. De manera ligeramente superior, los días que más se conduce son los viernes, por encima del jueves y miércoles.

Por otro lado, se puede ver un descenso tras el viernes donde el domingo se establece como el día de la semana que menos se pasa sobre la carretera. Se puede decir que el patrón se convierte en un ligero incremento del tiempo conduciendo desde el lunes, hasta que se llega al viernes donde decrece hasta el domingo.

- Tiempo de conducción durante la semana.

Acorde con las variables anteriores, la diferencia entre el tiempo de conducción entre los días de la semana y los fines de semana es notable.

GRÁFICO 21 – TIEMPO CONDUCIENDO ENTRE FINDES DE SEMANA/ENTRE SEMANA.



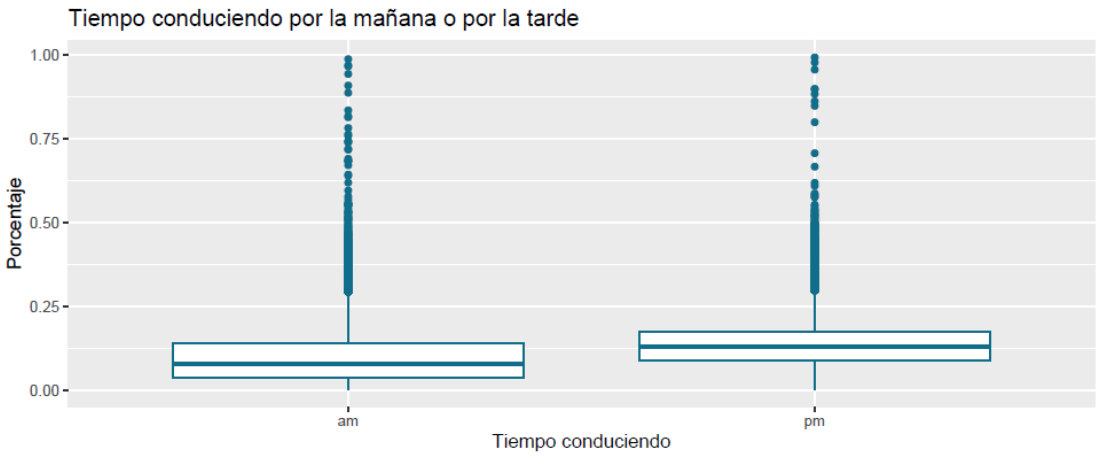
Fuente: Elaboración Propia

El 75% pasa más tiempo de conducción los días entre semana, mientras que el 25% restante del tiempo conducen los sábados y domingos. Este resultado, aparte de ser intuitivo, concuerda con el uso del vehículo que tienen los asegurados de la cartera. Donde el principal uso era *commute*, es decir, uso del auto para ir al trabajo.

- Tiempo que ha conducido entre el día y la tarde/noche.

La siguiente variable muestra del tiempo que ha pasado en carretera, cuánto ha sido entre las horas a.m. (00.00 – 11.59) que son por la mañana y p.m. (12.00 – 11.59) que son por la tarde y noche.

GRÁFICO 22 – TIEMPO CONDUCIENDO ENTRE AM/PM.



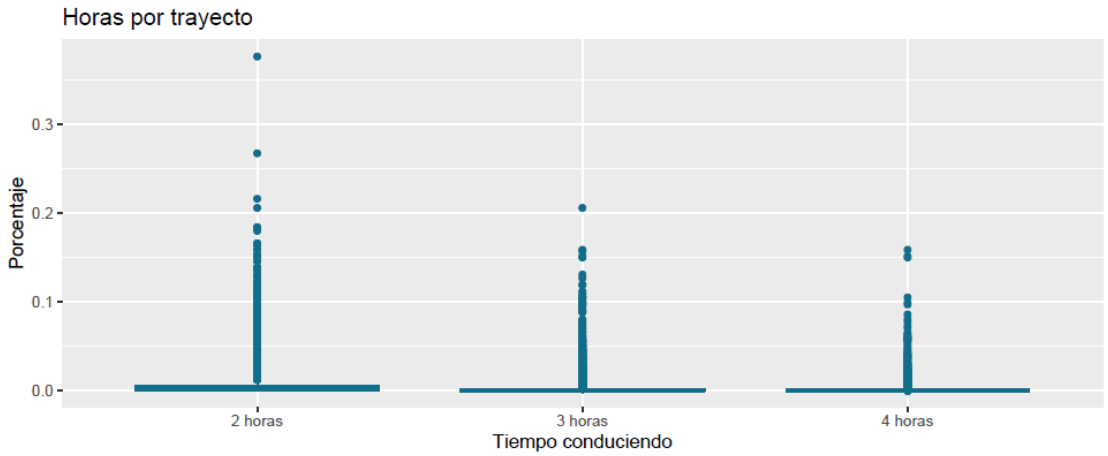
Fuente: Elaboración Propia

Del tiempo total de cada día, al final, pasan en la carretera en torno a un 14% por la tarde y un 10% por el día. El sentido de estos datos es que por la mañana se usa el coche para ir a trabajar y por la tarde, para volver del trabajo y para el resto de las actividades que se puedan tener, lo que hace que el porcentaje sea algo mayor en el rango p.m.

- Horas que pasa conduciendo.

El siguiente gráfico muestra los trayectos en los que las horas de conducción sobrepasan un tiempo establecido. Este tiempo en la base de datos se divide entre 2, 3 y 4 horas al volante.

GRÁFICO 23 – HORAS POR TRAYECTO.



Fuente: Elaboración Propia

Como se puede apreciar, las tres variables rondan el 0 siendo la de 2 horas la que tiene algunas pólizas con conductores que sí pasan una mayor parte del tiempo en torno a las 2 horas. En cuanto a las 3 y 4 horas, la mayoría no pasa tantas horas en el coche y, si lo hace, son muy pocas veces durante el año. Esas veces que sobrepasan las 2 horas al volante deben ser en momentos especiales donde realizan largos desplazamientos, ya sea por motivo vacacional o por trabajo.

5.2. Herramienta H2O en Rstudio.

Para trabajar con técnicas de ML en las variables telemáticas que incluye la base de datos, se va a usar un paquete de R llamado *H20*. Es un producto desarrollado por H2O.ai para trabajar con los principales algoritmos de *machine learning* con grandes cantidades de datos y requiere del uso de Java.

La herramienta dentro de R necesita iniciar un *cluster*, y todos los datos que se introduzcan a R se encontraran dentro de ese *cluster* de H2O, no en la memoria.

Antes de establecer los modelos, es importante tener en cuenta la optimización de los hiperparámetros. Estos, son valores que se establecen en los modelos de ML y, dependiendo del valor que tomen, los resultados pueden cambiar de manera significativa. Hay situaciones donde estos valores, por las características de los datos, no pueden aprenderse y se establecen de manera manual por el actuario, pero, con la práctica y la experiencia, ganan intuición sobre los mejores valores que pueden tomar dependiendo del proyecto en el que estén trabajando. Pero la forma más común de hallar los valores óptimos es a través de lo que se conoce como *model tunning*, que consiste en elegir entre diferentes posibilidades el que mejor se ajuste a lo que se pretenda modelizar.

5.3. Modelos GLM Regularizados.

La regularización se incluye en el modelo lineal generalizado a partir de los modelos *ridge regression, lasso* y *elastic net*. Esta regularización se incorpora para mejorar posibles problemas de sobreajuste, reducir la varianza y reducir los efectos que pudiese haber de correlaciones entre las variables.

- *Ridge regression* (l2): Consiste en un modelo lineal por mínimos cuadrados, pero añade una penalización en la suma de los coeficientes elevados al cuadrado. El

efecto que provoca es la reducción proporcional del valor de todos los coeficientes sin que lleguen a 0. El hiperparámetro que controla esta regularización es λ.

$$\left(\|\beta\|_2^2 = \sum_{k=1}^{p} \beta_k^2\right)$$

- *Lasso* (l1): También es un modelo lineal por mínimos cuadrados, pero penaliza la suma del valor absoluto de los coeficientes de regresión. Su efecto es el de forzar a que los coeficientes de los predictores tiendan a 0, por lo que, los que lleguen a este valor, ya no entrará en el modelo y por tanto se seleccionarán solo los más influyentes. Su hiperparámetro es λ, y cuanto mayor sea este valor, mayor será la regularización y más predictores estarán excluidos. Hay que tener en cuenta que cuando se modeliza con una función de enlace de una familia exponencial, esta regularización l1 no tiene una solución analítica para encontrar la máxima verosimilitud por lo que se utiliza un método de optimización.

$$\left(\|\beta\|_1 = \sum_{k=1}^{p} |\beta_k|\right)$$

- *Elastic net*. Este método de regularización combina los de *lasso* y *ridge*. El valor del hiperparámetro que lo controla es de $\alpha \in [0,1]$, cuanto más cerca de 0, dará más valor al método de *ridge* y, cuanto más cercano a 1, al de *lasso*.

$$\left(\alpha\lambda\|\beta\|_1 + \frac{1}{2}(1-\alpha)\|\beta\|_2^2\right)$$

5.3.1. Modelo de Frecuencia.

El proceso para la elaboración del GLM regularizado de frecuencia utiliza la misma fórmula que en la sección 3, es decir, misma variable dependiente, la función de enlace logarítmica y la distribución binomial negativa. Sin embargo, añade todos los factores de riesgo telemáticos junto a las variables clásicas.

Se comienza con una búsqueda de los hiperparámetros óptimos que seleccionen el mejor modelo entre las regularizaciones l1 y l2. Para ello, se establece un rango de $\alpha = (0, 0.1, 0.5, 0.95 \text{ y } 1)$ y el λ se busca de manera automática por métodos de optimización.

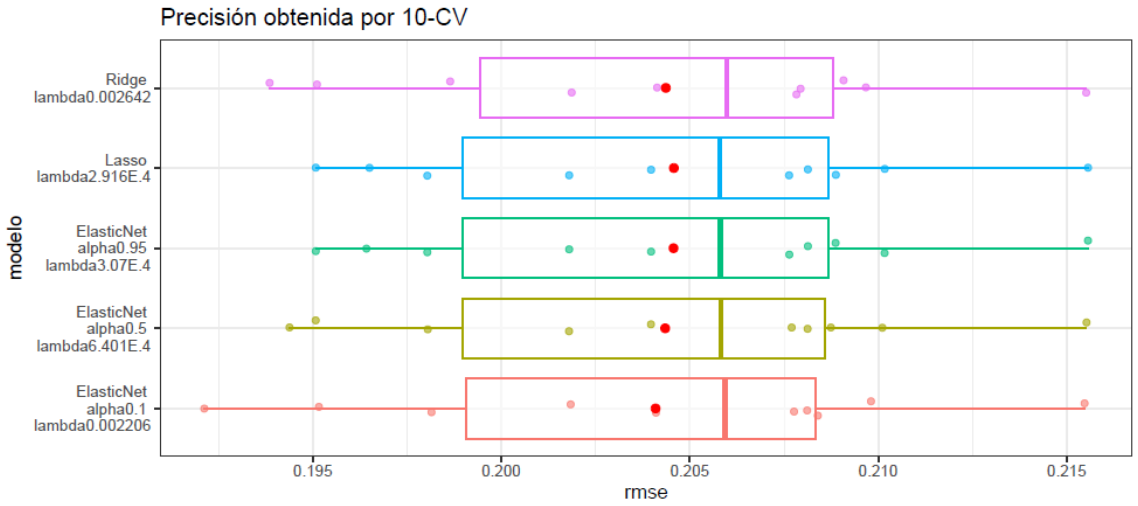
TABLA 11 – MÉTRICAS DE LOS MODELOS DE FRECUENCIA PARA CADA ALPHA

Modelo	alpha	model_ids	rmse	mse	mae	mean residual deviance	r^2
1	0,1	grid_glm_tel_model_5	0,2044	0,0418	0,0742	0,1488	0,0566
2	0,5	grid_glm_tel_model_4	0,2046	0,0419	0,0741	0,1488	0,0548
3	0	grid_glm_tel_model_2	0,2047	0,0419	0,0742	0,1488	0,0543
4	0,95	grid_glm_tel_model_1	0,2048	0,0419	0,0740	0,1488	0,0533
5	1	grid_glm_tel_model_3	0,2048	0,0419	0,0740	0,1489	0,0532

Fuente: Elaboración propia

La tabla anterior puede resultar una aproximación para ver qué valor de Alpha se ajusta mejor a los datos. Pero para estimar la capacidad predictiva de cada modelo se ha empleado un método de validación cruzada con 10 particiones y así obtener más datos de las métricas en diferentes carpetas.

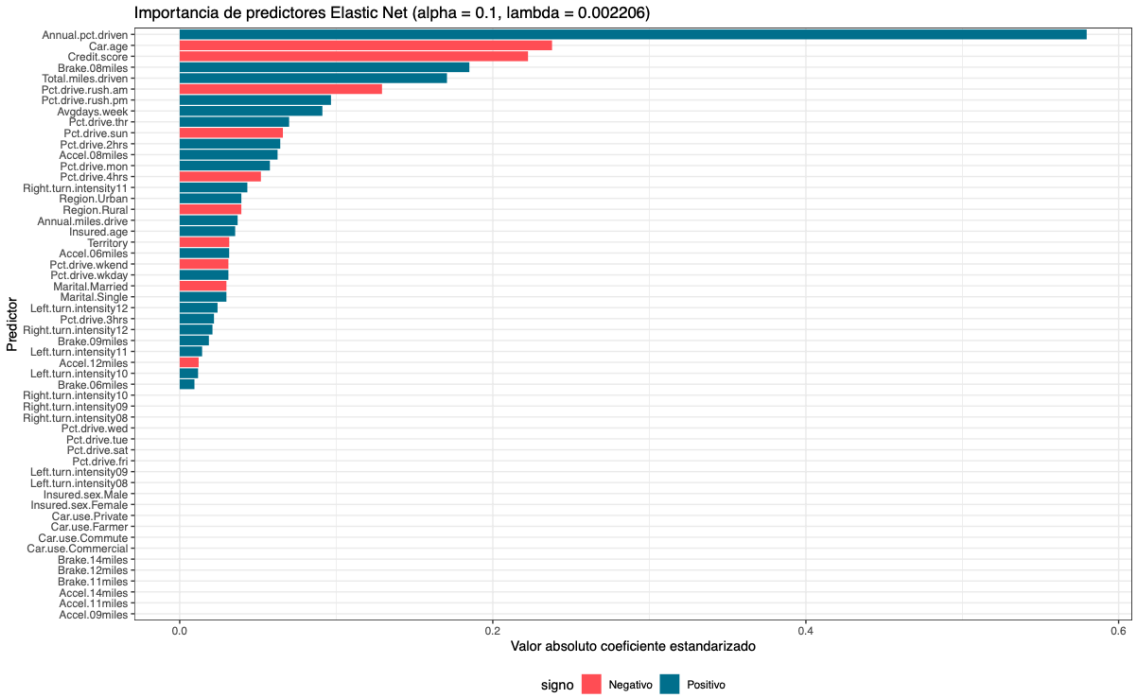
GRÁFICO 24 - COMPARATIVA DE MODELOS GLM REGULARIZADOS DE FRECUENCIA TRAS VALIDACIÓN CRUZADA



Fuente: Elaboración propia

Los resultados de los modelos son muy similares, pero junto a la información de la tabla y la gráfica se puede decir que el mejor modelo regularizado, incluyendo las variables telemáticas, es el modelo 1. Un *Elastic net* con un $\lambda = 0.002206$ y un $\alpha = 0{,}10$ lo que le da mayor peso a la regularización *ridge*.

GRÁFICO 25 – IMPORTANCIA DE PREDICTORES EN FRECUENCIA.



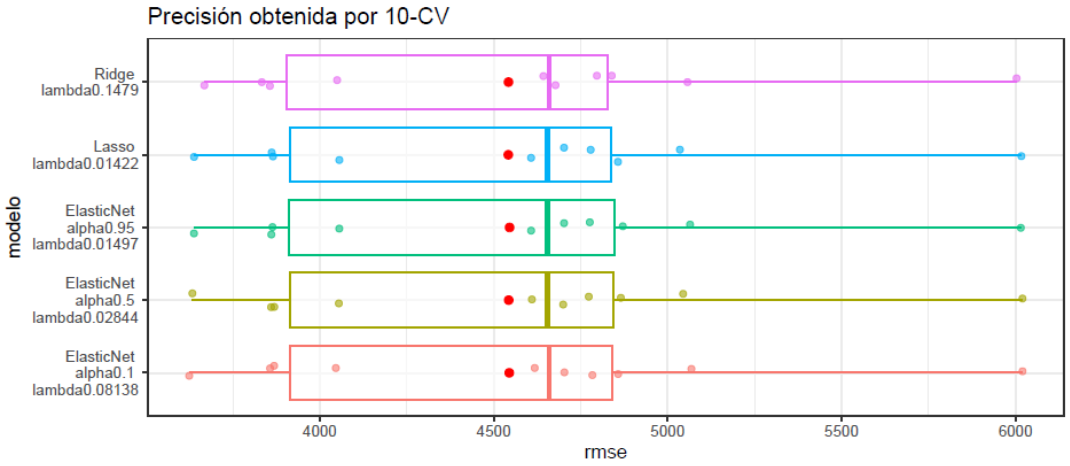
Fuente: Elaboración propia

Este modelo regularizado, se basa principalmente en reducir las correlaciones entre los factores, pero al tener también una parte de regularización l1, varios predictores van a tender a 0 por lo que se excluirán del modelo. Serían la mayoría variables telemáticas las que se quedan fuera por la regularización de *lasso*, manteniéndose las clásicas en el modelo menos las variables categóricas de *Car.use* e *Insured.sex* como ocurría en la modelización GLM/GAM. Se observa que la variable con más importancia es, con diferencia, el porcentaje de tiempo que se pasa en la carretera con un impacto positivo seguido de *Car.age* y *Credit.score*. De las variables telemáticas, también serían importantes en el modelo el número de frenazos de intensidad 08, el total de millas conduciendo y si ha sido por la mañana (am) o por la tarde/noche (pm).

5.3.2. Modelo de Severidad.

Para establecer el GLM regularizado de severidad se realiza el mismo proceso de búsqueda de los parámetros óptimos.

GRÁFICO 26 – COMPARATIVA DE MODELOS GLM REGULARIZADOS DE SEVERIDAD TRAS VALIDACIÓN CRUZADA.

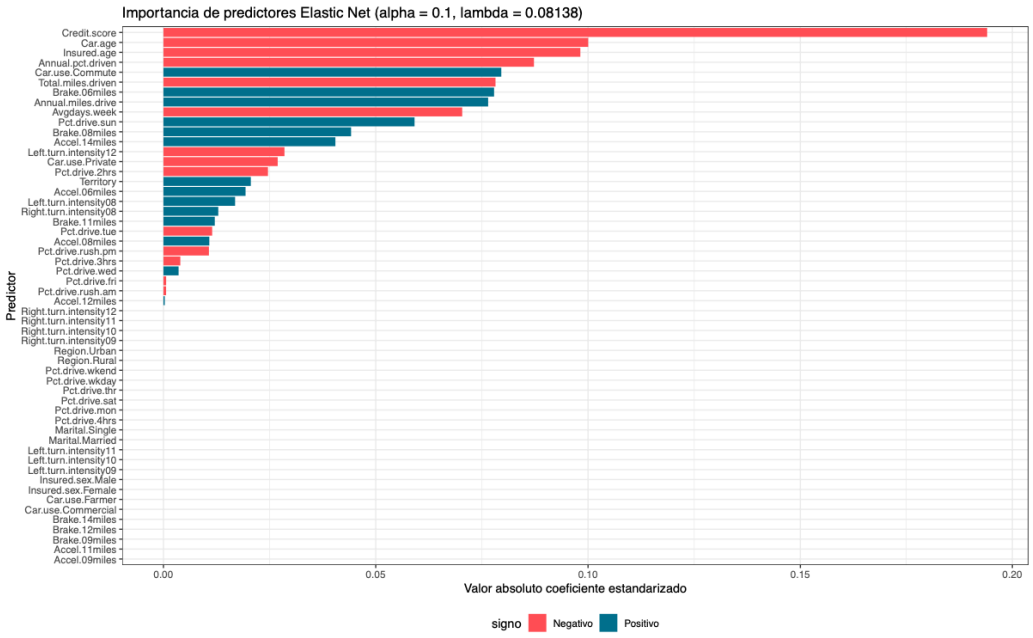


Fuente: Elaboración propia

Surgen 10 modelos para cada uno de los α establecidos y se observa que, de la validación cruzada utilizando 10 carpetas de datos, el que mejor sale, en media, es un *Elastic net* con parámetros λ = 0.08138 y un α = 0,10 lo que le da mayor peso a la regularización *ridge* pero permite la eliminación de los factores que no aportan al modelo.

El GLM de severidad con esos hiperparámetros da lugar a la siguiente clasificación de las variables según su importancia.

GRÁFICO 27 – IMPORTANCIA DE PREDICTORES EN SEVERIDAD.



Fuente: Elaboración propia

Como se podía intuir, la cantidad de factores que se quedan fuera del modelo es superior a lo que ocurre para la frecuencia. Esto es debido a la baja cantidad de datos que hay para estimar el modelo Gamma que hace que sea más complicada la modelización. Los regresores que acumulan mayor importancia en este caso son clásicos, pero también hay telemáticos.

5.4. Modelos Gradient Boosting Machine (GBM).

A partir de los GLM regularizados se han establecido qué variables son las más importantes y cuales no deberían incluirse en los modelos. Con esa información se estiman los GBM tanto para frecuencia como para severidad.

En primer lugar, se realiza un *tunning* de los siguientes hiperparámetros con un máximo de diez modelos, para así encontrar las posibles combinaciones que minimizan las principales métricas.

- Número de árboles (*ntree*): Entre 50 y 300 con una búsqueda de 50 en 50.
- Profundidad de los árboles (*max_depth*): entre 1 y 5 con una búsqueda de 1 en 1.
- Número mínimo de observaciones que debe tener cada nodo. Será fijo y comprende el 5% del total de observaciones.
- Tasa de aprendizaje (*learning_rate*): La búsqueda se hace con tres valores; 0,001, 0,01.
- Comportamiento estocástico (*sample_rate*): Se establecen entre 0,6 y 0,9 con una búsqueda de 0.1 en 0.1.

El resultado tras la búsqueda para el GBM de frecuencia es el siguiente.

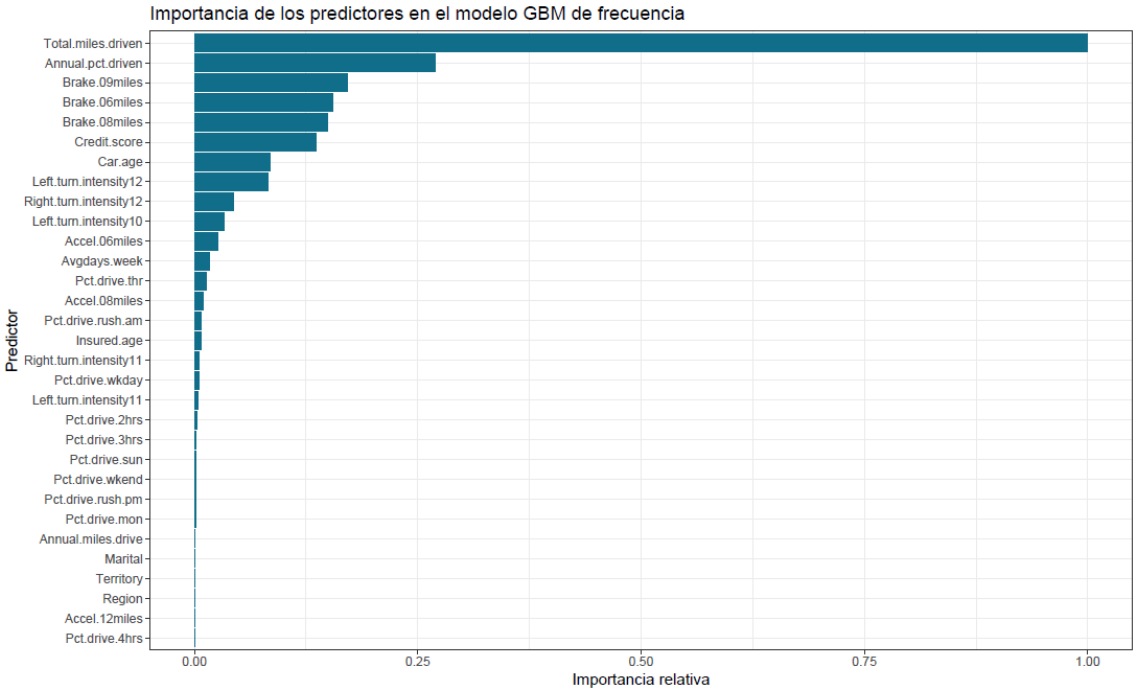
TABLA 12 – MODELOS OBTENIDOS GBM DE FRECUENCIA.

Modelo	learn_rate	max_depth	ntrees	sample_rate	model_ids	rmse	mse	mae	mean residual deviance	r^2
1	0,01	4	300	0,6	grid_gbm_freq_tel_model_1	0,2037	0,0415	0,0734	0,2959	0,0630
2	0,01	4	200	0,6	grid_gbm_freq_tel_model_10	0,2050	0,0420	0,0732	0,3007	0,0513
3	0,01	4	100	0,8	grid_gbm_freq_tel_model_2	0,2064	0,0426	0,0728	0,3080	0,0385
4	0,01	3	100	0,6	grid_gbm_freq_tel_model_5	0,2067	0,0427	0,0733	0,3103	0,0356
5	0,01	2	50	0,9	grid_gbm_freq_tel_model_8	0,2079	0,0432	0,0747	0,3206	0,0246
6	0,001	4	250	0,6	grid_gbm_freq_tel_model_7	0,2085	0,0435	0,0754	0,3255	0,0185
7	0,001	2	300	0,6	grid_gbm_freq_tel_model_4	0,2085	0,0435	0,0756	0,3260	0,0185
8	0,001	3	250	0,6	grid_gbm_freq_tel_model_3	0,2086	0,0435	0,0757	0,3260	0,0181
9	0,001	2	250	0,7	grid_gbm_freq_tel_model_9	0,2087	0,0435	0,0761	0,3274	0,0171
10	0,001	5	100	0,9	grid_gbm_freq_tel_model_6	0,2091	0,0437	0,0770	0,3319	0,0125

Fuente: Elaboración propia

Ese primer modelo obtiene globalmente las mejores métricas respecto a los demás y las variables que tienen más importancia son factores PAYD y PHYD. También, se puede ver que las variables clásicas que eran más significativas durante la modelización GAM/GLM son relevantes.

GRÁFICO 28 - IMPORTANCIA DE LOS PREDICTORES EN EL GBM DE FRECUENCIA.

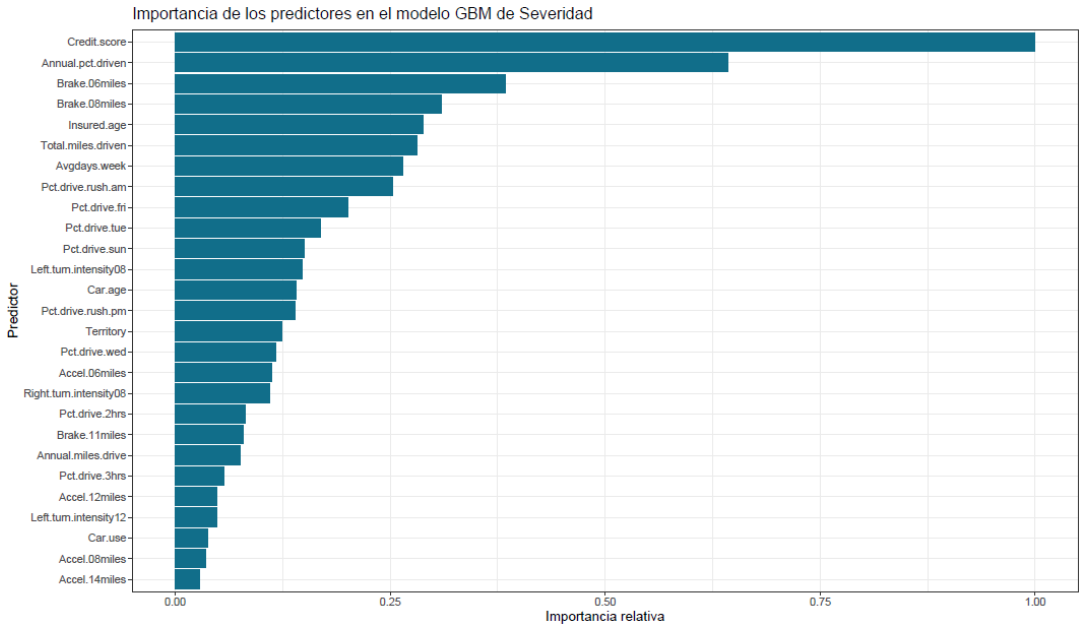


Fuente: Elaboración propia

El resultado final incluye un total de 30 variables, pero la mayor importancia se concentra en los primeros 6 factores.

A lo que el modelo de Severidad se refiere, la búsqueda de las combinaciones de los hiperparámetros es la misma y la combinación óptima coincide con la de frecuencia tal y como se ve en el Anexo P.

GRÁFICO 29 – IMPORTANCIA DE LOS PREDICTORES EN EL GBM DE SEVERIDAD.



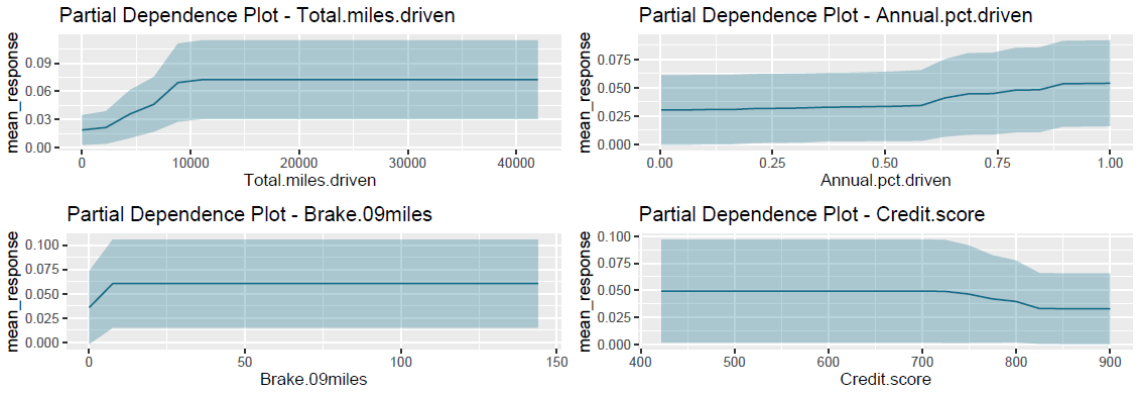
Fuente: Elaboración propia

En este caso, como era de esperar, el número de variables que entran es inferior al de frecuencia con un total de 27. La principal es *Credit Score*, pero el resto son factores telemáticos y estos tienen más impacto, en general, que en el modelo de frecuencia.

5.5. Partial dependence plots (pdp).

A continuación, se muestras los pdp de los factores que más importancia tienen respecto a los modelos de frecuencia y severidad.

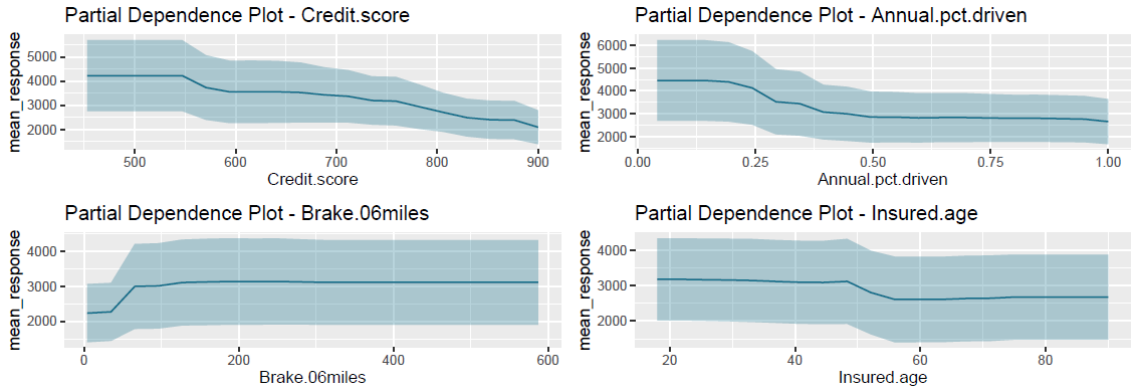
GRÁFICO 30 - PDP DE VARIABLES EN FRECUENCIA.



Fuente: Elaboración propia

Respecto al total de millas conducidas (arriba a la izquierda), la relación con la variable dependiente es creciente durante las primeras 10.000 millas, a partir de ese valor la relación se vuelve plana pudiendo ser por la baja caída en la exposición o porque a partir de un nivel deje de ser relevante para la variable respuesta. Lo mismo ocurre para el tiempo que se pasa en la carretera donde la baja exposición provoca que hasta la mitad de año, la relación sea plana mientras que empieza a incrementar teniendo una relación directa con la frecuencia de siniestros. La variable de numero de frenazos de intensidad 9 afecta en las primeras ocurrencias, después deja de aportar valor y al contrario ocurre con *credit score* donde la interpretación sería igual que en los apartados anteriores.

GRÁFICO 31 – PDP DE VARIABLES EN SEVERIDAD.



Fuente: Elaboración propia

Los dos factores que se encuentran en la parte superior tienen unas tendencias descendentes muy similares, mientras que el número de frenazos de intensidad 6 es parecida al caso de la frecuencia donde aumenta la severidad cuando aumentan los casos de fenazos en los primeros rangos. Sobre la edad del asegurado, se aprecia una caída en la relación con la severidad entre los 45 y 55 años.

5.6. Valor añadido de las variables telemáticas.

El objetivo es ver si añadir variables telemáticas a los modelos de frecuencia y severidad aporta valor a las predicciones. A parte de analizando las métricas que tienen como *output* los modelos, se va a realizar la comparativa desde un punto basado en la prima pura de los modelos suponiendo independencia.

Para poder hacer una comparativa y evaluar el valor añadido de incluir variables telemáticas, se comparan dos modelos GBM, el que ha sido modelizado en el apartado 5.4. y otro incluyendo solo los predictores clásicos.

Al realizar el GBM solo con los factores clásicos, se obtienen los modelos de la Tabla 12 tanto para frecuencia como para severidad. En los anexos se pueden ver ambas salidas de los GBMs y el orden de importancia de las variables que están alineados con los resultados de los GAMs/GLMs clásicos.

TABLA 13 – MODELOS GBM TRAS EL TUNNING DE HIPERPARÁMETROS

Modelo	learn_rate	max_depth	ntrees	sample_rate	model_ids
Frecuencia	0,01	4	300	0,6	grid_gbm_freq_cl_model_1
Severidad	0,01	4	300	0,6	grid_gbm_sev_cl_model_2

Fuente: Elaboración propia

Una vez estimados los modelos, ya se pueden realizar una comparativa entre ambos. Al haberse realizado con el mismo proceso de *tunning* de hiperparámetros y con el mismo proceso de modelización mediante métodos *ensemble* o combinados, las diferencias que se obtenga serán directamente por la inclusión de las variables telemáticas.

TABLA 14 – COMPARATIVA DE LAS MÉTRICAS EN LOS DATOS DE ENTRENAMIENTO.

	Frecuencia					Severidad				
Modelo	RMSE	MSE	MAE	RMSLE	MRD	RMSE	MSE	MAE	RMSLE	MRD
GBM clasicas	0,2084	0,0414	0,0775	0,1391	0,3263	4451,2320	19813465	2416,6800	1,2921	17,8952
GBM Telemáticas	0,2028	0,0411	0,0730	0,1348	0,2918	4170,4870	17392964	2203,8870	1,2113	17,7565

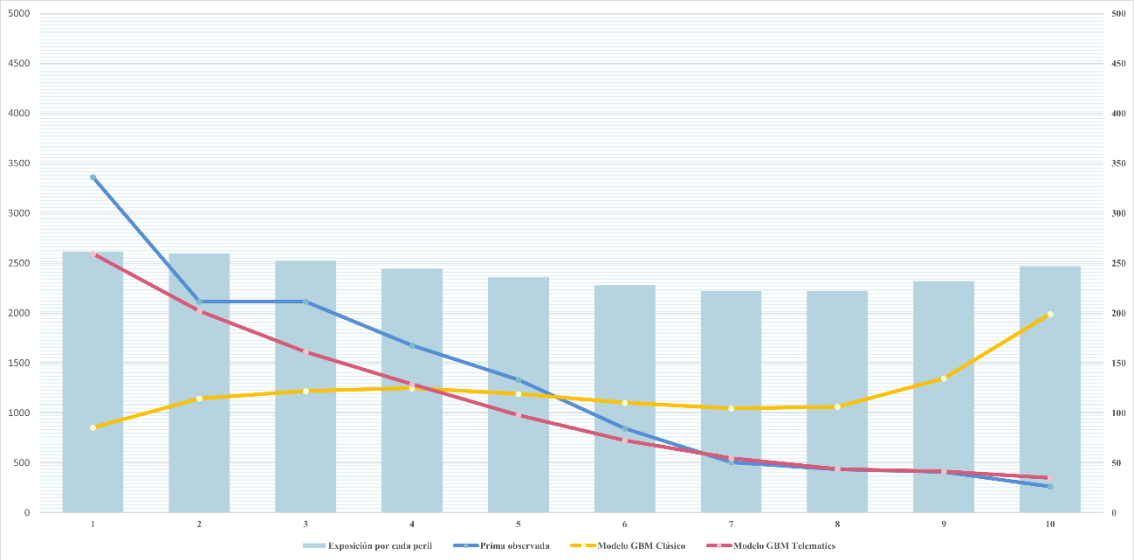
Fuente: Elaboración propia

Si se comparan las métricas en ambos modelos, añadir las variables telemáticas tanto a severidad como a frecuencia minimiza todas las métricas en los datos de entrenamiento.

5.6.1. Double lift charts.

Son gráficos que tratan de comparar modelos de predicción observando el ajuste que tienen y comparándolo con las medidas observadas. Va a servir para ver si, incluyendo los factores telemáticos, se modeliza con mayor precisión respecto a la prima observada, y cuáles son las partes en las que ambos modelos difieren más.

GRÁFICO 32 – DOUBLE LIFT CHART ENTRE LOS MODELOS GBM.



Fuente: Elaboración propia

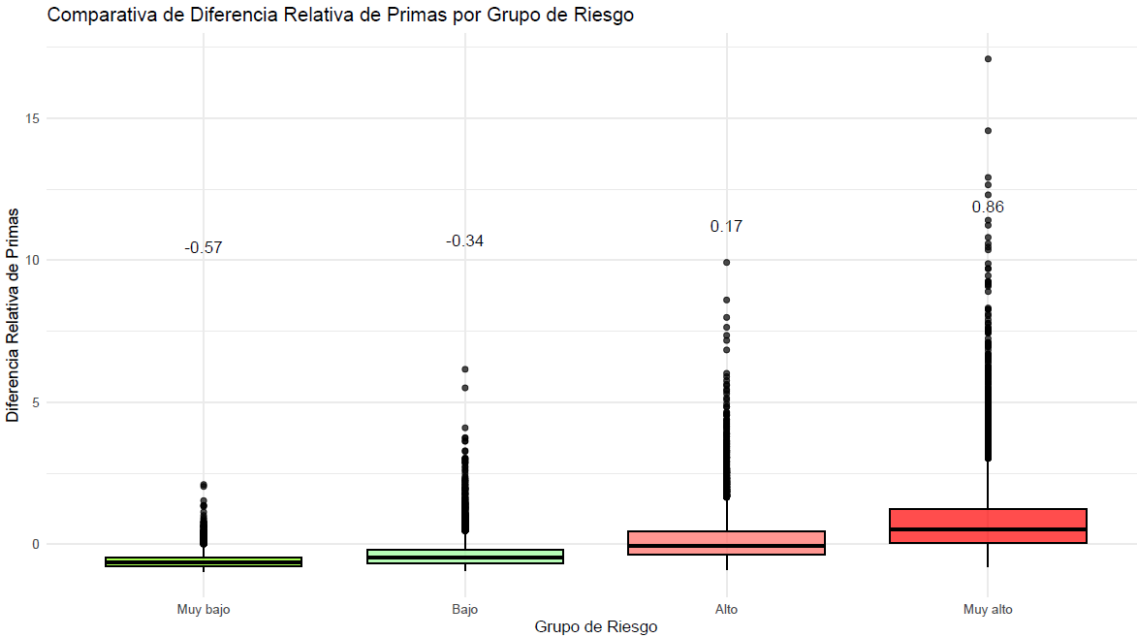
El gráfico se basa en la *performance* de los modelos, obtenidas a partir de los datos de *test*. En la comparativa se observa que el modelo con las variables telemáticas, la prima calculada a partir de las predicciones se ajusta mejor a los datos observados sobre todo en los grupos donde son más bajas. Las mayores diferencias con las estimadas mediante el GBM clásico se encuentran en los primeros y últimos grupos siendo superior la prima telemática en los primeros e inferior en los últimos.

5.6.2. Diferencias relativas entre las primas según los grupos de riesgo.

Por último, se realiza una asignación de riesgo para cada asegurado en función de las predicciones obtenidas tras la modelización. Esto consiste en calcular la función de distribución acumulada de las predicciones de frecuencia en los modelos para cada asegurado i de tal forma que $r_i = F_n\{f(x_i)\}$ siendo $f(x_i)$ las predicciones obtenidas en el modelo de frecuencia con los regresores x_i incluidos. Así, los valores de riesgo de cada asegurado están comprendidos entre [0,1] donde a menor (mayor) valor en la predicción de severidad, menor (mayor) riesgo.

Con la puntuación establecida, se agrupan en cuatro grupos de riesgo: Muy bajo [0, 0.25], Bajo (0.25, 0.5], Alto (0.5, 0.75], Muy alto (0.75, 1].

GRÁFICO 33 – COMPARATIVA DE LA DIFERENCIAS EN LA PRIMA POR GRUPOS DE RIESGO.



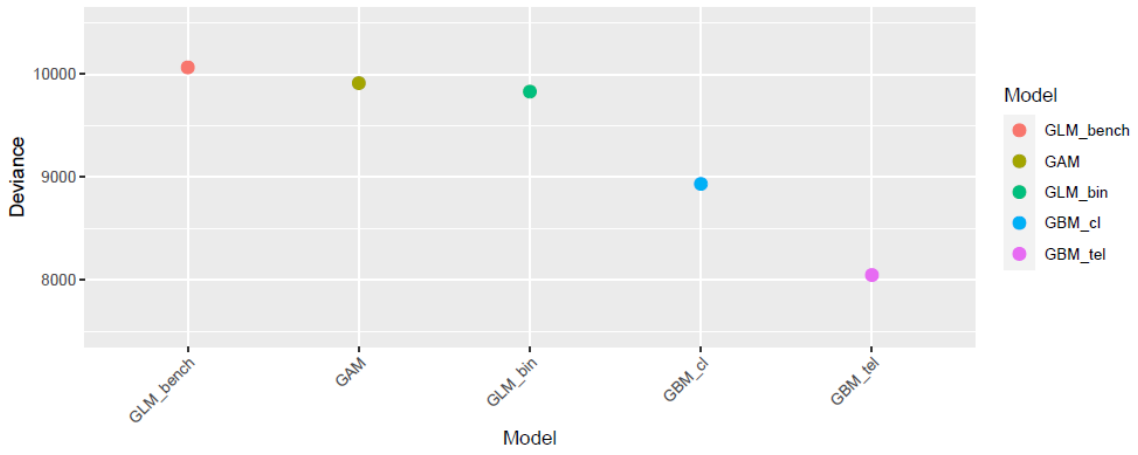
Fuente: Elaboración propia.

Tras la agrupación se han calculado las diferencias relativas entre las primas del modelo GBM clásico y GBM con las variables telemáticas. El resultado da lugar a que, en los grupos de riesgo bajo (alto) y muy bajo (muy alto), los modelos estimados con las variables telemáticas dan lugar a menores (mayores) primas encontrando una mayor diferencia cuanto menor (mayor) es el riesgo.

5.7. Comparativa de la deviance entre los modelos de frecuencia.

Si se comparan todos los modelos de frecuencia que se han estimado a través de la *deviance* en la muestra separada para realizar el *testing*, se obtiene el resultado visto en la gráfica 35.

GRÁFICO 34 – COMPARATIVA DE LA DEVIANCE EN LOS MODELOS DE FRECUENCIA



Fuente: Elaboración propia.

El modelo *GLM_bench* es un modelo base sin contar con los predictores, únicamente se añade el intercepto. Al comparar solo los modelos con los factores clásicos, se puede ver que el GLM tras las agrupaciones mejora levemente al GAM, pero es el GBM el que minimiza la *deviance* en mayor medida resultando así, el modelo que mejor predice.

Si después se añaden las variables telemáticas, la *deviance* vuelve a bajar por lo que se muestra una mejora en cuanto a la predicción y el ajuste respecto al GBM clásico.

6. CONCLUSIONES.

En la actualidad, el mundo de los seguros está experimentando una transformación significativa debido al crecimiento exponencial de los datos. Este cambio ha llevado a que los actuarios, se vean obligados a adaptarse y a adquirir competencias en el análisis de datos y el uso de técnicas mediante herramientas estadísticas capaces de soportar la cada vez la mayor cantidad de información que se maneja.

Esta inclusión del *big data*, también afecta en los procesos de tarificación de seguros de autos. No solo en la inclusión de avanzadas metodologías de modelización estadística mediante técnicas de *machine learning*, si no en la aparición de nuevos tipos de seguros como el caso de los seguros basados en el uso (UBI, por sus siglas en inglés). Estos aparecen por la incorporación de nuevos factores de riesgo gracias al acceso a nueva información a través de sistema incorporados en los coches que pueden cambiar la forma en la que se viene trabajando en los seguros de autos.

Durante la primera fase del proyecto, tras un estudio del ajuste de las variables respuesta tanto de frecuencia como de severidad, se ha concluido que las distribuciones que siguen los datos son la Binomial Negativa y la Gamma respectivamente. Tras el empleo de los Modelos Aditivos Generalizados (GAM) para capturar relaciones no lineales de los predictores continuos se obtuvieron los mejores modelos a partir de los criterios de información BIC y AIC de frecuencia y severidad. En ambos modelos, todas las variables continuas fueron significativas y de las categóricas, para frecuencia se mantuvo en el modelo la región y para la severidad ninguna fue significativa. Posteriormente, se agruparon las variables que entraron en los modelos mediante técnicas con el uso árboles de regresión donde sacaron los grupos óptimos. Este método combinaba la estimación del modelo GAM como variable dependiente y como regresor, los valores observados en los datos. Las agrupaciones que se obtienen se muestras consistentes con la distribución de la exposición en las variables y con las tendencias de las curvas.

Para finalizar con esta primera parte, se realizó un traspaso a los tradicionales Modelos Lineales Generalizados (GLM) donde en la Tabla 10 se muestran las métricas obtenidas. Se concluye de tales resultados que a pesar de aumentar las variables con las agrupaciones en los factores continuos, las métricas que penalizan la complejidad con el aumento en los regresores para modelar, en general han ido mejorando. Esto indica que el modelo

simplificado puede ser utilizado como un sustituto cercano en la práctica para el modelo más complejo y flexible. Además, se encontró una estrecha aproximación entre las primas calculadas a partir de los GLM resultantes y los GAM originales por lo que, la interpretabilidad y simpleza de los GLM prevalecen ante la flexibilidad de los GAM.

En la segunda parte del estudio, se exploró el impacto de la inclusión de variables telemáticas en la tarificación de seguros de autos frente a los factores clásicos. Se emplearon técnicas de *Gradient Boosting Machines* (GBM) para desarrollar modelos más complejos. Para ello, primero se ajustaron GLMs regularizados utilizando la búsqueda del parámetro Alpha con el uso de la técnica *cross-validation* que indicase el tipo de regularización óptima resultando ser en ambos modelos la combinación $l1$ y $l2$ con un Elastic Net de $\alpha = 0{,}10$ pero con diferentes valores de λ. Del total de 39 variables telemáticas junto a las 9 clásicas, para el modelo de frecuencia solo resultaron tener una importancia relativa 31 predictores mientras que en severidad, como era de esperar, fueron menos con un total de 27. Salvo algunas variables clásicas que quedaron fuera y que estaban alineadas con los resultados de la sección anterior como el caso del estado civil, el sexo o el tipo de uso del auto, la mayoría que se quedaron fuera fueron factores de datos telemáticos.

El resultado del modelo de frecuencia GBM dio como variable más relevante el total de millas conducidas en un año junto a los días que pasaba al año en carretera y el total de número de frenazos. En cuanto a severidad, la principal variable fue el *credit score* seguido del total de tiempo que pasa durante la carretera y también el número de frenazos.

Estas son variables dentro del grupo PAYD y PHYD las cuales en seguros tipo UBI se podrían utilizar para reducir las primas de los asegurados. Si se trata de un seguro de pago por uso, reducir el tiempo en carretera o el total de millas conducidas para este caso, reduciría la frecuencia y severidad como se ve en los *partial dependence plots* y por tanto las primas. Además, mejorar algunos hábitos como evitar los frenazos dejando mayor distancia de seguridad, o manteniendo las velocidades correctas parece que también tiene un impacto directo en el valor de la prima.

Los resultados mostraron que la inclusión de variables telemáticas mejoró tanto el ajuste del modelo como las diferencias entre los grupos de riesgo de los asegurados. Esto permitió calcular primas más precisas y ajustadas a los perfiles de riesgo individuales. Además, se utilizó el gráfico *double lift chart* para evaluar el impacto de las variables

telemáticas en la prima pura y se encontró una mejora en la aproximación a los valores observados de la cartera.

Respecto a los grupos de riesgo, cuando se trata del grupo con menor riesgo la diferencia relativa entre el valor de las primas respecto al modelo con solo predictores clásicos en media el un 57% inferior dando lugar a primas más bajas mientras en el grupo de mayor riesgo las primas con factores telemáticos daban un resultado superior en media de un 86%.

Para acabar, en cuento a todos los modelos ajustados de frecuencia, si se comparan los valores de la *deviance* como muestra el gráfico 34, el modelo de *machine learning GBM* con variables clásicas mejora a los estimados en la primera sección, pero se encuentra por debajo del que incluye los regresores telemáticos.

La inclusión a las ya extensas bases de datos de las compañías de seguros de los datos telemáticos puede mejorar las predicciones y establecer incentivos tanto económicos para sus clientes, como beneficios entorno a la seguridad vial para la sociedad. Sin embargo, en futuras líneas de investigación, seria interesante analizar si las mejoras en la predicción y la posibilidad de ofrecer descuentos o primas bajas a los asegurados cubren los gastos de adoptar esta tecnología. También se debe tener en cuenta la situación del mercado para establecer primas mínimas que permitan ser competitivos y analizar los recargos que se le aplican a los conductores que tengan más riesgo para competir con las aseguradoras que tengan sus carteras mutualizadas.

BIBLIOGRAFÍA

Actuarios Españoles. (24, pp. 123-147). https://www.actuarios.org/wp-content/uploads/2018/11/123_147_A06.pdf.

Antonio, K., & Beirlant, J. (2007). *Actuarial statistics with generalized linear mixed models. Insurance: Mathematics and Economics, 40(1), 58-76.*

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Cart. Classification and regression trees.*

Cómo la telemática podría afectar tu seguro de auto. (Julio, 2019). *Allstate*. La telemática y el seguro de auto | Allstate.

De la Vega, M. (2 de enero, 2023). Los retos del Sector Asegurador en un contexto de incertidumbre. *EY*. Los retos del Sector Asegurador en un contexto de incertidumbre (ey.com).

El precio del seguro del coche se dispara: los españoles pagan 680 millones más y seguirá subiendo. (23 de enero, 2023). *Cinco días*. https://cincodias.elpais.com/cincodias/2023/01/20/opinion/1674221870_664960.html

Eriksson, A. (2021, Junio). *A Comparison of Gradient Boosting Machines and Generalized Linear Models for Non-Life Insurance Pricing.*

Europe Insurance Telematics Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026). (2021). (s.f). *Mordor Intelligence.* https://www.mordorintelligence.com/industry-reports/europe-insurance-telematics-market#:~:text=The%20countries%20with%20the%20most,were%20interested%20in%20telematics%20insurance.

Goldburd, M., Khare, A., Tevet, D., & Guller, D. (2016). *Generalized linear models for insurance rating. Casualty Actuarial Society, CAS Monographs Series, 5.*

Guillen, M., & Pesantez-Narvaez, J. (2018, December). Machine learning y modelización predictiva para la tarificación en el seguro de automóviles. *Anales del Instituto de Actuarios Españoles* (No. 24, pp. 123-147).

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Overview of supervised learning. The elements of statistical learning: Data mining, inference, and prediction, 9-41*.

Henckaerts, R., Antonio, K., Clijsters, M., & Verbelen, R. (2018). *A data driven binning strategy for the construction of insurance tariff classes*. Scandinavian Actuarial Journal, 2018(8), 681-705.

Henckaerts, R. (2021). *Insurance pricing in the era of machine learning and telematics technology.*

Henckaerts, R., Côté, M. P., Antonio, K., & Verbelen, R. (2021). *Boosting insights in insurance tariff plans with tree-based machine learning methods.* North American Actuarial Journal, 25(2), 255-285.

INESE. (2018, noviembre 27). La telemática y el seguro de autos en España: un futuro por descubrir. *Future.* *https://future.inese.es/la-telematica-y-el-seguro-de-autos-en-espana-un-futuro-por-descubrir/*.

Jansson, Caspar. (Abril, 2022). Insurance Telematics in Europe and North America 6th Edition. *berginsight*. https://media.berginsight.com/2022/04/04173625/bi-insurancetelematics6-ps.pdf.

Jeong, H., Valdez, E. A., Ahn, J. Y., & Park, S. (2017). *Generalized linear mixed models for dependent compound risk models*. Available at SSRN 3045360.

Katrien Antonio. (9 de abril, 2020). *Data Science for Non Life Insurance: Telematics.* Youtube. https://www.youtube.com/watch?v=Ij8aAopugD.

King, R. (2022) Canada credit score: What that 3-digit number means. *Zolo*. https://www.zolo.ca/blog/canada-credit-score.

Mazorco, I. (18 de agosot, 2022). Seguro de autos: cómo funciona el sistema donde pagás según cuánto lo usás. *La nación*. Cómo funciona el sistema donde pagás según cuánto usás el auto - LA NACION.

Montero, H. (31 de enero, 2023). El precio del seguro del coche se dispara: los españoles pagan 680 millones más y seguirá subiendo. *La razón*. El precio del seguro del coche se dispara: los españoles pagan 680 millones más y seguirá subiendo (larazon.es).

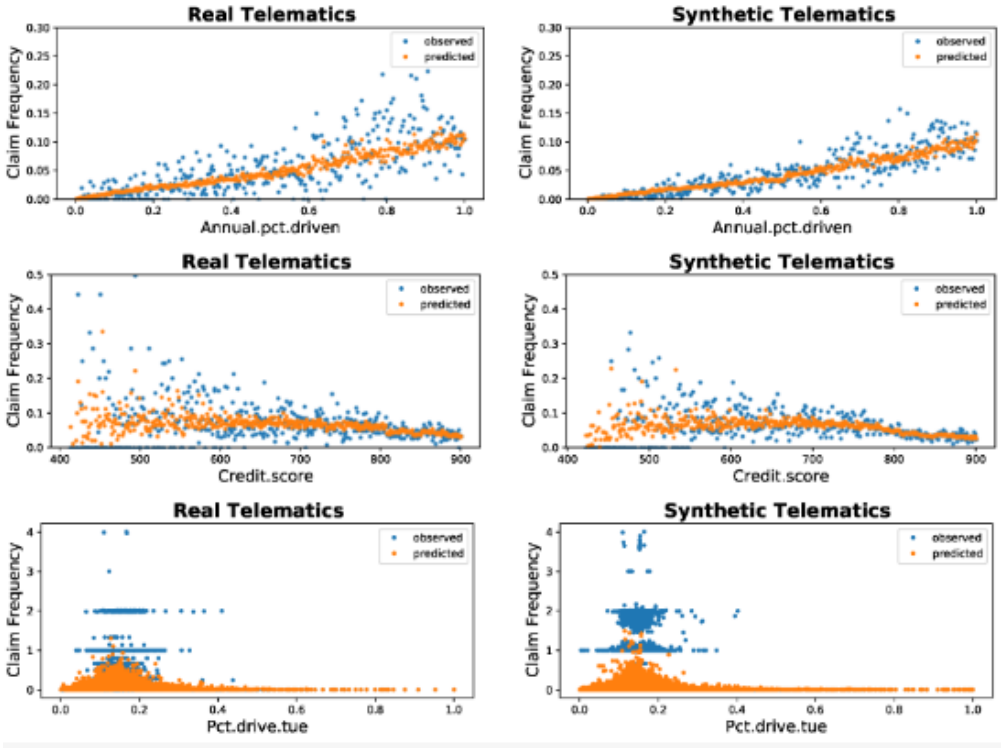
So, B., Boucher, J.-P., & Valdez, E. A. (2021). Synthetic Dataset Generation of Driver Telematics. Risks, 9(4), 58. MDPI AG. http://dx.doi.org/10.3390/risks9040058.

Tiwari, A. (13 de marzo, 2020). Modeling Insurance Claim Frequency: An illustrative guide to model insurance claim frequencies using generalized linear models in R. *Medium*. https://medium.com/swlh/modeling-insurance-claim-frequency-a776f3bf41dc

Tiwari, A. (30 de marzo, 2020). Modeling Insurance Claim Severity: An illustrative guide to model insurance claim severity using generalized linear models in Python & R. Medium. *Medium*. https://medium.com/swlh/modeling-insurance-claim-severity-b449ac426c23.

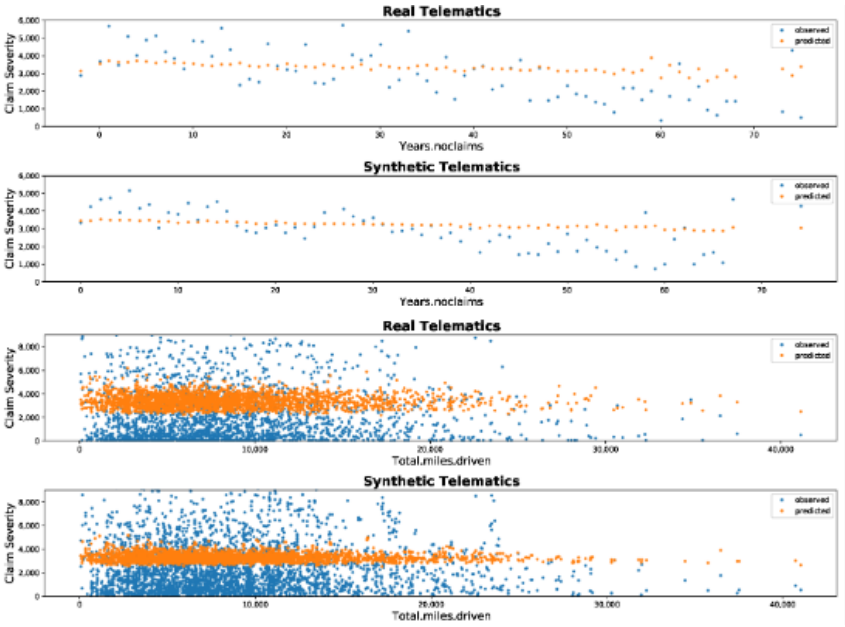
¿Qué es la telemática? (s.f). *Movildata*. https://movildata.com/recursos/que-es-la-telematica/

ANEXO A - FRECUENCIA MEDIA DE RECLAMACIONES UTILIZANDO CONJUNTOS DE DATOS REALES (IZQUIERDA) Y SINTÉTICOS (DERECHA)



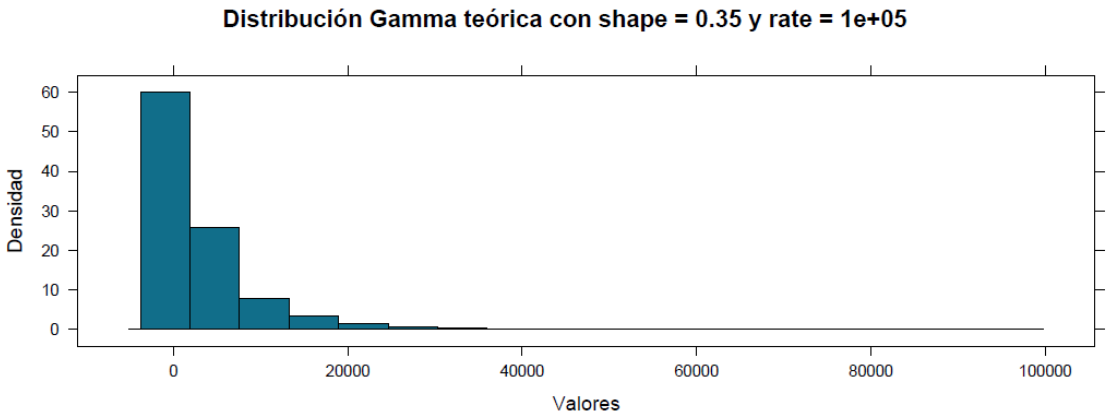
Fuente: So, B., Boucher, J. P., & Valdez, E. A. (2021).

ANEXO B - SEVERIDAD MEDIA UTILIZANDO CONJUNTOS DE DATOS REALES (1º Y 3º) Y SNTÉTICOS (2º Y 4º)

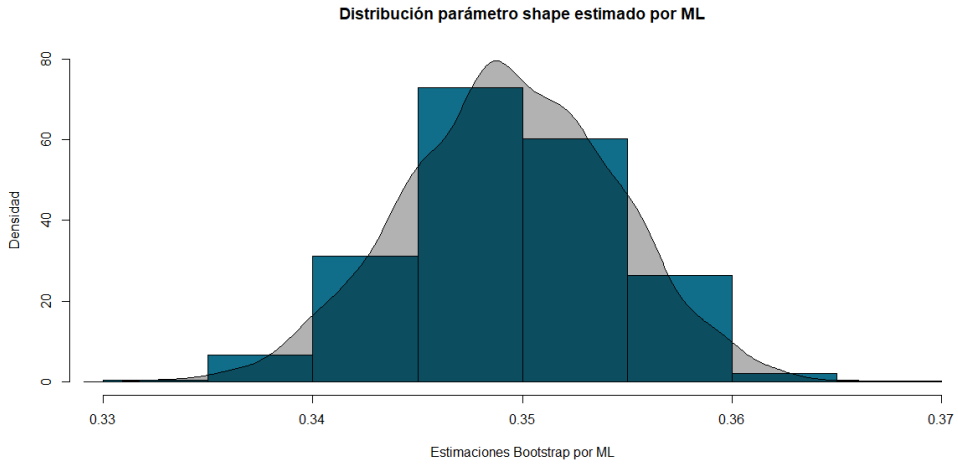


Fuente: So, B., Boucher, J. P., & Valdez, E. A. (2021).

ANEXO C - GAMMA MEDIANTE SIMULACIÓN DE MONTE-CARLO



ANEXO D – DISTRIBUCIÓN DEL PARÁMETRO AJUSTADO POR ML



Fuente: Elaboración propia

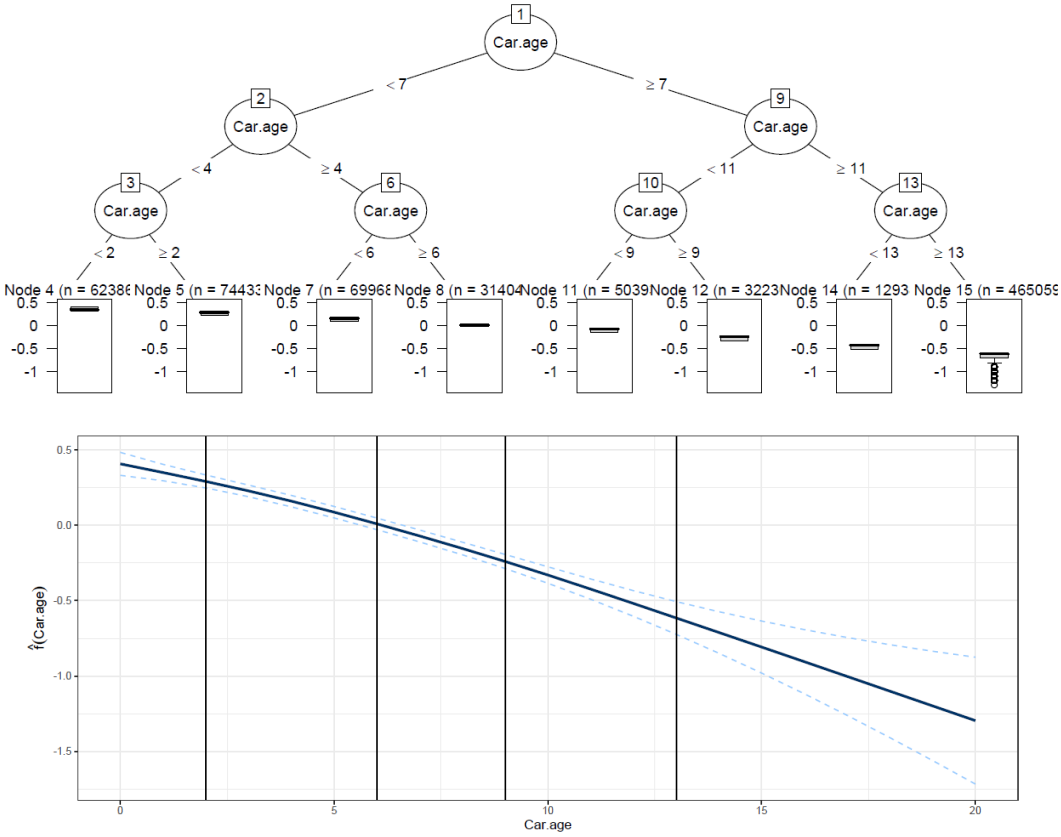
ANEXO E - TEST DE AJUSTE A UNA GAMMA

```
        Anderson-Darling test of goodness-of-fit
        Braun's adjustment using 316 groups
        Null hypothesis: Gamma distribution
        Parameters assumed to have been estimated from data

data:  gamma_ajustada
Anmax = 5.6, p-value = 0.376
```

Fuente: Elaboración propia

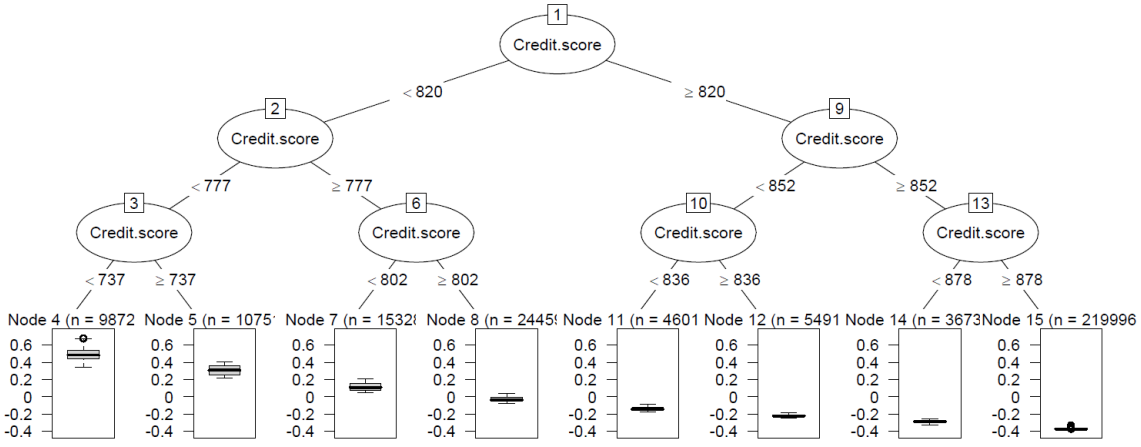
Fuente: Elaboración propia

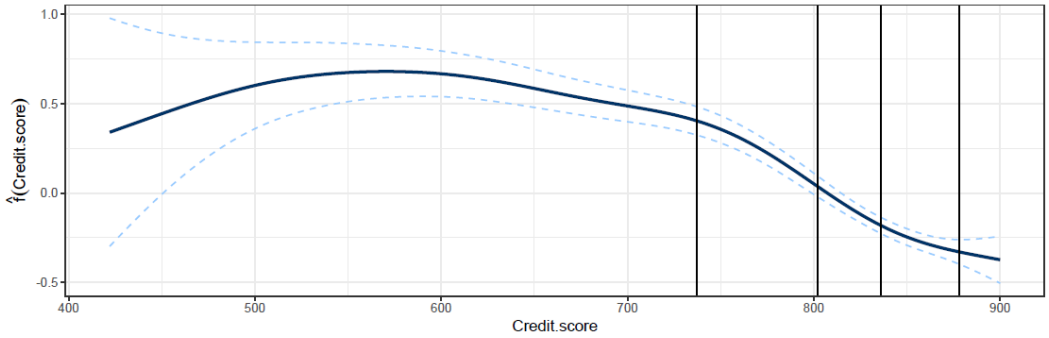
ANEXO F - AGRUPACIONES DE CAR.AGE EN FRECUENCIA



Fuente: Elaboración propia.

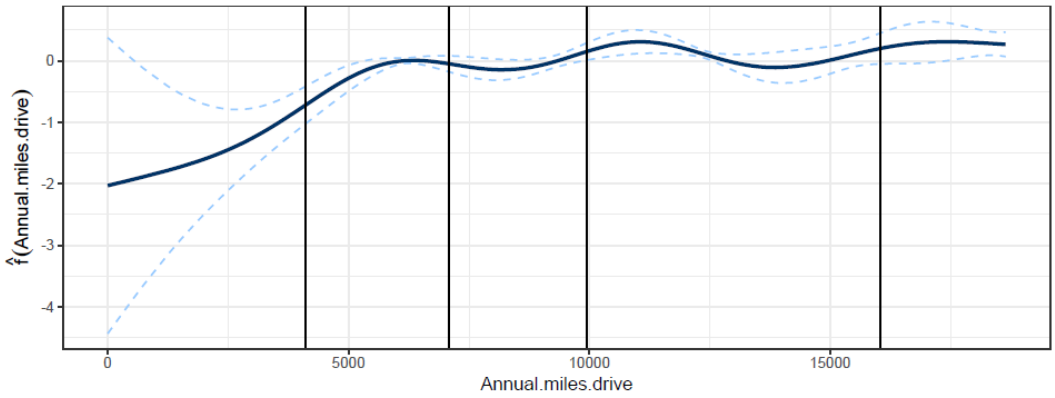
ANEXO G - AGRUPACIONES DE CREDIT.SCORE EN FRECUENCIA





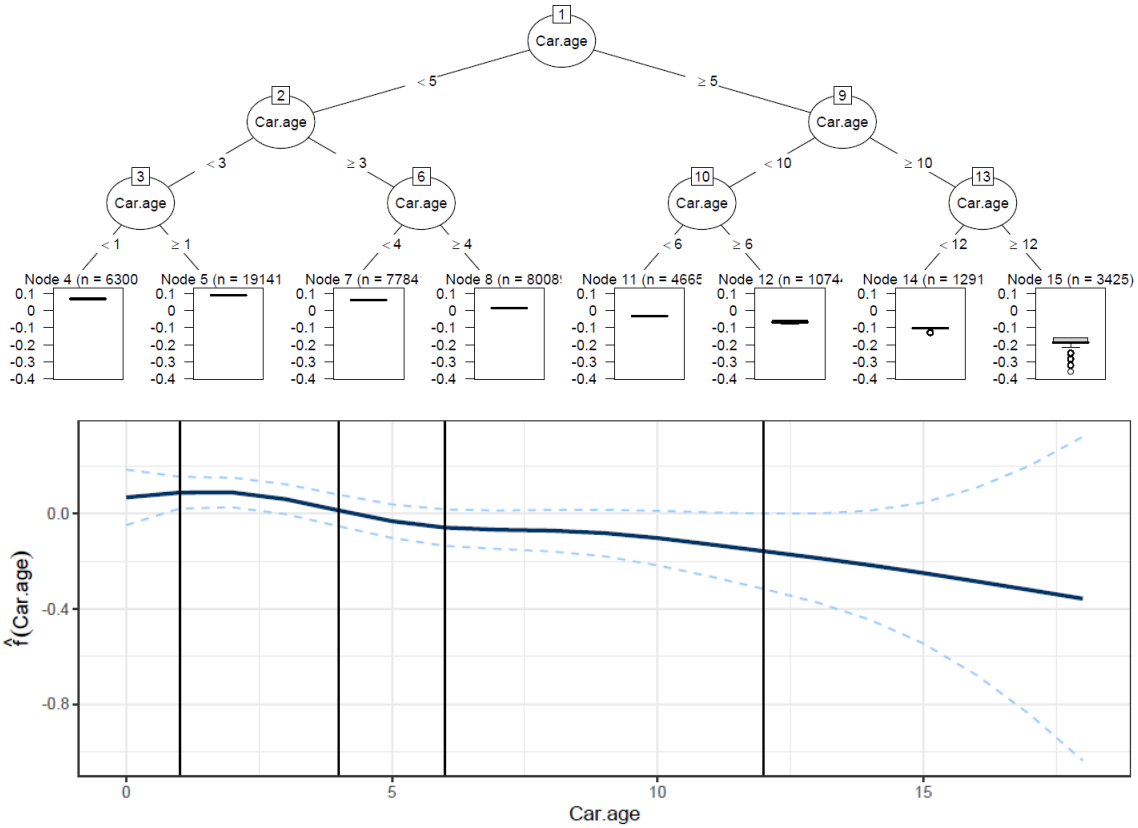
Fuente: Elaboración propia

ANEXO H - AGRUPACIONES DE ANNUAL.MILES.DRIVE EN FRECUENCIA



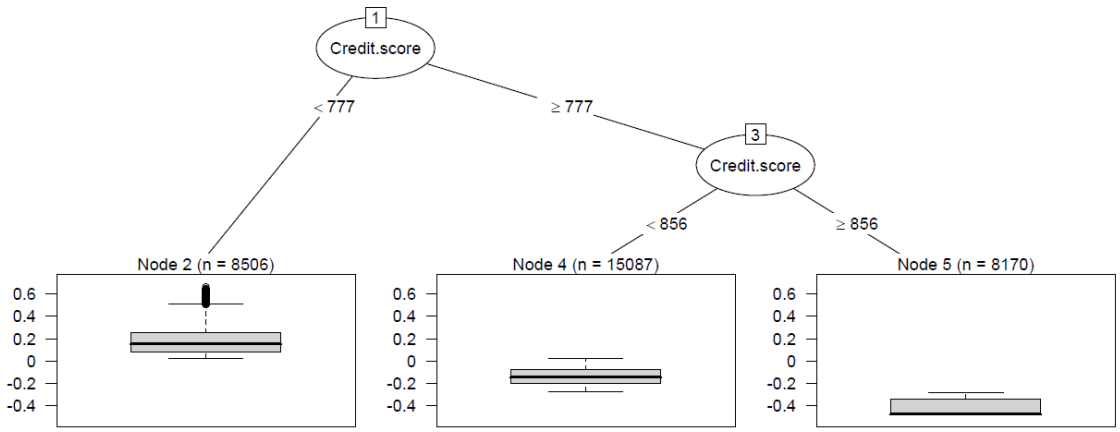
Fuente: Elaboración propia

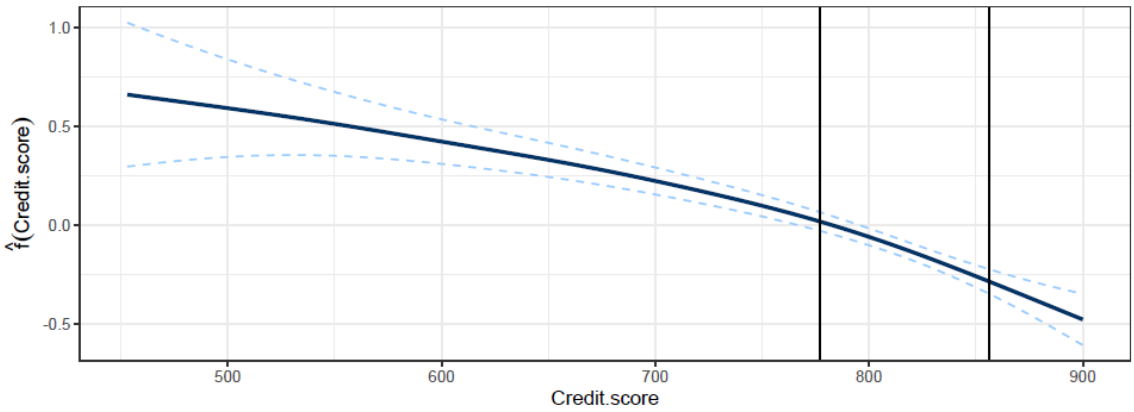
ANEXO I – AGRUPACIONES DE CAR.AGE EN SEVERIDAD



Fuente: Elaboración propia

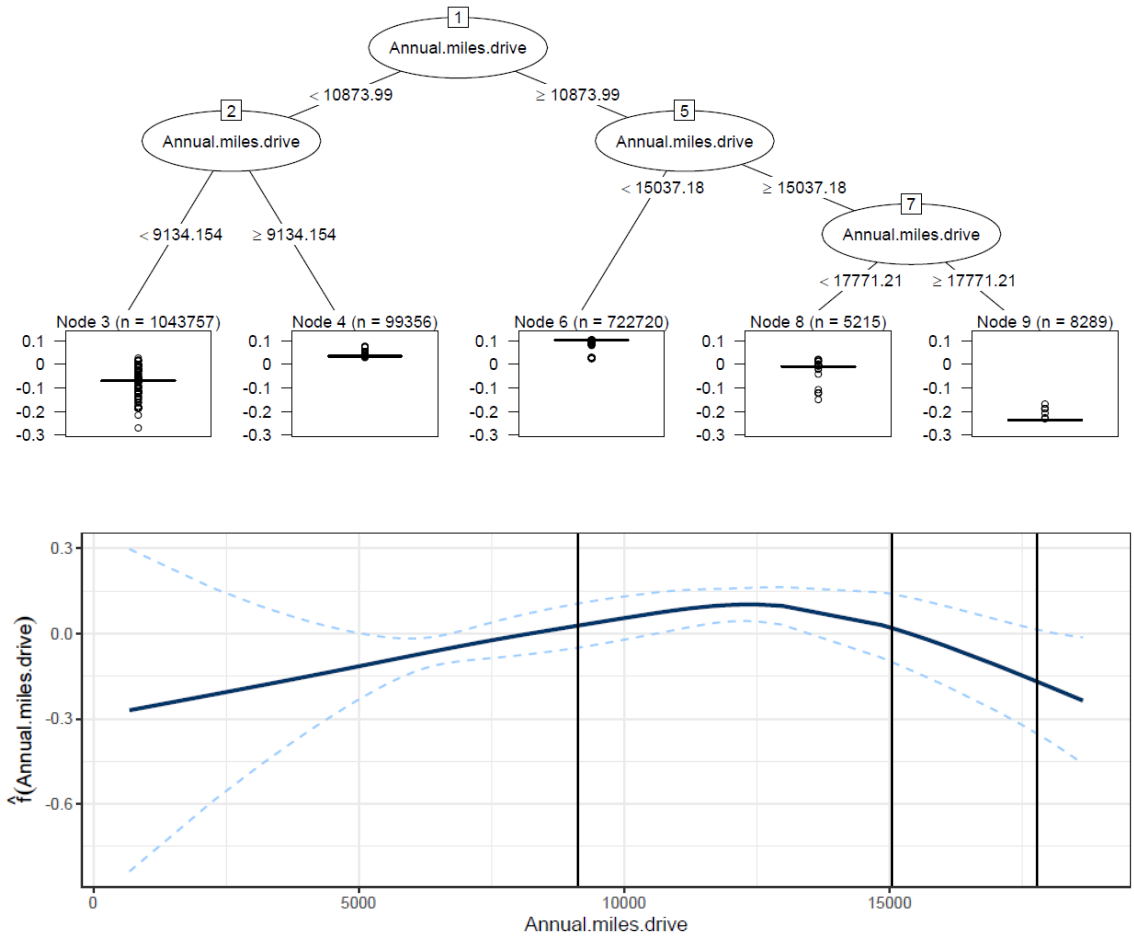
ANEXO J – AGRUPACIONES DE CREDIT.SCORE EN SEVERIDAD





Fuente: Elaboración propia

ANEXO K – AGRUPACIONES DE ANNUAL.MILES.DRIVE EN SEVERIDAD



Fuente: Elaboración propia

ANEXO L - GLM FRECUENCIA CON AGRUPACIONES

```
Call:
glm.nb(formula = as.formula("NB_Claim ~ Insured.age_bin + car.age_bin + Car.use + Credit.score_bin + Annual.miles.drive_bin +\n
                        Region + Territory"),
    data = train_classic_bin, control = glm.control(maxit = 1e+06),
    offset = log(Exposure), link = "log", init.theta = 1.208554047)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6160  -0.3168  -0.2526  -0.1894   3.8733

Coefficients:
                                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                             -3.2557980  0.0816506 -39.875  < 2e-16 ***
Insured.age_bin[16,26)                   0.4151806  0.0833274   4.983 6.28e-07 ***
Insured.age_bin[26,32)                   0.0437808  0.0656023   0.667 0.504538
Insured.age_bin[32,45)                  -0.1507393  0.0497988  -3.027 0.002470 **
Insured.age_bin[67,103]                 -0.2786860  0.0734774  -3.793 0.000149 ***
car.age_bin[0,2)                         0.1681444  0.0516274   3.257 0.001126 **
car.age_bin[6,9)                        -0.2633333  0.0532716  -4.943 7.68e-07 ***
car.age_bin[9,13)                       -0.5551679  0.0629847  -8.814  < 2e-16 ***
car.age_bin[13,20]                      -0.8747427  0.1163412  -7.519 5.53e-14 ***
Car.useCommercial                        0.1490559  0.1109738   1.343 0.179220
Car.useFarmer                           -0.1882792  0.2492202  -0.755 0.449965
Car.usePrivate                           0.0144691  0.0481922   0.300 0.763996
Credit.score_bin[422,737)                0.8100858  0.0600368  13.493  < 2e-16 ***
Credit.score_bin[737,802)                0.5762886  0.0621457   9.273  < 2e-16 ***
Credit.score_bin[802,836)                0.1616609  0.0651761   2.480 0.013125 *
Credit.score_bin[878,900]               -0.0567985  0.0870324  -0.653 0.514006
Annual.miles.drive_bin[0,4.1e+03)       -1.3621976  0.2804524  -4.857 1.19e-06 ***
Annual.miles.drive_bin[7.08e+03,9.94e+03) -0.0633448  0.0605588  -1.046 0.295559
Annual.miles.drive_bin[9.94e+03,1.6e+04)  0.0966602  0.0473535   2.041 0.041226 *
Annual.miles.drive_bin[1.6e+04,1.86e+04]  0.2605483  0.1078618   2.416 0.015710 *
RegionRural                             -0.2404418  0.0529554  -4.540 5.61e-06 ***
Territory                                0.0007365  0.0008325   0.885 0.376324
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(1.2086) family taken to be 1)

    Null deviance: 15526  on 67745  degrees of freedom
Residual deviance: 14823  on 67724  degrees of freedom
AIC: 21851

Number of Fisher Scoring iterations: 1


              Theta:  1.209
          Std. Err.:  0.233

 2 x log-likelihood:  -21804.641
```

Fuente: Elaboración propia

ANEXO M - GLM SEVERIDAD CON AGRUPACIONES

```
Call:
glm(formula = as.formula("AMT_avg  ~  Insured.age_bin + car.age_bin + Credit.score_bin + Annual.miles.drive_bin + Territory"),
    family = Gamma(link = "log"), data = AMT_claims_train_2_bin)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6497  -1.1245  -0.4606   0.2275   4.4730

Coefficients:
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                             8.381373   0.089697  93.441  < 2e-16 ***
Insured.age_bin[16,26)                  0.237776   0.107007   2.222 0.026366 *
Insured.age_bin[26,34)                  0.215104   0.075064   2.866 0.004196 **
Insured.age_bin[55,61)                 -0.063860   0.086637  -0.737 0.461132
Insured.age_bin[61,103]                -0.289928   0.075233  -3.854 0.000119 ***
car.age_bin[0,1)                       -0.145690   0.094026  -1.549 0.121396
car.age_bin[4,6)                       -0.186775   0.073356  -2.546 0.010950 *
car.age_bin[6,12)                      -0.220804   0.063880  -3.457 0.000556 ***
car.age_bin[12,20]                     -0.379747   0.127256  -2.984 0.002871 **
Credit.score_bin[777,856)              -0.315728   0.057981  -5.445 5.66e-08 ***
Credit.score_bin[856,900]              -0.607437   0.080287  -7.566 5.35e-14 ***
Annual.miles.drive_bin[0,9.13e+03)     -0.138090   0.055002  -2.511 0.012112 *
Annual.miles.drive_bin[1.5e+04,1.78e+04) -0.077496   0.144053  -0.538 0.590643
Annual.miles.drive_bin[1.78e+04,1.86e+04] -0.356313   0.138585  -2.571 0.010194 *
Territory                               0.001518   0.001119   1.357 0.175004
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 1.710064)

    Null deviance: 3542.7  on 2570  degrees of freedom
Residual deviance: 3243.7  on 2556  degrees of freedom
AIC: 46466

Number of Fisher Scoring iterations: 7
```

Fuente: Elaboración propia

ANEXO N - MODELO GLM REGULARIZADO FRECUENCIA

```
Model Details:
==============

H2ORegressionModel: glm
Model ID:  modelo_glm_tel
GLM Model: summary
            family link                                regularization number_of_predictors_total
1 negativebinomial  log Elastic Net (alpha = 0.1, lambda = 0.002147 )                         54
  number_of_active_predictors number_of_iterations  training_frame
1                          38                    6 datos_train_H2O

Coefficients: glm coefficients
               names coefficients standardized_coefficients
1          Intercept    -3.743188                 -3.439733
2 Car.use.Commercial     0.000000                  0.000000
3    Car.use.Commute    -0.036663                 -0.036663
4     Car.use.Farmer     0.000000                  0.000000
5    Car.use.Private     0.000000                  0.000000

---
                   names coefficients standardized_coefficients
50  Left.turn.intensity12     0.000922                  0.019612
51 Right.turn.intensity08     0.000000                  0.000000
52 Right.turn.intensity09     0.000000                  0.000000
53 Right.turn.intensity10     0.000000                  0.000000
54 Right.turn.intensity11     0.000188                  0.017084
55 Right.turn.intensity12     0.001026                  0.048468

H2ORegressionMetrics: glm
** Reported on training data. **

MSE:  0.04348989
RMSE:  0.2085423
MAE:  0.07895563
RMSLE:  0.1404572
Mean Residual Deviance :  0.1550507
R^2 :  0.06575247
Null Deviance :12533.24
Null D.o.F. :65960
Residual Deviance :10227.3
Residual D.o.F. :65922
AIC :21143.12
```

Fuente: Elaboración propia

ANEXO Ñ - MODELO GLM REGULARIZADO SEVERIDAD

```
Model Details:
==============

H2ORegressionModel: glm
Model ID:  modelo_glm_tel_sev
GLM Model: summary
  family link                                regularization number_of_predictors_total number_of_active_predictors
1  gamma  log Elastic Net (alpha = 0.1, lambda = 0.05937 )                          54                           26
  number_of_iterations      training_frame
1                    7 datos_train_h2o_sev

Coefficients: glm coefficients
               names coefficients standardized_coefficients
1          Intercept    10.818250                  7.953699
2 Car.use.Commercial     0.000000                  0.000000
3    Car.use.Commute     0.074661                  0.074661
4     Car.use.Farmer     0.000000                  0.000000
5    Car.use.Private     0.000000                  0.000000

---
                   names coefficients standardized_coefficients
50  Left.turn.intensity12    -0.002136                 -0.064240
51 Right.turn.intensity08     0.000000                  0.000000
52 Right.turn.intensity09     0.000000                  0.000000
53 Right.turn.intensity10     0.000000                  0.000000
54 Right.turn.intensity11     0.000000                  0.000000
55 Right.turn.intensity12     0.000000                  0.000000

H2ORegressionMetrics: glm
** Reported on training data. **

MSE:  20170756
RMSE:  4491.187
MAE:  2487.888
RMSLE:  1.334445
Mean Residual Deviance :  1.170192
R^2 :  0.09685809
Null Deviance :3536.155
Null D.o.F. :2492
Residual Deviance :3074.094
Residual D.o.F. :2466
AIC :NaN
```

Fuente: Elaboración propia

ANEXO O – IMPORTANCIA DE LOS PREDICTORES EN GBM DE FRECUENCIA

	variable	relative_importance	scaled_importance	percentage
1	Total,miles,driven	766,2778	1,0000	0,3848
2	Brake,08miles	270,9036	0,3535	0,1360
3	Annual,pct,driven	233,6405	0,3049	0,1173
4	Car,age	135,7034	0,1771	0,0681
5	Credit,score	128,2353	0,1673	0,0644
6	Left,turn,intensity12	67,1990	0,0877	0,0337
7	Pct,drive,thr	47,3990	0,0619	0,0238
8	Accel,06miles	43,1427	0,0563	0,0217
9	Brake,11miles	37,4924	0,0489	0,0188
10	Left,turn,intensity10	35,9492	0,0469	0,0181
11	Pct,drive,rush,am	27,7637	0,0362	0,0139
12	Accel,08miles	23,0039	0,0300	0,0116
13	Brake,12miles	19,6180	0,0256	0,0099
14	Left,turn,intensity08	19,3533	0,0253	0,0097
15	Avgdays,week	19,2751	0,0252	0,0097
16	Pct,drive,wkday	19,0793	0,0249	0,0096
17	Insured,age	16,8975	0,0221	0,0085
18	Territory	12,8671	0,0168	0,0065
19	Pct,drive,3hrs	9,2250	0,0120	0,0046
20	Right,turn,intensity12	9,0637	0,0118	0,0046
21	Pct,drive,2hrs	8,4857	0,0111	0,0043
22	Right,turn,intensity11	8,3795	0,0109	0,0042
23	Pct,drive,mon	6,7360	0,0088	0,0034
24	Pct,drive,sun	5,8635	0,0077	0,0029
25	Pct,drive,wed	4,1250	0,0054	0,0021
26	Left,turn,intensity11	3,4266	0,0045	0,0017
27	Annual,miles,drive	3,3415	0,0044	0,0017
28	Pct,drive,wkend	2,6313	0,0034	0,0013
29	Pct,drive,tue	2,3048	0,0030	0,0012
30	Marital	1,9005	0,0025	0,0010
31	Pct,drive,4hrs	1,0289	0,0013	0,0005
32	Pct,drive,rush,pm	0,6438	0,0008	0,0003
33	Car,use	0,5434	0,0007	0,0003
34	Region	0,0000	0,0000	0,0000
35	Accel,11miles	0,0000	0,0000	0,0000
36	Accel,12miles	0,0000	0,0000	0,0000

Fuente: Elaboración propia

ANEXO P – MÉTRICAS DE LOS MODELOS GBM DE SEVERIDAD.

Modelo	learn_rate	max_depth	ntrees	sample_rate	model_ids	rmse	mse	mae	R^2
1	0,05	4	110	0,7	grid_gbm_sev_tel_model_1	4234,7985	17933518,5441	2205,6991	0,1970
2	0,05	5	70	0,9	grid_gbm_sev_tel_model_6	4292,3357	18424145,5575	2186,8510	0,1751
3	0,05	4	110	0,9	grid_gbm_sev_tel_model_2	4315,8943	18626943,2244	2194,9566	0,1660
4	0,05	2	50	0,7	grid_gbm_sev_tel_model_4	4528,4542	20506897,3065	2430,9437	0,0818
5	0,01	2	120	0,7	grid_gbm_sev_tel_model_9	4591,,51783	21082036,0086	2494,6104	0,0561
6	0,01	3	50	0,7	grid_gbm_sev_tel_model_5	4608,9634	21242543,9012	2526,0208	0,0489
7	0,01	2	20	0,9	grid_gbm_sev_tel_model_8	4693,3693	22027715,0904	2609,0075	0,0137
8	0,001	4	100	0,7	grid_gbm_sev_tel_model_7	4695,3749	22046544,9741	2611,9410	0,0129
9	0,001	4	70	0,7	grid_gbm_sev_tel_model_10	4706,3563	22149789,9718	2619,5829	0,0083
10	0,001	3	40	0,7	grid_gbm_sev_tel_model_3	4717,4764	22254584,0012	2635,5992	0,0036

Fuente: Elaboración propia

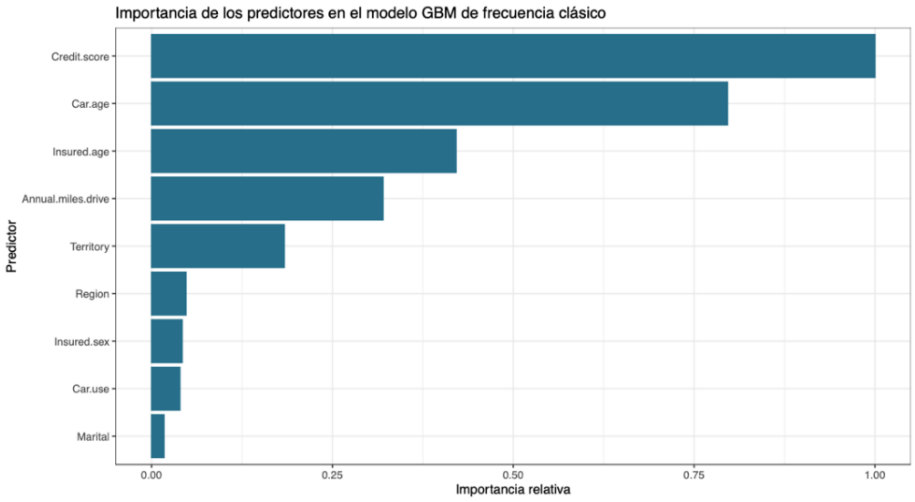
ANEXO Q – IMPORTANCIA DE LOS PREDICTORES EN GBM DE FRECUENCIA

	variable	relative_importance	scaled_importance	percentage
1	Credit,score	2779,7244	1,0000	0,1952
2	Annual,pct,driven	1343,1971	0,4832	0,0943
3	Avgdays,week	1181,6615	0,4251	0,0830
4	Brake,06miles	857,0109	0,3083	0,0602
5	Pct,drive,mon	828,9276	0,2982	0,0582
6	Brake,08miles	796,7631	0,2866	0,0560
7	Left,turn,intensity08	754,5048	0,2714	0,0530
8	Pct,drive,sun	615,6436	0,2215	0,0432
9	Car,age	572,5925	0,2060	0,0402
10	Pct,drive,sat	552,8067	0,1989	0,0388
11	Total,miles,driven	535,6572	0,1927	0,0376
12	Insured,age	522,3093	0,1879	0,0367
13	Left,turn,intensity12	521,6947	0,1877	0,0366
14	Territory	501,0189	0,1802	0,0352
15	Pct,drive,rush,am	385,8803	0,1388	0,0271
16	Accel,06miles	347,0237	0,1248	0,0244
17	Pct,drive,2hrs	293,1043	0,1054	0,0206
18	Annual,miles,drive	255,4700	0,0919	0,0179
19	Brake,11miles	144,3994	0,0519	0,0101
20	Accel,12miles	110,4738	0,0397	0,0078
21	Accel,08miles	108,0886	0,0389	0,0076
22	Accel,14miles	83,4187	0,0300	0,0059
23	Pct,drive,4hrs	60,3088	0,0217	0,0042
24	Car,use	50,0872	0,0180	0,0035
25	Marital	35,3419	0,0127	0,0025

Fuente: Elaboración propia

ANEXO R – GBM VARIABLES CLÁSICAS

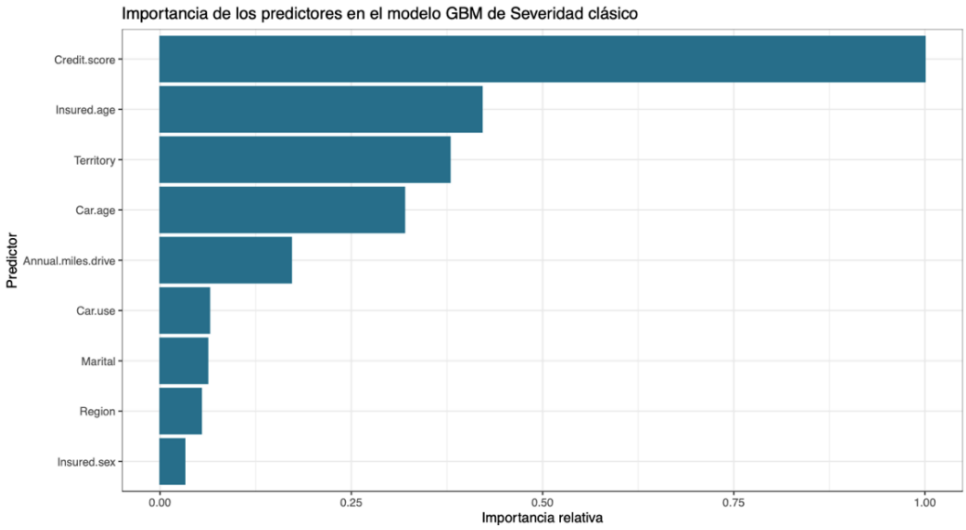
```
Model Details:
==============

H2ORegressionModel: gbm
Model ID:  grid_gbm_freq_cl_model_2
Model Summary:
  number_of_trees number_of_internal_trees model_size_in_bytes min_depth max_depth mean_depth min_leaves
1             157                      157               24862         0         4    3.97452          1
  max_leaves mean_leaves
1         13     7.95541


H2ORegressionMetrics: gbm
** Reported on training data. **

MSE:  0.04543633
RMSE:  0.213158
MAE:  0.08272019
RMSLE:  0.1430702
Mean Residual Deviance :  0.3411801



H2ORegressionMetrics: gbm
** Reported on cross-validation data. **
** 5-fold cross-validation on training data (Metrics computed for combined holdout predictions) **

MSE:  0.04575271
RMSE:  0.2138988
MAE:  0.08299707
RMSLE:  0.1437472
Mean Residual Deviance :  0.3468599


Cross-Validation Metrics Summary:
                            mean       sd cv_1_valid cv_2_valid cv_3_valid cv_4_valid cv_5_valid
mae                     0.083007 0.002012   0.082065   0.082523   0.082409   0.086536   0.081502
mean_residual_deviance  0.346949 0.019202   0.337655   0.345784   0.343118   0.379297   0.328892
mse                     0.045765 0.002898   0.044389   0.045549   0.045712   0.050480   0.042695
r2                      0.016939 0.002186   0.019299   0.014463   0.018472   0.017656   0.014804
residual_deviance       0.346949 0.019202   0.337655   0.345784   0.343118   0.379297   0.328892
rmse                    0.213844 0.006701   0.210688   0.213422   0.213805   0.224678   0.206627
rmsle                   0.143716 0.004329   0.141853   0.143136   0.142963   0.151036   0.139590
```



Fuente: Elaboración propia

ANEXO S – GBM DE SEVERIDAD VARIABLES CLÁSICAS

```
H2ORegressionModel: gbm
Model ID:  grid_gbm_sev_cl_model_1
Model Summary:
  number_of_trees number_of_internal_trees model_size_in_bytes min_depth max_depth mean_depth min_leaves
1             110                      110              22193         4         4    4.00000          6
  max_leaves mean_leaves
1         16    11.43636


H2ORegressionMetrics: gbm
** Reported on training data. **

MSE:  17539725
RMSE:  4188.045
MAE:  2214.296
RMSLE:  1.224144
Mean Residual Deviance :  17.80611



H2ORegressionMetrics: gbm
** Reported on cross-validation data. **
** 5-fold cross-validation on training data (Metrics computed for combined holdout predictions) **

MSE:  19794384
RMSE:  4449.088
MAE:  2373.729
RMSLE:  1.27956
Mean Residual Deviance :  17.99627


Cross-Validation Metrics Summary:
                                mean            sd      cv_1_valid      cv_2_valid      cv_3_valid
mae                       2382.271500    240.102040     2491.282700     2125.473000     2562.719200
mean_residual_deviance      18.003530      0.162529       18.055367       17.739857       18.040518
mse                   19977572.000000 5201408.000000 23455740.000000 13302991.000000 24502450.000000
r2                            0.111208       0.025223        0.110043        0.110388        0.141629
residual_deviance            18.003530      0.162529       18.055367       17.739857       18.040518
rmse                       4436.950000    603.164370     4843.113000     3647.326400     4949.995000
rmsle                         1.276509       0.084275        1.314309        1.358202        1.261994
                          cv_4_valid      cv_5_valid
mae                       2121.327100    2610.555700
mean_residual_deviance      17.999317      18.182600
mse                   15426570.000000 23200112.000000
r2                            0.121619       0.072362
residual_deviance           17.999317      18.182600
rmse                       3927.667200    4816.649400
rmsle                         1.138623       1.309420
```



Fuente: Elaboración propia

ANEXO T - CÓDIGO RSTUDIO

```
rm(list = ls())


packages <- c("tidyverse", "dplyr", "mgcv", "evtree",
"classInt","kableExtra", "rgdal", "RColorBrewer", "grid",
"goftest",
              "gridExtra", "visreg", "sf", "fitdistrplus",
"leaflet", "fastDummies", "AER", "pROC", "ineq", "openxlsx",
"ggplotify",
              "MuMIn", "stats","MASS",
"rpart","repr","rpart.plot","rsample","caret","corrplot","tidymo
dels","vcd", "h2o")
suppressMessages(packages <- lapply(packages, FUN = function(x)
{
  if (!require(x, character.only = TRUE)) {
    install.packages(x)
    library(x, character.only = TRUE)
  }
}))


data <- as_tibble(read.csv(file.choose(), header = TRUE))
data_AMT_claim1 <- subset(data, AMT_Claim > 0) ## Los claim
amount que son mayores que 0
data_AMT_claim2 <- subset(data_AMT_claim1, AMT_Claim <=
quantile(data_AMT_claim1$AMT_Claim,0.99)) ## Los claim amount
que son mayores que 0 quitando outliers


######## Análisis Estadístico variables clásicas ########
KULbg <- "#116E8A"
col <- KULbg
fill <- KULbg
ylab <- "Frec Relativa"
##### Análisis de las variables clásicas #####


## Funciones para realizar los gráficos ##
ggplot.hist <- function(DT, variable, xlab, binwidth){
```

```
  ggplot(data = DT, aes(variable)) + theme_bw() +
    geom_histogram(aes(y = (..count..)/sum(..count..)), binwidth
= binwidth, col = col, fill = fill, alpha = 0.7) +
    labs(x = xlab, y = ylab)
}

ggplot.bar <- function(DT, variable, xlab){
  ggplot(data = DT, aes(as.factor(variable))) + theme_bw() +
    geom_bar(aes(y = (..count..)/sum(..count..)), col = col,
fill = fill, alpha = 0.7) +
    labs(x = xlab, y = ylab) +
    geom_text(stat='count',
aes(label=paste0(round(prop.table(..count..),4)*100,
"%"),y=(..count..)/sum(..count..)),
              vjust=1.3, nudge_y = 0.2)
}

ggplot.dens <- function(DT, variable, xlab){
  ggplot(data = DT, aes(variable)) + theme_bw() +
    geom_density(col = col, fill = fill, alpha = 0.7) + labs(x =
xlab, y = "Density")
}

## gráficos de las variables respuesta
graficos_response <- grid.arrange(
  plot_NB_Claim <- ggplot.bar(data, data$NB_Claim, "Numero de
siniestros"),
  plot_AMT_Claim <- ggplot.dens(data_AMT_claim1,
data_AMT_claim1$AMT_Claim, "Valor de Siniestros"),
  ncol = 2
)

## gráficos variables categóricas ##
graficos_classic1 <- grid.arrange(
  plot_marital <- ggplot.bar(data, data$Marital, "Estado
Civil"),
  plot_sex <- ggplot.bar(data, data$Insured.sex, "Sexo"),
```

```
  plot_car_use <-ggplot.bar(data, data$Car.use, "Uso del
Coche"),
  plot_region <-ggplot.bar(data, data$Region, "Región donde vive
el asegurado"),
  ncol = 2)


graficos_classic2 <- grid.arrange(
  plot_Duration <- ggplot.hist(data, data$Duration,
"Exposición", 2),
  plot_age <- ggplot.hist(data, data$Insured.age, "Edad del
Asegurado", 2),
  plot_car_age <- ggplot.hist(data, data$Car.age, "Antigüedad
coche", 2),
  plot_credit_score <- ggplot.hist(data, data$Credit.score,
"Credit Score", 2),
  plot_years_no_claims <-ggplot.hist(data, data$Years.noclaims,
"Años sin siniestros",2),
  plot_Territory <- ggplot.hist(data, data$Territory,
"Territorio",2),
  plot_annual_miles_drive <- ggplot.dens(data,
data$Annual.miles.drive, "Millas conducidas en 1 año"),
  ncol = 4
)


##### Preparación del dato #####
datos <- as_tibble(data)
datos$Duration <- ifelse(datos$Duration == 366, 365,
datos$Duration)
datos$Duration <- (datos$Duration)/365
datos <- rename(datos, Exposure = Duration)
datos <- subset(datos, !(datos$NB_Claim > 0 &
datos$AMT_Claim==0))
datos <- datos[datos$Car.age >= 0,]
datos <- datos[datos$Annual.miles.drive <=
quantile(datos$Annual.miles.drive,0.99),]
datos <- subset(datos, AMT_Claim <= 55000) ## Los claim amount
que son mayores que 0 quitando outliers
```

```
datos$AMT_avg <- ifelse(datos$NB_Claim == 0, 0,
datos$AMT_Claim/datos$NB_Claim)


datos$Insured.sex <- as.factor(datos$Insured.sex)

datos$Marital <- as.factor(datos$Marital)

datos$Car.use <- as.factor(datos$Car.use)

datos$Region <- as.factor(datos$Region)

datos$Car.use <- as.factor(datos$Car.use)


AMT_Claims1 <- subset(datos, AMT_Claim > 0) ## Los claim amount
que son mayores que 0


##0.0474

emp_freq <- datos %>%

  summarize(emp_freq = sum(NB_Claim) / sum(Exposure))


m <- sum(datos$NB_Claim)/sum(datos$Exposure)

m

##0.0506258

var <- sum((datos$NB_Claim - m * datos$Exposure)^2)/

  sum(datos$Exposure)


###### Análisis del ajuste de Frecuencia y Severidad ######

#### Análisis Distribución freq --> Binomial Negativa ####

freq_classic_poiss <- glm(NB_Claim ~ 1, data = datos, offset =
log(Exposure), family = poisson(link="log"))

freq_classic_bin_nega <- glm.nb(NB_Claim ~ 1, data = datos,
offset=log(Exposure), link = "log"  ,control =
glm.control(maxit=1000000))


resultados_freq <- data.frame(Modelo = c("Poisson", "Binomial
Negativa"), AIC = c(AIC(freq_classic_poiss),
AIC(freq_classic_bin_nega)), BIC = c(BIC(freq_classic_poiss),
BIC(freq_classic_bin_nega)))

resultados_freq
```

```
##Test de Dispersión ## Ho es No disperso. Rechazo y alpha es > 0. Es Sobredispersa.
dispersiontest(freq_classic_poiss,trafo=1) ##Test Poisson


PoisModel<- goodfit(freq_classic_poiss$y, type = "poisson")
plot(PoisModel, type = "standing", xlab="count", main = "Poisson Model")
plot(PoisModel, xlab="count", main = "Rootogram - Poisson")


nbinomModel <- goodfit(freq_classic_bin_nega$y, type="nbinomial")
plot(nbinomModel, type = "standing", xlab="count", main = "Negative Binomial Model")
plot(nbinomModel, xlab="count", main = "Rootogram - Negative Binomial")


summary(nbinomModel) ##No rechazo Ho por lo que se ajusta a una distribución Binomial Negativa. pval = 0.2099


#Log likelyhood para ambos modelos
models <- list("Pois" = freq_classic_poiss, "NB" = freq_classic_bin_nega)
df_log <- datos.frame(rbind(logLik = sapply(models, function(x) round(logLik(x), digits = 0)),
                            Df = sapply(models, function(x) attr(logLik(x), "df"))))
df_log


#### Análisis Distribución Sev  --> Gamma ####
graficos_sev <- grid.arrange(
  ggplot.dens(AMT_Claims1_out, AMT_Claims1_out$AMT_Claim/AMT_Claims1_out$NB_Claim, "Severidad"),
  ggplot.dens(AMT_Claims1, AMT_Claims1$AMT_Claim/AMT_Claims1$NB_Claim, "Severidad sin outlier"),
  ncol = 2)
```

```
sev_classic_gauss <- glm(AMT_avg ~ 1, data = AMT_Claims1,
family=gaussian(link="log"))

summary(sev_classic_gauss)

plot(sev_classic_gauss)


sev_classic_gamma <- glm(AMT_avg ~ 1, data = AMT_Claims1,
family=Gamma(link="log"))

summary(sev_classic_gamma)

plot(sev_classic_gamma)

gamma_fit <- fitdist((sev_classic_gamma$residuals), "gamma")

gofstat(gamma_fit)


resultados_sev <- data.frame(Modelo = c("Log-Normal", "Gamma"),
AIC = c(AIC(sev_classic_gauss), AIC(sev_classic_gamma)), BIC =
c(BIC(sev_classic_gauss), BIC(sev_classic_gamma)))

resultados_sev


### calculo calculo Sesgo, MSE y consistencia de una gamma ###

polizas <- length(datos$NB_Claim) #97174

Siniestros <- sum(AMT_Claims1$NB_Claim) #3934

sim<-1000


parametro=0.35

p = 0.0001

N =100000

teorica_gamma<-rgamma(N,parametro,p)

histogram(teorica_gamma, col = col, xlab = "Valores", ylab =
"Densidad",

          main = paste0("Distribución Gamma teórica con shape =
", parametro, " y rate = ", N))

summary(AMT_Claims1$AMT_Claim/AMT_Claims1$NB_Claim)

summary(teorica_gamma)


M_sim<-matrix(0,sim,1)

PM_sim<-matrix(0,sim,1)

ML_sim<-matrix(0,sim,1)
```

```
bar <- txtProgressBar(0,sim,style=3)
for (i in 1:sim){
  muestra<-sample(teorica_gamma,Siniestros,replace=FALSE)
  #M□todo de los momentos
  dif<-function(param){
    r1<-(param[1]/param[2]-mean(muestra))^2
    r2<-(param[1]/(param[2]^2)-var(muestra))^2
    return(r1+r2)
  }
  MM<-optim(c(1,0.5),dif,method="L-BFGS-B")
  M_sim[i]<-MM$par[1]
  #M□todo Percentile Matching
  param<-c()
  dif<-function(param){
    r1<-(qgamma(0.1,param[1],param[2])-quantile(muestra,0.1))^2
    r2<-(qgamma(0.8,param[1],param[2])-quantile(muestra,0.8))^2
    return(r1+r2)
  }
  PM<-
optim(c(mean(muestra)^2/var(muestra),mean(muestra)/var(muestra))
,dif,method="L-BFGS-B", lower=0.0001)
  PM_sim[i]<-PM$par[1]

  #M□todo M□ximo Verosimilitud
  param<-c()
  LL <- function(param) {
    -sum(dgamma(muestra, param[1], param[2], log=TRUE))
  }
  ML <-
optim(c(mean(muestra)^2/var(muestra),mean(muestra)/var(muestra))
,LL,method="L-BFGS-B", lower=0.0001)
  ML_sim[i]<-ML$par[1]
  setTxtProgressBar(bar, i)
}
#Sesgo
```

```
Bias_M=parametro-mean(M_sim)
Bias_PM=parametro-mean(PM_sim)
Bias_ML=parametro-mean(ML_sim)
#MSE
MSE_M=Bias_M^2+var(M_sim)
MSE_PM=Bias_PM^2+var(PM_sim)
MSE_ML=Bias_ML^2+var(ML_sim)

Resultados<-
matrix(c(Bias_M,Bias_PM,Bias_ML,MSE_M,MSE_PM,MSE_ML),3,2)
rownames(Resultados)<-c("Momentos","PM","ML")
colnames(Resultados)<-c("sesgo","MSE")
Resultados

##Consistencia
limite=1000
min=50

M_sim<-matrix(0,limite-min,1)
PM_sim<-matrix(0,limite-min,1)
ML_sim<-matrix(0,limite-min,1)
bar <- txtProgressBar(0,limite,style=3)
i=1
for (n in min:limite){
  muestra<-sample(teorica_gamma,n,replace=FALSE)
  #M□todo de los momentos
  dif<-function(param){
    r1<-(param[1]/param[2]-mean(muestra))^2
    r2<-(param[1]/(param[2]^2)-var(muestra))^2
    return(r1+r2)
  }
  MM<-optim(c(1,0.5),dif,method="L-BFGS-B")
  M_sim[i]<-MM$par[1]
  #M□todo Percentile Matching
```

```
  param<-c()
  dif<-function(param){
    r1<-(qgamma(0.1,param[1],param[2])-quantile(muestra,0.1))^2
    r2<-(qgamma(0.8,param[1],param[2])-quantile(muestra,0.8))^2
    return(r1+r2)
  }
  PM<-
optim(c(mean(muestra)^2/var(muestra),mean(muestra)/var(muestra))
,dif,method="L-BFGS-B",lower = 0.0001)
  PM_sim[i]<-PM$par[1]
  #M□todo M□ximo Verosimilitud
  param<-c()
  LL <- function(param) {
    -sum(dgamma(muestra, param[1], param[2], log=TRUE))
  }
  ML <-
optim(c(mean(muestra)^2/var(muestra),mean(muestra)/var(muestra))
,LL,method="L-BFGS-B", lower = 0.0001)
  ML_sim[i]<-ML$par[1]
  setTxtProgressBar(bar, n)
  i=i+1
}
#Consistencia
par(mfrow = c(1,2))
ts.plot(M_sim)
abline(parametro,0,col="red")
ts.plot(PM_sim)
abline(parametro,0,col="red")
ts.plot(ML_sim)
abline(parametro,0,col="red")
#par(mfrow = c(1,1))

### Validación Hipótesis ###
sim = 5000
estimaciones_boot_MM<-c()
```

```
estimaciones_boot_PM<-c()
estimaciones_boot_ML<-c()
muestra_b<-sample(teorica_gamma,Siniestros,FALSE)
bar <- txtProgressBar(0,sim,style=3)
for (b in 1:sim){
  muestra_boot<-sample(muestra_b,Siniestros,TRUE)
  #M◆todo de los momentos
  dif<-function(param){
    r1<-(param[1]/param[2]-mean(muestra_boot))^2
    r2<-(param[1]/(param[2]^2)-var(muestra_boot))^2
    return(r1+r2)
  }
  MM<-optim(c(1,0.5),dif,method="L-BFGS-B")
  estimaciones_boot_MM[b]<-MM$par[1]

  #M◆todo Percentile Matching
  param<-c()
  dif<-function(param){
    r1<-(qgamma(0.1,param[1],param[2])-
quantile(muestra_boot,0.1))^2
    r2<-(qgamma(0.8,param[1],param[2])-
quantile(muestra_boot,0.8))^2
    return(r1+r2)
  }
  PM<-
optim(c(mean(muestra_boot)^2/var(muestra_boot),mean(muestra_boot
)/var(muestra_boot)),dif,method="L-BFGS-B",lower=0.0001)
  estimaciones_boot_PM[b]<-PM$par[1]

  LL <- function(param) {
    -sum(dgamma(muestra_boot, param[1], param[2], log=TRUE))
  }
  ML <-
optim(c(mean(muestra_boot)^2/var(muestra_boot),mean(muestra_boot
)/var(muestra_boot)),LL,method="L-BFGS-B",lower=0.0001)
  estimaciones_boot_ML[b]<-ML$par[1]
```

```
  setTxtProgressBar(bar, b)
}

Resultados_estimador_gamma<-
matrix(c(quantile(estimaciones_boot_MM,0.025),quantile(estimacio
nes_boot_PM,0.025),quantile(estimaciones_boot_ML,0.025),quantile
(estimaciones_boot_MM,0.975),quantile(estimaciones_boot_PM,0.975
),quantile(estimaciones_boot_ML,0.975)),3,2)
rownames(Resultados_estimador_gamma)<-c("Momentos","PM","ML")
colnames(Resultados_estimador_gamma)<-c("2.5%","97.5%")
Resultados_estimador_gamma

hist(estimaciones_boot_ML, freq = F, col=col, ylim = c(0,80),
xlab = "Estimaciones Bootstrap por ML", ylab = "Densidad",
     main = paste0("Distribución parámetro shape estimado por
ML"))
polygon(density(estimaciones_boot_ML),col = rgb(0, 0, 0, 0.3))

### sacamos los parametros por ML de nuestra muestra
AMT_Claims1$sev ##
AMT_Claims1$sev <- AMT_Claims1$AMT_Claim/AMT_Claims1$NB_Claim
l_gamma<-mean(AMT_Claims1$sev)^2/(var(AMT_Claims1$sev))
p_gamma<-mean(AMT_Claims1$sev)/(var(AMT_Claims1$sev))
#Máxima Verosimilitud
param<-c()
LL_gamma<- function(param) {
  -sum(dgamma(AMT_Claims1$sev, param[1], param[2], log=TRUE))
}
ML_gamma<- optim(c(l_gamma,p_gamma),LL_gamma,method="L-BFGS-
B",lower=0.00001)
ML_gamma$par

## Test de ajuste
escenarios<-100000
gamma_ajustada<-
rgamma(escenarios,ML_gamma$par[1],ML_gamma$par[2])
```

```
ad.test(gamma_ajustada, "pgamma",
ML_gamma$par[1],ML_gamma$par[2],estimated=TRUE) ##No rechazo la
Ho por lo que es una Gamma con esos parámetros



#### Splitting 70/30 y modelización ####

set.seed(5678)

rsample <- initial_split(datos,prop = 0.70)

train <- training(rsample)

test <- testing(rsample)


#### GAM Variables clásicas####

train_classic = train[c("Exposure", "Insured.age",
"Insured.sex", "Car.age", "Marital", "Years.noclaims",
                        "Car.use", "Region", "Credit.score",
"Annual.miles.drive", "Territory", "NB_Claim", "AMT_Claim")]


#str(dummies_classic)

dummies_classic <- dummy_columns(train_classic,
                                 remove_most_frequent_dummy =
TRUE,
                                 remove_selected_columns = TRUE)


dummies_classic_cont <- dummies_classic[c("Insured.age",
"Car.age", "Years.noclaims",
                                          "Credit.score",
"Annual.miles.drive", "Territory")]

correlaciones_pearson <- cor(dummies_classic_cont)

corrplot(correlaciones_pearson, method="circle")

cor(dummies_classic_cont$Insured.age,dummies_classic_cont$Years.
noclaims) #0.8285616


dummies_classic_cat <- dummies_classic[c("Insured.sex_Female",
"Marital_Single", "Car.use_Commercial", "Car.use_Farmer",
                                        "Car.use_Private",
"Region_Rural")]
```

```
cramer_train1 <-
table(dummies_classic$Marital_Single,dummies_classic$Region_Rura
l)

cramer1 <- assocstats(cramer_train1)$cramer

cramer1#0.3436816


cramer_train2 <-
table(dummies_classic$Insured.age,dummies_classic$Car.use_Privat
e)

cramer2 <- assocstats(cramer_train2)$cramer

cramer2#0.5339835


train_classic <- train_classic[, -
which(names(train_classic)=="Years.noclaims")]

dummies_classic <- dummies_classic[, -
which(names(dummies_classic)=="Years.noclaims")]


train_classic$Car.use <- relevel(train_classic$Car.use, ref =
"Commute")

train_classic$Marital <- relevel(train_classic$Marital, ref =
"Married")

train_classic$Insured.sex <- relevel(train_classic$Insured.sex,
ref = "Male")

train_classic$Region <- relevel(train_classic$Region, ref =
"Urban")


#### Frecuencia ####

# Función para graficar los plots de los modelos GAM

ggplot.gam <- function(model, variable, gam_term,

                       xlabel, ylabel){

  pred <- predict(model, type = "terms", se = TRUE)

  col_index <- which(colnames(pred$fit)==gam_term)

  x <- variable

  b <- pred$fit[, col_index]

  l <- pred$fit[, col_index] -

    qnorm(0.975) * pred$se.fit[, col_index]

  u <- pred$fit[, col_index] +
```

```
    qnorm(0.975) * pred$se.fit[, col_index]
  df <- unique(data.frame(x, b, l, u))
  p <- ggplot(df, aes(x = x))
  p <- p + geom_line(aes(y = b), size = 1,
                     col = "#003366")
  p <- p + geom_line(aes(y = l), size = 0.5,
                     linetype = 2, col = "#99CCFF")
  p <- p + geom_line(aes(y = u), size = 0.5,
                     linetype = 2, col = "#99CCFF")
  p <- p + xlab(xlabel) + ylab(ylabel) + theme_bw()
  p
}


## Proceso de automatización para encontrar el mejor GAM de
Frecuencia ##
vars <- c("s(Insured.age)", "Insured.sex ", "s(Car.age)",
"Marital", "Car.use",
          "Region", "s(Credit.score)", "s(Annual.miles.drive)",
"s(Territory)")


combos <- unlist(lapply(seq_along(vars), function(i) combn(vars,
i, simplify = FALSE)), recursive = FALSE)
metrics <- data.frame(variables = character(), AIC = numeric(),
BIC = numeric(), stringsAsFactors = FALSE)
bar<-txtProgressBar(0,length(combos),style=3)


for (i in seq_along(combos)) {
  # construye la fórmulas para el modelo
  formula <- as.formula(paste("NB_Claim ~", paste(combos[[i]],
collapse = "+")))
  # ajusta el modelo GAM
  model <- gam(formula, data = train_classic, offset =
log(Exposure), family = nb(link = 'log'))
  # guarda las métricas AIC y BIC del modelo en la tabla
  metrics[i, "variables"] <- paste(combos[[i]], collapse = "+")
  metrics[i, "AIC"] <- AIC(model)
```

```
  metrics[i, "BIC"] <- BIC(model)
  setTxtProgressBar(bar,i)
}

# muestra la tabla de métricas
head(metrics %>% arrange(AIC))

# mejor model gam freq
model_gam_freq <- gam(NB_Claim ~ s(Insured.age) + s(Car.age) +
Region + s(Credit.score) + s(Annual.miles.drive)
                      + s(Territory),
                     data = train_classic, offset =
log(Exposure), family = nb(link = 'log'))
summary(model_gam_freq)
gam.check(model_gam_freq, k.rep=1000)
AIC(model_gam_freq)
BIC(model_gam_freq)

plot_model_gam_freq_Insured.age <- ggplot.gam(model_gam_freq,
train_classic$Insured.age,
                                              "s(Insured.age)",
"Insured.age",

expression(hat(f)(Insured.age)))
plot_model_gam_freq_Car.age <- ggplot.gam(model_gam_freq,
train_classic$Car.age,
                                          "s(Car.age)",
"Car.age",

expression(hat(f)(Car.age)))
plot_model_gam_freq_Credit.score <- ggplot.gam(model_gam_freq,
train_classic$Credit.score,
                                               "s(Credit.score)",
"Credit.score",

expression(hat(f)(Credit.score)))
plot_model_gam_freq_Annual.miles.drive <-
ggplot.gam(model_gam_freq, train_classic$Annual.miles.drive,
```

```
"s(Annual.miles.drive)", "Annual.miles.drive",

expression(hat(f)(Annual.miles.drive)))
plot_model_gam_freq <- grid.arrange(
  plot_model_gam_freq_Insured.age,
  plot_model_gam_freq_Car.age,
  plot_model_gam_freq_Credit.score,
  plot_model_gam_freq_Annual.miles.drive,
  ncol = 2
)


pdt_gam_freq_train <- predict.gam(model_gam_freq,train, type =
"response")
pdt_gam_freq_test <- predict.gam(model_gam_freq, test, type =
"response")


deviance_gam_freq <- -2*sum(dnbinom(test$NB_Claim, size = 1.272,
mu = pdt_gam_freq_test, log = TRUE))
deviance_gam_freq


#### Agrupaciones mediante árboles de regresión en frecuencia
####
getGAMdata_single = function(model, term, var, varname){
  pred <- predict(model, type = "terms", terms = term)
  dt_pred <- tibble("x" = var, pred)
  dt_pred <- arrange(dt_pred, x)
  names(dt_pred) = c("x", "s")
  dt_unique <- unique(dt_pred)
  dt_exp <- dt_pred %>% group_by(x) %>% summarize(tot = n())
  dt_exp <- dt_exp[c("x", "tot")]
  GAM_data <- left_join(dt_unique, dt_exp)
  names(GAM_data) <- c(varname, "s", "tot")
  GAM_data <- GAM_data[which(GAM_data$tot != 0), ]
  return(GAM_data)
}
```

```
splits_evtree = function(evtreemodel, GAMvar, DTvar){
  preds <- predict(evtreemodel, type = "node")
  nodes <- data.frame("x" = GAMvar, "nodes" = preds)
  nodes$change <- c(0, pmin(1, diff(nodes$nodes)))
  splits_evtree <- unique(c(min(DTvar),
                            nodes$x[which(nodes$change==1)],
                            max(DTvar)))
  return(splits_evtree)
}

#Insured.age
train_classic_bin <- train_classic

ctrl.freq <- evtree.control(
  minbucket = 0.10*nrow(train_classic_bin),
  minsplit = 14000,
  alpha = 500, maxdepth = 3)

freq_gam_Insured.age <- getGAMdata_single(model_gam_freq,
"s(Insured.age)", train_classic_bin$Insured.age, "Insured.age")
evtree_freq_Insured.age <- evtree(s ~ Insured.age,
                                  data = freq_gam_Insured.age,
                                  weights = tot,
                                  control = ctrl.freq)
evtree_freq_Insured.age
plot(evtree_freq_Insured.age)

freq_splits_Insured.age <- c(min(train_classic_bin$Insured.age),
26, 32, 45, 67, max(train_classic_bin$Insured.age))
train_classic_bin$Insured.age_bin <-
cut(train_classic_bin$Insured.age, freq_splits_Insured.age,
right = FALSE, include.lowest = TRUE)
summary(train_classic_bin$Insured.age_bin)
train_classic_bin$Insured.age_bin <-
relevel(train_classic_bin$Insured.age_bin, ref = "[45,67)")
```

```
plot_freq_bin_Insured.age <- ggplot.gam(model_gam_freq,

train_classic_bin$Insured.age,
                                        "s(Insured.age)",
"Insured.age",

expression(hat(f)(Insured.age)))
plot_freq_bin_Insured.age <- plot_freq_bin_Insured.age +
  geom_vline(xintercept =

freq_splits_Insured.age[2:(length(freq_splits_Insured.age)-1)])
plot_freq_bin_Insured.age


#Car.age
freq_gam_car.age <- getGAMdata_single(model_gam_freq,
"s(Car.age)", train_classic_bin$Car.age, "Car.age")


evtree_freq_car.age <- evtree(s ~ Car.age,
                              data = freq_gam_car.age,
                              weights = tot,
                              control = ctrl.freq)
evtree_freq_car.age
plot(evtree_freq_car.age)


freq_splits_car.age <- c(min(train_classic_bin$Car.age), 2, 6,
9, 13, max(train_classic_bin$Car.age))
train_classic_bin$car.age_bin <- cut(train_classic_bin$Car.age,
freq_splits_car.age, right = FALSE, include.lowest = TRUE)
summary(train_classic_bin$car.age_bin)
train_classic_bin$car.age_bin <-
relevel(train_classic_bin$car.age_bin, ref = "[2,6)")


plot_freq_bin_Car.age <- ggplot.gam(model_gam_freq,
                                    train_classic_bin$Car.age,
                                    "s(Car.age)", "Car.age",
                                    expression(hat(f)(Car.age)))
```

```
plot_freq_bin_Car.age <- plot_freq_bin_Car.age +
  geom_vline(xintercept =

freq_splits_car.age[2:(length(freq_splits_car.age)-1)])
plot_freq_bin_Car.age


#Credit.score
freq_gam_Credit.score <- getGAMdata_single(model_gam_freq,
"s(Credit.score)", train_classic_bin$Credit.score,
"Credit.score")
evtree_freq_Credit.score <- evtree(s ~ Credit.score,
                                   data = freq_gam_Credit.score,
                                   weights = tot,
                                   control = ctrl.freq)
evtree_freq_Credit.score
plot(evtree_freq_Credit.score)


freq_splits_Credit.score <-
c(min(train_classic_bin$Credit.score), 737, 802, 836, 878,
max(train_classic_bin$Credit.score))
train_classic_bin$Credit.score_bin <-
cut(train_classic_bin$Credit.score, freq_splits_Credit.score,
right = FALSE, include.lowest = TRUE)
summary(train_classic_bin$Credit.score_bin)
train_classic_bin$Credit.score_bin <-
relevel(train_classic_bin$Credit.score_bin, ref = "[836,878)")


plot_freq_bin_Credit.score <- ggplot.gam(model_gam_freq,

train_classic_bin$Credit.score,
                                         "s(Credit.score)",
"Credit.score",

expression(hat(f)(Credit.score)))
plot_freq_bin_Credit.score <- plot_freq_bin_Credit.score +
  geom_vline(xintercept =
```

```
freq_splits_Credit.score[2:(length(freq_splits_Credit.score)-
1)])

plot_freq_bin_Credit.score


#Annual.miles.drive

freq_gam_Annual.miles.drive <- getGAMdata_single(model_gam_freq,
"s(Annual.miles.drive)", train_classic_bin$Annual.miles.drive,
"Annual.miles.drive")

evtree_freq_Annual.miles.drive <- evtree(s ~ Annual.miles.drive,
                                         data =
freq_gam_Annual.miles.drive,
                                         weights = tot,
                                         control = ctrl.freq)

evtree_freq_Annual.miles.drive

plot(evtree_freq_Annual.miles.drive)


freq_splits_Annual.miles.drive <- c(0, 4101, 7084, 9942, 16031,
max(train_classic_bin$Annual.miles.drive))

train_classic_bin$Annual.miles.drive_bin <-
cut(train_classic_bin$Annual.miles.drive,
freq_splits_Annual.miles.drive, right = FALSE, include.lowest =
TRUE)

summary(train_classic_bin$Annual.miles.drive_bin)

train_classic_bin$Annual.miles.drive_bin <-
relevel(train_classic_bin$Annual.miles.drive_bin, ref =
"[4.1e+03,7.08e+03)")


plot_freq_bin_Annual.miles.drive <- ggplot.gam(model_gam_freq,


train_classic_bin$Annual.miles.drive,


"s(Annual.miles.drive)", "Annual.miles.drive",


expression(hat(f)(Annual.miles.drive)))

plot_freq_bin_Annual.miles.drive <-
plot_freq_bin_Annual.miles.drive +

  geom_vline(xintercept =
```

```
freq_splits_Annual.miles.drive[2:(length(freq_splits_Annual.mile
s.drive)-1)])

plot_freq_bin_Annual.miles.drive


#### Modelo frec tras las agrupaciones ####


model_glm_freq_bin <- glm.nb(as.formula('NB_Claim ~
Insured.age_bin + car.age_bin + Credit.score_bin +
Annual.miles.drive_bin +
                                        Region + Territory'),
                             data = train_classic_bin,
offset=log(Exposure), link = "log", control =
glm.control(maxit=1000000))

summary(model_glm_freq_bin)

AIC(model_glm_freq_bin)

BIC(model_glm_freq_bin)


#Hacer las agrupaciones en los datos de test

test_classic_bin <- test


test_classic_bin$Insured.age_bin <-
cut(test_classic_bin$Insured.age, freq_splits_Insured.age, right
= FALSE, include.lowest = TRUE)

summary(test_classic_bin$Insured.age_bin)

test_classic_bin$Insured.age_bin <-
relevel(test_classic_bin$Insured.age_bin, ref = "[45,67)")


test_classic_bin$car.age_bin <- cut(test_classic_bin$Car.age,
freq_splits_car.age, right = FALSE, include.lowest = TRUE)

summary(test_classic_bin$car.age_bin)

test_classic_bin$car.age_bin <-
relevel(test_classic_bin$car.age_bin, ref = "[2,6)")


test_classic_bin$Credit.score_bin <-
cut(test_classic_bin$Credit.score, freq_splits_Credit.score,
right = FALSE, include.lowest = TRUE)

summary(test_classic_bin$Credit.score_bin)
```

```
test_classic_bin$Credit.score_bin <-
relevel(test_classic_bin$Credit.score_bin, ref = "[836,878)")


test_classic_bin$Annual.miles.drive_bin <-
cut(test_classic_bin$Annual.miles.drive,
freq_splits_Annual.miles.drive, right = FALSE, include.lowest =
TRUE)

summary(test_classic_bin$Annual.miles.drive_bin)

test_classic_bin$Annual.miles.drive_bin <-
relevel(test_classic_bin$Annual.miles.drive_bin, ref =
"[4.1e+03,7.08e+03)")


pdt_glm_freq_bin_test <- predict.glm(model_glm_freq_bin,
test_classic_bin, type = "response")


deviance_glm_bin_freq <- -
2*sum(dnbinom(test_classic_bin$NB_Claim, size =
model_glm_freq_bin$theta, mu = pdt_glm_freq_bin_test, log =
TRUE))

deviance_glm_bin_freq


#### Severidad ####

gam_train_classic_sev = train[c("Exposure", "Insured.age",
"Insured.sex", "Car.age", "Marital",

                            "Car.use", "Region", "Credit.score",
"Annual.miles.drive", "Territory", "AMT_avg", "AMT_Claim")]


gam_train_classic_sev$Car.use <-
relevel(gam_train_classic_sev$Car.use, ref = "Commute")

gam_train_classic_sev$Marital <-
relevel(gam_train_classic_sev$Marital, ref = "Married")

gam_train_classic_sev$Insured.sex <-
relevel(gam_train_classic_sev$Insured.sex, ref = "Male")

gam_train_classic_sev$Region <-
relevel(gam_train_classic_sev$Region, ref = "Urban")


AMT_claims_train <- subset(gam_train_classic_sev, AMT_Claim > 0)
## Los claim amount que son mayores que 0
```

```
# Automatizar para encontrar el mejor GAM de Gamma #
vars <- c("s(Insured.age)", "Insured.sex ", "s(Car.age)",
"Marital", "Car.use",
          "Region", "s(Credit.score)", "s(Annual.miles.drive)",
"s(Territory)")
combos <- unlist(lapply(seq_along(vars), function(i) combn(vars,
i, simplify = FALSE)), recursive = FALSE)
metrics <- data.frame(variables = character(), AIC = numeric(),
BIC = numeric(), stringsAsFactors = FALSE)


bar<-txtProgressBar(0,length(combos),style=3)
for (i in seq_along(combos)) {
  # construye la fórmula para el modelo
  formula <- as.formula(paste("AMT_avg ~", paste(combos[[i]],
collapse = "+")))
  # ajusta el modelo GAM
  model <- gam(formula, data = AMT_claims_train, family =
Gamma(link = 'log'))
  # guarda las métricas AIC y BIC del modelo en la tabla
  metrics[i, "variables"] <- paste(combos[[i]], collapse = "+")
  metrics[i, "AIC"] <- AIC(model)
  metrics[i, "BIC"] <- BIC(model)
  setTxtProgressBar(bar,i)
}
# muestra la tabla de métricas (Cuanto mas pequeñas, mejor
ajuste.)
head(metrics %>% arrange(AIC))


#### mejor modelo Sev ####
model_gam_sev<- gam(as.formula('AMT_avg  ~  s(Insured.age) +
s(Car.age) + s(Credit.score) + s(Annual.miles.drive) +
                               s(Territory)'),
                 data = AMT_claims_train, family = Gamma(link =
'log'))
summary(model_gam_sev)
AIC(model_gam_sev)
BIC(model_gam_sev)
```

```
plot_model_gam_sev_Insured.age <- ggplot.gam(model_gam_sev,
AMT_claims_train$Insured.age,
                                             "s(Insured.age)",
"Insured.age",

expression(hat(f)(Insured.age)))
plot_model_gam_sev_Car.age <- ggplot.gam(model_gam_sev,
AMT_claims_train$Car.age,
                                           "s(Car.age)",
"Car.age",

expression(hat(f)(Car.age)))
plot_model_gam_sev_Credit.score <- ggplot.gam(model_gam_sev,
AMT_claims_train$Credit.score,
                                        "s(Credit.score)",
"Credit.score",

expression(hat(f)(Credit.score)))
plot_model_gam_sev_Annual.miles.drive <-
ggplot.gam(model_gam_sev, AMT_claims_train$Annual.miles.drive,

"s(Annual.miles.drive)", "Annual.miles.drive",

expression(hat(f)(Annual.miles.drive)))
plot_model_gam_sev <- grid.arrange(
  plot_model_gam_sev_Insured.age,
  plot_model_gam_sev_Car.age,
  plot_model_gam_sev_Credit.score,
  plot_model_gam_sev_Annual.miles.drive,
  ncol = 2
)


pdt_gam_sev_train <- predict.gam(model_gam_sev, train, type =
"response")
pdt_gam_sev_test <- predict.gam(model_gam_sev, test, type =
"response")
```

```
AMT_Claims_test <- subset(test, AMT_Claim > 0) ## Los claim
amount que son mayores que 0


#### Agrupaciones en severidad ####
AMT_claims_train_2_bin <- AMT_claims_train


ctrl.sev <- evtree.control(
  minbucket = 0.10*nrow(AMT_claims_train_2_bin),
  minsplit = 600,
  alpha = 500, maxdepth = 3)


#Insured.age
sev_gam_Insured.age <- getGAMdata_single(model_gam_sev,
"s(Insured.age)", AMT_claims_train_2_bin$Insured.age,
"Insured.age")
evtree_sev_Insured.age <- evtree(s ~ Insured.age,
                                 data = sev_gam_Insured.age,
                                 weights = tot,
                                 control = ctrl.sev)
plot(evtree_sev_Insured.age)


sev_splits_Insured.age <- c(min(train_classic$Insured.age), 26,
34, 55, 61, max(train_classic$Insured.age))
AMT_claims_train_2_bin$Insured.age_bin <-
cut(AMT_claims_train_2_bin$Insured.age, sev_splits_Insured.age,
right = FALSE, include.lowest = TRUE)
summary(AMT_claims_train_2_bin$Insured.age_bin)
AMT_claims_train_2_bin$Insured.age_bin <-
relevel(AMT_claims_train_2_bin$Insured.age_bin, ref = "[34,55)")


plot_sev_bin_Insured.age <- ggplot.gam(model_gam_sev,

AMT_claims_train_2_bin$Insured.age,
                                       "s(Insured.age)",
"Insured.age",

expression(hat(f)(Insured.age)))
```

```
plot_sev_bin_Insured.age <- plot_sev_bin_Insured.age +
  geom_vline(xintercept =

sev_splits_Insured.age[2:(length(sev_splits_Insured.age)-1)])
plot_sev_bin_Insured.age


#Car.age
sev_gam_car.age <- getGAMdata_single(model_gam_sev,
"s(Car.age)", AMT_claims_train_2_bin$Car.age, "Car.age")
evtree_sev_car.age <- evtree(s ~ Car.age,
                             data = sev_gam_car.age,
                             weights = tot,
                             control = ctrl.sev)
evtree_sev_car.age
plot(evtree_sev_car.age)


sev_splits_car.age <- c(min(train_classic$Car.age), 1, 4, 6, 12,
max(train_classic$Car.age))
AMT_claims_train_2_bin$car.age_bin <-
cut(AMT_claims_train_2_bin$Car.age, sev_splits_car.age, right =
FALSE, include.lowest = TRUE)
summary(AMT_claims_train_2_bin$car.age_bin)
AMT_claims_train_2_bin$car.age_bin <-
relevel(AMT_claims_train_2_bin$car.age_bin, ref = "[1,4)")


plot_sev_bin_Car.age <- ggplot.gam(model_gam_sev,

AMT_claims_train_2_bin$Car.age,
                                   "s(Car.age)", "Car.age",
                                   expression(hat(f)(Car.age)))
plot_sev_bin_Car.age <- plot_sev_bin_Car.age +
  geom_vline(xintercept =
               sev_splits_car.age[2:(length(sev_splits_car.age)-
1)])
plot_sev_bin_Car.age


#Credit.score
```

```
sev_gam_Credit.score <- getGAMdata_single(model_gam_sev,
"s(Credit.score)", AMT_claims_train_2_bin$Credit.score,
"Credit.score")

evtree_sev_Credit.score <- evtree(s ~ Credit.score,

                                  data = sev_gam_Credit.score,

                                  weights = tot,

                                  control = ctrl.sev)

evtree_sev_Credit.score

plot(evtree_sev_Credit.score)


sev_splits_Credit.score <- c(min(train_classic$Credit.score),
777, 856, max(train_classic$Credit.score))

AMT_claims_train_2_bin$Credit.score_bin <-
cut(AMT_claims_train_2_bin$Credit.score,
sev_splits_Credit.score, right = FALSE, include.lowest = TRUE)

summary(AMT_claims_train_2_bin$Credit.score_bin)

AMT_claims_train_2_bin$Credit.score_bin <-
relevel(AMT_claims_train_2_bin$Credit.score_bin, ref =
"[422,777)")


plot_sev_bin_Credit.score <- ggplot.gam(model_gam_sev,


AMT_claims_train_2_bin$Credit.score,

                                        "s(Credit.score)",
"Credit.score",


expression(hat(f)(Credit.score)))

plot_sev_bin_Credit.score <- plot_sev_bin_Credit.score +

  geom_vline(xintercept =


sev_splits_Credit.score[2:(length(sev_splits_Credit.score)-1)])

plot_sev_bin_Credit.score


#Annual.miles.drive

sev_gam_Annual.miles.drive <- getGAMdata_single(model_gam_sev,
"s(Annual.miles.drive)",
AMT_claims_train_2_bin$Annual.miles.drive, "Annual.miles.drive")

evtree_sev_Annual.miles.drive <- evtree(s ~ Annual.miles.drive,
```

```
                                    data =
sev_gam_Annual.miles.drive,
                                    weights = tot,
                                    control = ctrl.sev)
evtree_sev_Annual.miles.drive
plot(evtree_sev_Annual.miles.drive)


sev_splits_Annual.miles.drive <- c(0, 9134, 15037, 17771,
max(train_classic$Annual.miles.drive))
AMT_claims_train_2_bin$Annual.miles.drive_bin <-
cut(AMT_claims_train_2_bin$Annual.miles.drive,
sev_splits_Annual.miles.drive, right = FALSE, include.lowest =
TRUE)
summary(AMT_claims_train_2_bin$Annual.miles.drive_bin)
AMT_claims_train_2_bin$Annual.miles.drive_bin <-
relevel(AMT_claims_train_2_bin$Annual.miles.drive_bin, ref =
"[9.13e+03,1.5e+04)")


plot_sev_bin_Annual.miles.drive <- ggplot.gam(model_gam_sev,

AMT_claims_train_2_bin$Annual.miles.drive,
                                        "s(Annual.miles.drive)",
"Annual.miles.drive",

expression(hat(f)(Annual.miles.drive)))
plot_sev_bin_Annual.miles.drive <-
plot_sev_bin_Annual.miles.drive +
  geom_vline(xintercept =

sev_splits_Annual.miles.drive[2:(length(sev_splits_Annual.miles.
drive)-1)])
plot_sev_bin_Annual.miles.drive


#### Modelo glm tras agrupaciones ####
model_glm_sev_bin<- glm(as.formula('AMT_avg  ~  Insured.age_bin
+ car.age_bin + Credit.score_bin + Annual.miles.drive_bin +
Territory'),
                    data = AMT_claims_train_2_bin, family =
Gamma(link = "log"))
```

```
summary(model_glm_sev_bin)

AIC(model_glm_sev_bin)

BIC(model_glm_sev_bin)


# Agrupar en train data #

train_classic_bin$Insured.age_bin <-
cut(train_classic_bin$Insured.age, sev_splits_Insured.age, right
= FALSE, include.lowest = TRUE)

summary(train_classic_bin$Insured.age_bin)

train_classic_bin$Insured.age_bin <-
relevel(train_classic_bin$Insured.age_bin, ref = "[34,55)")


train_classic_bin$car.age_bin <- cut(train_classic_bin$Car.age,
sev_splits_car.age, right = FALSE, include.lowest = TRUE)

summary(train_classic_bin$car.age_bin)

train_classic_bin$car.age_bin <-
relevel(train_classic_bin$car.age_bin, ref = "[6,12)")


train_classic_bin$Credit.score_bin <-
cut(train_classic_bin$Credit.score, sev_splits_Credit.score,
right = FALSE, include.lowest = TRUE)

summary(train_classic_bin$Credit.score_bin)

train_classic_bin$Credit.score_bin <-
relevel(train_classic_bin$Credit.score_bin, ref = "[777,856)")


train_classic_bin$Annual.miles.drive_bin <-
cut(train_classic_bin$Annual.miles.drive,
sev_splits_Annual.miles.drive, right = FALSE, include.lowest =
TRUE)

summary(train_classic_bin$Annual.miles.drive_bin)

train_classic_bin$Annual.miles.drive_bin <-
relevel(train_classic_bin$Annual.miles.drive_bin, ref =
"[0,9.13e+03)")


pdt_glm_sev_bin_train <- predict.glm(model_glm_sev_bin,
train_classic_bin, type = "response")


# Reagrupar en los datos test data #
```

```
test_classic_bin$Insured.age_bin <-
cut(test_classic_bin$Insured.age, sev_splits_Insured.age, right
= FALSE, include.lowest = TRUE)

summary(test_classic_bin$Insured.age_bin)

test_classic_bin$Insured.age_bin <-
relevel(test_classic_bin$Insured.age_bin, ref = "[34,55)")


test_classic_bin$car.age_bin <- cut(test_classic_bin$Car.age,
sev_splits_car.age, right = FALSE, include.lowest = TRUE)

summary(test_classic_bin$car.age_bin)

test_classic_bin$car.age_bin <-
relevel(test_classic_bin$car.age_bin, ref = "[6,12)")


test_classic_bin$Credit.score_bin <-
cut(test_classic_bin$Credit.score, sev_splits_Credit.score,
right = FALSE, include.lowest = TRUE)

summary(test_classic_bin$Credit.score_bin)

test_classic_bin$Credit.score_bin <-
relevel(test_classic_bin$Credit.score_bin, ref = "[777,856)")


test_classic_bin$Annual.miles.drive_bin <-
cut(test_classic_bin$Annual.miles.drive,
sev_splits_Annual.miles.drive, right = FALSE, include.lowest =
TRUE)

summary(test_classic_bin$Annual.miles.drive_bin)

test_classic_bin$Annual.miles.drive_bin <-
relevel(test_classic_bin$Annual.miles.drive_bin, ref =
"[0,9.13e+03)")


pdt_glm_sev_bin_test <- predict.glm(model_glm_sev_bin,
test_classic_bin, type = "response")


#### PRIMA GLM_BIN y GAM ####


test_classic_bin$predict_freq_gam <- pdt_gam_freq_test

test_classic_bin$predict_freq_glm_bin <- pdt_glm_freq_bin_test


test_classic_bin$predict_sev_gam <- pdt_gam_sev_test
```

```
test_classic_bin$predict_sev_glm_bin <- pdt_glm_sev_bin_test


test_classic_bin$gam_premium <-
test_classic_bin$predict_freq_gam*test_classic_bin$predict_sev_g
am
test_classic_bin$glm_premium <-
test_classic_bin$predict_freq_glm_bin*test_classic_bin$predict_s
ev_glm_bin


premium_table <- data.frame(Modelo = c("GLM", "GAM"),
                            Prima_cartera =
c(sum(test_classic_bin$glm_premium),
sum(test_classic_bin$gam_premium)))
premium_table


# Crear el gráfico
premium_df <- data.frame(Modelo = rep(c("GLM", "GAM"),
                                      times =
c(nrow(test_classic_bin), nrow(test_classic_bin))),
                         Premium =
c(test_classic_bin$glm_premium, test_classic_bin$gam_premium))


ggplot(premium_df, aes(x = Modelo, y = Premium)) +
  geom_violin() +
  geom_boxplot(width = 0.1, fill = fill) +
  labs(title = "Prima pura por modelo",
       x = "Modelo", y = "Prima pura")


summary(test_classic_bin$gam_premium)
summary(test_classic_bin$glm_premium)



############## TELEMÁTICA ##############


###### Análisis Estadístico variables telemáticas ######
ggplot.bar_tel <- function(DT, variable, xlab){
  ggplot(data = DT, aes(as.factor(variable))) + theme_bw() +
```

```
    geom_bar(aes(y = (..count..)/sum(..count..)), col = col,
fill = fill, alpha = 0.7) + labs(x = xlab, y = ylab)
}


#### Variables telemáticas Pay-how-you-drive (PHYD). ####


#Acelerones
datos_tel <- subset(datos, datos$Accel.06miles <
quantile(datos$Accel.06miles,0.99))


datos_Pct.Accel <- gather(datos_tel, key = "Intensidad", value =
"Accel",
                          Accel.06miles:Accel.12miles)
datos_Accel <- datos_Pct.Accel %>%
  mutate(Intensidad = case_when(Intensidad == "Accel.06miles" ~
"06",
                                Intensidad == "Accel.08miles" ~
"08",
                                Intensidad == "Accel.09miles" ~
"09",
                                Intensidad == "Accel.11miles" ~
"11",
                                Intensidad == "Accel.12miles" ~
"12",
                                Intensidad == "Accel.14miles" ~
"14"))
datos_Accel$Intensidad <- factor(datos_Accel$Intensidad, levels
= c("06", "08", "09", "11", "12", "14"))
ggplot(datos_Accel, aes(x = Intensidad, y = Accel)) +
  geom_boxplot(col=col) +
  labs(x = "Nivel de intensidad", y = "Número de acelerones") +
  ggtitle("Boxplots de numero de acelerones según su
intensidad")


summary(datos_tel$Accel.06miles)
summary(datos_tel$Accel.08miles)
summary(datos_tel$Accel.09miles)
summary(datos_tel$Accel.11miles)
```

```
summary(datos_tel$Accel.12miles)
summary(datos_tel$Accel.14miles)


#Breaks
datos_tel <- subset(datos_tel, datos_tel$Brake.06miles <
quantile(datos_tel$Brake.06miles,0.999))


datos_Pct.Brake <- gather(datos_tel, key = "Intensidad", value =
"Brake",
                          Brake.06miles:Brake.12miles)
datos_Brake <- datos_Pct.Brake %>%
  mutate(Intensidad = case_when(Intensidad == "Brake.06miles" ~
"06",
                                Intensidad == "Brake.08miles" ~
"08",
                                Intensidad == "Brake.09miles" ~
"09",
                                Intensidad == "Brake.11miles" ~
"11",
                                Intensidad == "Brake.12miles" ~
"12",
                                Intensidad == "Brake.14miles" ~
"14"))
datos_Brake$Intensidad <- factor(datos_Brake$Intensidad, levels
= c("06", "08", "09", "11", "12", "14"))
ggplot(datos_Brake, aes(x = Intensidad, y = Brake)) +
  geom_boxplot(col=col) +
  labs(x = "Nivel de intensidad", y = "Número de frenazos") +
  ggtitle("Boxplots de numero de frenazos según su intensidad")


summary(datos_tel$Brake.06miles)


graficos_Left.turn.intensityxx <- grid.arrange(
  plot_Brake.08miles <- ggplot.bar_tel(datos,
datos$Left.turn.intensity08, "Numero de giros a la izquierda con
intensidad 8 cada 1000 millas"),
```

```
  plot_Brake.09miles <- ggplot.bar_tel(datos,
datos$Left.turn.intensity09, "Numero de giros a la izquierda con
intensidad 9 cada 1000 millas"),

  plot_Brake.10miles <- ggplot.bar_tel(datos,
datos$Left.turn.intensity10, "Numero de giros a la izquierda con
intensidad 10 cada 1000 millas"),

  plot_Brake.11miles <- ggplot.bar_tel(datos,
datos$Left.turn.intensity11, "Numero de giros a la izquierda con
intensidad 11 cada 1000 millas"),

  plot_Brake.12miles <- ggplot.bar_tel(datos,
datos$Left.turn.intensity12, "Numero de giros a la izquierda con
intensidad 12 cada 1000 millas"),

  ncol = 2

)


#Right turns

datos_tel <- subset(datos_tel, datos_tel$Right.turn.intensity08
< quantile(datos_tel$Right.turn.intensity08,0.99))


datos_Pct.Right.turn <- gather(datos_tel, key = "Intensidad",
value = "Right.turn",


Right.turn.intensity08:Right.turn.intensity12)

datos_Right.turn <- datos_Pct.Right.turn %>%

  mutate(Intensidad = case_when(Intensidad ==
"Right.turn.intensity08" ~ "08",

                                Intensidad ==
"Right.turn.intensity09" ~ "09",

                                Intensidad ==
"Right.turn.intensity10" ~ "10",

                                Intensidad ==
"Right.turn.intensity11" ~ "11",

                                Intensidad ==
"Right.turn.intensity12" ~ "12"))

datos_Right.turn$Intensidad <-
factor(datos_Right.turn$Intensidad, levels = c("08", "09", "10",
"11", "12"))

ggplot(datos_Right.turn, aes(x = Intensidad, y = Right.turn)) +

  geom_boxplot(col=col) +

  labs(x = "Nivel de intensidad", y = "Giros a derechas") +
```

```
  ggtitle("Boxplots de giros a derechas según su intensidad")



##Left turns
datos_tel <- subset(datos_tel, datos_tel$Left.turn.intensity08 <
quantile(datos_tel$Left.turn.intensity08,0.99))


datos_Pct.Left.turn <- gather(datos_tel, key = "Intensidad",
value = "Left.turn",

Left.turn.intensity08:Left.turn.intensity12)
datos_Left.turn <- datos_Pct.Left.turn %>%
  mutate(Intensidad = case_when(Intensidad ==
"Left.turn.intensity08" ~ "08",
                                Intensidad ==
"Left.turn.intensity09" ~ "09",
                                Intensidad ==
"Left.turn.intensity10" ~ "10",
                                Intensidad ==
"Left.turn.intensity11" ~ "11",
                                Intensidad ==
"Left.turn.intensity12" ~ "12"))
datos_Left.turn$Intensidad <- factor(datos_Left.turn$Intensidad,
levels = c("08", "09", "10", "11", "12"))


ggplot(datos_Left.turn, aes(x = Intensidad, y = Left.turn)) +
  geom_boxplot(col=col) +
  labs(x = "Nivel de intensidad", y = "Giros a izquierdas") +
  ggtitle("Boxplots de giros a izquierdas según su intensidad")



#### Variables telemáticas Pay-as-you-drive (PAYD). ####
graficos_telematics_1 <- grid.arrange(
  plot_Avgdays.week <-ggplot.hist(datos_tel,
datos_tel$Avgdays.week, "Dias que conduce por semana",2),
  plot_Annual.pct.driven <- ggplot.bar_tel(datos_tel,
datos_tel$Annual.pct.driven, "Tiempo al año en la carretera"),
```

```
  plot_Total.miles.driven <- ggplot.hist(datos_tel,
datos_tel$Total.miles.driven, "Distancia recorrida en
Millas",2),
  ncol = 3
)


## Dias que conduce a la semana
datos_tel_Pct.drive <- gather(datos_tel, key = "Dia", value =
"Pct.drive",
                              Pct.drive.mon:Pct.drive.sun)
datos_tel_Pct.drive <- datos_tel_Pct.drive %>%
  mutate(Dia = case_when(Dia == "Pct.drive.mon" ~ "lunes",
                         Dia == "Pct.drive.tue" ~ "martes",
                         Dia == "Pct.drive.wed" ~ "miércoles",
                         Dia == "Pct.drive.thr" ~ "jueves",
                         Dia == "Pct.drive.fri" ~ "viernes",
                         Dia == "Pct.drive.sat" ~ "sábado",
                         Dia == "Pct.drive.sun" ~ "domingo"))
datos_tel_Pct.drive$Dia <- factor(datos_tel_Pct.drive$Dia,
levels = c("lunes", "martes", "miércoles", "jueves", "viernes",
"sábado", "domingo"))
ggplot(datos_tel_Pct.drive, aes(x = Dia, y = Pct.drive)) +
  geom_boxplot(col=col) +
  labs(x = "Día de la semana", y = "Pct.drive") +
  ggtitle("Boxplots de Pct.drive por día de la semana")


## Tiempo que pasa conduciendo entre semana o los findes de
semana
datos_tel_drive_hrs_long <- datos_tel %>%
  gather(key = "variable", value = "value",
Pct.drive.2hrs:Pct.drive.4hrs)
ggplot(datos_tel_drive_hrs_long, aes(x = variable, y = value)) +
  geom_boxplot(col = col) +
  labs(x = "Tiempo conduciendo", y = "Porcentaje") +
  scale_x_discrete(labels = c("2 horas",
                              "3 horas",
```

```
                                "4 horas")) +
  ggtitle("Horas por trayecto")


graficos_Pct.drive.xhrs <- grid.arrange(
  plot_Pct.drive.2hrs <- ggplot.hist(datos_tel,
datos_tel$Pct.drive.2hrs*100, "% de tiempo que conduce 2h",2),
  plot_Pct.drive.3hrs <- ggplot.hist(datos_tel,
datos_tel$Pct.drive.3hrs*100, "% de tiempo que conduce 3h",2),
  plot_Pct.drive.4hrs <- ggplot.hist(datos_tel,
datos_tel$Pct.drive.4hrs*100, "% de tiempo que conduce 4h",2),
  ncol = 3
)


## Tiempo que pasa conduciendo entre semana o los findes de
semana
datos_tel_drive_week_long <- datos_tel %>%
  gather(key = "variable", value = "value",
Pct.drive.wkday:Pct.drive.wkend)
ggplot(datos_tel_drive_week_long, aes(x = variable, y = value))
+
  geom_boxplot(col = col) +
  labs(x = "Tiempo conduciendo", y = "Porcentaje") +
  scale_x_discrete(labels = c("Entre semana",
                              "Findes de semana")) +
  ggtitle("Tiempo conduciendo entre semana o findes de semana")


## Tiempo que pasa conduciendo por la mañana y por la noche
datos_tel_drive_rush_long <- datos_tel %>%
  gather(key = "variable", value = "value",
Pct.drive.rush.am:Pct.drive.rush.pm)
ggplot(datos_tel_drive_rush_long, aes(x = variable, y = value))
+
  geom_boxplot(col = col) +
  labs(x = "Tiempo conduciendo", y = "Porcentaje") +
  scale_x_discrete(labels = c("am",
                              "pm")) +
  ggtitle("Tiempo conduciendo por la mañana o por la tarde")
```

```
########## H2O ##########
options(timeout = 600)
h2o.init(
  ip = "localhost",
  nthreads = -1,
  max_mem_size = "6g"
)
# Eliminan los datos del cluster por si ya estaba iniciado.
h2o.removeAll()
h2o.no_progress()


datos_tel$log_Exposure <- log(datos_tel$Exposure)
datos_tel <- datos_tel[, -
which(names(datos_tel)=="Years.noclaims")]


datos_h2o <- as.h2o(x = datos_tel, destination_frame =
"datos_h2o")
particiones     <- h2o.splitFrame(data = datos_h2o, ratios =
c(0.7), seed = 123)
datos_train_h2o <- h2o.assign(data = particiones[[1]], key =
"datos_train_H2O")
datos_test_h2o  <- h2o.assign(data = particiones[[2]], key =
"datos_test_H2O")


datos_train_h2o_df <- as.data.frame(datos_train_h2o)
datos_test_h2o_df <- as.data.frame(datos_test_h2o)


# Definir los posibles valores de los parámetros de ajuste para
la búsqueda/tunning en cuadrícula
hyper_params <- list(
  ntrees = seq(from = 50, to = 300, by = 50),
  max_depth = seq(from = 1, to = 5, by = 1),
  learn_rate = c(0.001, 0.01),
  sample_rate = seq(from = 0.6, to = 0.9, by = 0.1)
```

```
)

# Definir criterios de parada
search_criteria <- list(
  strategy = "RandomDiscrete",
  max_models = 10,
  seed = 1234,
  stopping_rounds = 5,
  stopping_tolerance = 0.001,
  stopping_metric = "rmse"
)

##### GBM Clasic #####

#### Frequency ####
predictores_cl <- c("Insured.age", "Insured.sex", "Car.age",
"Marital",
                    "Car.use", "Region", "Credit.score",
"Annual.miles.drive", "Territory")

# Realizar la búsqueda de cuadrícula aleatoria con validación
cruzada en 5 folds
grid_gbm_freq_cl <- h2o.grid(
  algorithm = "gbm",
  distribution = "poisson",
  grid_id = "grid_gbm_freq_cl",
  x = predictores_cl,
  y = "NB_Claim",
  offset_column = "log_Exposure",
  training_frame = datos_train_h2o,
  min_rows = nrow.H2OFrame(datos_train_h2o)*0.05,
  nfolds = 5,
  hyper_params = hyper_params,
  search_criteria = search_criteria
)
```

```
resultados_grid_gbm_freq_cl <- h2o.getGrid(
  grid_id = "grid_gbm_freq_cl",
  sort_by = "rmse",
  decreasing = FALSE
)
print(resultados_grid_gbm_freq_cl)
unlist(grid_gbm_freq_cl@model_ids)

best_gbm_model_freq_cl <-
h2o.getModel(grid_gbm_freq_cl@model_ids[[1]])
varimp_freq_cl <- h2o.varimp(best_gbm_model_freq_cl)
summary(best_gbm_model_freq_cl, plotit = FALSE)

varimp_freq_cl <- as.data.frame(varimp_freq_cl)
ggplot(data = varimp_freq_cl,
       aes(x = reorder(variable, scaled_importance), y =
scaled_importance)) +
  geom_col(fill = fill, color = col) +
  coord_flip() +
  labs(title = "Importancia de los predictores en el modelo GBM
de frecuencia clásico",
       x = "Predictor",
       y = "Importancia relativa") +
  theme_bw()

performance_gbm_freq_cl <- h2o.performance(model =
best_gbm_model_freq_cl, train = TRUE)
predict_gbm_freq_cl_train <- h2o.predict(object =
best_gbm_model_freq_cl, newdata = datos_train_h2o)
summary(predict_gbm_freq_cl_train)

performance_gbm_freq_cl_test <- h2o.performance(model =
best_gbm_model_freq_cl, newdat = datos_test_h2o)
predict_gbm_freq_cl_test <- h2o.predict(object =
best_gbm_model_freq_cl, newdata = datos_test_h2o)
```

```
#### Severidad ####
datos_train_h2o_sev <- as.data.frame(datos_train_h2o)
datos_train_h2o_sev <- subset(datos_train_h2o_sev, AMT_Claim >
0)
datos_train_h2o_sev <- as.h2o(datos_train_h2o_sev,
destination_frame = "datos_train_h2o_sev")


# Realizar la búsqueda de cuadrícula aleatoria con validación
cruzada en 5 folds
grid_gbm_sev_cl <- h2o.grid(
  algorithm = "gbm",
  distribution= "gamma",
  grid_id = "grid_gbm_sev_cl",
  x = predictores_cl,
  y = "AMT_avg",
  min_rows = nrow.H2OFrame(datos_train_h2o_sev)*0.01,
  training_frame = datos_train_h2o_sev,
  nfolds = 5,
  hyper_params = hyper_params,
  search_criteria = search_criteria
)


resultados_grid_gbm_sev_cl <- h2o.getGrid(
  grid_id = "grid_gbm_sev_cl",
  sort_by = "rmse",
  decreasing = FALSE
)
print(resultados_grid_gbm_sev_cl)
unlist(grid_gbm_sev_cl@model_ids)


best_gbm_model_sev_cl <-
h2o.getModel(grid_gbm_sev_cl@model_ids[[1]])
varimp_sev_cl <- h2o.varimp(best_gbm_model_sev_cl)
summary(best_gbm_model_sev_cl, plotit = FALSE)
```

```
varimp_sev_cl <- as.data.frame(varimp_sev_cl)

ggplot(data = varimp_sev_cl,

       aes(x = reorder(variable, scaled_importance), y =
scaled_importance)) +

  geom_col(fill = fill, color = col) +

  coord_flip() +

  labs(title = "Importancia de los predictores en el modelo GBM
de Severidad clásico",

       x = "Predictor",

       y = "Importancia relativa") +

  theme_bw()


performance_gbm_sev_cl <- h2o.performance(model =
best_gbm_model_sev_cl, train = TRUE)

predict_gbm_sev_cl_train <- h2o.predict(object =
best_gbm_model_sev_cl, newdata = datos_train_h2o)

summary(predict_gbm_sev_cl_train)


performance_gbm_sev_cl_test <- h2o.performance(model =
best_gbm_model_sev_cl, newdat = datos_test_h2o)

predict_gbm_sev_cl_test <- h2o.predict(object =
best_gbm_model_sev_cl, newdata = datos_test_h2o)


#### Prima GBM cl ####

#test

predict_gbm_freq_cl_test_df <-
as.data.frame(predict_gbm_freq_cl_test)

predict_gbm_sev_cl_test_df <-
as.data.frame(predict_gbm_sev_cl_test)


datos_test_h2o_df$gbm_freq_cl <-
predict_gbm_freq_cl_test_df$predict

datos_test_h2o_df$gbm_sev_cl <-
predict_gbm_sev_cl_test_df$predict

datos_test_h2o_df$prima_gbm_cl <- datos_test_h2o_df$gbm_freq_cl
* datos_test_h2o_df$gbm_sev_cl

summary(datos_test_h2o_df$prima_gbm_cl)

sum(datos_test_h2o_df$prima_gbm_cl)
```

```
# Calcular la deviance en la muestra out-of-sample
deviance_gbm_freq <- -2 * sum(dpois(datos_test_h2o_df$NB_Claim,
lambda = predict_gbm_freq_cl_test_df$predict, log = TRUE))

###### Modelo GLM regularizado ######
#### Frecuencia ####

# Para este modelo se emplean todos los predictores.
predictores_tel <- setdiff(h2o.colnames(datos_h2o),
c("AMT_Claim", "NB_Claim","Exposure","log_Exposure","AMT_avg"))

# Valores de alpha que se van a comparar.
hiperparametros <- list(alpha = c(0, 0.1, 0.5, 0.95, 1))

grid_glm_tel <- h2o.grid(
  # Algoritmo y parámetros
  algorithm      = "glm",
  family = "negativebinomial",
  link   = "log",
  offset_column = "log_Exposure",
  # Variable respuesta y predictores
  y = "NB_Claim",
  x = predictores_tel,
  # Datos de entrenamiento
  training_frame = datos_train_h2o,
  # Preprocesado
  standardize    = TRUE,
  missing_values_handling = "Skip",
  ignore_const_cols = TRUE,
  # Hiperparámetros
  hyper_params    = hiperparametros,
  # Tipo de búsqueda
  search_criteria = list(strategy = "RandomDiscrete",
                         max_models = 5,
```

```
                          seed = 1234),
  lambda_search   = TRUE,
  solver          = "AUTO",
  # Estrategia de validación para seleccionar el mejor modelo
  seed            = 123,
  nfolds          = 10,
  keep_cross_validation_predictions = FALSE,
  grid_id         = "grid_glm_tel"
)

resultados_grid_tel <- h2o.getGrid(
  grid_id = "grid_glm_tel",
  sort_by = "mean_residual_deviance",
  decreasing = FALSE
)
print(resultados_grid_tel)

id_modelos_tel <- unlist(resultados_grid_tel@model_ids)
rmse_xvalidacion_tel <- vector(mode = "list", length =
length(id_modelos_tel))

# Se recorre cada modelo almacenado en el grid y se extraen la
métrica (rmse)
for (i in seq_along(id_modelos_tel)) {
  modelo <- h2o.getModel(resultados_grid_tel@model_ids[[i]])
  metricas_xvalidacion_modelo <-
modelo@model$cross_validation_metrics_summary
  names(rmse_xvalidacion_tel)[i]   <-
modelo@model$model_summary$regularization
  rmse_xvalidacion_tel[[i]] <-
as.numeric(metricas_xvalidacion_modelo["rmse", -c(1,2)])
}

# Se eliminan los espacios en blanco del nombre de los modelos.
names(rmse_xvalidacion_tel) <- str_remove_all(string =
names(rmse_xvalidacion_tel),
```

```
                                                  pattern = "[ )=]")
names(rmse_xvalidacion_tel) <- str_replace_all(string =
names(rmse_xvalidacion_tel),
                                                   pattern = "[(,]",
                                                   replacement =
"_")
# Se convierte la lista en dataframe.
rmse_xvalidacion_tel_df <- as.data.frame(rmse_xvalidacion_tel)
%>%
  mutate(resample = row_number()) %>%
  gather(key = "modelo", value = "rmse", -resample) %>%
  mutate(modelo = str_replace_all(string = modelo,
                                  pattern = "_" ,
                                  replacement = " \n "))
# Gráfico
ggplot(data = rmse_xvalidacion_tel_df, aes(x = modelo, y = rmse,
color = modelo)) +
  geom_boxplot(alpha = 0.6, outlier.shape = NA) +
  geom_jitter(width = 0.1, alpha = 0.6) +
  stat_summary(fun.y = "mean", colour = "red", size = 2, geom =
"point") +
  theme_bw() +
  labs(title = "Precisión obtenida por 10-CV") +
  coord_flip() +
  theme(legend.position = "none")


#Tenemos los parámetros óptimos: Elastic Net (alpha = 0.1,
lambda = 0.002206)
modelo_glm_tel <- h2o.glm(
  y = "NB_Claim",
  x = predictores_tel,
  training_frame = datos_train_h2o,
  family = "negativebinomial",
  link   = "log",
  offset_column = "log_Exposure",
  standardize  = TRUE,
```

```
  balance_classes   = FALSE,
  ignore_const_cols = TRUE,
  missing_values_handling = "Skip",
  lambda = 0.002206,
  alpha  = 0.1,
  # Validación cruzada de 5 folds para estimar el error
  seed = 123,
  nfolds = 5,
  keep_cross_validation_predictions = FALSE,
  model_id = "modelo_glm_tel"
)
modelo_glm_tel


as.data.frame(modelo_glm_tel@model$coefficients_table) %>%
head()
# Predictores incluidos.
names(modelo_glm_tel@model$coefficients[modelo_glm_tel@model$coe
fficients_table != 0])
coeficientes_tel_freq <-
as.data.frame(modelo_glm_tel@model$coefficients_table)


# Se excluye el intercept.
coeficientes_tel_freq <- coeficientes_tel_freq %>% filter(names
!= "Intercept")
# Se calcula el valor absoluto.
coeficientes_tel_freq <- coeficientes_tel_freq %>%
  mutate(abs_stand_coef = abs(standardized_coefficients))
# Se añade una variable con el signo del coeficiente.
coeficientes_tel_freq <- coeficientes_tel_freq %>%
  mutate(signo = if_else(standardized_coefficients > 0,
                         "Positivo",
                         "Negativo"))


#Gráfico de los predictores
ggplot(data = coeficientes_tel_freq,
       aes(x = reorder(names, abs_stand_coef),
```

```
          y = abs_stand_coef,
          fill = signo)) +
  geom_col() +
  coord_flip() +
  labs(title = "Importancia de predictores Elastic Net (alpha =
0.1, lambda = 0.002206)",
       x = "Predictor",
       y = "Valor absoluto coeficiente estandarizado") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_manual(values = c("#FF6961", "#116E8A"))

#### Severidad ####
grid_glm_tel_sev <- h2o.grid(
  # Algoritmo y parámetros
  algorithm = "glm",
  family = "gamma",
  link   = "log",
  # Variable respuesta y predictores
  y = "AMT_avg",
  x = predictores_tel,
  # Datos de entrenamiento
  training_frame = datos_train_h2o_sev,
  # Preprocesado
  standardize    = TRUE,
  missing_values_handling = "Skip",
  ignore_const_cols = TRUE,
  # Hiperparámetros
  hyper_params    = hiperparametros,
  lambda_search   = TRUE,
  # Tipo de búsqueda
  search_criteria = list(strategy = "RandomDiscrete",
                         max_models = 5,
                         seed = 1234),
  solver          = "AUTO",
```

```
  # Estrategia de validación para seleccionar el mejor modelo
  seed            = 123,
  nfolds          = 10,
  keep_cross_validation_predictions = FALSE,
  grid_id = "grid_glm_tel_sev"
)

resultados_grid_tel_sev <- h2o.getGrid(
  grid_id = "grid_glm_tel_sev",
  sort_by = "rmse",
  decreasing = TRUE
)
print(resultados_grid_tel_sev)

id_modelos_tel_sev <- unlist(resultados_grid_tel_sev@model_ids)
rmse_xvalidacion_tel_sev <- vector(mode = "list", length =
length(id_modelos_tel_sev))

# Se recorre cada modelo almacenado en el grid y se extraen la
métrica (rmse) obtenida en cada partición.
for (i in seq_along(id_modelos_tel_sev)) {
  modelo <- h2o.getModel(resultados_grid_tel_sev@model_ids[[i]])
  metricas_xvalidacion_modelo <-
modelo@model$cross_validation_metrics_summary
  names(rmse_xvalidacion_tel_sev)[i]   <-
modelo@model$model_summary$regularization
  rmse_xvalidacion_tel_sev[[i]] <-
as.numeric(metricas_xvalidacion_modelo["rmse", -c(1,2)])
}

# Se eliminan los espacios en blanco del nombre de los modelos.
names(rmse_xvalidacion_tel_sev) <- str_remove_all(string =
names(rmse_xvalidacion_tel_sev),
                                                  pattern = "[
)=]")
names(rmse_xvalidacion_tel_sev) <- str_replace_all(string =
names(rmse_xvalidacion_tel_sev),
```

```
                                                    pattern =
"[(,]",
                                                    replacement =
"_")
# Se convierte la lista en dataframe.
rmse_xvalidacion_tel_sev_df <-
as.data.frame(rmse_xvalidacion_tel_sev) %>%
  mutate(resample = row_number()) %>%
  gather(key = "modelo", value = "rmse", -resample) %>%
  mutate(modelo = str_replace_all(string = modelo,
                                  pattern = "_" ,
                                  replacement = " \n "))
# Gráfico
ggplot(data = rmse_xvalidacion_tel_sev_df, aes(x = modelo, y =
rmse, color = modelo)) +
  geom_boxplot(alpha = 0.6, outlier.shape = NA) +
  geom_jitter(width = 0.1, alpha = 0.6) +
  stat_summary(fun.y = "mean", colour = "red", size = 2, geom =
"point") +
  theme_bw() +
  labs(title = "Precisión obtenida por 10-CV") +
  coord_flip() +
  theme(legend.position = "none")


#Tenemos los parámetros óptimos: Elastic Net (alpha = 0.1,
lambda = 0.08138)
modelo_glm_tel_sev <- h2o.glm(
  y = "AMT_avg",
  x = predictores_tel,
  training_frame = datos_train_h2o_sev,
  family = "gamma",
  link   = "log",
  standardize  = TRUE,
  balance_classes   = FALSE,
  ignore_const_cols = TRUE,
  missing_values_handling = "Skip",
```

```
  lambda = 0.08138,
  alpha  = 0.1,
  # Validación cruzada de 5 folds para estimar el error del
modelo.
  seed = 123,
  nfolds = 5,
  # Reparto estratificado de las observaciones en la creación de
las particiones.
  #fold_assignment = "Stratified",
  keep_cross_validation_predictions = FALSE,
  model_id = "modelo_glm_tel_sev"
)
print(modelo_glm_tel_sev)


as.data.frame(modelo_glm_tel_sev@model$coefficients_table) %>%
head()
# Predictores incluidos.
names(modelo_glm_tel_sev@model$coefficients[modelo_glm_tel_sev@m
odel$coefficients_table != 0])
coeficientes_tel_sev <-
as.data.frame(modelo_glm_tel_sev@model$coefficients_table)
# Se excluye el intercept.
coeficientes_tel_sev <- coeficientes_tel_sev %>% filter(names !=
"Intercept")
# Se calcula el valor absoluto.
coeficientes_tel_sev <- coeficientes_tel_sev %>%
  mutate(abs_stand_coef = abs(standardized_coefficients))
# Se añade una variable con el signo del coeficiente.
coeficientes_tel_sev <- coeficientes_tel_sev %>%
  mutate(signo = if_else(standardized_coefficients > 0,
                         "Positivo",
                         "Negativo"))
# Gráfico con los predictores
ggplot(data = coeficientes_tel_sev,
       aes(x = reorder(names, abs_stand_coef),
           y = abs_stand_coef,
```

```
          fill = signo)) +
  geom_col() +
  coord_flip() +
  labs(title = "Importancia de predictores Elastic Net (alpha =
0.1, lambda = 0.08138)",
       x = "Predictor",
       y = "Valor absoluto coeficiente estandarizado") +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_fill_manual(values = c("#FF6961", "#116E8A"))


###### Modelo GBM tel ######


#### Frecuencia ####
predictores_gbm_freq <- subset(coeficientes_tel_freq,
abs(standardized_coefficients) != 0)$names
predictores_gbm_freq <-
predictores_gbm_freq[!predictores_gbm_freq %in%
c("Marital.Married", "Region.Rural")]
predictores_gbm_freq <- gsub("Marital.Single", "Marital",
predictores_gbm_freq)
predictores_gbm_freq <- gsub("Region.Urban", "Region",
predictores_gbm_freq)


# Realizar la búsqueda de cuadrícula aleatoria con validación
cruzada en 5 folds
grid_gbm_freq_tel <- h2o.grid(
  algorithm = "gbm",
  distribution = "poisson",
  grid_id = "grid_gbm_freq_tel",
  x = predictores_gbm_freq,
  y = "NB_Claim",
  offset_column = "log_Exposure",
  training_frame = datos_train_h2o,
  min_rows = nrow.H2OFrame(datos_train_h2o)*0.05,
  nfolds = 5,
```

```
  hyper_params = hyper_params,
  search_criteria = search_criteria
)

resultados_grid_gbm_freq_tel <- h2o.getGrid(
  grid_id = "grid_gbm_freq_tel",
  sort_by = "rmse",
  decreasing = FALSE
)
print(resultados_grid_gbm_freq_tel)
unlist(grid_gbm_freq_tel@model_ids)

best_gbm_model_freq_tel <-
h2o.getModel(grid_gbm_freq_tel@model_ids[[1]])
varimp_freq_tel <- h2o.varimp(best_gbm_model_freq_tel)
summary(best_gbm_model_freq_tel, plotit = FALSE)
varimp_freq_tel <- as.data.frame(varimp_freq_tel)

ggplot(data = varimp_freq_tel,
       aes(x = reorder(variable, scaled_importance), y =
scaled_importance)) +
  geom_col(fill = fill, color = col) +
  coord_flip() +
  labs(title = "Importancia de los predictores en el modelo GBM
de frecuencia",
       x = "Predictor",
       y = "Importancia relativa") +
  theme_bw()

performance_gbm_freq_tel<- h2o.performance(model =
best_gbm_model_freq_tel, train = TRUE)
predict_gbm_freq_tel_train <- h2o.predict(object =
best_gbm_model_freq_tel, newdata = datos_train_h2o)
summary(predict_gbm_freq_tel_train)
```

```
performance_gbm_freq_tel_test <- h2o.performance(model =
best_gbm_model_freq_tel, newdat = datos_test_h2o)

predict_gbm_freq_tel_test <- h2o.predict(object =
best_gbm_model_freq_tel, newdata = datos_test_h2o)

summary(predict_gbm_freq_tel_test)


# Calcular el PDP de las variables

pdp_plot <- function(model, data, x_var, y_var) {

  pdp_data <- h2o.partialPlot(object = model, data = data, cols
= x_var)

  ggplot(pdp_data, aes_string(x = x_var, y = y_var)) +

    geom_line(color = KULbg) +

    geom_ribbon(aes(ymin = mean_response - stddev_response, ymax
= mean_response + stddev_response), fill = KULbg, alpha = 0.3) +

    labs(title = paste("Partial Dependence Plot -", x_var))

}


pdp_Total.miles.driven_plot_freq_tel <-
pdp_plot(best_gbm_model_freq_tel, datos_train_h2o,
"Total.miles.driven", "mean_response")

pdp_Annual.pct.driven_freq_plot_freq_tel <-
pdp_plot(best_gbm_model_freq_tel, datos_train_h2o,
"Annual.pct.driven", "mean_response")

pdp_Brake.09miles_plot_freq_tel <-
pdp_plot(best_gbm_model_freq_tel, datos_train_h2o,
"Brake.09miles", "mean_response")

pdp_Credit.score_plot_freq_tel <-
pdp_plot(best_gbm_model_freq_tel, datos_train_h2o,
"Credit.score", "mean_response")


combined_pdp_freq_tel <- grid.arrange(

  pdp_Total.miles.driven_plot_freq_tel,

  pdp_Annual.pct.driven_freq_plot_freq_tel,

  pdp_Brake.09miles_plot_freq_tel,

  pdp_Credit.score_plot_freq_tel,

  nrow = 2

)
```

```
#### Severidad ####
predictores_gbm_sev <- subset(coeficientes_tel_sev,
abs(standardized_coefficients) != 0)$names
predictores_gbm_sev <- predictores_gbm_sev[!predictores_gbm_sev
%in% c("Car.use.Commute")]
predictores_gbm_sev <- gsub("Car.use.Private", "Car.use",
predictores_gbm_sev)

grid_gbm_sev_tel <- h2o.grid(
  algorithm = "gbm",
  distribution= "gamma",
  grid_id = "grid_gbm_sev_tel",
  x = predictores_gbm_sev,
  y = "AMT_avg",
  training_frame = datos_train_h2o_sev,
  min_rows = nrow.H2OFrame(datos_train_h2o_sev)*0.01,
  nfolds = 5,
  hyper_params = hyper_params,
  search_criteria = search_criteria
)

resultados_grid_gbm_sev_tel <- h2o.getGrid(
  grid_id = "grid_gbm_sev_tel",
  sort_by = "rmse",
  decreasing = FALSE
)
print(resultados_grid_gbm_sev_tel)
unlist(grid_gbm_sev_tel@model_ids)

best_gbm_model_sev_tel <-
h2o.getModel(grid_gbm_sev_tel@model_ids[[1]])
varimp_sev_tel <- h2o.varimp(best_gbm_model_sev_tel)
print(varimp_sev_tel)
varimp_sev_tel <- as.data.frame(varimp_sev_tel)

ggplot(data = varimp_sev_tel,
```

```
       aes(x = reorder(variable, scaled_importance), y =
scaled_importance)) +
  geom_col(fill = fill, color = col) +
  coord_flip() +
  labs(title = "Importancia de los predictores en el modelo GBM
de Severidad",
       x = "Predictor",
       y = "Importancia relativa") +
  theme_bw()


performance_gbm_sev_tel_train <- h2o.performance(model =
best_gbm_model_sev_tel, train = TRUE)
predict_gbm_sev_tel_train <- h2o.predict(object =
best_gbm_model_sev_tel, newdata = datos_train_h2o)
summary(predict_gbm_sev_tel_train)


performance_gbm_sev_tel_test <- h2o.performance(model =
best_gbm_model_sev_tel, newdat = datos_test_h2o)
predict_gbm_sev_tel_test <- h2o.predict(object =
best_gbm_model_sev_tel, newdata = datos_test_h2o)
summary(predict_gbm_sev_tel_test)


pdp_Credit.score_plot_sev_tel <-
pdp_plot(best_gbm_model_sev_tel, datos_train_h2o_sev,
"Credit.score", "mean_response")
pdp_Annual.pct.driven_plot_sev_tel <-
pdp_plot(best_gbm_model_sev_tel, datos_train_h2o_sev,
"Annual.pct.driven", "mean_response")
pdp_Brake.06miles_plot_sev_tel <-
pdp_plot(best_gbm_model_sev_tel, datos_train_h2o_sev,
"Brake.06miles", "mean_response")
pdp_Insured.age_plot_sev_tel <- pdp_plot(best_gbm_model_sev_tel,
datos_train_h2o_sev, "Insured.age", "mean_response")


combined_pdp_sev_tel <- grid.arrange(
  pdp_Credit.score_plot_sev_tel,
  pdp_Annual.pct.driven_plot_sev_tel,
  pdp_Brake.06miles_plot_sev_tel,
```

```
  pdp_Insured.age_plot_sev_tel,
  nrow = 2
)

#### Prima GBM Tel ####
#test
predict_gbm_freq_tel_test_df <-
as.data.frame(predict_gbm_freq_tel_test)
predict_gbm_sev_tel_test_df <-
as.data.frame(predict_gbm_sev_tel_test)

datos_test_h2o_df$gbm_freq_tel <-
predict_gbm_freq_tel_test_df$predict
datos_test_h2o_df$gbm_sev_tel <-
predict_gbm_sev_tel_test_df$predict
datos_test_h2o_df$prima_gbm_tel <-
datos_test_h2o_df$gbm_freq_tel * datos_test_h2o_df$gbm_sev_tel
summary(datos_test_h2o_df$prima_gbm_tel)
sum(datos_test_h2o_df$prima_gbm_tel)

# Cálculo de deviance en gbm tel
deviance_gbm_freq_tel <- -2 *
sum(dpois(datos_test_h2o_df$NB_Claim, lambda =
predict_gbm_freq_tel_test_df$predict, log = TRUE))

#Dar valores de riesgo a cada asegurado en función de su función
de frecuencia
# Risk Score - classic
#test
v_predict_gbm_freq_cl_test <-
as.vector(predict_gbm_freq_cl_test)
ecdf_gbm_cl_test <- ecdf(v_predict_gbm_freq_cl_test)
risk_score_gbm_cl_test <-
ecdf_gbm_cl_test(v_predict_gbm_freq_cl_test)
summary(risk_score_gbm_cl_test)

datos_test_h2o_df$risk_score_gbm_cl <- risk_score_gbm_cl_test
```

```
# Risk Score - telematics

#test

v_predict_gbm_freq_tel_test <-
as.vector(predict_gbm_freq_tel_test)

ecdf_gbm_tel <- ecdf(v_predict_gbm_freq_tel_test)

risk_score_gbm_tel_test <-
ecdf_gbm_tel(v_predict_gbm_freq_tel_test)

summary(risk_score_gbm_tel_test)


datos_test_h2o_df$risk_score_gbm_tel <- risk_score_gbm_tel_test


#Gráfico de las diferencias relativas

puntos_corte <- c(0, 0.25, 0.5, 0.75, 1)

datos_test_h2o_df$grupo_riesgo <-
cut(datos_test_h2o_df$risk_score_gbm_tel, breaks = puntos_corte,
labels = c("Muy bajo", "Bajo", "Alto", "Muy alto"),
include.lowest = TRUE, right = FALSE)


datos_test_h2o_df$diferencia_primas <-
(datos_test_h2o_df$prima_gbm_tel -
datos_test_h2o_df$prima_gbm_cl) / datos_test_h2o_df$prima_gbm_cl

comparativa_grupos_riesgo <- aggregate(diferencia_primas ~
grupo_riesgo, datos_test_h2o_df, mean)


ggplot(datos_test_h2o_df, aes(x = grupo_riesgo, y =
diferencia_primas, fill = grupo_riesgo)) +

  geom_boxplot(color = "black", alpha = 0.7) +

  geom_text(data = comparativa_grupos_riesgo, aes(x =
grupo_riesgo, y = diferencia_primas, label =
round(diferencia_primas, 2)),

            color = "black", size = 4, vjust = -19) +

  scale_fill_manual(values = c("#7FFF00", "#98FB98", "#FF6961",
"red"), guide = "none") +

  labs(x = "Grupo de Riesgo", y = "Diferencia Relativa de
Primas") +

  ggtitle("Comparativa de Diferencia Relativa de Primas por
Grupo de Riesgo") +
```

```
  theme_minimal()


#### Double lift Chart de GBM ####
datos_test_h2o_lc <- datos_test_h2o_df


summary(datos_test_h2o_lc$prima_gbm_tel)
summary(datos_test_h2o_lc$prima_gbm_cl)


datos_test_h2o_lc$diff <-
datos_test_h2o_lc$prima_gbm_cl/datos_test_h2o_lc$prima_gbm_tel
datos_test_h2o_lc <- datos_test_h2o_lc %>% arrange(diff)
datos_test_h2o_lc <- datos_test_h2o_lc %>%
  mutate(exposure_cumsum = cumsum(Exposure))


datos_test_h2o_lc$decile <-
cut(datos_test_h2o_lc$exposure_cumsum, breaks =
quantile(datos_test_h2o_lc$exposure_cumsum, probs = seq(0, 1,
0.1)),
                                  labels = FALSE, include.lowest =
TRUE)


datos_test_h2o_lc <- datos_test_h2o_lc %>%
  group_by(decile) %>%
  mutate(exposure_group = sum(Exposure)) %>%
  ungroup()


write.xlsx(datos_test_h2o_lc,file
="/Users/Miguibr/Desktop/TFM/GBM_dlc_v5.xlsx", rowNames = FALSE)



# Gráfico comparativo sobre la deviance de todos los modelos.
#Modelo BENCH
bench_bin_nega_train <- glm.nb(NB_Claim ~ 1, data = train,
offset=log(Exposure), link = "log"  ,control =
glm.control(maxit=1000000))
pdt_bench_test <- predict.glm(bench_bin_nega_train, test, type =
"response")
```

```
deviance_bench <- -2*sum(dnbinom(test$NB_Claim, size =
bench_bin_nega_train$theta, mu = pdt_bench_test, log = TRUE))
deviance_bench


deviance_df <- data.frame(Model = c("GLM_bench", "GAM",
"GLM_bin", "GBM_cl", "GBM_tel" ),
                          Deviance = c(deviance_bench,
deviance_gam_freq, deviance_glm_bin_freq, deviance_gbm_freq,
deviance_gbm_freq_tel))
deviance_df$Model <- factor(deviance_df$Model, levels =
c("GLM_bench", "GAM", "GLM_bin", "GBM_cl", "GBM_tel"))


ggplot(deviance_df, aes(x = Model, y = Deviance, color = Model))
+
  geom_point(size = 3) +
  labs(x = "Model", y = "Deviance") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylim(7500, 10500)
```