uc3m | Universidad Carlos III de Madrid

Máster Universitario en Ciencias Actuariales y Financieras
2021-2022

Trabajo Fin de Máster

Comparativa de los modelos GLM y GBM para la tarificación de una cartera de autos

Gonzalo Gutiérrez Meléndez

Tutores

José Miguel Rodríguez-Pardo del Castillo

Jesús Ramón Simón del Potro

15 junio 2022

DETECCIÓN DEL PLAGIO

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

RESUMEN

El sector asegurador se encuentra en una constante evolución en todos los ámbitos gracias especialmente a los avances tecnológicos. En la rama de la tarificación, la búsqueda constante de una mejor modelización del riesgo ha generado la creación y uso de distintos algoritmos para capturar de la manera más eficiente y exacta el riesgo de los clientes.

En este estudio se lleva a cabo la comparativa entre el método clásico de modelización lineal generalizado frente a la nueva corriente de modelos basados en Machine Learning llamados Gradient Boosting.

Los resultados obtenidos muestran una mayor capacidad predictiva para el método más novedoso y una mayor granularidad para ajustar las primas de los asegurados. Ambos métodos son válidos de cara al estudio académico, y observando los resultados, todo parece indicar que es el momento de iniciar el cambio utilizando con mayor asiduidad los nuevos modelos de Machine Learning.

También se lleva a cabo un análisis de negocio comparando una metodología de mutualización frente a una segmentación en los precios con el fin de mantener la cartera sana económicamente.

Palabras clave: Modelización, *Machine Learning*, tarificación, negocio.

ABSTRACT

The insurance sector is constantly evolving thanks especially to technological advances. In the pricing section, the constant search for better modelling has led to the creation and use of different algorithms to capture customers risk in the most efficient and accurate way.

In this study, we compare the classic generalized linear modelling method against the new trend based on machine learning called Gradient Boosting models.

The results obtained show a greater predictive capacity for the most innovative method and a greater granularity to adjust the premiums of the insurers. Both methods are valid for academic study, and observing these results, everything seems to indicate that now is the moment to initiate the change and use more frequently the Machine Learning models.

We also carry out a comparative business analysis between a mutualization and a segmentation price methodology in order to maintain an economically healthy portfolio.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

El mercado financiero está compuesto por tres sectores: sector financiero, de valores y asegurador. Respecto al estudio histórico de los mismos es notoria la falta de investigación respecto al último pilar, aunque en los últimos años se ha llevado a cabo una gran exploración con obras que exponen los avances de los últimos dos siglos, como puede ser la aprobación de la ley de seguro de accidentes de trabajo de 1932 (Iparraguire, 1934) o el efecto generado por la Ley de seguridad Social de 1963 (Velarde, Guindos y Lázaro, 1963).

El sector asegurador en España comienza a crecer a partir del siglo XIX con la creación de las primeras sociedades anónimas ya que es la única sociedad mercantil capacitada para poder acumular suficiente capital como para poder abordar grandes riesgos. El comienzo del sector fue pausado debido a distintos factores como podían ser la lenta reacción de las instituciones, la falta de consolidación de los principios básicos de los mercados o la falta de información de manera que pudiese haber una mayor inversión. Además, España no podía crear ideas propias debido a la falta de capital, tanto nacional, como extranjera, ya que países como Francia o Inglaterra estaban mucho más desarrollados en estos ámbitos. A pesar de ello, en el último tercio de siglo gracias a que se establece la responsabilidad limitada empieza a surgir la modernización económica y el crecimiento financiero en España continúa aumentando.

A medida que incrementaban el número de sociedades anónimas también lo hacían las compañías de seguros como puede ser la Compañía Malagueña de Seguros Marítimos, constituida en 1836, o la Compañía General Española de Seguros "La Española", constituida en 1841.

A lo largo de este siglo es importante mencionar ciertos avances como puede ser la Ley de 27 de enero de 1848, donde las sociedades deben revalidar su existencia, la creación del Registro Mercantil en el año 1886 o la Estadística del registro mercantil en 1899. Gracias a estos avances se puede comenzar a establecer el ritmo de creación de las sociedades en España, aunque gran parte de los datos se han perdido, podemos encontrar información acerca de empresas en distintos sectores gracias a revistas y publicaciones del momento, como puede ser *Estadística administrativa de la contribución industrial y de comercio.*

El crecimiento del sector asegurador en España comienza a partir del último tercio del siglo XIX gracias a tres factores, el primero de ellos, la liberalización de la constitución de sociedades anónimas llevada a cabo en 1869. El segundo factor fue el incremento de sociedades gracias a la liberalización que generó una mayor inversión tanto nacional como internacional. Por último, la progresiva unión entre el sector bancario y el asegurador.

Con la llegada del siglo XX la actividad aseguradora continuó profesionalizándose. Sin embargo, continuaba sin estar tan desarrollada como en otros países europeos. La Guerra Civil y la posterior dictadura generaron una caída del sector y posterior estancamiento en su crecimiento y no fue hasta la década de 1980, con la entrada de España en la Unión Europea cuando tuvo un nuevo impulso. La inclusión en la Unión Europea generó un gran crecimiento en todos los sectores gracias a la internacionalización.

Desde entonces la industria aseguradora ha continuado su crecimiento especializándose en todas las líneas de negocio posible con la idea de gestionar los riesgos que sus clientes les transfieren a través de un contrato que une ambas partes a cambio de una prima establecida. El primer seguro se llevó a cabo en la industria marítima y actualmente abarca cualquier aspecto cotidiano.

Como consecuencia de este crecimiento, los riesgos que las compañías deben manejar varían dependiendo de cada situación y por ello es importante capturar correctamente, no solo su comportamiento, si no las causas del mismo.

La industria aseguradora tiene dos ramas básicas de actividad que vienen definidas por la naturaleza de los siniestros, la primera de ellas son los seguros de Vida, que incluye una gama de productos muy variada. Pueden ser seguros individuales y colectivos, están formados por coberturas de fallecimiento o invalidez o seguros de ahorro, como pueden ser planes de pensión, etc. Por otro lado, encontramos los seguros de No Vida, también abarcan un espectro muy amplio ya que comprenden el resto de las coberturas, como puede ser el seguro de vivienda o el seguro de autos, que será el que analizaremos y trataremos con más profundidad en este trabajo.

Actualmente, en España ambas partes son realmente importantes, ya que el 22% de los españoles tiene un seguro de salud, el 65% de las viviendas está asegurada y la obligatoriedad de los seguros de automóviles, con una flota en el año 2019 de 31.8 millones, genera una masa realmente importante de contratos.

La expansión de la industria aseguradora ha causado una profesionalización del sector originando un estudio del mercado más optimizado y un enfoque dirigido al cliente. Además, los grandes avances tecnológicos han ayudado a tener una mayor eficiencia pudiendo así mejorar los cálculos actuariales y financieros en las distintas ramas del negocio.

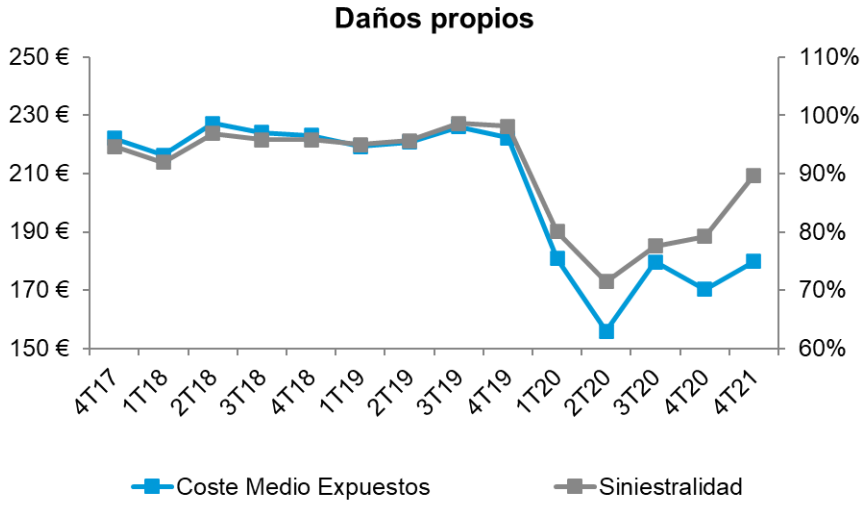
Gracias al informe ICEA (Investigación Cooperativa entre Entidades Aseguradoras y Fondos de Pensiones) que se realiza periódicamente podemos llevar a cabo un análisis sobre el seguro del automóvil en la actualidad. La información mostrada por el estudio muestra con claridad cómo se encuentra el mercado actualmente, ya que cuenta con 29 entidades participantes, obteniendo una cuota de mercado del 94.1%. En torno al 88% de las pólizas pertenecen a automóviles o furgonetas (1º categoría), el 7% pertenece a camiones, vehículos industriales, autocares, etc. (2º categoría). Y el último 5% pertenece a motocicletas o ciclomotores (3º categoría).

La siniestralidad media se encuentra en un 66,3%, siendo los vehículos de primera categoría los que mayor siniestralidad muestran (67%), seguidos por los camiones y

vehículos de gran tamaño (65%) y las motocicletas entorno al 55%. Siendo daños propios (daños causados en el vehículo tras un accidente o un siniestro) y responsabilidad civil (daño generado sobre un tercero) las coberturas más representativas. Los vehículos que mayor siniestralidad muestran son los autocares, seguidos por las furgonetas.

Como podemos ver en ambos gráficos, el efecto causado por la pandemia ha generado una disminución notoria de la siniestralidad en los dos casos, y, a medida que se recupera la normalidad y el uso del vehículo vuelve a aumentar, hay un incremento a lo largo del año 2021.

***Figura 1**. Coste medio de Daños propios.*



FUENTE= ICEA

***Figura 2**. Coste medio de Responsabilidad civil*

FUENTE: ICEA

Respecto la frecuencia en las tres categorías los daños propios son muy notorios, al igual que los de responsabilidad civil. Destaca el hecho de que los asegurados sin franquicia en las dos primeras categorías generan mayores daños propios que aquellos que sí la tienen. Por tanto, se puede considerar como una medida efectiva de las aseguradoras la inclusión de las franquicias con el fin de reducir los daños propios.

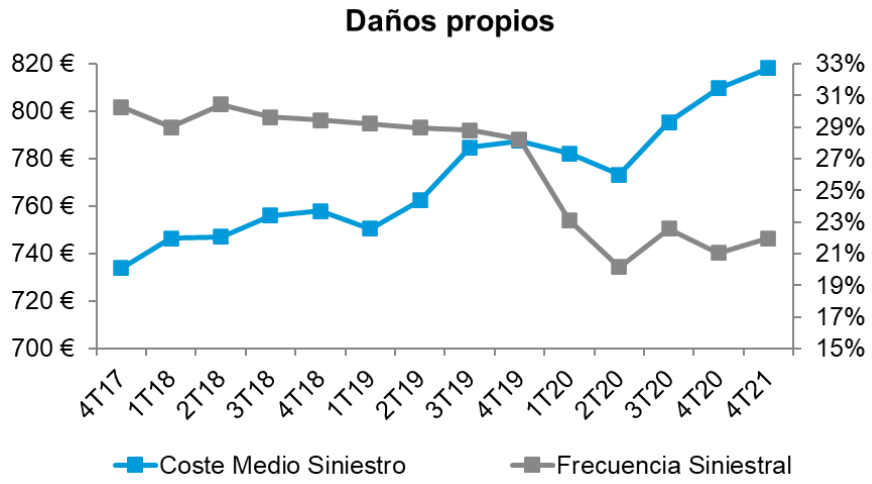
***Tabla 1**. Frecuencia de Daños propios*

Daños Propios	Con Franquicia	Sin Franquicia
1ª Categoría	14,85%	37,54%
2º Categoría	11,46%	22,24%
3ª Categoría	4,64%	1,55%

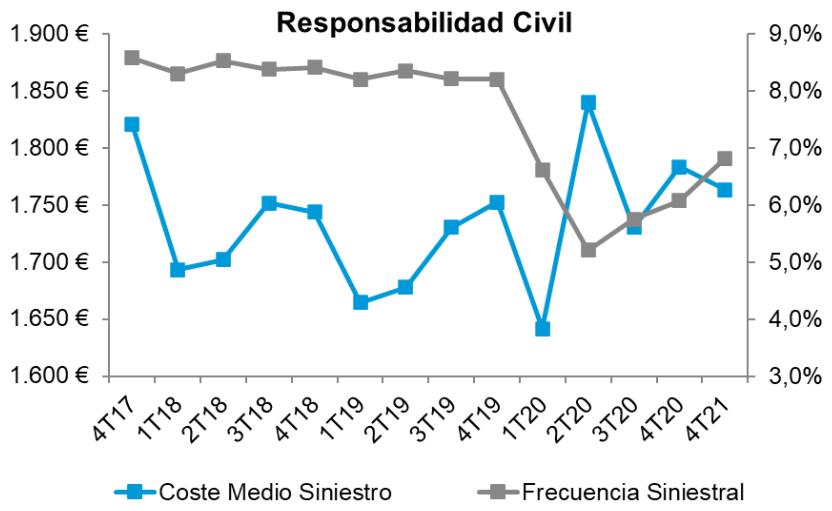
FUENTE: ICEA

Observando las figuras, al igual que en la siniestralidad el efecto de la pandemia es significativo, con una reducción de la frecuencia, pero no así del coste medio que ha aumentado en ambos casos. Una posible explicación del incremento de los costes medios puede ser el uso de una mayor tecnología de los vehículos que generan una mano de obra más cualificado para su arreglo y por tanto un mayor coste, otra posibilidad puede ser que debido a la cuarentena sufrida en el año 2020 y la disminución del uso del vehículo los conductores hayan perdido cierta pericia o control de sus vehículos. Además, con el incremento del IPC lo más probable es que los costes medios continúen subiendo.

***Figura 3.** Frecuencia y Coste medio de Daños propios*



FUENTE: ICEA



FUENTE: ICEA

En los dos gráficos previos se muestra el coste medio de los siniestros, pero se muestran las tres categorías aglomeradas. Con la siguiente tabla apreciamos como se distribuyen los distintos costes medios dependiendo del tipo de vehículo y el tipo de siniestro, siendo los vehículos pesados aquellos con mayores costes tanto en responsabilidad civil como en daños propios. El coste medio superior en los daños propios con franquicia se debe a que aquellos siniestros cuyo coste se encuentra por debajo del límite marcado por la franquicia la compañía no se hace cargo y, por tanto, únicamente paga lo siniestros con un coste más elevado.

***Tabla 2**. Distribución de costes medios*

	Responsabilidad civil		Daños propios	
	Leves	Graves	Con franquicia	Sin franquicia
1ª Categoría	2.874,76	81.351,83	934,92	703,27
2º Categoría	4.518,19	109.220,09	2.939,59	1.694,96
3ª Categoría	1.723,55	55.258,55	1.382,87	1.219,08

FUENTE: ICEA

El objetivo de la tarificación es llevar a cabo un estudio que sea capaz de capturar adecuadamente el comportamiento de los riesgos que asumen, con el fin de conocer las posibles pérdidas a las que se enfrenta la aseguradora por indemnizar a sus clientes, de acuerdo con los términos de la póliza.

La tarificación es un aspecto que a lo largo de los últimos años y gracias a los avances mencionados ha mejorado ostensiblemente. Gracias al estudio de mercado las

aseguradoras descubren nuevas variables que pueden llegar a afectar al riesgo de los asegurados, de esta manera son capaces de modelizar el riesgo de una forma más exacta. El avance tecnológico ayuda a la implementación de nuevas técnicas estadísticas, además, el crecimiento del uso de tanto la programación como del Machine Learning han ayudado a obtener una mayor eficiencia en la obtención de resultados.

1.1 OBJETIVOS Y DESCRIPCIÓN DEL ESTUDIO

La intención de este trabajo es la creación de una tarifa a partir de la modelización de la frecuencia y la severidad de los siniestros. Para ello, se llevarán a cabo distintas técnicas, es decir, se hará una comparativa entre distintas maneras de tarificar, la más utilizada en la industria actuarial actual, modelización mediante regresiones generales (GLM) y, por otro lado, utilizando técnicas de Machine Learning, se aplicará la modelización con Gradient Boosting (GBM). Para realizar el estudio dividimos la muestra de manera aleatoria en 2 partes, la primera llamada training y que es el 80% de la misma, y la segunda parte, que nos servirá para validar los resultados, que contiene el 20% de los datos, llamada test.

Antes de comenzar la modelización se debe hacer una exploración de la base de datos con el fin de entender las variables que contiene, ya que puede haber variables que estén incompletas, por valores vacíos, valores punta o valores incorrectos. Una vez se ha limpiado la base de datos se debe analizar la misma para ver las distintas características de cada una de las variables. Todo este estudio se llevará a cabo en el punto tres del trabajo.

Tras el estudio de la base de datos, se puede comenzar la modelización. En la actualidad el método más utilizado es el GLM, pero poco a poco se va introduciendo en el negocio el segundo método gracias a los avances tecnológicos. Mediante este estudio comprobaremos en las distintas fases de creación de la tarifa distintos aspectos, como pueden ser qué modelo predice de una forma más exacta, cuál es más útil, más eficiente, etc. El análisis y creación de ambos modelos se llevará a cabo en el entorno de programación Python, utilizando las bibliotecas más actualizadas, en especial para el caso donde tengamos que aplicar técnicas de Machine Learning.

El GLM, primer método que aplicaremos en el estudio es la generalización de la regresión lineal, muy utilizada en ámbitos de investigación como la economía o la medicina. La cualidad que destaca respecto a los modelos de regresión lineal es el hecho de que su distribución de errores no sigue una distribución normal, es decir, no se asume que sigue una distribución normal.

Respecto a la modelización GBM, es una técnica utilizada para generar modelos de predicción como un conjunto de '*weak learners'*, que en su mayoría se trata de árboles de decisión. Un modelo de árboles de Gradient Boosting se construye por etapas de manera que puede corregir los errores cometidos en árboles previos.

En ambos casos se crearán dos modelos, uno para la frecuencia y otro para la severidad, y a partir de ellos se puede obtener la prima pura mediante la creación de los modelos Burning Cost. Ahora es el momento de comparar los resultados conseguidos. Para ello se realizará un análisis por cada variable observando las posibles variaciones entre una posibilidad y otra. La creación, estudio y análisis de los distintos modelos se lleva a cabo en el apartado cuatro, respecto a la comparativa entre las distintas maneras de tarificar se definirá en el punto cinco.

El último paso, que se tratará en el sexto capítulo, es el análisis de impacto y el punto de vista del negocio. En este punto debemos decidir como evoluciona nuestra cartera. Debemos analizar el ratio de siniestralidad y como deben ser las nuevas primas para tratar de mantener sana a nivel económico la cartera.

1.2 RESULTADOS OBTENIDOS

La modelización mediante Gradient Boosting ha mejorado a la metodología clásica, tanto en la frecuencia como en la severidad. Destaca en especial la capacidad para la segmentación, otorgando una menor prima pura a los mejores conductores y una prima más elevada a aquellos con un riesgo más elevado.

A lo largo del trabajo se han encontrado diferencias entre ambas modelizaciones, como puede ser una mayor dificultad y un mayor tiempo en el caso del GLM de encontrar un buen modelo que capture el riesgo adecuadamente, donde todos sus coeficientes sean significativos.

Una de las grandes diferencias que se han confirmado, mediante el uso de los gráficos de dependencia parcial, es la capacidad de la nueva modelización para capturar riesgos no lineales, otorgando un mayor ajuste a este método.

Mediante el uso de clústeres se ha llevado a cabo una comparación entre las distintas modelizaciones de la frecuencia, la severidad y, por último, la prima pura o *Burning Cost*. En ellos se ha obtenido resultados similares para la frecuencia, una mayor diferencia para la severidad y finalmente, diferencias más abultadas para la prima pura.

A parte del análisis académico llevado a cabo es importante tratar de aterrizar los resultados al mundo empresarial, eso es lo que se ha hecho en el sexto apartado comparando una posible renovación de cartera mediante dos metodologías, la mutualización de la prima, donde todos reciben el mismo incremento de la prima, o una segmentación de esta donde dependiendo del clúster al que los clientes están asociados reciben una bonificación o un recargo.

En definitiva, ambas modelizaciones son válidas de cara al estudio académico, siendo la metodología basada en machine learning aquella a la que se ajusta mejor el riesgo. Además, muestra una mejor granularidad, es decir, muestra una mejor segmentación de la prima dependiendo del riesgo del cliente. Por ello, tras analizar esta comparativa es el momento de comenzar a utilizar esta nueva metodología a nivel profesional.

2. BASE TEÓRICA DEL ESTUDIO

El Machine Learning es la combinación de la estadística con la ciencia de computación y se aplica en diversas casuísticas como pueden ser estudios con multitud de variables, ante falta de valores y datos o relaciones no lineales. Existen diversas ventajas como puede ser la automatización y reducción de errores, la precisión de los cálculos o la fiabilidad de las predicciones.

Esta última cobra gran importancia ya que mejora los métodos clásicos y se utiliza a nivel empresarial con gran frecuencia. Existen distintas cinco de análisis predictivo:

1. Regresión: Consiste en predecir un valor cuantitativo.
2. Clasificación: Reside en predecir un valor cualitativo.
3. Aprendizaje no supervisado: Consiste en separar la muestra en grupos con características similares (Clustering).
4. Análisis de supervivencia: Predice el tiempo antes de un suceso
5. Series temporales: Radica en predecir valores futuros usando históricos.

Estos cinco grupos se pueden dividir en dos, aprendizaje supervisado y aprendizaje no supervisado. El primero de ellos agrupa las regresiones, los árboles, los métodos de clasificación, etc. El aprendizaje supervisado consiste en reproducir o predecir a través de la información del pasado.

Respecto al aprendizaje no supervisado incluye el clustering y los análisis de componente principal. En este caso se trata de hacer agrupaciones dentro de la muestra con características similares. Una vez tenemos los conjuntos de datos se pueden tratar de forma homogénea.

En este estudio únicamente analizaremos métodos del aprendizaje supervisado, en concreto, modelos de regresión lineales generalizados y modelos Gradient Boosting.

2.1 MODELOS LINEALES GENERALIZADOS

Los modelos de predicción han sido estudiados no sólo en múltiples ramas del ámbito de la investigación, sino también en el mundo empresarial. En el sector asegurador su implementación para la creación de tarifas surge en la década de 1980. Dada su importancia, la modelización ha mejorado con el paso del tiempo, ya que ser capaces de capturar de la mejor manera posible el riesgo posibilita una mejor estimación y por tanto mayor competitividad en el sector.

Los modelos lineales generalizados son formulados en primera instancia por los estadísticos John Nelder y Robert Wedderburn, quienes trataron de unificar distintos modelos estadísticos como la regresión logística o la regresión lineal; para ello, llevaron a cabo un método de mínimos cuadrados ponderados para la estimación de máxima verosimilitud de los distintos parámetros del modelo.

La idea principal de la regresión lineal es la predicción de la variable dependiente a partir de la combinación lineal del conjunto de variables independientes utilizadas. Pero esto significa que un cambio constante en las variables explicativas genera una variación constante en la explicada, es decir, la variable explicada sigue una distribución normal; esto es algo que no siempre ocurre, por ejemplo, la distribución más común para el número de siniestros es la Poisson.

En estos modelos se plantean tres supuestos, el primero de ellos es que los errores se distribuyen normalmente, la segunda característica es que la varianza es constante, es decir, son modelos homocedásticos y, en tercer lugar, la variable dependiente se relaciona linealmente con la variable independiente. Estos supuestos pueden no cumplirse siempre, por ello, una solución muy común es la transformación mediante logaritmos.

Mediante los modelos lineales generalizados se trata de ampliar el espectro de manera que no se tomen hipótesis que puedan afectar el resultado y poder así realizar una mejor predicción. Las tres características más importantes son:

- Variable aleatoria **Y**, que sigue una función de distribución perteneciente a la familia exponencial
- Un predictor lineal o componente sistémica: $\eta = \boldsymbol{X\beta}$
- Una función de enlace g tal que $E(\boldsymbol{Y}) = \mu = g^{-1}(\eta)$

Respecto la variable aleatoria Y, en nuestro caso, para la frecuencia será el número de accidentes y para la severidad será el coste total de los siniestros. Cada una de estas variables tiene una distribución: para el primer caso la distribución más común es la Poisson, ya que es realmente útil para calcular sucesos que tienen pocas posibilidades de ocurrir en un periodo determinado de tiempo -por ejemplo: que un asegurado tenga una gran cantidad de accidentes a lo largo de un año. La distribución más común de la severidad es la Gamma, utilizada para modelizar comportamientos de variables aleatorias con asimetrías positivas.

Los predictores lineales están compuestos por coeficientes ($\boldsymbol{\beta}$) y la matriz de variables independientes (X), es decir, el predictor lineal alberga la información del modelo.

$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \cdots + \beta_n * X_n \qquad (1)$$

La última característica es la función de enlace. Su labor es relacionar el predictor lineal y la media de la función de distribución. Dependiendo de las características, se pueden utilizar distintas funciones como pueden ser la log, logit o probit.

$$\mathrm{g}\,(Y) = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \cdots + \beta_n * X_n \qquad (2)$$

Una vez creamos el modelo debemos fijarnos en varios detalles: el primero de ellos es si los coeficientes son significativos. Cuando nos encontramos con variables no significativas es preferible eliminarlas para tratar de evitar un posible sesgo sobre las demás. El siguiente aspecto es la desvianza (D^2). Es una medida de bondad del ajuste de los modelos lineales generalizados, es decir, el nivel de varianza explicada por el modelo. Es similar a la suma de cuadrados residual de los modelos lineales, donde los valores más

elevados muestran un peor ajuste. Aquellos modelos con menor desvianza serán los que mejor ajusten y por tanto los escogidos.

Se debe comparar la desvianza del modelo nulo, con el modelo escogido, así se puede calcular la incertidumbre explicada por el modelo. El modelo nulo es aquel que únicamente incluye la constante (β_0).

$$D^2 = \frac{Desvianza\ nula - Desvianza\ residual}{Desvianza\ residual} * 100 \quad (3)$$

Para cada tipo de GLM se realizarán distintos modelos, siendo los criterios de Akaike y el criterio bayesiano de Schwartz, la mejor manera para comparar las cualidades de los distintos modelos. Así pues, gracias a estos dos métodos y a la desvianza seremos capaces de escoger el mejor modelo.

Por último, se deberán analizar los residuos de los distintos modelos mediante distintos histogramas, como puede ser el q-q plot, que permite contrastar la distribución de los residuos.

Todo este proceso de revisión sobre el modelo se lleva a cabo porque el mercado asegurador es muy dinámico, es decir, se obtiene nueva información continuamente de manera que se debe comprobar si realmente el modelo está funcionando correctamente o si por el contrario no está recogiendo los distintos riesgos correctamente.

2.2 MODELOS GRADIENT BOOSTING

En la actualidad, las compañías están comenzando a introducir otra forma de modelar basada en árboles de decisión. Con esta nueva técnica no se trata de eliminar la regresión generalizada, únicamente se pretende mejorar el uso actual de la modelización.

Existen multitud de ventajas para el uso de esta nueva forma de analizar y obtener información del dato: se seleccionan los predictores de forma automática. La selección de las variables más significativas e importantes se detecta fácilmente, al contrario que en modelos generalizados lineales, los valores punta no generan un sesgo tan elevado. Su comprensión es sencilla. No es necesario que las variables explicadas deban seguir una distribución determinada, ya que no se aplican métodos paramétricos, etc.

Por el contrario, también hay desventajas para su uso, como pueden ser variaciones en la muestra que generan grandes cambios tanto en los resultados como en la segmentación de las variables en los árboles de decisión; tendencia al '*overfitting*' o sobreajuste; creación de árboles muy complejos que genera dificultad de entendimiento.

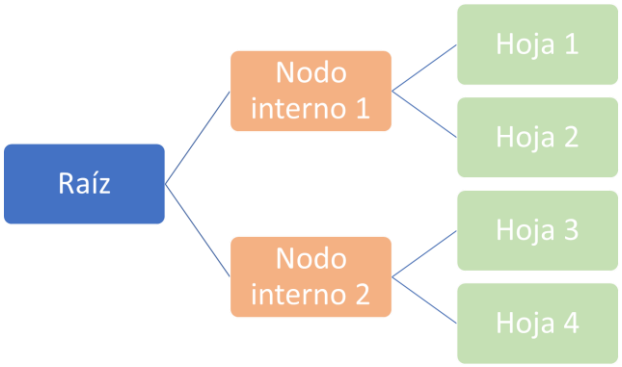
2.2.1 ÁRBOLES DE DECISIÓN

Para poder explicar el funcionamiento de los modelos Gradient Boosting debemos explicar la base teórica en la que se sustentan. Son un conjunto de árboles de decisión que tratan de mejorar los errores de los árboles anteriores. La idea principal es dividir la muestra en distintos subconjuntos con características similares. Mediante multitud de divisiones, basadas en condicionales sobre los valores de las distintas variables y una jerarquía en función de dichas variables se obtienen los resultados finales, que son las predicciones del árbol de decisión.

La estructura de los árboles de decisión se divide en tres partes. En primer lugar, Raíz o Nodo raíz, en segundo lugar, nodos internos o intermedios, y, finalmente, hojas o nodos terminales.

Los árboles de decisión pueden ser árboles de clasificación o de regresión, existen dos grandes diferencias. La primera de ellas se debe a la variable que debe predecir, si la variable es cualitativa, por ejemplo, detección de fraude, es un árbol de clasificación, si la variable es cuantitativa, por ejemplo, el número de siniestros en un año es un árbol de regresión. La segunda discrepancia surge en las hojas del árbol: en el caso de los árboles de clasificación el valor es la moda de las observaciones; sin embargo, en los árboles de regresión el valor es la media de las observaciones que han caído en esa hoja.

Figura 4. *Partes del árbol de decisión*



FUENTE: Elaboración propia.

Los árboles de decisión tienen 4 características que deben analizarse con el fin de obtener un algoritmo lo más eficiente posible. Debe haber un mínimo de observaciones para dividir entre nodos u hojas, ya que puede ocurrir que se segmente tanto la muestra que al final no se esté prediciendo si no mostrando los resultados de la muestra y, por tanto, generando un mal modelo. Es decir, a mayor número mínimo de observaciones menor aprendizaje tendrá el modelo sobre valores específicos, pero, si es muy elevado no predecirá adecuadamente. Para saber cuál es la muestra mínima que debe haber por nodo u hoja se debe utilizar la validación cruzada.

En segundo lugar, la profundidad del árbol también es importante; a mayor longitud, mayor aprendizaje y mayor exactitud con la muestra. Se debe validar mediante validación cruzada. La tercera característica, muy relacionada con la anterior, es el número máximo

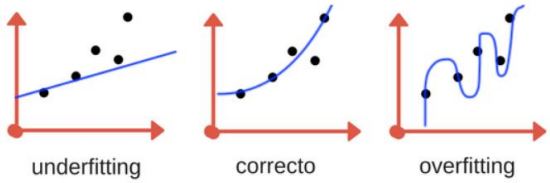
de hojas, ya que puede elegirse si se prefiere profundidad del árbol o un número máximo de hojas.

Y, por último, el número máximo de atributos a considerar para la ramificación. Esta característica es especialmente utilizada en los bosques de decisión, pero consiste, con el fin de obtener una mayor eficiencia, en elegir aleatoriamente entre las variables explicativas un número de estas para realizar el árbol. Se escogen la raíz cuadrada del número total de variables, aunque puede ir incrementándose hasta un 40% del total.

En nuestro caso, queremos predecir tanto la frecuencia del número de siniestros de cada asegurado como la severidad de éstos, mediante las características de las distintas variables de la base de datos. La idea del uso de los árboles de decisión no es ajustar bien, si no predecir bien. Un ajuste perfecto tiene una varianza muy elevada y ante variaciones en la muestra su predicción puede no ser correcta, surge de esta manera el problema del sobreajuste. El caso contrario, llamado subajuste, es aquel en el que con nuestro modelo no somos capaces de segmentar la muestra de forma adecuada.

Para reconocer estos problemas debemos subdividir la muestra, al igual que en el caso de los modelos GLM, en una parte de entrenamiento, que en nuestro caso será el 80% de los datos, y una segunda parte, el otro 20%, que llamaremos parte de validación o test y con los que comprobaremos los resultados de la parte de entrenamiento. La manera para tratar de eliminar los problemas será probando mediante distintas características (que analizaremos y explicaremos más adelante) del algoritmo que utilizaremos, como puede ser la tasa de aprendizaje o la profundidad del árbol y del bosque.

***Figura 5**. Ejemplos de predicción.*



FUENTE: Elaboración propia

Tras mostrar la composición, las características y los problemas que pueden generarse si nuestro árbol se diseña de manera errónea, debemos explicar cómo se crean los árboles de decisión. En primer lugar, al igual que para la modelización GLM, tenemos una variable explicada (y_i), y variables explicativas (x_i). Como es de esperar, no todas las variables segmentan igual la muestra, por ello, antes de comenzar debemos probar qué variables son mejores. Debemos realizar pequeños árboles (de únicamente dos hojas) para cada variable y medir el nivel de impureza de la división. Para medir este nivel de impureza el método más utilizado es Gini, aunque también hay otros métodos como la ganancia de información o la detección automática de interacciones mediante chi-cuadrado, que, aunque no se utilizará a lo largo del estudio debe ser mencionada.

El primer método es utilizado para predecir la probabilidad de que un extracto de tu muestra haya sido clasificado erróneamente por un nodo específico. La fórmula para el cálculo de las hojas se muestra en la ecuación 4, una vez se obtienen estos resultados se debe calcular la impureza de Gini para el nodo. Se debe tener en cuenta el peso de cada hoja, como observamos en la ecuación 5. Así pues, la mejor variable será aquella con un Gini menor.

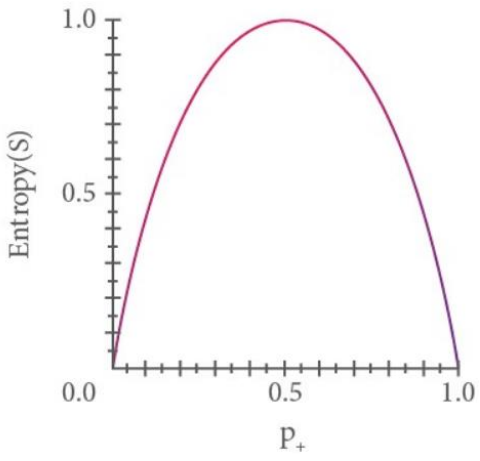
$$Gini = 1 - (Prob.acertar)^2 - (Prob.\ fallo)^2 \quad (4)$$

$$Impureza\ de\ Gini = \left(\frac{Muestra\ hoja_1}{Muestra\ total}\right) * Gini_1 + \left(\frac{Muestra\ hoja_2}{Muestra\ total}\right) * Gini_2 \quad (5)$$

El segundo método es la ganancia de información, también conocida como divergencia de Kullback-Leibler. Para entender este concepto se debe explicar el significado de entropía. La entropía mide la impureza de la muestra, por ejemplo, si en tu muestra hay 10 hombres y 0 mujeres, la entropía es 0. Sin embargo, si tuviésemos 5 hombres y 5 mujeres, nuestra entropía sería 1. Por ello, cuanto menor es la entropía mayor ganancia de información se obtiene.

$$Entropía = -p * log_2(p) - q * log_2(q)^1 \quad (6)$$

Figura 6. *Entropía*



FUENTE = Elaboración propia

$$Ganancia\ de\ información = 1 - Entropía \quad (7)$$

Una vez sabemos cuál es la variable que debemos usar en un primer momento comenzamos la división o segmentación del árbol. Esta segmentación se lleva a cabo de distinta manera si la variable escogida es categórica o si es numérica. En el caso de ser categórica, por ejemplo, el sexo de los asegurados se divide si pertenece a un sexo u otro, o los colores de los coches, donde ya no es una variable binaria, se segmenta si pertenece

[1] P hace referencia a la probabilidad de acierto y Q a la probabilidad de fallo.

a uno o al resto, o si pertenece a dos tipos de color o al resto, etc. Siempre teniendo que calcular la impureza de Gini para escoger el mejor corte de la variable.

Las variables numéricas, por ejemplo, el peso de los asegurados, se debe escoger el mejor punto de corte, para ello, se realiza la media entre los distintos datos de peso que existen en nuestra base de datos. A continuación, como ya hemos explicado previamente se realiza la impureza de Gini y se escoge el menor resultado.

Para la creación de más nodos se deben seguir los pasos explicados previamente, tanto para escoger qué variable debe ser la siguiente como para saber cómo dividirla. Llegados a este punto es el momento de saber cuándo debe parar la profundidad del árbol, como se ha explicado brevemente, gracias a la validación cruzada, se sabrá cual es la profundidad más eficiente. A pesar de ello, existen límites como el número mínimo de datos que debe haber en cada hoja o que si al dividir el nodo la impureza de Gini es superior no debe crearse esa nueva segmentación.

2.2.2 BOSQUES ALEATORIOS

Los árboles de decisión tienen una sencilla aplicación y comprensión, pero debido a su rigidez ante nuevos datos y su posterior falta de precisión no son la mejor herramienta para el aprendizaje predictivo. Con el fin de mejorar estos resultados surge la idea de los Bosques aleatorios tratando de mantener la sencillez de los árboles de decisión e incrementando la flexibilidad de la modelización.

Los bosques aleatorios son un conjunto de árboles de decisión, pero generados de una manera ligeramente distinta. Para explicar la diferencia debemos analizar el concepto del algoritmo Bootstrap. Se trata de tomar de forma aleatoria valores de la muestra inicial con el fin de obtener una muestra nueva con la misma distribución que la inicial.

En la siguiente tabla se muestra un ejemplo sencillo con únicamente una muestra de cuatro individuos y dos variables. Como podemos ver, una vez se escoge un dato de la muestra (entendiendo como dato toda la información de cada individuo) este no se elimina de la muestra inicial y puede volver a ser escogido. En este caso el segundo individuo no ha sido escogido pero la mujer ha sido escogida en dos ocasiones.

Tabla 3. *Ejemplo Bootstrap*

Base de datos inicial

sexo	Edad	Número siniestros
M	33	0
M	37	1
M	80	1
F	55	1

→

Base de datos con Bootstrap

sexo	Edad	Número siniestros
F	55	1
M	33	0
M	80	1
F	55	1

FUENTE: Elaboración propia

Una vez tenemos ya la nueva muestra de datos con el mismo tamaño que la muestra inicial debemos crear el primer árbol de decisión, aplicando la misma metodología explicada en el apartado anterior, pero con un número inferior de variables. Normalmente se utiliza la raíz del número de variables. Para la raíz se escoge entre todas las variables que se encuentran en la selección y a medida que se avanza en los nodos no se pueden utilizar las variables utilizadas previamente.

Finalmente, ya tenemos el árbol terminado y solo falta repetir este proceso tantas veces como sea necesario. Para saber cuál es la predicción final se deben obtener los resultados de todos los árboles creados y en caso de que la variable explicada sea categórica se escogerá la moda y en caso de que sea variable numérica se escogerá la media.[2]

Cuando se han creado las distintas muestras mediante Bootstrap se permitió que hubiese datos duplicados, es decir, existen datos que no se utilizaron. Éstos se denominan datos fuera de la bolsa y son utilizados para comprobar si el bosque creado es bueno o no. Para ello, introducimos esta muestra en sus respectivos árboles de decisión y comprobamos si han sido estimados de manera correcta o no. De esta manera se puede medir la proporción de la muestra que ha sido correctamente predicha.[3]

Además, aparte de comprobar la exactitud del bosque creado se puede tratar de mejorar la estimación mediante variaciones en el número de variables para hacer el Bootstrap (siempre partiendo de la base de la raíz del número de variables), el número de árboles que conforman el bosque y al igual que en el apartado anterior modificando las características de los árboles.

2.2.3 ADABOOST

Adaboost es una metodología donde se aplican los conceptos tanto de los árboles de decisión como de los bosques aleatorios. Para definir este nuevo método debemos mencionar sus cuatro conceptos claves:

1. Profundidad determinada de los árboles.
2. Combinación de ‘weak learners’.
3. Orden de importancia en los tocones.[4]
4. Se tienen en cuenta los errores de los tocones previos.

La primera característica difiere de los bosques aleatorios donde no había una profundidad determinada, en este caso cada árbol tiene únicamente un nodo y 2 hojas, a lo que se denomina tocón.

El uso de un único tocón no es útil ya que su poder de predicción es muy escaso, por ello se considera a cada tocón como ‘weak learner’. La idea principal es obtener una mayor

[2] El uso de Bootstrap y el conjunto de árboles de decisión para tomar una decisión se llama “Bagging”.
[3] La proporción de muestra clasificada de manera incorrecta se denomina ‘Error fuera de la bolsa’ (Out of the bag error).
[4] También llamados ‘Stumps’.

precisión gracias a la combinación de éstos, es decir, crear un bosque de tocones. Pero no todos los tocones tienen la misma relevancia, su orden de creación es importante, esto es algo que difiere al bosque aleatorio explicado previamente. Además, como explicaremos a continuación, cada tocón tiene en cuenta los errores de los casos previos.

Al igual que en los casos previos debemos averiguar con qué variable debemos comenzar la predicción. En primer lugar, otorgamos el mismo peso a cada dato de nuestra muestra (en nuestro caso sería el mismo peso a cada asegurado, siendo n la muestra total cada asegurado tendrá un peso de $\frac{1}{n}$). A medida que vayamos generando distintos tocones los pesos variarán.

Para la creación del primer tocón se debe hacer igual que en los árboles de decisión donde se elige la variable en función del menor Gini o por la ganancia de información. Una vez se obtiene la mejor variable, se debe hallar cuánta información aporta. El error total de cada tocón es la suma de pesos asociados a la muestra mal predicha.

$$Información\ aportada = \frac{1}{2}\log\left(\frac{1 - error\ total}{error\ total}\right) \qquad (8)$$

A continuación, llega el momento de modificar los pesos: incrementando los incorrectos y disminuyendo aquellos en los que acertó en un primer momento. Una vez se cambian los pesos como indican las ecuaciones debe normalizarse para que la suma total sea igual a 1.

$$Peso_{incorrectos} = Peso\ inicial * e^{información\ aportada} \qquad (9)$$

$$Peso_{correctos} = Peso\ inicial * e^{-información\ aportada} \qquad (10)$$

Tras la modificación, con el fin de otorgar mayor importancia a los errores cometidos, se debe realizar una nueva base de datos del mismo tamaño. Para poder hacerla tomamos los pesos como una distribución acumulada y, mediante números aleatorios entre 0 y 1, obtenemos la nueva base de datos, así pues, al tener mayor peso los errores, existe una mayor probabilidad de que en esta nueva muestra aparezcan. Los pesos de esta nueva base de datos vuelven a ser los originales. De esta forma, los errores afectan a los siguientes tocones.

Finalmente, la manera para saber la predicción de esta modelización es sumando la información aportada para cada opción y escoger aquella más elevada.

2.2.4 ÁRBOLES GRADIENT BOOSTING

La razón por la que se han explicado previamente los árboles de decisión, los bosques aleatorios y el algoritmo Adaboost es porque la modelización Gradient boosting es una unión de todas ellas.

Esta nueva metodología es muy similar a la vista previamente, donde, al igual que antes, cada tocón se basaba en los errores del anterior con el fin de poder mejorar la predicción. En el caso actual, existen ciertas diferencias, en primer lugar, ya no es un conjunto de tocones o 'Stumps', son árboles de decisión. Además, el primer paso pasa a ser el cálculo de la media, si trabajamos con variables continuas, o la moda, si estamos utilizando variables categóricas.

Así pues, la primera predicción es dicha media o moda, pero la predicción no termina aquí. Ahora se realiza el primer árbol teniendo en cuenta los errores de la primera predicción. Dichos errores, llamados pseudo residuos, son la diferencia entre la muestra observada y la predicción realizada. Dicho árbol no trata de predecir la variable explicada, si no los pseudo residuos.

Una vez tenemos la media y la predicción de los residuos, es importante mencionar el ratio de aprendizaje. Con un ratio de aprendizaje igual a 1 la predicción no es buena ya que pese a existir poco sesgo la varianza es muy elevada y ante un cambio de la muestra las predicciones no serás adecuadas. Por el lado contrario, cuanto menor sea el ratio de aprendizaje mayor número de árboles será necesario para aproximarse a una predicción adecuada pero la varianza disminuirá.

$$Predicción = \overline{y} + Ratio\ aprendizaje * pseudo\ residuos \qquad (11)$$

De esta manera realizando este proceso en multitud de ocasiones se obtiene el resultado final. Al igual que en los árboles de decisión, las características, también llamadas hiperparámetros, pueden modificarse con el fin de una mayor eficiencia en la predicción. Las características principales son el número de hojas por árbol, el número de árboles y la profundidad del árbol.

Una vez se ha realizado el modelo Gradient boosting debemos comprobar que es un buen modelo, para ello se lleva a cabo la validación cruzada donde se trata de realizar un análisis sobre las características.

2.2.5 VALIDACIÓN CRUZADA

Una vez hemos realizado nuestro primer modelo debemos saber si es óptimo o si por el contrario se puede mejorar. La validación cruzada es una técnica para evaluar los distintos modelos realizados con el fin de saber cual es el mejor. Para ello existen diversos métodos, pero en este estudio realizaremos el método de K-fold. Para esta comparativa se utiliza únicamente la parte de la muestra que hemos denominado train. Esta muestra la dividimos en k submuestras, en nuestro caso 5. La idea principal de la validación cruzada es tratar de evitar el sobre ajuste o infra ajuste, es decir, debemos obtener un modelo que estime lo mejor posible, no un modelo descriptivo.

El objetivo del procedimiento es comprobar la capacidad del modelo para predecir nuevos datos que no se utilizaron en la estimación y, por tanto, evaluar cómo se ajustan a un conjunto de datos desconocidos.

3. EXPLORACIÓN DE LA BASE DE DATOS

3.1 ANÁLISIS DE LAS VARIABLES

La base de datos utilizada en este estudio trata de simular el entorno que encontraríamos en una compañía, es decir, el tamaño de esta debe ser lo suficientemente grande como para poder utilizar las distintas herramientas estadísticas y obtener resultados que sean significativas y representativos.

El conjunto de datos que vamos a utilizar contiene un total de 13 variables con un total de 60.392 datos. Las variables son:

- **Fecha_vencimiento**: Fecha de vencimiento de la póliza.
- **Sexo**: Género del conductor principal. Variable Alfanumérica, , transformada en numérica para la modelización GBM.
- **Edad_conductor_agrupada**: Edad del conductor principal agrupada, siendo 1 el intervalo de conductores más jóvenes y 6 el de edad más avanzada. Variable Numérica.
- **Fecha_nacimiento**: fecha de nacimiento del conductor principal. Variable Numérica.
- **Credit_Scoring**: Muestra la Solvencia del asegurado, cuánto más bajo mayor nivel de insolvencia. Variable Numérica
- **Area_residencia**: Área geográfica de la residencia del conductor principal. Variable numérica.
- **Indice_trafico**: Condiciones de tráfico, siendo 0 las mejores condiciones y 100 la media esperada. Variable numérica.
- **Antiguedad_vehículo**: Antigüedad del vehículo. Variable categórica. 1 son los más jóvenes y 4 los coches más antiguos. Variable numérica.
- **B7_tipo_vehiculo**: Los distintos tipos de vehículos que encontramos en la cartera. Variable alfanumérica, transformada en numérica para la modelización GBM.
- **Valor_vehiculo**: Valor del vehículo en euros.
- **Oficina**: Área geográfica de la oficina. Variable numérica.
- **Num_siniestros**: Número de siniestros ocurridos. Variable numérica.
- **Coste_siniestro_total**: Coste total del siniestro. Variable numérica.

Siendo las dos últimas variables las que se utilizarán para modelar tanto la frecuencia como la severidad. Es importante mencionar la inclusión de variables a partir de esta base de datos, en primer lugar, la creación de la variable exposición, que muestra el periodo de póliza que aún no se ha consumido. Otra variable incluida es la edad, gracias a tener la fecha de nacimiento es posible hacerlo, además se ha completado la variable 'edad_conductor_agrupada' ya que había algunos valores vacíos. Dicha variable segmenta por décadas las edades de los conductores principales de la cartera.

Para el estudio debemos saber cuáles son las características de las variables. Contamos con variables cuantitativas y cualitativas. La primera de ellas puede expresarse mediante números y la segunda no. Dentro de cada una de ellas podemos encontrar variables categóricas, que contienen un número limitado de valores distintos. Éstas pueden ser nominales, ocurre cuando dichos valores representan categorías sin clasificación (por ejemplo, en la variable tipo de vehículo no es preferible una furgoneta que una autocaravana). Por otro lado, puede haber variables que sí tengan una clasificación, las cuales se denominan ordinales (por ejemplo, número de accidentes, siendo 0 la mejor opción y 5 la peor).

En nuestra base de datos partimos con cinco variables cuantitativas. En el caso de Credit Scoring e índice de tráfico no tenemos toda la información posible pero más adelante se analiza y se expone cómo deben tratarse dichos datos. La edad de los clientes en esta cartera contiene un gran variedad, desde noveles hasta individuos con 95 años. El valor del vehículo muestra una gran amplitud entre el menor valor y el más abultado, generando una gran desviación. El coste del siniestro, al igual que la variable anterior también tiene una gran desviación; en este caso, para el futuro cálculo de los modelos de la severidad deberemos tener en cuenta posibles valores punta que puedan desvirtuar una estimación adecuada. Podría llamar la atención una media tan cercana a 0 ya que el máximo es tan elevado, esto se debe al gran número de pólizas que no han tenido siniestros, por ello aparte de analizar el coste medio también se debe estudiar el coste total de la cartera.

***Tabla 4**. Información de las variables*

Variables	Cuantía	Sin datos	Máximo	Minimo	Media
Credit Scoring	57.591	2.801	850	301	668
Índice tráfico	56.889	3.503	207	0	106
Valor vehículo	60.392	0	380.160	1.980	19.560
Edad	60.392	0	95	18	48
Coste del siniestro	60.392	0	69.479	0	645

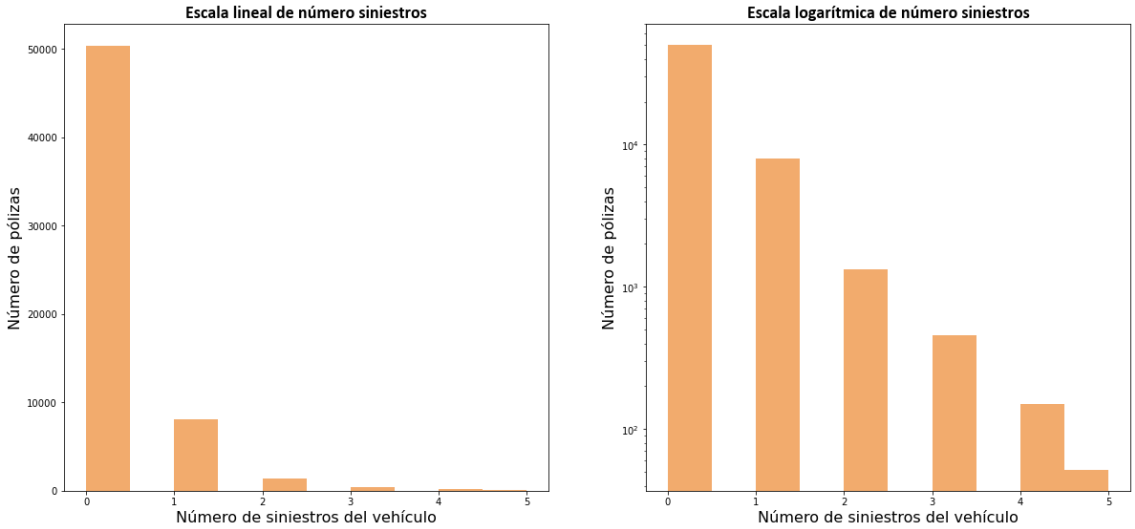
FUENTE: Elaboración propia

Tras exponer las variables de la base de datos debemos ver cómo son las características de estas. Así sabremos cómo modificarlas en caso de falta de significatividad en los modelos.

Comenzamos analizando la frecuencia de accidentes. Como es de esperar y como se puede apreciar, gran parte de la masa de la cartera no ha tenido accidentes. Esto genera que el resto de número de siniestros apenas pueda percibirse, por ello realizamos una escala logarítmica, de forma que se pueden apreciar las proporciones de cada siniestro, siendo descendiente el número de los mismos a medida que aumenta el número de siniestros. Del total de 60.392 pólizas, 50.362 no han tenido ningún siniestro. 8.034 han sufrido un único siniestro, el total de pólizas con 2 siniestros es de 1.336. Por último, con 3, 4 y 5 siniestros ha habido 459, 149 y 52 pólizas respectivamente.

Debemos saber si realmente esta variable tiene una distribución Poisson, para ello comparamos su media y su varianza. Los resultados muestran 0.21 y 0.3, por tanto es un resultado bastante cercano.

Figura 7. *Escala del número de siniestros*

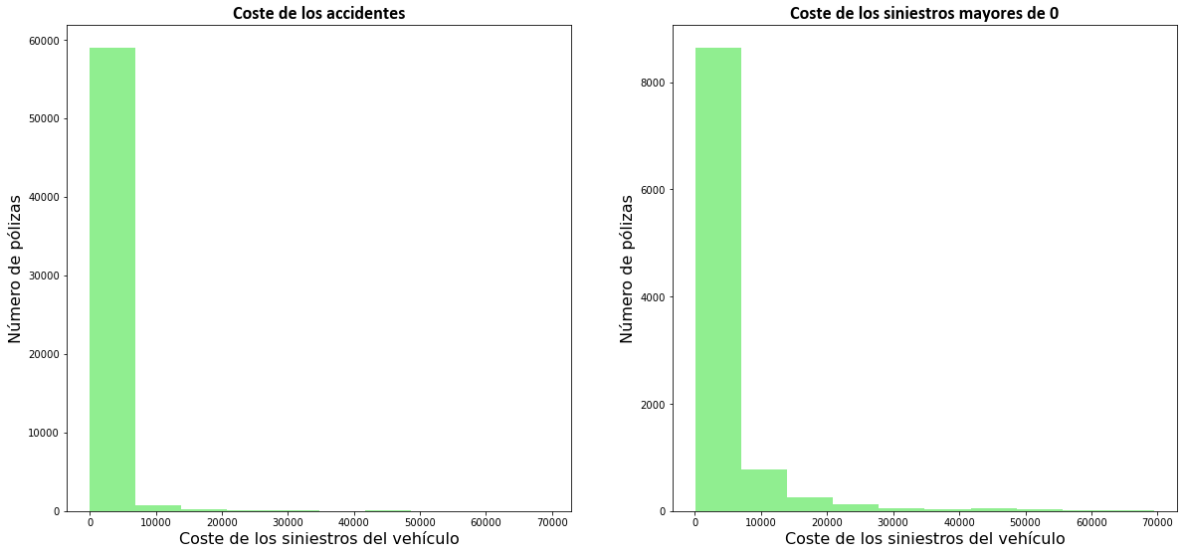


FUENTE: Elaboración propia

La variable coste de los siniestros concuerda con la frecuencia ya que la mayor parte de pólizas no ha sufrido un accidente y por ello el coste es 0. En el segundo gráfico eliminamos las pólizas que no han sufrido accidente con el fin de observar con mayor facilidad el resto de la muestra y podemos apreciar que la mayoría de los siniestros son inferiores a 10.000€. A pesar de ello, observamos siniestros con costes variados siendo el más elevado de 69.478€.

Para llevar a cabo la modelización es importante saber como se distribuyen los costes de los siniestros ya que el hecho de tener una distribución con una cola larga, como es este caso donde la gran parte de la masa de siniestros se concentra por debajo de los 10.000€ pero tenemos casos más elevados puede generar una tendencia alcista de los resultados, es decir, se puede llegar a generar sesgo en los resultados donde se estima que los costes son más elevados de lo que realmente son.

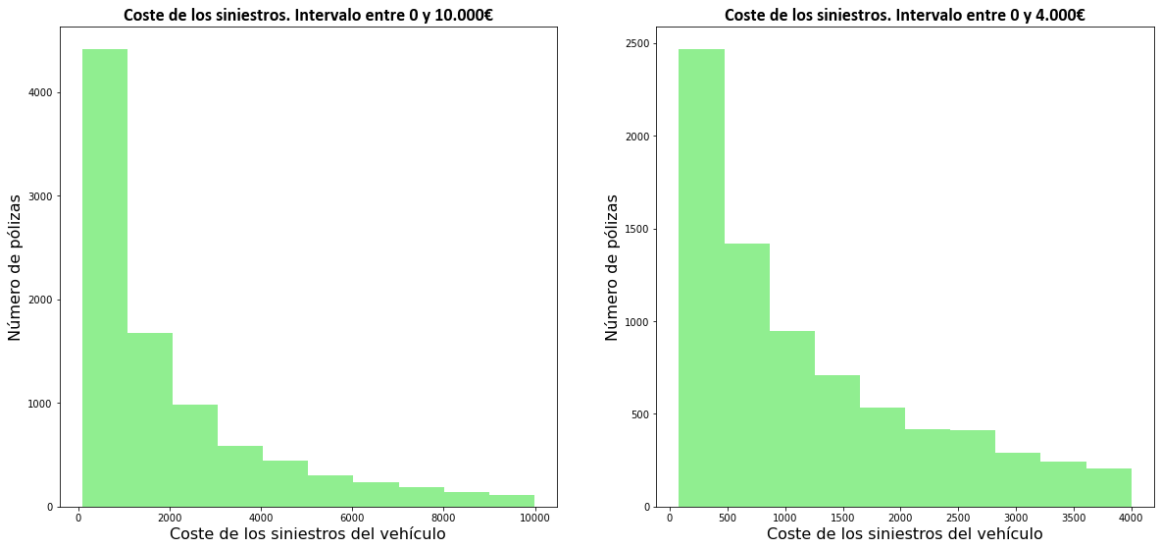
Figura 8. *Escala del coste de los siniestros*



FUENTE: Elaboración propia

Al tratarse de una distribución de cola larga con valores muy lejanos a la media, para poder analizar correctamente la variable mostramos el siguiente gráfico donde eliminamos los valores por encima de 10.000€ y de 4.000€ para observar como se distribuye en estos intervalos el coste. Donde como podemos ver, la mayor parte de los costes se encuentra por debajo de los mil euros.

Figura 9*. Intervalo de costes de siniestralidad*



FUENTE: Elaboración propia

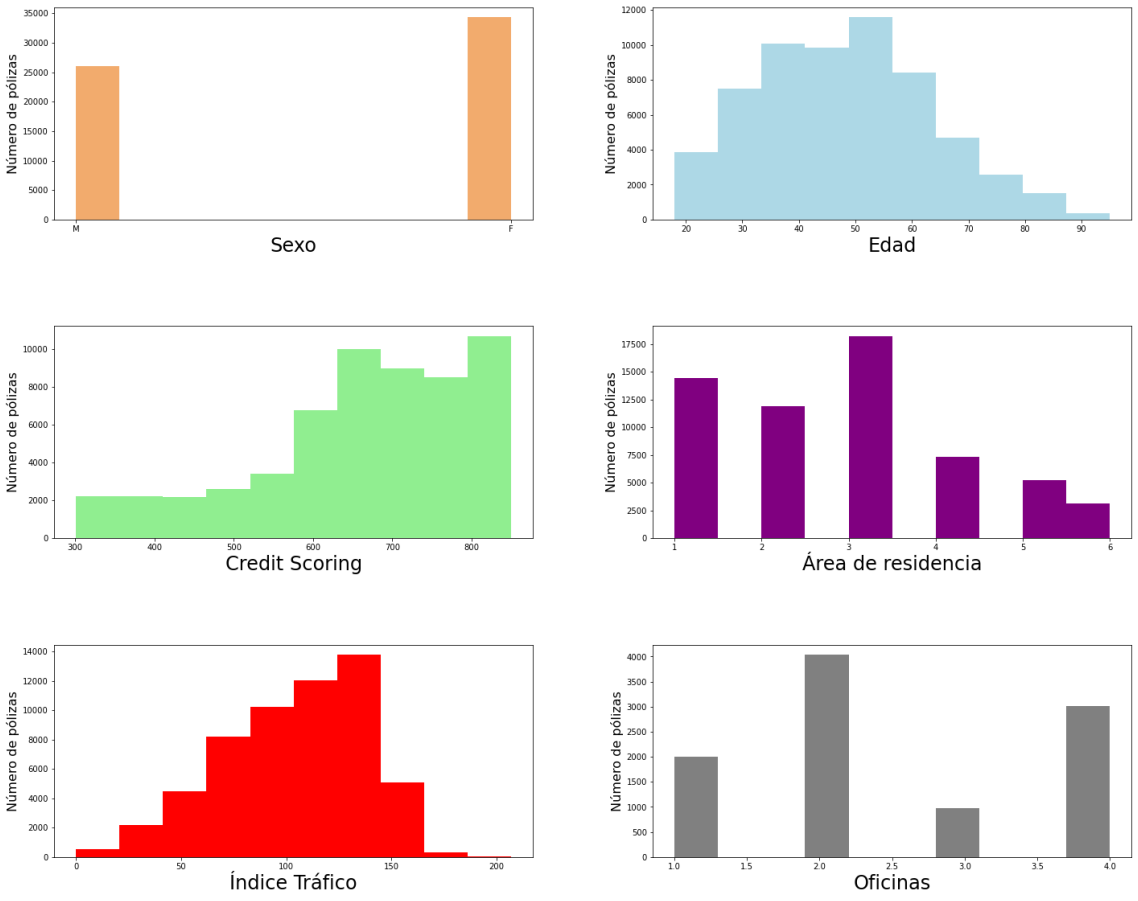
Como podemos ver en el siguiente gráfico, no se exponen las 6 variables que no están relacionadas con los vehículos, si no con el individuo que ha contratado la póliza.

Destacan el mayor número de mujeres y una edad promedio de 48 años, donde hay jóvenes de 18 años y ancianos de 95 años. Respecto a la variable credit Scoring, muestra, como se ha mencionado previamente, la salud crediticia de los individuos. Como podemos ver, gran parte de la población se muestra en la parte derecha, por tanto, no deberían tener problemas de default.

Respecto al área de residencia, la base de datos no muestra si se refiere a ciudades o a comunidades autónomas, pero observando los datos podemos ver que la zona más poblada o donde más asegurados hay es la tercera y el área 6 es donde menor número hay. El índice de tráfico muestra un crecimiento lineal desde cero, las mejores condiciones hasta casi 150 y desde ese punto se aprecia un descenso drástico, siendo el máximo 207. Por último las oficinas es una varible con un gran número de valores vacíos, por tanto es una variable que a priori no puede ofrecernos información de valor, a pesar de ello, el área número 2 es el canal de venta más utilizado. Una posible solución para los valores vacíos sería generar un nuevo área, llamado 5. Otra solución podría ser incluirlos en el área 2 ya que es el de mayor tamaño pero podría no ser correcto ya que la masa de valores vacios es más grande.

De cara al posterior estudio, debemos mencionar que al tratarse de un análisis académico se utilizan variables, como es el sexo de los asegurados, que no pueden utilizarse en el negocio asegurador ya que es discriminatorio

Figura 10. *Distribución de las variables personales*

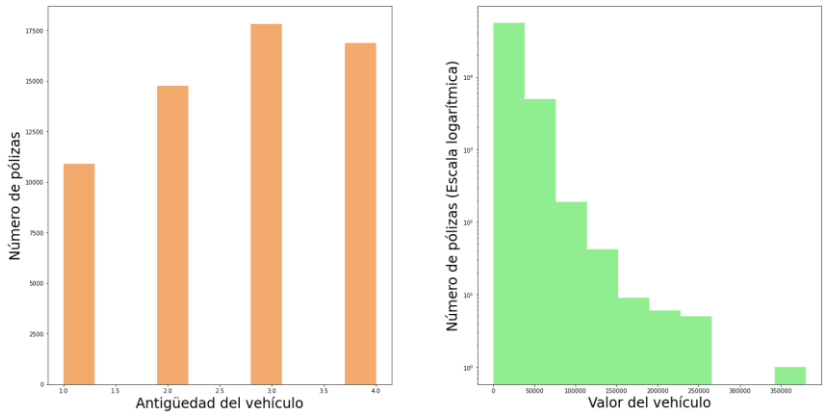


FUENTE: Elaboración propia

Las variables relacionadas con los automóviles son tres: su antigüedad, el valor y el tipo de vehículo. Respecto a la primera de ellas, podemos ver que se divide en cuatro. Si tomamos este valor como años, estaríamos tratando con vehículos muy jóvenes, y lo más lógico sería pensar que estamos analizando una cartera de renting. Otra posibilidad es que la base de datos este compuesta por intervalos, siendo aquellos que tienen un uno los vehículos de menor antigüedad.

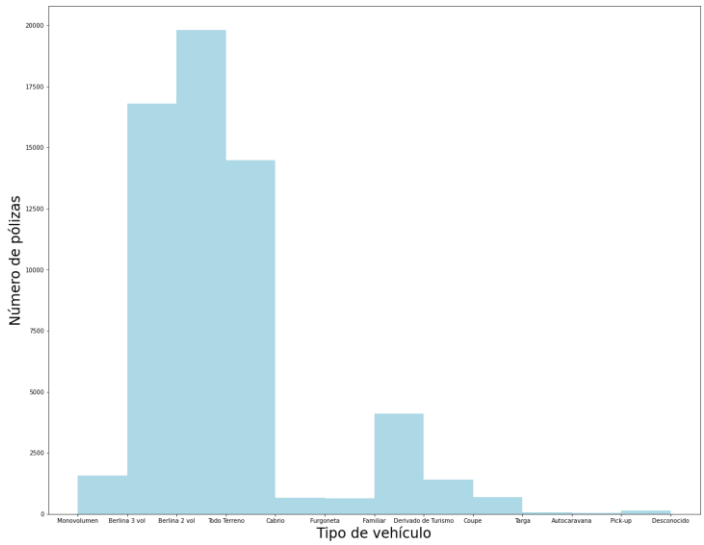
La segunda variable, muestra una gran masa de vehículos de la cartera con un valor inferior a los 20.000€, siendo la media exactamente 19.560€. Se debe mencionar que existen valores punta, teniendo un valor de 380.160€ el coche más caro. Finalmente, la cartera muestra una amplia variación en los tipos de vehículo contando con un total de 9. Encontramos, berlinas de dos y tres volúmenes, todoterrenos, familiares, monovolúmenes, derivados de turismo, coupés, cabrios, furgonetas, pick-ups, targas, autocaravanas y desconocidos, descritos por orden de tamaño en la base de datos, pero no así en el gráfico inferior. Los vehículos desconocidos se agruparán más adelante con otro tipo de vehículo tras analizar variables como la exposición o el coste medio. Aunque al igual que con variables previas podemos encontrar posibles soluciones, como puede ser su inclusión con la mediana, que sería berlinas de 2 volúmenes.

Figura 11. *Variables del vehículo*



FUENTE: Elaboración propia

Figura 12. *Tipo de vehículo*



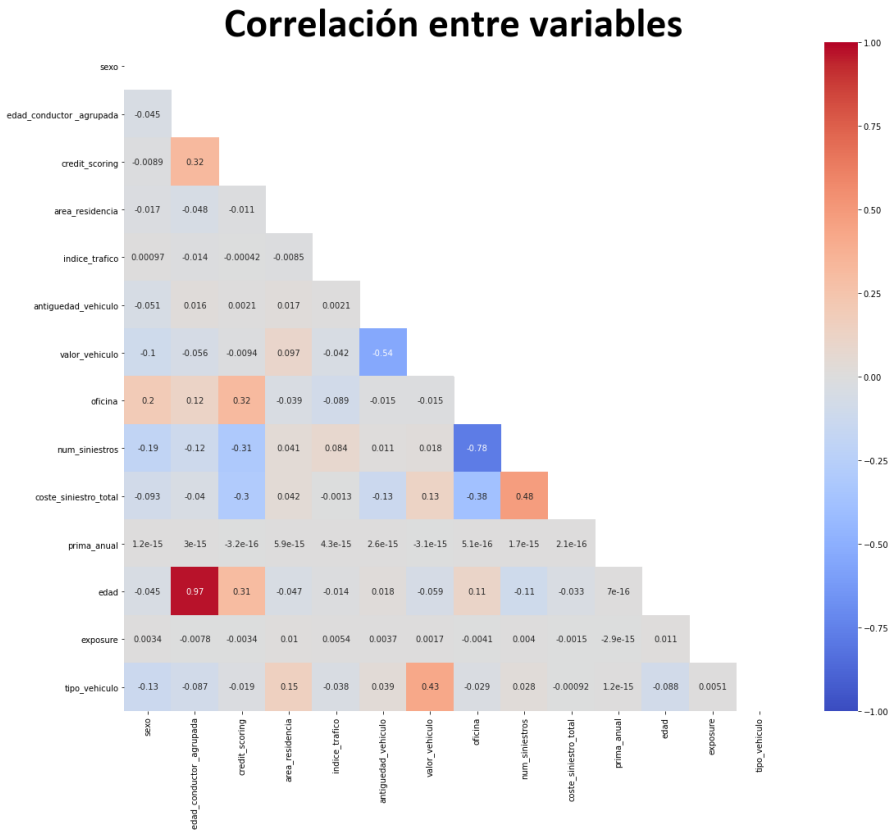
FUENTE: Elaboración propia

Tras analizar cómo es cada variable, para el posterior estudio y creación de modelos, es importante ver la correlación existente entre las variables. Del siguiente gráfico podemos confirmar ciertas ideas. En primer lugar, existen correlaciones que no aportan mucha información, como puede ser entre las variables edad y edad agrupada del conductor (agrupa por décadas la edad del conductor) o las variables número de siniestro y oficina, donde como ya hemos visto, la mayor parte de la variable oficina no tenía registro y ha sido sustituida por el número 5.

El sexo, convertida a variable dummy donde la mujer toma el valor 1 y el hombre el 0, apenas tiene relación con las variables, únicamente destaca el número de siniestros donde las mujeres tienen menor propensión al número de accidentes. Por otro lado, Credit Scoring destaca por tener correlación con la edad, a mayor edad, mayor nivel crediticio y menor riesgo de impago. Y también con el número de siniestros y su coste, siendo aquellos con mayor riesgo de impago los que mayor número de accidentes y coste generan. El área de residencia y el índice de tráfico no muestran apenas relación con el resto de las variables.

En relación con las variables de los vehículos existe gran relación entre ellas, en especial entre el valor del vehículo y su antigüedad (a mayor antigüedad menor es el precio). El tipo de vehículo también está relacionado con su valor, como es de esperar.

Figura 13. *Correlación entre variables*



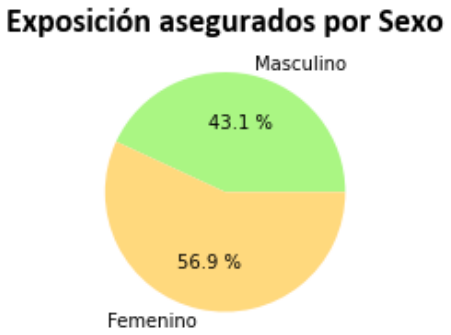
FUENTE: Elaboración propia

3.2 ANÁLISIS DE LA EXPOSICIÓN

La exposición es una variable no incluida en la base de datos pero que es de suma importancia, ya que muestra en forma de ratio el periodo que resta de póliza. De esta forma, utilizamos la exposición como filtro para obtener más información de cada variable. Mediante este estudio, tratamos de completar y ampliar la información de las variables que tratamos.

Comenzamos este análisis por la variable sexo, donde como podemos ver la exposición del género femenino supera al masculino en casi 14 puntos porcentuales. Puede parecer una gran diferencia, pero realmente es muy parejo dado que hay una mayor cantidad de mujeres que de hombres en la cartera que estamos analizando.

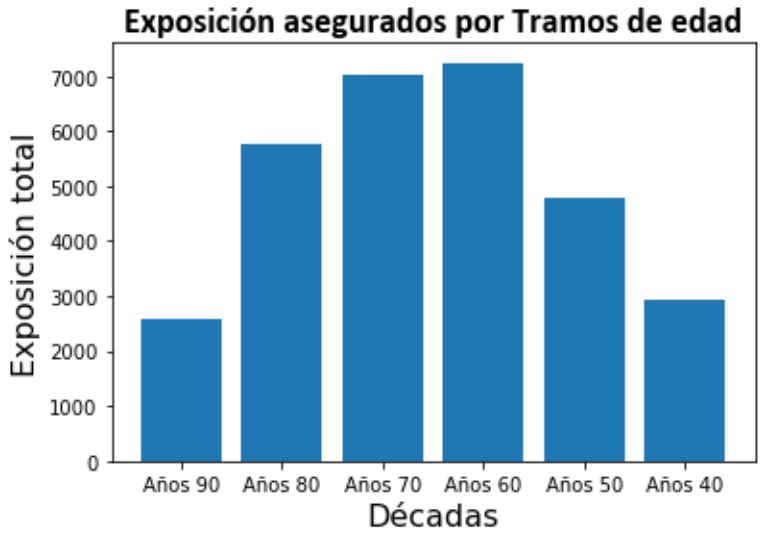
Figura 14. Exposición asegurados. Variable Sexo



FUENTE: Elaboración propia

Decidimos llevar a cabo este análisis sobre la edad mediante la variable edad del conductor agrupada, que únicamente anexiona las edades de los conductores por décadas. Tanto la década de los años 70 como la de los años 60 son las que mayor exposición tiene algo que concuerda con el gráfico previo sobre las edades.

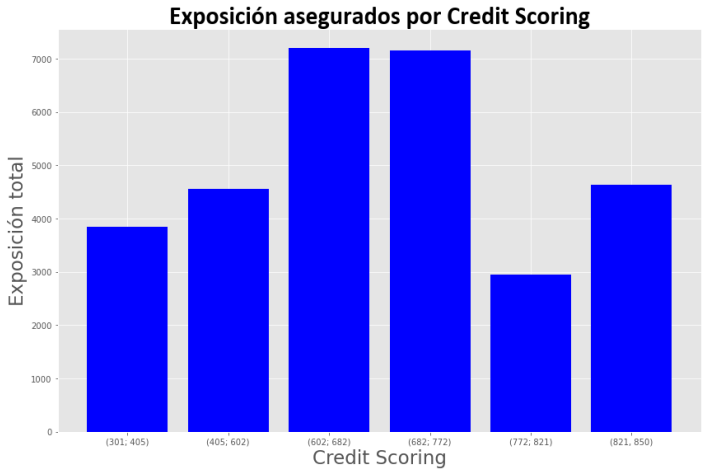
Figura 15. Exposición asegurados. Variable tramos de edad



FUENTE: Elaboración propia

El Credit Scoring, que como hemos visto previamente muestra la capacidad crediticia del cliente destaca enormemente en los valores centrales. Se ha dividido la muestra mediante intervalos. Dichos intervalos son el percentil 10, percentil 25, percentil 50, percentil 75, percentil 90 y finalmente, el 10 por ciento más elevado. La inclusión del percentil 10 y 90 se lleva a cabo para poder analizar los extremos de una manera más cercana. Podemos apreciar que los extremos son muy similares, es decir, el porcentaje de pólizas restante entre aquellos con una alta calidad crediticia y aquellos con una mayor probabilidad de insolvencia es muy similar.

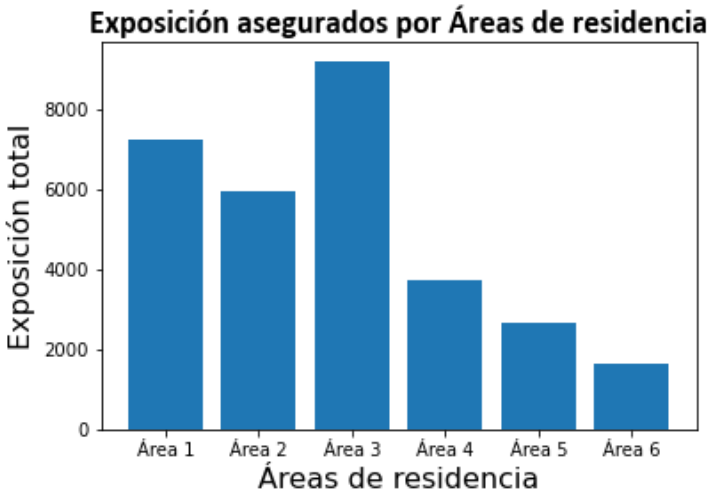
Figura 16. *Exposición asegurados. Variable Credit Scoring*



FUENTE: Elaboración propia

Se puede apreciar que el área con menor población es aquella que tiene una menor exposición, por tanto, en caso de que los habitantes de esa área tengan algún tipo de problema puede afectar en menor medida al resto de la cartera que si ocurre en el área con una mayor exposición, como es el caso del área tres.

Figura 17*. Exposición asegurados. Variable áreas de residencia.*

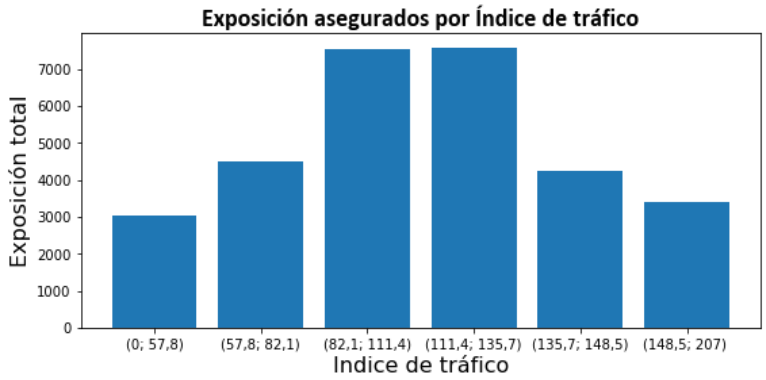


FUENTE: Elaboración propia

Como es de esperar, y observando el gráfico donde definíamos la variable índice de tráfico, obtenemos unos resultados muy similares, donde la exposición más elevada está cerca del 100, que como se expone previamente es la media esperada.

Al igual que en el Credit Scoring se ha llevado a cabo una división por intervalos de la variable de manera que se pueda analizar el dato de una manera más precisa.

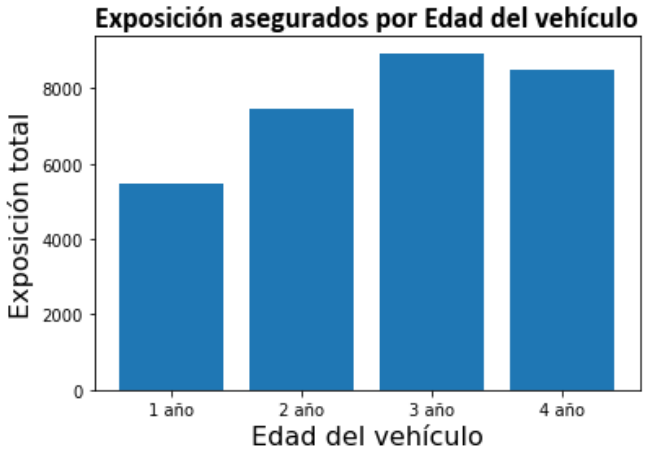
Figura 18. *Exposición asegurados. Variable Índice de tráfico*



FUENTE: Elaboración propia

En este caso observamos algo curioso, ya que hay una mayor exposición en los vehículos con mayor número de años. Esto debe analizarse a continuación junto con los costes medios y totales, ya que los coches más antiguos tienden a romperse con mayor frecuencia, pero no tiene por qué significar un mayor coste.

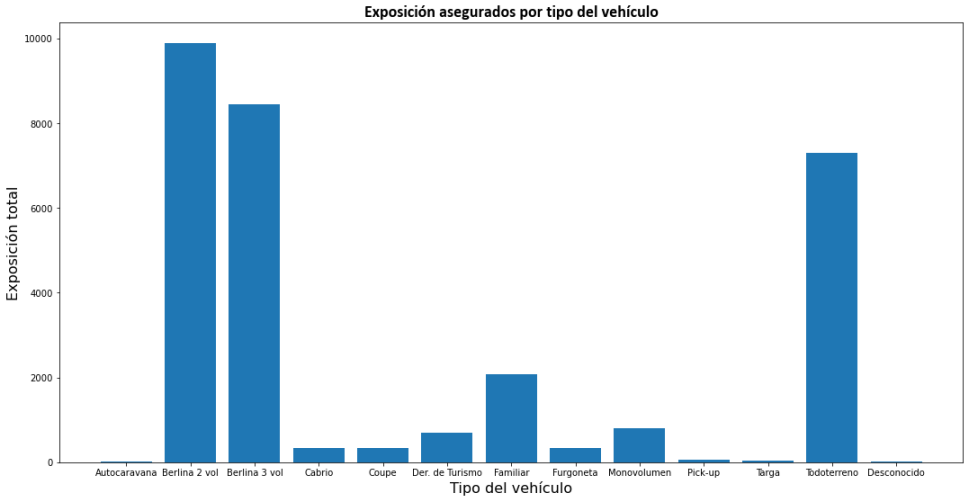
Figura 19. *Exposición asegurados. Variable edad del vehículo*



FUENTE: Elaboración propia

Al igual que en el caso anterior, es importante analizar esta información de forma conjunta con los costes medios y totales, ya que pese a tener una menor exposición en vehículos como pueden ser las furgonetas o coches deportivos, pueden llegar a generar unos costes muy elevados.

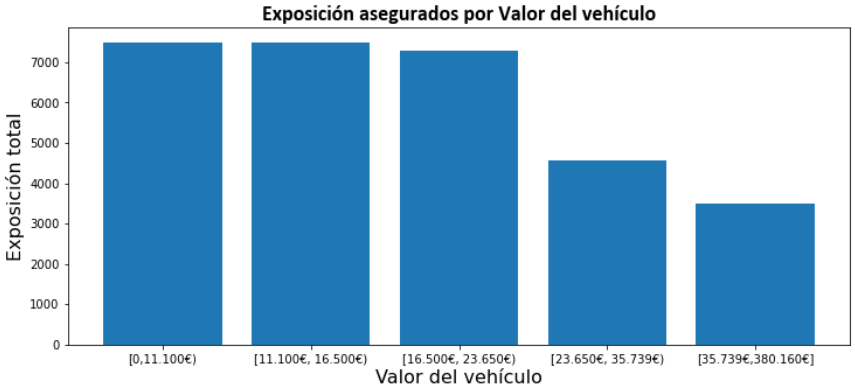
Figura 20. *Exposición asegurados. Variable tipo de vehículo*



FUENTE: Elaboración propia

En este caso, no se ha dividido la muestra de la misma manera, ya que en la parte inferior el valor entre el percentil 10 y 25 apenas variaba. En el otro extremo si que se lleva a cabo de manera que se pueden observar valores mucho más alejados. Respecto al gráfico observamos una disminución paulatina de la exposición desde los valores más pequeños hacia los más grandes.

Figura 21. *Exposición asegurados. Variable valor del vehículo*



FUENTE: Elaboración propia

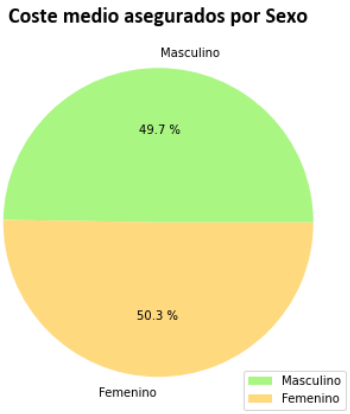
3.3 ANÁLISIS DEL COSTE MEDIO Y TOTAL

Antes de tratar cómo afecta el Coste medio a las variables debemos saber que el coste medio de la cartera es de 645€. Es decir, de media cada vehículo tiene un costo de

reparación por dicho valor. Por otro lado, el coste total es de 38.945.347,26 €. habiendo un total de 12.939 siniestros siendo el más caro de ellos de 69.478€.

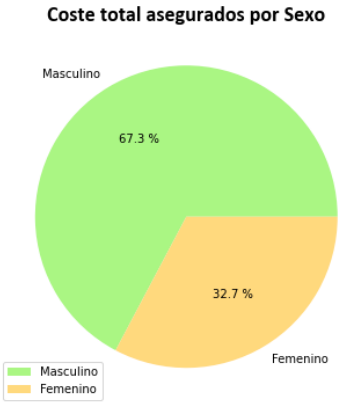
Comenzamos por el sexo, donde cómo podemos apreciar, el género masculino, pese a ser un porcentaje menor, en la cartera tiene un mayor coste total de siniestros, pero menor coste medio, es decir, sus daños son de menor coste, pero tiene una mayor frecuencia.

***Figura 22**. Coste medio asegurados. Variable Sexo*



FUENTE: Elaboración propia

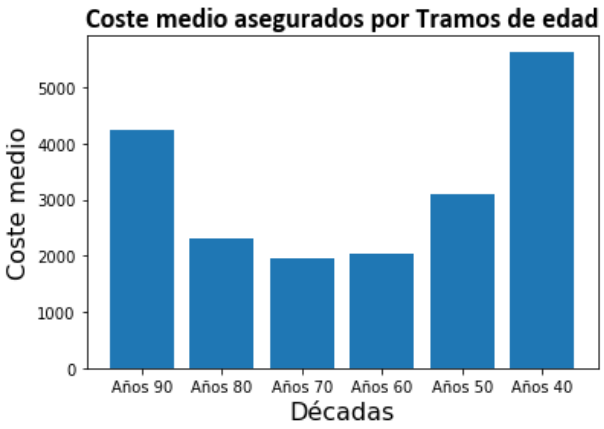
***Figura 23.** Coste total asegurados. Variable Sexo*



FUENTE: Elaboración propia

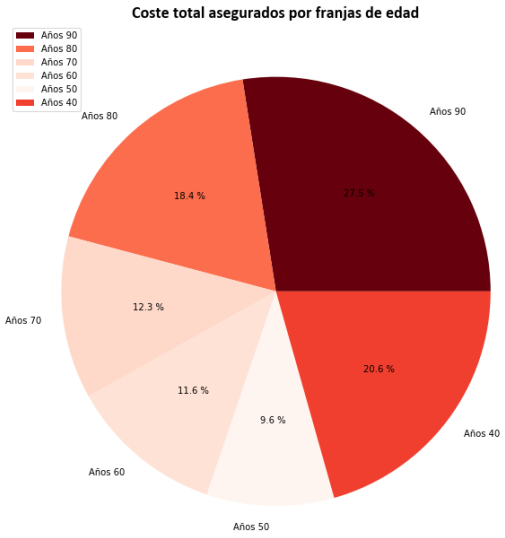
El coste medio más elevado es para las personas nacidas tanto en los años 40 como antes, pero no el coste total ya que son un menor número de personas. El intervalo con un mayor coste total es de los nacidos en la década de los años 90. Es algo previsible dada su juventud y la poca experiencia que tienen, siendo los segundos con un mayor coste medio. Aquellos individuos nacidos en los años 60 son los que menor coste total como menor coste medio generan.

***Figura 24**. Coste medio asegurados. Variable tramos de edad*



FUENTE: Elaboración propia

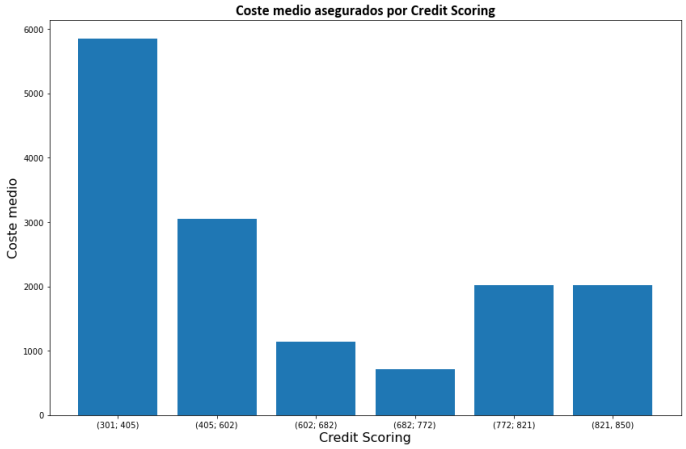
***Figura 25.** Coste total asegurados. Variable tramos de edad*



FUENTE: Elaboración propia

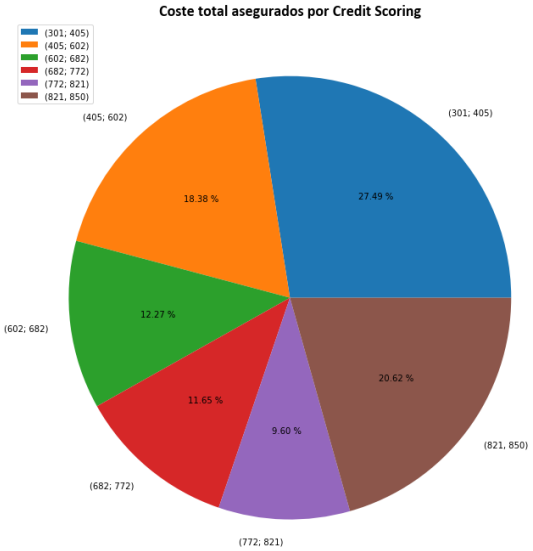
La tercera variable es el credit Scoring, que muestra el riesgo crediticio de los clientes, donde podemos apreciar que aquellos con un mayor riesgo a la quiebra financiera mayor coste medio tienen y mayor coste total generan (27.5%). Por el lado contrario, los que tienen menor riesgo crediticio tienen un coste medio menor pero son los segundos con mayor coste total (20.6%), posiblemente se deba a que tienen vehículos de alta gama, por lo que tienen pocos accidentes pero de un mayor coste.

Figura 26. *Coste medio asegurados. Variable Credit Scoring*



FUENTE: Elaboración propia

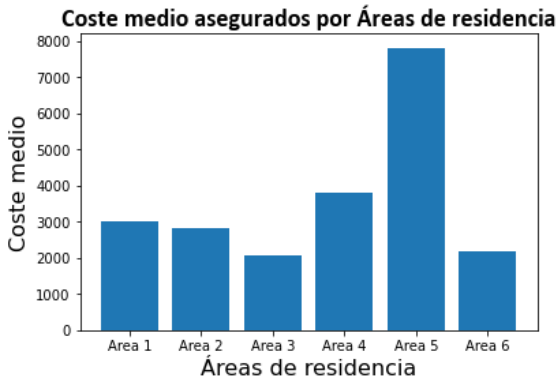
Figura 27. *Coste total asegurados. Variable Credit Scoring*



FUENTE: Elaboración propia

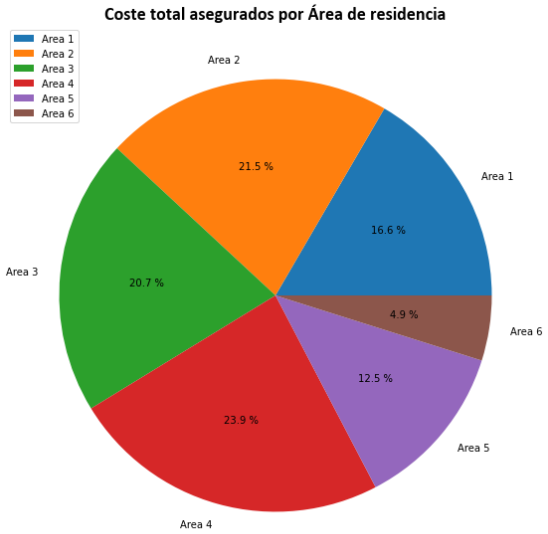
El área de residencia más poblada es la tercera con más de 18.000 individuos, pero como podemos apreciar son los que menor coste medio tienen y los terceros en coste total, es decir, tienen siniestros poco costosos. Observamos un coste medio muy elevado en la quinta área respecto al resto de zonas lo que acaba generando un coste total superior al 1 por ciento, aunque su población sea únicamente el 8.3 por ciento de la cartera. Analizando el gráfico acecra de los costes totales observamos que los costes totales de la cuarta zona son los más elevados pese a ser una localidad con menos de la mitad de habitantes en la cartera que la tercera área.

Figura 28. *Coste medio asegurados. Variable Área de residencia*



FUENTE: Elaboración propia

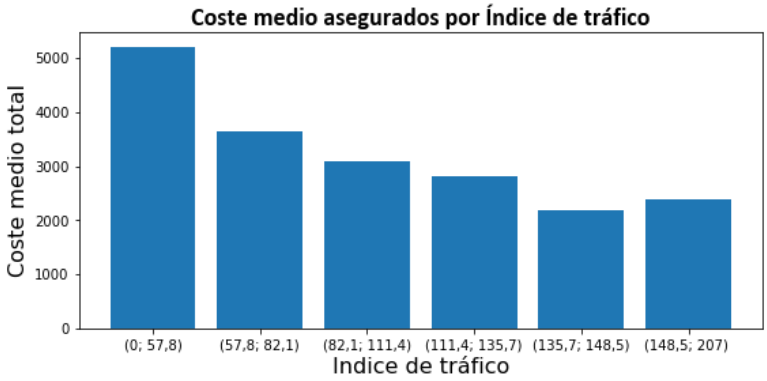
Figura 29. *Coste total. variable Área de residencia*



FUENTE: Elaboración propia

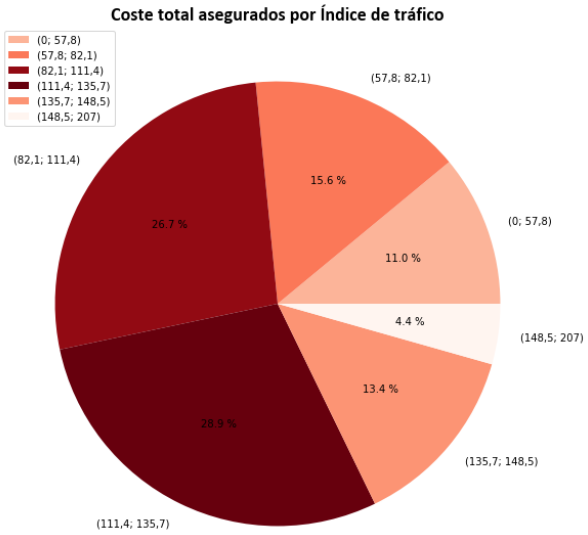
Como ya se ha explicado previamente la división de esta variable es por cuantiles, siendo 0 las mejores condiciones de tráfico. El coste medio del cuantil 10 es muy elevado pero su coste total es el segundo más pequeño, de aquí podemos dilucidar que el número de accidentes es pequeño, pero estos son costosos. A medida que empeoran las condiciones de tráfico el coste medio disminuye, salvo en el último intervalo, donde se encuentran las peores condiciones. El segundo y tercer intervalo, aquellos que están cerca de la media, superan el 50 por ciento del coste total, pese a ser únicamente el 40 por ciento de los individuos de la póliza.

Figura 30. *Coste medio asegurados. variable Índice de tráfico*



FUENTE: Elaboración propia

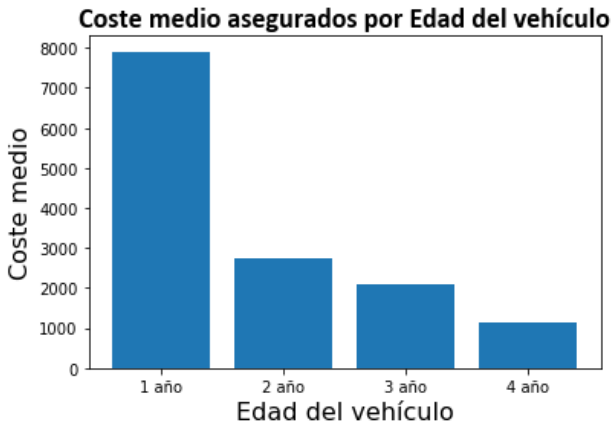
Figura 31. *Coste total asegurados. variable Índice de tráfico*



FUENTE: Elaboración propia

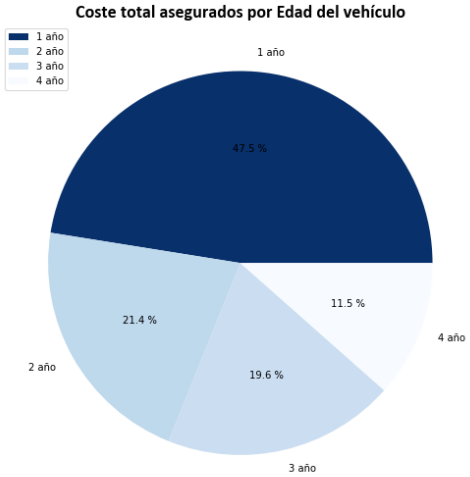
Debemos mencionar la gran diferencia tanto en coste medio como en coste total de los vehículos del primer intervalo respecto a los demás. Es lógico pensar que los vehículos más jóvenes puedan tener unos costes medios más elevados debido a la tecnología que puedan albergar, pero el hecho de que los costes totales lleguen a casi el 50 por ciento muestra una severidad muy elevada de estos vehículos, además únicamente son un quinto de la cartera. En el lado opuesto, los vehículos más antiguos, muestran un coste medio cercano a los 1.000€ y un coste total de únicamente el 11.5% del total.

***Figura 32**. Coste medio asegurados. Variable Edad del vehículo.*



FUENTE: Elaboración propia

***Figura 33**. Coste total asegurados. Variable Edad del vehículo*



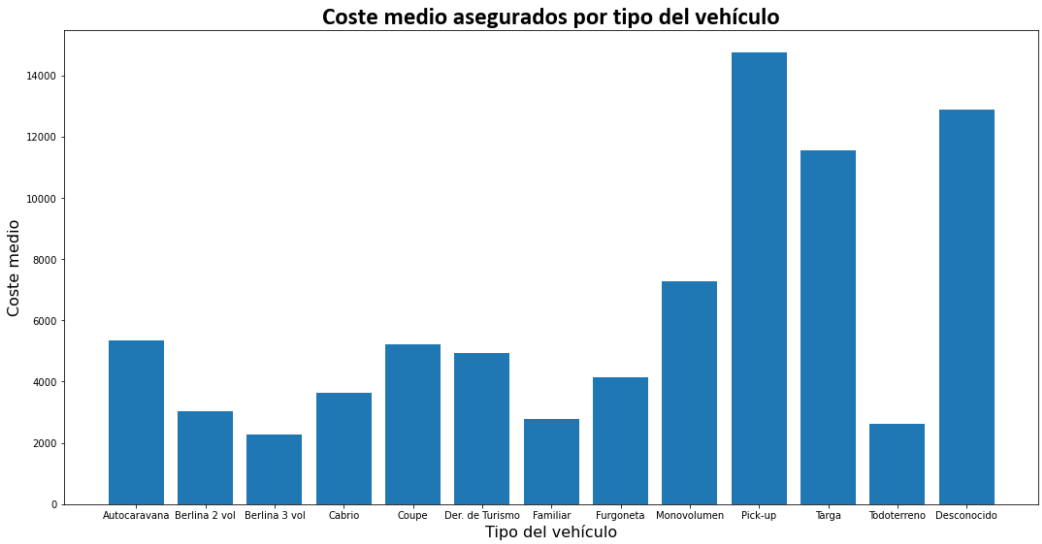
FUENTE: Elaboración propia

El Coste medio varía mucho entre los distintos tipos de vehículo. El más elevado son las Pick-up, que únicamente representan el 1% del coste total, pero muestran que cada siniestro es de un coste elevado. Lo mismo ocurre con los vehículos Targa, que al ser coches deportivos tienen un coste medio de siniestro muy elevado. Los vehículos 'desconocidos', es decir, aquellos de los que se desconoce el tipo de vehículo también tienen un coste medio muy elevado, pero como su masa es muy pequeña apenas son el 0.23% del coste total.

Las Berlinas de 2 volúmenes son los que mayor coste generan en la cartera, pero a su vez se aprecia un coste medio pequeño similar a otros vehículos como las berlinas de 3 volúmenes, a los familiares o los todoterrenos, es decir, la mayoría de los vehículos de la cartera tienen un coste medio cercano a los 2.000€ cuando tienen un siniestro. Los

automóviles todoterrenos y berlinas de 3 volúmenes son los siguientes con un mayor coste total.

Figura 34. *Coste medio asegurados. variable Tipo de vehículo*



FUENTE: Elaboración propia

Tabla 5. *Coste total asegurados. variable tipo de vehículo*

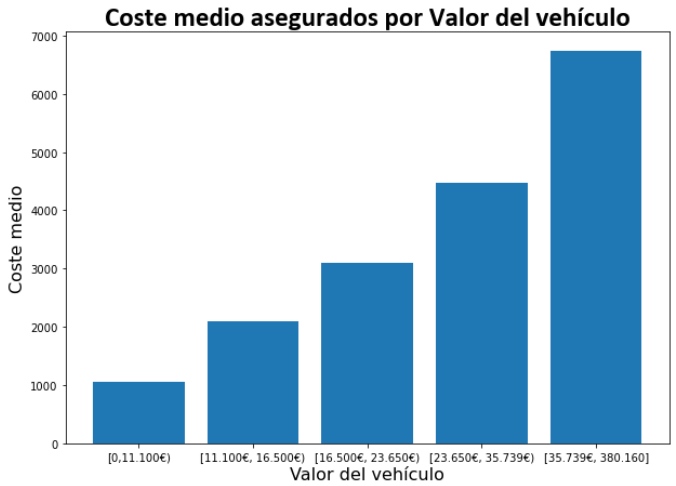
Tipo de vehículo	Coste total	Porcentaje
Berlina 2 vol	11.381.330	29,22%
Todo Terreno	8.802.568	22,60%
Berlina 3 vol	7.330.967	18,82%
Monovolumen	3.657.888	9,39%
Familiar	3.563.090	9,15%
Derivado de Turismo	1.621.048	4,16%
Coupe	890.539	2,29%
Cabrio	633.715	1,63%
Furgoneta	398.100	1,02%
Pick-up	339.213	0,87%
Targa	161.702	0,42%
Desconocido	90.205	0,23%
Autocaravana	74.980	0,19%
Total	**38.945.347**	**100,00%**

FUENTE: Elaboración propia

Al igual que en casos anteriores se ha dividido la muestra en intervalos por cuantiles. El primer cuantil, donde los vehículos son más baratos el coste medio de los siniestros es

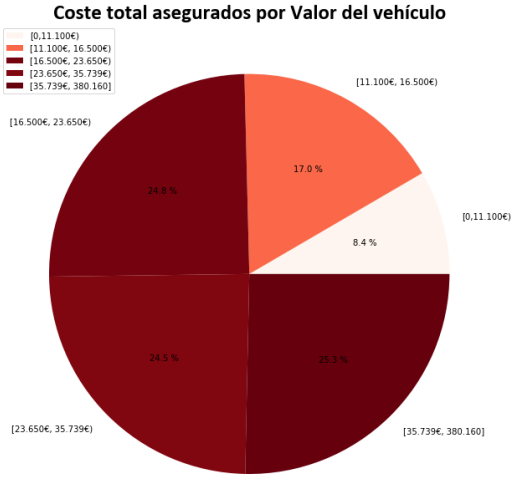
muy inferior al resto de intervalos, al igual que el coste total. Esto puede entenderse ya que los automóviles más baratos son más económicos de arreglar. Por ello a medida que aumenta el valor de estos, su coste medio aumenta progresivamente siendo los tres últimos los que mayor coste total generan. La cartera tiene 6.028 vehículos con un valor superior a los 35.739€, es decir, un 10 por ciento de la masa total que genera el 25.3% de los costes totales.

Figura 35. *Coste medio asegurados. Variable Valor del vehículo*



FUENTE: Elaboración propia

Figura 36. *Coste total asegurados. Variable valor del vehículo*



FUENTE: Elaboración propia

3.4 ANÁLISIS DE VARIABLES

En este apartado voy a analizar las variables que incluían valores vacíos o incorrectos. Como hemos visto previamente algunas de estas variables ya han sido modificadas como puede ser el caso del índice de tráfico, donde aquellas pólizas que no contenían esta información se han incluido en la moda de la variable. Otra variable modificada es la oficina de venta, donde la gran mayoría de valores se encontraban vacíos. En este caso se ha generado un nuevo valor, el 5, indicando un nuevo canal de venta.

Credit Scoring y tipo de vehículo son las dos variables restantes que contienen valores vacíos o incorrectos, como es el caso de la segunda variable con el tipo de vehículo 'Desconocido'. En ambos casos se ha optado por sustituir dicha información por la moda de cada variable. Analizando las características de estas, se ha observado que la exposición era muy similar y por tanto no era concluyente. Respecto al coste medio tampoco se encontró nada concluyente por lo que los 2.801 valores pasan a tomar el mismo valor de la moda.

Los vehículos 'Desconocidos' únicamente son 22, por ello la muestra no es representativa. Pese a ello se ha llevado a cabo un estudio sobre su coste medio encontrándose entre los automóviles Targa y Pick-up. Al igual que ocurre con su valor, pero a pesar de ello se estima que lo más conveniente ante la falta de masa es considerarlos como el vehículo que más abunda en la cartera, berlina de 2 volúmenes.

Tabla 6. *Estudio vehículos desconocidos*

Tipo de vehículo	Coste Medio	Valor medio del vehículo
Desconocido	12.886	54.019,00
Targa	11.550	84.532,00
Pick-up	14.748	30.707,00

FUENTE: Elaboración propia

4. CREACIÓN DE MODELOS

Después de analizar y modificar las variables que conforman la base de datos, en este apartado voy a exponer las técnicas de tarificación que se van a aplicar a lo largo del estudio. Además, mostraré las posibles variaciones que se realizarán a las variables con el fin de obtener una modelización lo más correcta posible. Finalmente, al término de cada apartado, se verán los distintos resultados que se vayan obteniendo.

4.1 MODELO GLM

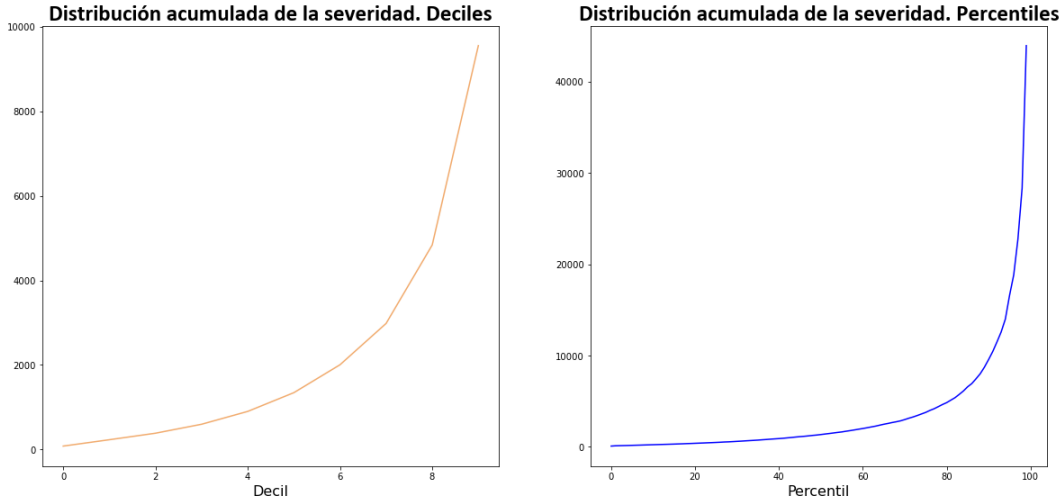
Como se ha expresado previamente la base de datos se divide en dos partes, por un lado, la muestra de entrenamiento (80% de la muestra total) y, por otro lado, la muestra de comprobación (20% de la muestra total). De esta manera, la primera muestra cuenta con un total de 48.314 datos y la segunda con 12.078.

Existe la posibilidad de que haya valores punta en los costes del siniestro, es decir, siniestros poco comunes de un alto coste. Este tipo de siniestros pueden deberse a diversas razones como puede ser el siniestro de un coche de alta gama, un atropello o asegurados con una severidad muy elevada. Estos valores atípicos pueden generar que la cola de la distribución sea mucho más alargada generando unos resultados en la modelización del coste de la siniestralidad más abultados de lo esperado.

Para realizar una estimación adecuada, tanto para la frecuencia como para la severidad estos valores serán eliminados, y finalmente en apartados posteriores se analizará como deben tratarse de cara al negocio asegurador. El problema que generan estos datos en la modelización es principalmente la transformación de la distribución de las variables generando valores mucho más elevados del coste. En caso de incluir los datos, la cola derecha de la distribución es mucho más pronunciada ya que existen valores muy alejados del resto.

Para analizar este hecho se muestran los siguientes gráficos. El primero de ellos divide los distintos costes por deciles y el segundo por percentiles obteniendo un resultado mucho más preciso. Como podemos apreciar en el primer gráfico a partir del séptimo decil comienza un incremento agudo de los costes. En el gráfico derecho observamos el mismo crecimiento siendo un crecimiento vertical en los últimos percentiles.

***Figura 37**. Distribución de costes de la cartera*

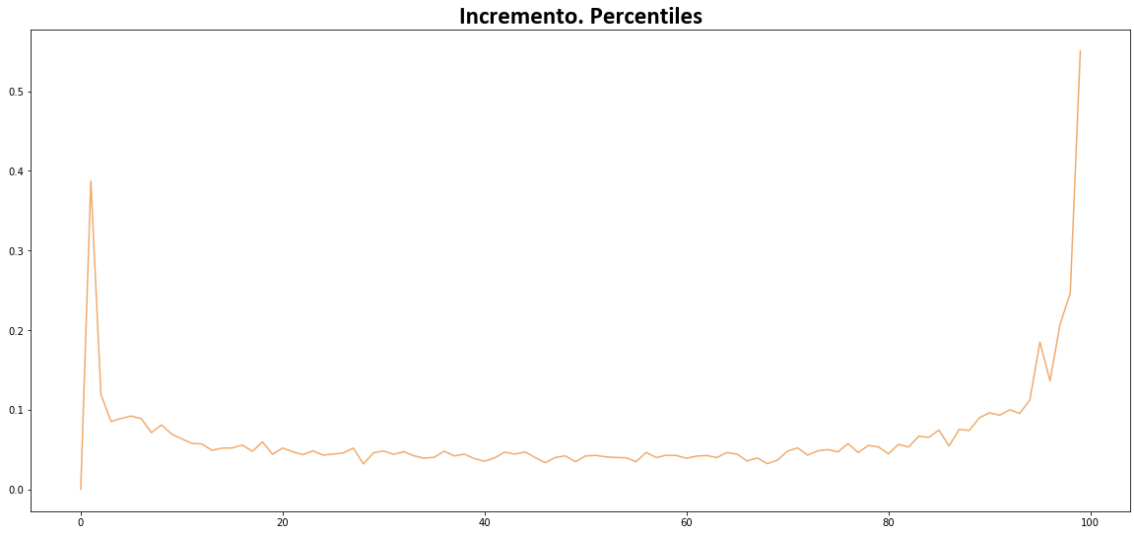


FUENTE: Elaboración propia

De esta manera hemos comprobado que efectivamente hay valores atípicos. El siguiente paso que debemos hacer es saber qué punto es el más adecuado para cortar. Para ello realizamos incrementos de los percentiles. En el primer gráfico debemos analizar la parte de la derecha donde comienza un aumento exponencial de los incrementos de los percentiles, ese es el punto donde debemos eliminar los costes de cara a la modelización. En el segundo gráfico apreciamos el resultado tras eliminar los últimos cuatro percentiles de los siniestros, un total de 506 pólizas que respecto el total de 60.392 es el 0,8% de la muestra que generaba un total de 15.337.557,68€ de costes.

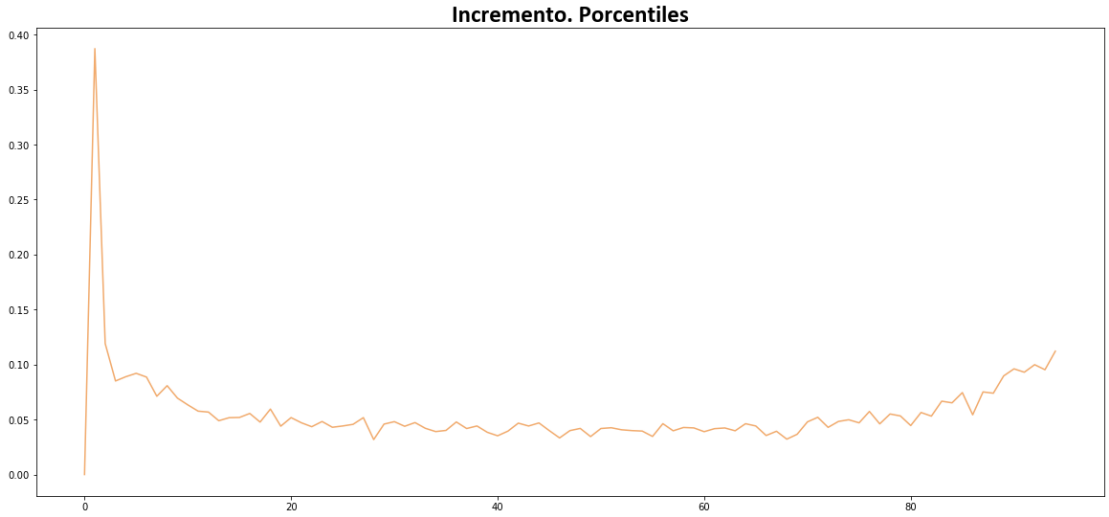
.

***Figura 38.** Incremento por percentiles del coste*



FUENTE: Elaboración propia

***Figura 39**. Incremento por percentiles sin valores punta*



FUENTE: Elaboración propia

De esta manera, tras eliminar los valores que podrían generar sesgo a lo largo del estudio, nos quedamos con una muestra total de 59.886 pólizas, es decir, 47.912 para el estudio de entrenamiento y 11.974 para el test.

4.1.1 MODELO GLM PARA LA FRECUENCIA

La creación de este modelo se hace en función del número de siniestros. Se ha utilizado la función GLM perteneciente a la herramienta de Statsmodel de Python. El proceso llevado a cabo es la inclusión de variables paso a paso, considerando su significatividad y en caso contrario, analizar una posible modificación de esta para tratar de obtener un resultado positivo. De esta forma obtendríamos el modelo univariante final. Ajustaremos la variable independiente a una distribución Poisson con observaciones ponderadas, además utilizaremos la función log como función de enlace.

En la siguiente tabla mostramos el primer modelo como ejemplo. La tabla tiene dos partes diferenciadas, la primera de ellas muestra las características del modelo, la variable dependiente, que en este caso es el número de siniestros, el modelo utilizado, GLM, la familia del modelo, el número de observaciones, los grados de libertad, la desviación, etc.

La segunda parte son los resultados obtenidos, donde en primer lugar, aparece el intercepto y las variables utilizadas, sus coeficientes, sus errores estándar, su nivel de significatividad (que como podemos ver en este caso, la variable es significativa), y el intervalo de confianza.

$$Número\ de\ siniestros = \beta_0 + \beta_1 * Sexo \qquad (12)$$

Tabla 7. *Ejemplo de modelización GLM.*

Generalized linear model results

Dep. Variable	num_siniestro_total	No. Observations	47912			
Model:	GLM	Df Residuals	47910			
Model Family:	Poisson	Df model	1			
Link Function:	log	Scale	1			
Method:	IRLS	Log-Likelihood:	-28047			
Date:	Sun, 22 May 2022	Deviance:	39883			
Time:	18:28:14	Pearson chi2:	1,81E+05			
No. Iterations	20					
Covariance Type	nonrobust					

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-1,4885	0.018	-82.700	0,000	-1.524	-1.453
C(sexo,Treatment(reference="F"))[T.M]	1,0158	0.022	46.433	0,000	0.973	1.059

FUENTE: Elaboración propia

Un aspecto importante que se debe mencionar en estos modelos es el valor de referencia utilizado en variables categóricas. En este caso el valor de referencia es el sexo femenino, por tanto, si estuviésemos analizando a una mujer el coeficiente sería 0. Por el contrario, si estuviésemos analizando el número de siniestros de un hombre habría que tener en cuenta el coeficiente ($\beta_1 = 1.0158$), es decir, este modelo muestra una mayor frecuencia para los hombres que para las mujeres.

La significatividad de las variables es muy importante ya que aportan valor a la modelización. En el caso de tener un modelo con una variable que no es significativa es preferible eliminarla ya que puede estar afectando a otras variables obteniendo resultados no deseados. Para saber si una variable es válida debemos fijarnos la cuarta columna de la segunda parte de la tabla. Hay distintos niveles de significatividad, pero siempre se tratará de obtener resultados menores a 0.001 o en su defecto a 0.005.

Este modelo es muy sencillo y alberga gran sesgo por ello debemos mejorarlo incluyendo un mayor número de variables, pero a pesar de ello la intuición es correcta, ya que en promedio en esta cartera los hombres duplican la frecuencia de accidentes de las mujeres.

A medida que se van realizando diversos modelos deben compararse para saber cuál de ellos es preferible, para ello utilizamos el criterio AIC de Akaike y el criterio BIC desde el enfoque bayesiano. El primer criterio, mediante el cálculo mostrado en la siguiente ecuación escoge el mejor modelo siendo aquel que maximiza la verosimilitud esperada (L). Mediante este método se consigue una penalización sobre los modelos con una mayor número de variables (k), para así tratar de evitar el sobreajuste, también llamado 'overfitting' en inglés.

$$AIC = 2 * k + 2 * \ln(L) \qquad (13)$$

El segundo criterio, creado por Schwarz, escoge como mejor modelo a aquel que maximice la probabilidad a posteriori suponiendo que las probabilidades a priori son iguales para todos los modelos. Al igual que en el caso anterior, esta alternativa para comparar modelos tiene una penalización para evitar el sobreajuste mayor al caso previo, ya que no se multiplica por 2 sino por el logaritmo neperiano de la muestra (n). La k es el número de parámetros, Ln (L) es la función de log-verosimilitud.

$$BIC = \ln(n) * k + 2 * \ln(L) \quad (14)$$

Mediante estos criterios se trata de conseguir el equilibrio entre la complejidad de los modelos con la inclusión de variables y el rendimiento que éstos son capaces de aportar. El mejor resultado es el menor, en este primer modelo el AIC y BIC obtenidos son 56.098, y -476.448. En los modelos siguientes se espera que los resultados sean menores y, por tanto, mejores.

Con el fin de obtener los mejores resultados se han llevado a cabo diversos modelos, comparándose entre ellos mediante los criterios explicados previamente. En el modelo univariante escogido las variables incluidas son:

- **Sexo**: Al tratarse de un estudio académico incluimos la variable, tomando como valor base el género femenino.
- **Edad**: Variable no incluida en la base de datos inicial pero creada a partir de la fecha de nacimiento.
- **Edad_mayores70**: Variable dicotómica que toma valor 1 para los individuos con más de 70 años y 0 para el resto.
- **Credit_scoring_mod**: Se divide la variable inicial en intervalos de 50 en 50. Comenzando en 350 y acabando en 750.
- **Credit_scoring_mod2**: Variación de la variable inicial. Toma valor 0 para los individuos con mayor riesgo de impago, toma valor 1 para aquellos con riesgo medio y 2 para aquellos con menores problemas económicos.
- **Área_residencia_23_46**: Fusiona las ciudades 2 y 3 y las ciudades 4 y 6.
- Índice de tráfico modificado: esta variable se ha segmentado según los cuantiles calculados y explicados previamente.
- **Valor_vehiculo**: Se utiliza la variable inicial.

Tabla 8. *Modelo de la frecuencia GLM*

Generalized linear model results

Dep. Variable	num_siniestro_total	No. Observations	47912
Model:	GLM	Df Residuals	47910
Model Family:	Poisson	Df model	10
Link Function:	log	Scale	1
Method:	IRLS	Log-Likelihood:	-25404
Date:	Sun, 22 May 2022	Deviance:	34597
Time:	18:28:14	Pearson chi2:	1,68E+05
No. Iterations	6		
Covariance Type	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	1,1336	0,113	10	0,000	0,911	1.356
C(sexo,Treatment(reference="F"))[T.M]	1,0610	0,022	48	0,000	1.018	1.104
C(edad_mayores70,Treatment(reference=0))[T.1]	0,5903	0,05	12	0,000	0,491	0,689
C(area_residencia_23_46,Treatment(reference=3))[T.1]	-0,3428	0,033	-10	0,000	-0,407	-0,278
C(area_residencia_23_46,Treatment(reference=3))[T.4]	0,3179	0,025	13	0,000	0,268	0,368
C(area_residencia_23_46,Treatment(reference=3))[T.5]	-0,651	0,059	-11	0,000	-0,766	-0,536
edad	-0,0182	0,001	-21	0,000	-0,02	-0,017
credit_scoring_mod	-0,0029	0	-13	0,000	-0,003	-0,002
credit_scoring_mod2	-0,243	0,038	-6	0,000	-0,318	-0,168
indice_trafico_mod	0,0033	0,000	9	0,000	0,003	0,004
valor_vehiculo	-4,69E-03	8,43E-07	-6	0,000	-6,34E-06	-3,04E-06

FUENTE: Elaboración propia

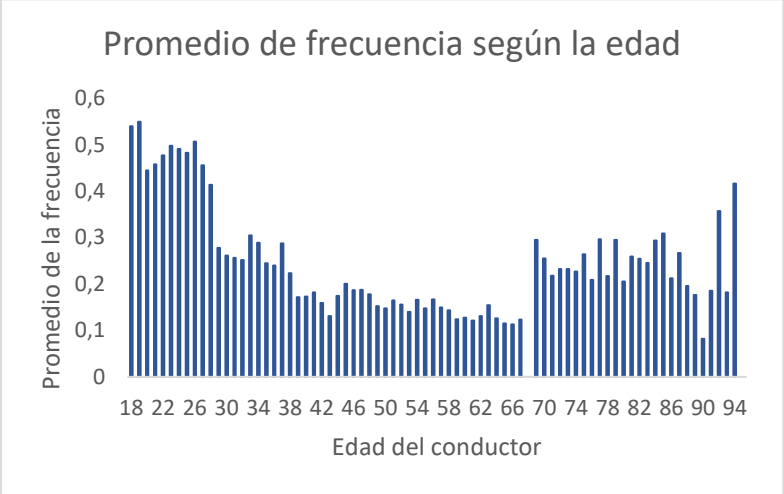
Como hemos explicado con el ejemplo del primer modelo, debemos fijarnos en la significatividad de las variables. Todas y cada una de ellas son significativas y, por tanto, válidas para el estudio.

Se debe realizar un breve análisis ex-post para saber si las variables mantienen la intuición esperada o no. La primera variable, el sexo, muestra un coeficiente positivo, es decir, en el caso de pertenecer al género masculino, el modelo muestra una mayor frecuencia. Si comparamos esta intuición con el análisis ex-ante comprobamos que coincide el análisis ya que los hombres en esta cartera tienden a tener un mayor número de siniestros.

La edad tiene dos variables, la primera de ellas muestra la edad de los asegurados y la segunda es una variable dicotómica que analiza el efecto de los mayores de 70 años. El coeficiente es negativo para la edad, es decir, a medida que eres mayor hay menos propensión a tener un siniestro. Al observar el gráfico siguiente observamos cierta tendencia ascendente para los individuos de mayor edad. Incluimos la segunda variable para capturar dicho incremento. Observamos un coeficiente positivo, por tanto, está capturando bien la tendencia.

Existe un pequeño repunte para los conductores entre 50 y 60 años, una de las razones por las que esto puede ocurrir es que es el momento en el que sus hijos obtienen el carné de conducir y por ello comienzan a tener más siniestros. Respecto al crecimiento tras la edad de jubilación el incremento se debe principalmente a la disminución de habilidades de los conductores.

Figura 40. *Promedio frecuencia. Variable edad.*

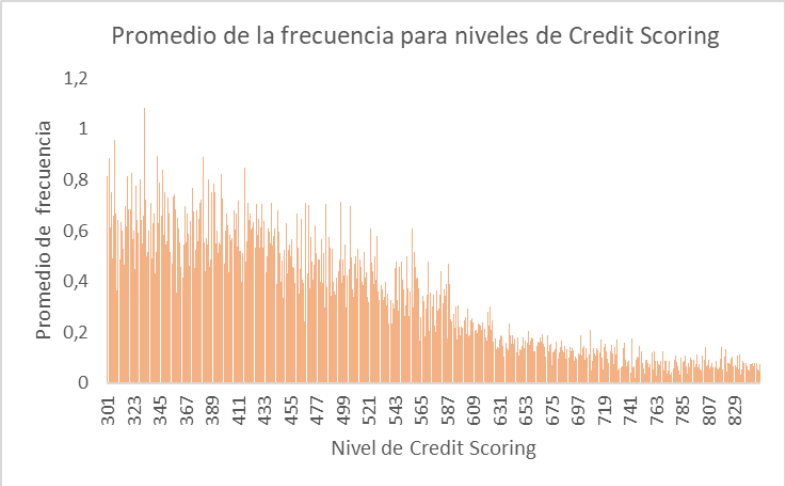


FUENTE: Elaboración propia

La siguiente variable, también parametrizada en dos variables, es el riesgo crediticio de los asegurados. En la siguiente figura se aprecia un descenso continuado de la frecuencia media a medida que disminuye el riesgo de impago, con la primera variable tratamos de segmentar dicha tendencia y otorgando distintos valores dependiendo del nivel crediticio. El coeficiente es negativo, al igual que la pendiente, por tanto, podemos afirmar que la intuición es correcta, a mayor nivel crediticio menor probabilidad de sufrir un accidente.

La idea de incluir la segunda variable subyace en otorgar mayor importancia a aquellos individuos con distintos niveles de riesgo crediticio. De esta manera, aquellos con un buen riesgo tienen un segundo coeficiente negativo ya que como podemos ver en la figura hay gran diferencia entre los extremos.

Figura 41*. Promedio de la frecuencia. Variable Credit Scoring*



FUENTE: Elaboración propia

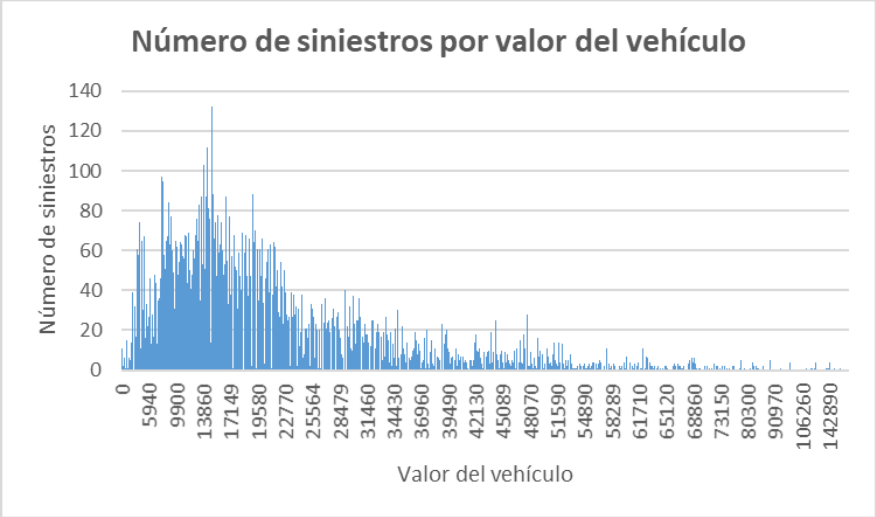
La siguiente variable es el área de residencia. Para llevar a cabo la decisión de unir las ciudades 2 y 3 y 4 y 6 hemos observado el promedio de la frecuencia de accidentes. Como podemos ver en la tabla tienen una frecuencia similar entre ellas. Tomamos como base las ciudades 2 y 3, por tanto, para tener una buena modelización, las áreas 1 y 5 deberán tener un coeficiente negativo y la unión 4 y 6 positivo. Efectivamente este es el resultado obtenido. Esta unión se ha llevado a cabo ante la falta de significatividad al incluir las ciudades de manera independiente.

Tabla 9. *Promedio de la frecuencia. Variable Área de residencia*

Área de residencia	Promedio frecuencia
1	14,75%
2	24,81%
3	21,45%
4	33,41%
5	11,79%
6	27,43%
Promedio general	21,43%

La penúltima variable del modelo es el valor del vehículo. Observamos un coeficiente negativo, es decir, en caso de tener un vehículo de mayor gama, la probabilidad de sufrir un siniestro es menor, aunque como podemos apreciar es un número muy cercano a 0. En el gráfico observamos el número de accidentes dependiendo del valor del vehículo, y aunque es cierto que la mayor parte de la muestra se concentra donde mayor número de accidentes hay, se aprecia un decrecimiento paulatino a medida que aumenta el valor.

***Figura 42**. Número de siniestros. Variable valor del vehículo*



FUENTE: Elaboración propia

La última variable del modelo es el índice de tráfico. Aquellos que tienen un mayor número son los que peor calificación tienen, por tanto, al tener un coeficiente positivo se espera que sean aquellos que peor calidad de conducción tienen y, por tanto, los que mayor frecuencia de siniestros pueden llegar a tener.

Para comprobar que este modelo es válido no solo sirve observar que todos los coeficientes son significativos si no que también lo son generando el mismo modelo, pero con la muestra del test. Como podemos ver en la siguiente tabla mostramos que efectivamente los resultados vuelven a ser significativos, por tanto, estamos ante un modelo válido.

Tabla 10. *Modelo GLM para la frecuencia. Muestra Test.*

Generalized linear model results

Dep. Variable	num_siniestro_total	No. Observations	11978
Model:	GLM	Df Residuals	11967
Model Family:	Poisson	Df model	10
Link Function:	log	Scale	1
Method:	IRLS	Log-Likelihood:	-6552
Date:	Sun, 22 May 2022	Deviance:	8912,5
Time:	18:28:14	Pearson chi2:	4,93E+04
No. Iterations	6		
Covariance Type	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	1,2661	0,225	6	0,000	0,826	1.706
C(sexo,Treatment(reference="F"))[T.M]	1,0648	0,043	25	0,000	1	1.149
C(edad_mayores70,Treatment(reference=0))[T.1]	0,6847	0,098	7	0,000	0,493	0,877
C(area_residencia_23_46,Treatment(reference=3))[T.1]	-0,2403	0,063	-4	0,000	-0,363	-0,117
C(area_residencia_23_46,Treatment(reference=3))[T.4]	0,32	0,051	6	0,000	0,221	0,419
C(area_residencia_23_46,Treatment(reference=3))[T.5]	-0,6317	0,114	-6	0,000	-0,856	-0,408
edad	-0,019	0,002	-11	0,000	-0,022	-0,016
credit_scoring_mod	-0,0028	0	-6	0,000	-0,004	-0,002
credit_scoring_mod2	-0,2524	0,075	-3	0,001	-0,4	-0,105
indice_trafico_mod	0,0028	0,001	4	0,000	0,001	0,004
valor_vehiculo	-7,88E-06	1,75E-06	-5	0.000	-1,13E-05	-4,45E-06

FUENTE: Elaboración propia

Como podemos ver en la tabla de los resultados la desvianza del modelo es 34.597, de forma que su error generalizado es 0.722. Estos resultados nos servirán para comparar más adelante con el modelo calculado mediante Gradient Boosting. Respecto al AIC y BIC se ha utilizado para desechar otros modelos no incluidos, pero de manera informativa los resultados han sido para el AIC 50.830 y para el BIC -481.637, que comparándolos con el modelo de prueba se puede apreciar una gran mejoría ya que ambos valores son menores. Además, como la desvianza es menor al '*Pearson chi-squared'* podemos aceptar el ajuste.

4.1.2 MODELO GLM PARA LA SEVERIDAD

En este apartado tratamos de explicar la severidad o coste de los siniestros a partir de las distintas variables de nuestra base de datos. Para este modelo es importante tener en cuenta la posibilidad de la existencia de valores punta o valores atípicos que puedan distorsionar los resultados. Por ello, al comienzo de este apartado hemos llevado a cabo ese estudio eliminando las 506 pólizas con costes por encima de los 16.500€. Esta extracción de datos se realiza antes del estudio de la frecuencia para partir con el mismo número de datos en ambos casos. El estudio se lleva a cabo mediante una modelización gamma, con una función log como función de enlace.

A posteriori, tras realizar los distintos modelos, cuando se exponga la perspectiva de negocio, se dará una explicación empresarial de cómo tratar a las pólizas que no se han podido incluir en el estudio dada su alta siniestralidad.

Al igual que para la frecuencia, la muestra se divide en dos partes, un 80% es utilizado para entrenar el modelo y el otro 20% es utilizado para las comprobaciones. Para obtener el mejor modelo se van introduciendo variables una a una y analizando los resultados.

Se ha llevado a cabo un estudio para saber cuál es la distribución que mejor se ajusta a nuestro modelo. La distribución escogida es aquella con menor AIC y BIC (prevalece el resultado del segundo método sobre el primero). Como podemos ver en la tabla, se han estudiado diversas distribuciones siendo la Gamma la escogida.

Figura 43. *Comparativa de distribuciones*

	Suma error cuadrático	AIC	BIC
Gamma	0,90	1.682,88	-159.601,90
Exponencial	0,92	1.683,83	-159.466,90
Beta	1,33	1.657,36	-156.912,08
Cauchy	5,07	1.774,54	-147.517,98
Logistic	5,91	1.700,77	-146.453,82

FUENTE: Elaboración propia

Se han llevado a cabo distintos modelos muy similares donde la modificación se encontraba en las variables, ya que la función de enlace ('*link function*') siempre se ha utilizado la Gamma. El modelo final escogido tiene las siguientes variables:

- **Credit_scoring_mod2**: Al igual que en el modelo de la frecuencia dividimos en tres a la muestra del riesgo crediticio, siendo 0 los más riesgosos.
- **Valor_vehiculo_tramos**: Dividimos en tres partes los precios de los vehículos. Con un valor 0 tenemos los coches con un valor dentro del percentil 25, con valor igual a 1 aquellos que se encuentran entre el 25 y 75, y con un dos los coches de alta gama, aquellos que se encuentran por encima del percentil 75.
- **Área_residencia_36**: En este caso, únicamente fusionamos las ciudades 3 y 6, tomando como base la tercera ciudad.
- **Edad_tramos**: Esta variable es muy similar a la del valor del vehículo, dividiendo la edad en los mismos intervalos.
- **Antigüedad_vehiculo_23**: Unificamos los coches que se encuentran en el segundo y tercer periodo, tomándolos como base.

Tabla 11. *Modelo de la severidad GLM*

Generalized linear model results

Dep. Variable	coste_siniestro_total	No. Observations	7622
Model:	GLM	Df Residuals	7609
Model Family:	Gamma	Df model	12
Link Function:	log	Scale	12.467
Method:	IRLS	Log-Likelihood:	-64984
Date:	Sun, 22 May 2022	Deviance:	5777,6
Time:	13:06:12	Pearson chi2:	9.49e+03
No. Iterations	20		
Covariance Type	nonrobust		

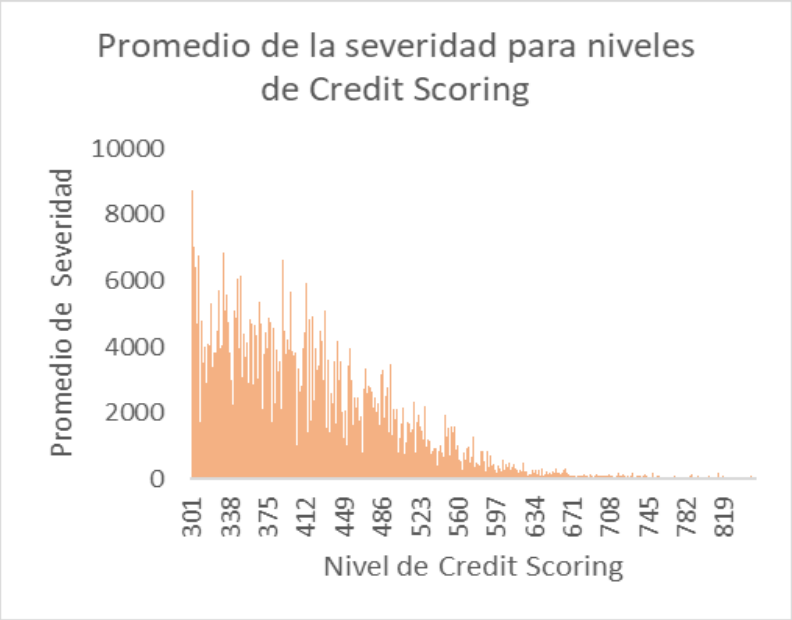
	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	6,3273	0,035	181,485	0,000	6	6.396
C(credit_scoring_mod2,Treatment(reference=2))[T.0]	1,4519	0,032	45,784	0,000	1	1.514
C(credit_scoring_mod2,Treatment(reference=2))[T.1]	0,6032	0,031	19,260	0,000	0,542	0,665
C(valor_vehículo_tramos,Treatment(reference=1))[T.0]	-0,3732	0,035	-10,619	0,000	-0,442	-0,304
C(valor_vehículo_tramos,Treatment(reference=1))[T.2]	0,3169	0,033	9,595	0,000	0,252	0,382
C(area_residencia_36,Treatment(reference=3))[T.1]	0,2953	0,037	7,899	0,000	0,222	0,369
C(area_residencia_36,Treatment(reference=3))[T.2]	0,3387	0,034	9,947	0,000	0,272	0,405
C(area_residencia_36,Treatment(reference=3))[T.4]	0,5908	0,037	16,149	0,000	0,519	0,663
C(area_residencia_36,Treatment(reference=3))[T.5]	10.615	0,068	15,580	0,000	0,928	1.195
C(edad_tramos,Treatment(reference=1))[T.0]	0,419	0,03	13,831	0,000	0,36	0,478
C(edad_tramos,Treatment(reference=1))[T.2]	0,4854	0,034	14,407	0,000	0,419	0,551
C(antiguedad_vehiculo_23,Treatment(reference=2))[T.1]	0,8132	0,038	21,198	0,000	0,738	0,888
C(antiguedad_vehiculo_23,Treatment(reference=2))[T.4]	-0,3546	0,033	-10,631	0,000	-0,42	-0,289

FUENTE: Elaboración propia

Al igual que en el caso anterior, expondremos un análisis ex-post para comprobar si realmente el modelo muestra los resultados esperados o sí, por el contrario, no obtenemos las tendencias esperadas, averiguar la razón.

Comenzamos con la primera variable, el riesgo de crédito. Observando el gráfico observamos una pendiente negativa similar al caso de la frecuencia. Es decir, a medida que disminuye el riesgo crediticio el coste de la siniestralidad es menor. Observamos como ambas variables coinciden, en el caso de credit_scoring_2 el coeficiente también es negativo. En el caso de Credit_Scoring_mod_2 ambos coeficientes son positivos porque la base son los asegurados con mejor nivel financiero.

Figura 44. *Promedio de la severidad. Variable Credit Scoring*



FUENTE: Elaboración propia

El valor del vehículo por tramos muestra que aquellos vehículos más costosos tienen un mayor coste cuando sufren un siniestro que aquellos que son más baratos. Un resultado plausible ya que los vehículos más lujosos suelen tener piezas más específicas y mejor acabado generando así un mayor coste de reparación.

En este caso, la variable del área de residencia de nuestros asegurados se segmenta de una manera distinta al caso de la frecuencia. Por un lado, tomamos como referencia las ciudades tres y seis, que como hemos explicado, han sido fusionadas. Esta unión se lleva a cabo por la falta de significatividad de la última zona y por la cercanía en resultados del coste promedio de la severidad como podemos ver en la tabla. Además, al anexionarla con la ciudad más poblada se mitigan las posibles diferencias en el promedio del coste.

Los coeficientes deben ser todos positivos ya que el promedio del coste es más elevado en todas ellas. Siendo el coeficiente más pequeño el de la primera ciudad dado la cercanía de costes medios. Al estar tratando con una base de datos más pequeña que la inicial ya que hemos apartado aquellas pólizas sin siniestros y aquellas con siniestros punta, observamos como la quinta ciudad únicamente cuenta con 201 observaciones, posiblemente por ello su coeficiente es más elevado que el de la cuarta zona, pese a esperar un resultado contrario. A pesar de ello, los resultados no son tan alejados.

***Tabla 12**. Promedio de la severidad. Variable Área de residencia*

Área de residencia	Promedio Coste Severidad
1	446,68
2	702,87
3	441,08
4	1271,47
5	921,03
6	597,21

FUENTE: Elaboración propia

La edad se ha segmentado en tres tramos. Como ya se ha expresado a lo largo del estudio las zonas más complejas se encuentran en los extremos donde los jóvenes tienden a tener más accidentes y por tanto mayores costes y, en el otro extremo, las personas mayores, al tener menos reflejos, tienden a tener mayor número de accidentes. Por ello, al ser coeficientes positivos coincide con el análisis ex-ante.

La última variable del modelo es la edad del vehículo. El costo de las reparaciones de los vehículos más viejos es más barato ya que la tecnología no es tan innovadora y los procesos de reparación son más rápidos de llevar a cabo. Por ello, ambos coeficientes son correctos ya que los vehículos del primer periodo cuando tienen un siniestro este es más caro.

Al igual que en frecuencia mostramos en la siguiente tabla que los coeficientes son significativos también para la muestra del test. De esta forma, el resultado obtenido es válido.

***Tabla 13**. Modelo GLM para la severidad. Muestra test.*

Generalized linear model results

Dep. Variable	coste_siniestro_total	No. Observations	1906
Model:	GLM	Df Residuals	1893
Model Family:	Gamma	Df model	12
Link Function:	log	log Scale	1,2830
Method:	IRLS	Log-Likelihood:	-16368
Date:	Sun, 22 May 2022	Deviance:	1450,9
Time:	13:06:12	Pearson chi2:	2.43e+03
No. Iterations	22		
Covariance Type	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	6,33	0.071	88,69	0.000	6,19	6,47
C(credit_scoring_mod2,Treatment(reference=2))[T.0]	1,41	0.065	21,80	0.000	1,28	1,53
C(credit_scoring_mod2,Treatment(reference=2))[T.1]	0.5766	0.064	9,07	0.000	0.452	0.701
C(valor_vehículo_tramos,Treatment(reference=1))[T.0]	-0.3485	0.073	-4,81	0.000	-0.491	-0.206
C(valor_vehículo_tramos,Treatment(reference=1))[T.2]	0.4243	0.066	6,41	0.000	0.295	0.554
C(area_residencia_36,Treatment(reference=3))[T.1]	0.2341	0.078	3,01	0.003	0.082	0.387
C(area_residencia_36,Treatment(reference=3))[T.2]	0.4737	0.069	6,83	0.000	0.338	0.610
C(area_residencia_36,Treatment(reference=3))[T.4]	0.6589	0.073	9,02	0.000	0.516	0.802
C(area_residencia_36,Treatment(reference=3))[T.5]	0.8598	0.129	6,66	0.000	0.607	1,11
C(edad_tramos,Treatment(reference=1))[T.0]	0.4995	0.061	8,19	0.000	0.380	0.619
C(edad_tramos,Treatment(reference=1))[T.2]	0.5101	0.070	7,34	0.000	0.374	0.646
C(antiguedad_vehiculo_23,Treatment(reference=2))[T.1]	0.7912	0.075	10,48	0.000	0.643	0.939
C(antiguedad_vehiculo_23,Treatment(reference=2))[T.4]	-0.4584	0.068	-6,75	0.000	-0.592	-0.325

FUENTE: Elaboración propia

Por último, debemos analizar las características de nuestro modelo. Como podemos ver en la tabla de los resultados la desvianza del modelo es 5777, de forma que su error generalizado es 0.758. Estos resultados nos servirán para comparar más adelante con el modelo calculado mediante Gradient Boosting. Respecto al AIC y BIC se ha utilizado para desechar otros modelos no incluidos, pero de manera informativa los resultados han sido para el AIC 129.994y para el BIC -62.237. Además, como la desvianza es menor al '*Pearson chi-squared'* podemos aceptar el ajuste.

4.1.3 PRIMA PURA

Una vez hemos creado ambos modelos, debemos otorgar valores a la frecuencia y severidad modeladas para cada una de nuestras pólizas. Tras hallar estos valores, para la frecuencia tendremos una ponderación para saber cuál es el número de siniestros de cada asegurado según la exposición de cada uno. Por otro lado, en la severidad obtendremos el coste expresado en u.m. dadas las características del asegurado. Previamente se debe realizar una transformación para los distintos coeficientes, ya que se han calculado en base logarítmica. Realizamos la exponenciación de los coeficientes, tanto para el modelo de la frecuencia como de la severidad.

La severidad de aquellos asegurados que no han tenido ningún siniestro y por tanto no han entrado en la muestra para la creación del modelo, pasan a tener un coste esperado igual al coeficiente β_0.

Finalmente, teniendo tanto la frecuencia modelada como la severidad modelada, únicamente debemos multiplicar un valor por otro de manera que obtengamos la prima pura o Burning Cost.

Al tener una cartera de únicamente un año, no sabemos cómo se comportan a largo plazo los asegurados, por ello, no podemos utilizar Bonus Malus en función del número de siniestros y el coste de estos de años anteriores.

Mostramos un ejemplo aleatorio de 5 pólizas donde incluimos la frecuencia esperada, el coste esperado y la Prima pura o Burning Cost.

***Tabla 14**. Ejemplo de pólizas modeladas*

Póliza	frecuencia Modelada	Coste modelado	Burning Cost (Prima Pura)
28	0,0775	559,64	43,40
29	0,1060	559,64	59,34
30	2,5025	11.251,05	28.155,90
31	0,0778	2.072,54	161,16
32	0,3652	559,64	204,36

FUENTE: Elaboración propia

4.2 MODELO GBM

Para poder llevar a cabo una comparativa entre las distintas casuísticas utilizaremos los mismos datos, es decir, al igual que en la creación de los modelos lineales generalizados debemos eliminar los valores punta de los siniestros. Además, evitamos cierto sesgo por los siniestros de alto coste.

4.2.1 MODELO GBM PARA LA FRECUENCIA

La creación de este modelo se hace en función del número de siniestros. Se ha utilizado la función LGBM perteneciente a la herramienta de Sklearn de Python.

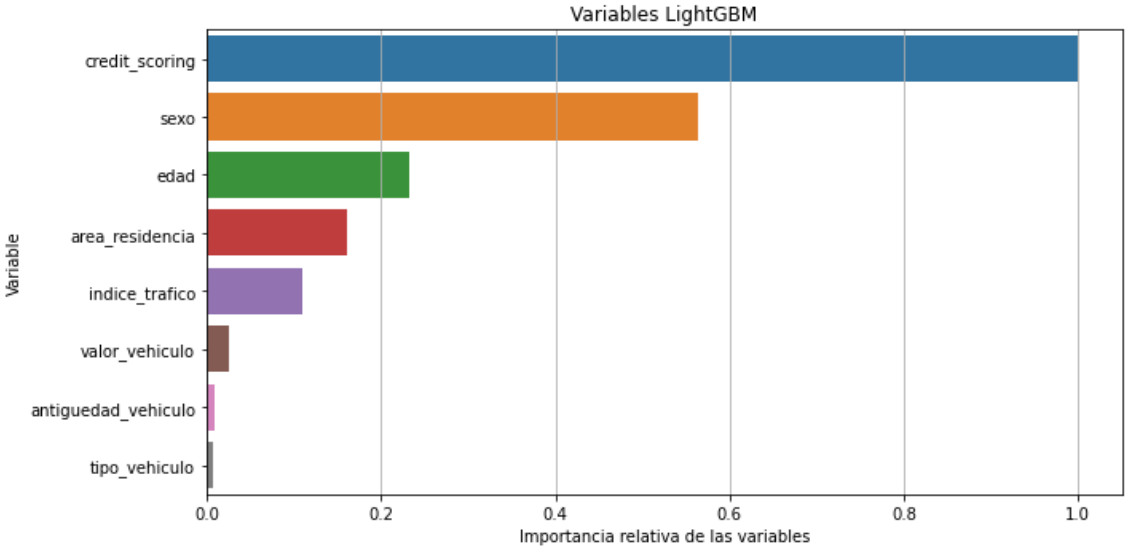
Como hemos explicado previamente estos modelos tienen distintas características que pueden modificarse con el fin de obtener una mejor estimación y tratar de evitar el sobre ajuste. Comenzamos con el modelo de frecuencia donde debemos capturar la estimación sobre el número de siniestros que tienen los asegurados de la cartera. Generamos un modelo para analizar un primer resultado con el que poder avanzar a la validación cruzada y así saber cuál es el mejor modelo.

Este primer modelo tiene una profundidad de los árboles de 3 nodos, una tasa de aprendizaje del error de 0.1, cuenta con 200 árboles y la función objetivo, al igual que la modelización GLM, es la distribución de Poisson. Algo interesante es ver cuáles son las

variables más importantes en este caso y compararlas no solo con el modelo escogido final si no también con las variables escogidas con la otra modelización.

Como se puede ver en la figura 44 vemos como el Credit Scoring es realmente importante, seguido por el sexo y la edad. En el lado opuesto, tanto el tipo de vehículo como su valor no aportan gran información al modelo.[5]

***Figura 45**. Importancia relativa de las variables. Modelo Frecuencia GBM*

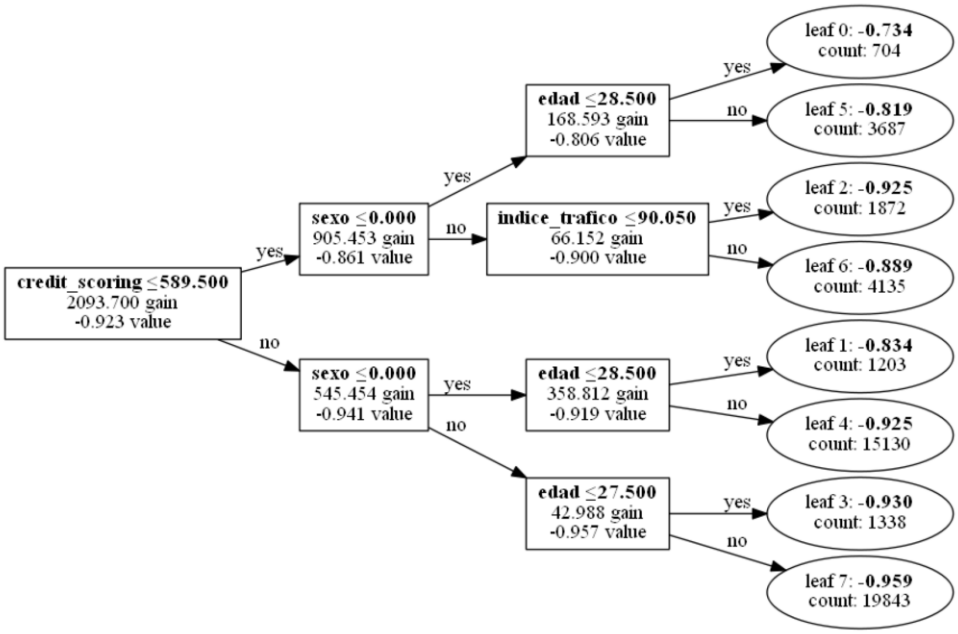


FUENTE: Elaboración propia

A continuación, se muestra el primer árbol generado y el último. La diferencia más notoria es la ganancia de información entre ambos gracias a que cada árbol tiene en cuenta los errores de los modelos previos. Pese a tratarse de árboles con profundidad tan corta, únicamente 3 nodos, observamos el uso de casi todas las variables, ya que dependiendo en cómo se haya realizado el primer corte puede que variables que a priori no aportan tanta información sean útiles, por ello, no debemos desecharlas antes de analizar los resultados.

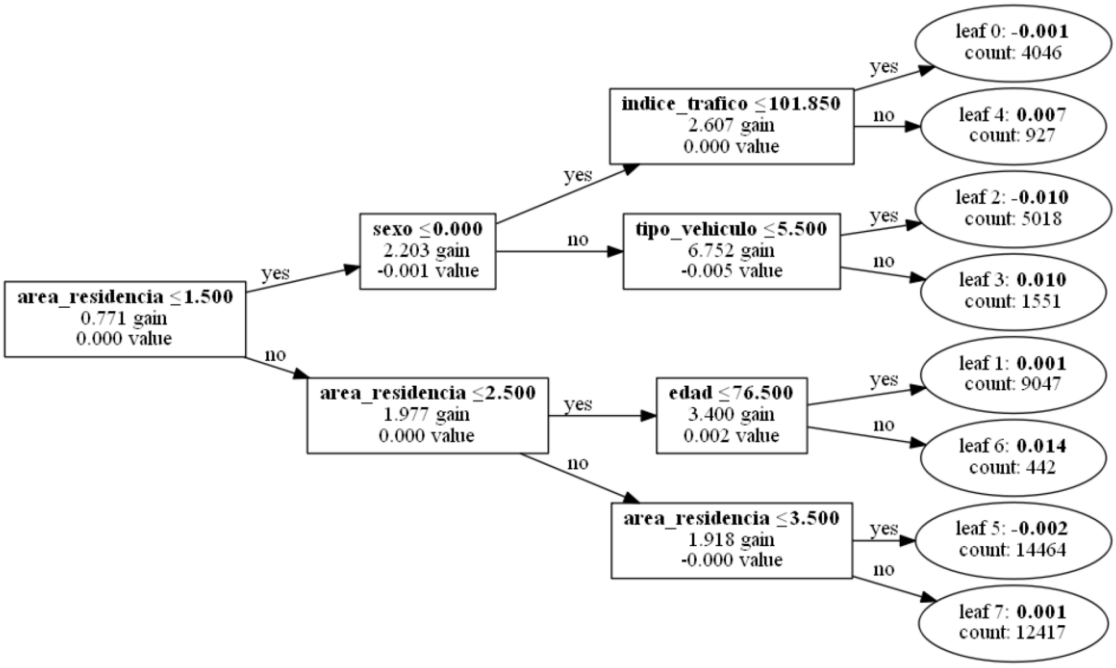
[5] La importancia relativa de las variables se calcula mediante la función feature_importances_ de SKlearn donde se calculan todas las ganancias de información de cada variable y se ordenan por su importancia.

Figura 46. *Árbol número 1. GBM de frecuencia*



FUENTE: Elaboración propia

Figura 47*. Último árbol de regresión. GBM frecuencia*



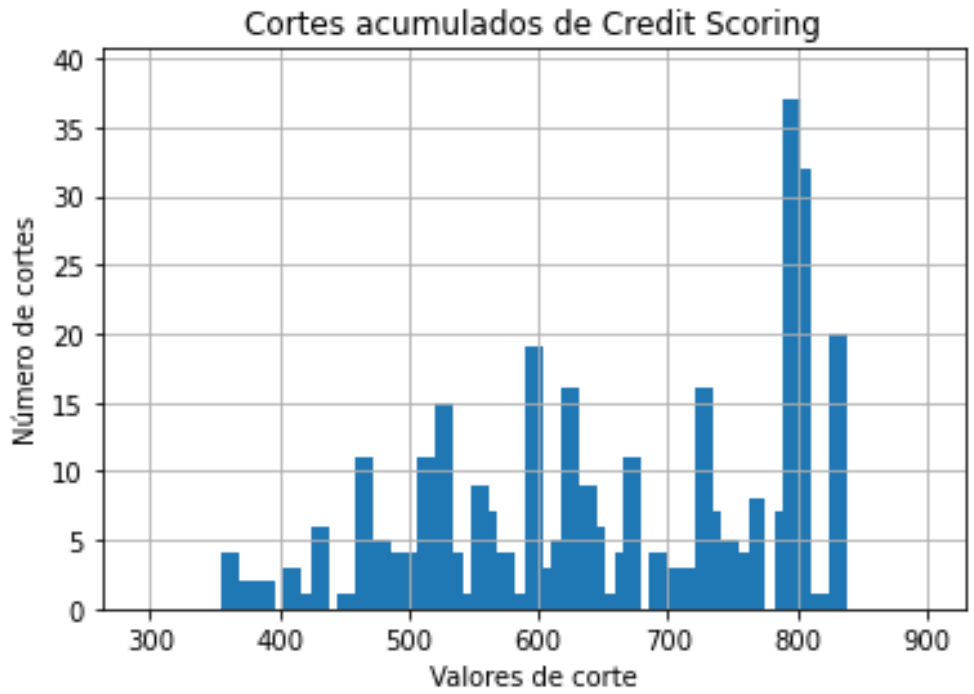
FUENTE: Elaboración propia

Una vez hemos visto que variables son las más importantes y como son los distintos árboles, es interesante saber cómo se distribuyen los distintos cortes en cada variable para

saber dónde otorga importancia la modelización en cada variable. Mostramos la variable Credit Scoring como ejemplo, aunque a medida que avancemos el análisis mostraremos los distintos cortes para los modelos finales.

Observamos como la mayoría de los cortes se llevan a cabo en el centro aunque el punto de corte más utilizado 800, donde se encuentra la mediana. Observamos una tendencia alcista hacia los números más elevados pero los extremos apenas tienen cortes, algo obvio ya que apenas genera información.

Figura 48. *Número de cortes acumulados. Variable Credit Scoring*



FUENTE: Elaboración propia

Tras haber explicado brevemente como se muestran las características de esta modelización debemos pasar a la validación cruzada con el fin de obtener el mejor modelo.

La validación cruzada tiene diversos métodos de aplicación, uno de ellos y el que aplicaremos en el estudio se llama Kfolds. Consiste en subdividir la muestra de entrenamiento en ‘k’ partes, en nuestro caso 5, y comprobar si se está modelizando bien. Este corte se lleva a cabo en la muestra de entrenamiento y no con la parte del test para no otorgar información al algoritmo de dicha parte ya que podría alterar las estimaciones puesto que ya conocería los resultados de aquello con lo que queremos comprobar si funciona.

Para calcular el mejor modelo realizaremos distintos modelos para las tres características más importantes. Para la profundidad del árbol se probarán cuatro métricas distintas [3, 4, 5, 6]. Para el número de árboles utilizaremos 500, 750 y 1000 árboles. Finalmente, para el ratio de aprendizaje utilizaremos otras cuatro posibilidades [0.005, 0.01, 0.02, 0.05].

De esta manera, habremos creado 48 modelos distintos. El mejor modelo será aquel con menor error generalizado. Como podemos ver en la siguiente tabla un ratio de aprendizaje de 0.05, con 750 árboles y una profundidad de 3 nodos es el árbol con menor error generalizado, por tanto, es el modelo escogido. Se debe mencionar el hecho de que los peores modelos se han obtenido con una tasa de aprendizaje de 0.005, posiblemente debido a un aprendizaje muy lento para el resto de las características, ya que la mejor modelización con esa tasa es con un árbol de 6 nodos y un conjunto de 1000 estimadores o árboles.

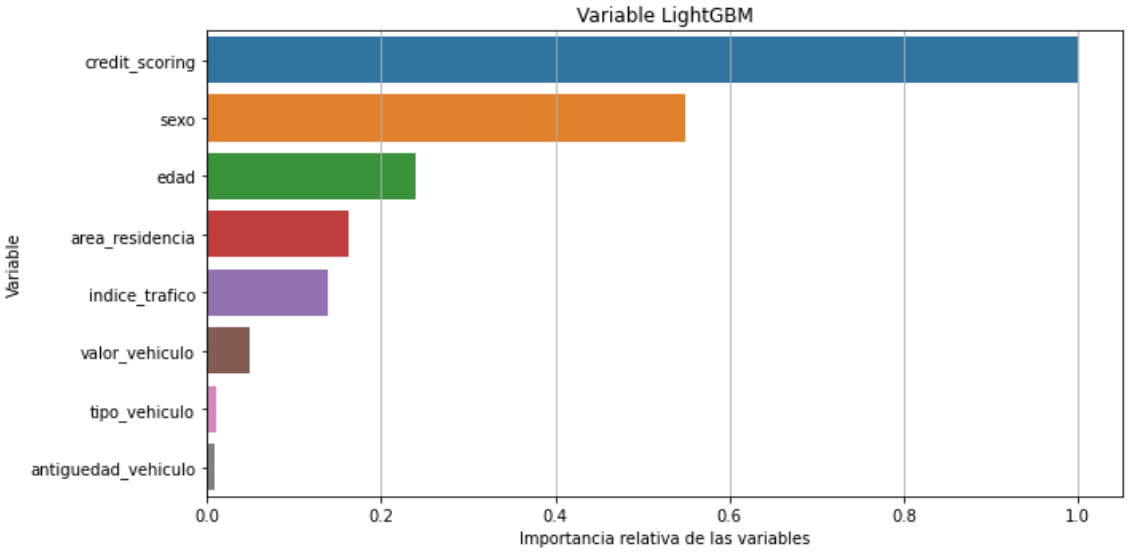
Tabla 15. *Comparativa de resultados. Modelos de frecuencia GBM*

Ratio aprendizaje	Nº de árboles	Profundidad	Error generalizado
0,05	750	3	0,70281
0,05	1.000	3	0,70281
0,05	500	3	0,70285
0,02	1.000	3	0,70313
0,05	1.000	4	0,70329
0,05	750	4	0,70329
0,05	500	4	0,70329
0,02	1.000	4	0,70349
0,02	750	4	0,70358
0,02	750	3	0,70361

FUENTE: Elaboración propia

Una vez ya tenemos nuestro modelo pasamos a mostrar los resultados de este. En primer lugar, las variables más importantes y que mayor ganancia de información aportan son las que vemos en la siguiente tabla. En comparación con el primer modelo mostrado observamos que el orden de las variables es el mismo pero la variable sexo tiene menor importancia relativa, es decir, el resto de las variables aportan más información que en el caso previo.

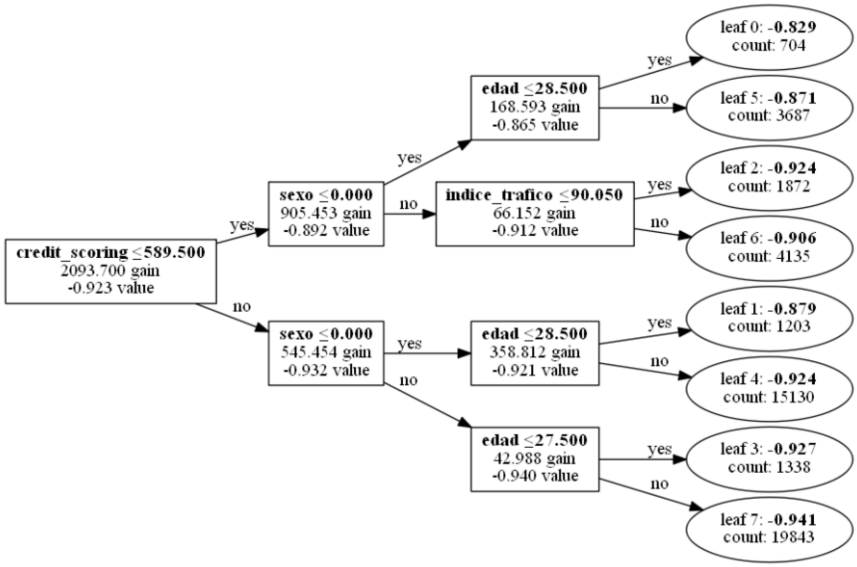
Figura 49. *Importancia relativa de variables. Modelo de frecuencia GBM*



FUENTE: Elaboración propia

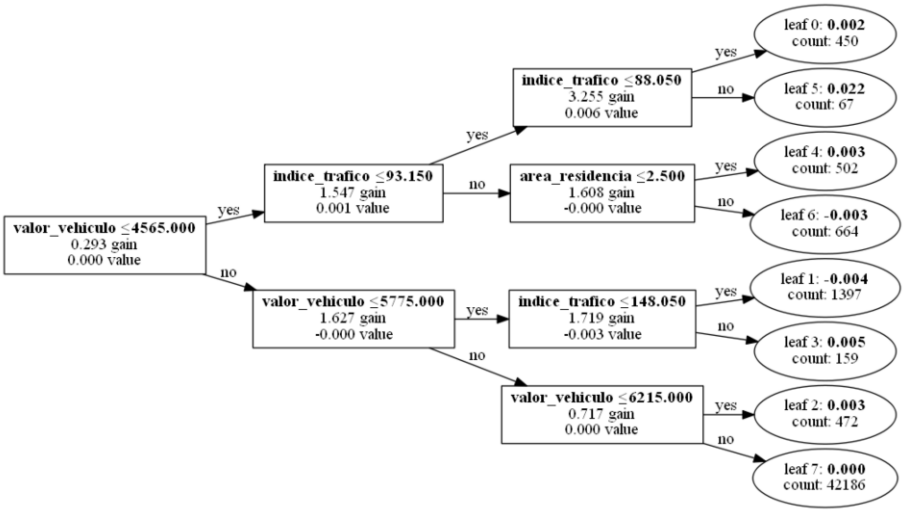
Respecto a los distintos árboles mostramos el primero y el último al igual que antes y observamos el uso de las distintas variables. Al haber generado 750 árboles, apenas podemos obtener información de únicamente dos, pero si podemos apreciar la tendencia de reducción de ganancia de información a medida que se crea un mayor número de árboles. El hecho de que a medida que se crean árboles estos generen menor ganancia de información no significa que sean malos o que su predicción es errónea, simplemente se están acercando a la frontera eficiente, por ello, lo importante es que la tasa de incremento de información se mantenga siempre por encima de un mínimo y poder así evitar el sobreajuste.

Figura 50. *Primer árbol del modelo de frecuencia GBM*



FUENTE: Elaboración propia

***Figura 51**. Último árbol del modelo de frecuencia GBM*

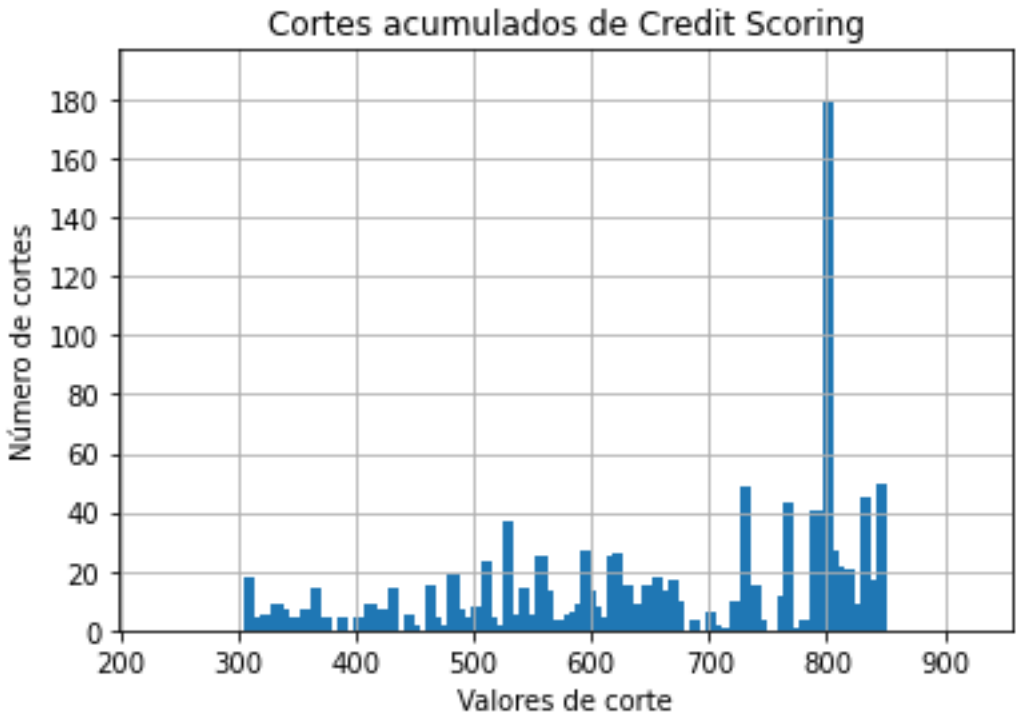


FUENTE: Elaboración propia

Respecto a los cortes generados, mostramos la primera y tercera variable ya que la variable sexo al ser dicotómica apenas aporta información. Respecto a la variable crediticia los resultados son más homogéneos a lo largo de la variable, aunque sigue existiendo un máximo en la mediana. En comparación con el primer caso visto los extremos tienen un mayor número de cortes lo que muestra un mayor análisis de las colas. Posiblemente y debido a un mayor número de casos con un valor crediticio más elevado los cortes se dan más en la parte derecha de la variable.

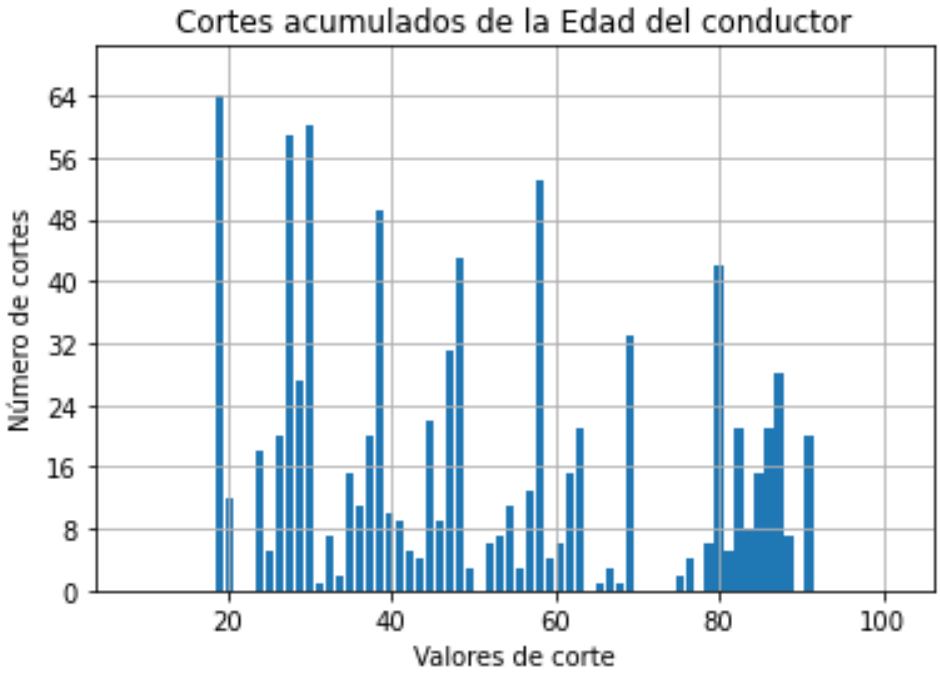
Respecto a la variable de la edad, como ya hemos analizado a lo largo del trabajo los puntos clave se muestran en los conductores jóvenes y en los conductores de edad más avanzada, algo que se aprecia en el número de cortes donde destacan ciertas edades como los menores de veinte años o aquellos con edades superiores a los ochenta.

Figura 52. *Número de cortes acumulados. Variable Credit Scoring*



FUENTE: Elaboración propia

Figura 53*. Número de cortes acumulados. Variable edad del conductor*



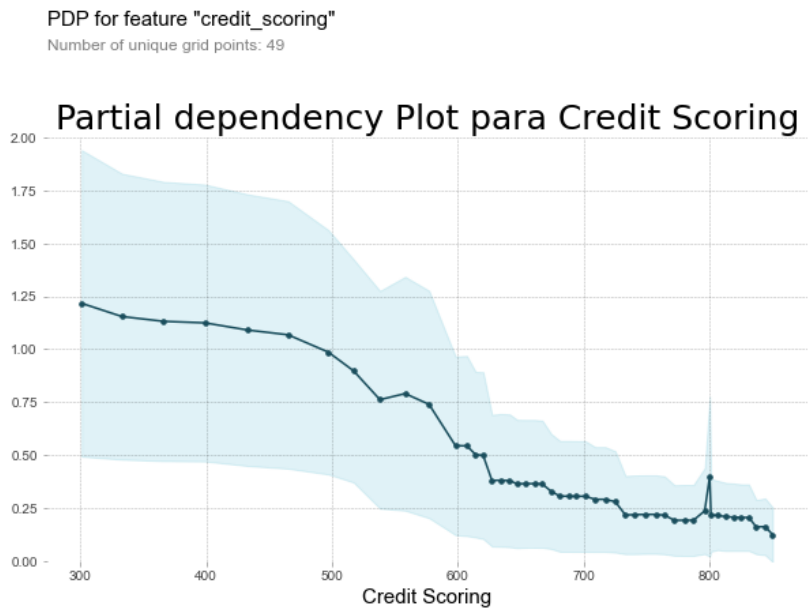
Para finalizar el estudio sobre la modelización de la frecuencia debemos obtener los gráficos de dependencia parcial para las distintas variables. Mediante estos gráficos se pretende analizar el efecto que genera cada variable sobre la frecuencia de accidentes.

En primer lugar, observamos una clara tendencia negativa partiendo de aquellos individuos con mayor probabilidad de impago, que tienden a tener un mayor número de accidentes hacia aquellos con mayor salud económica que disminuyen su probabilidad de tener accidentes.

Respecto a la edad se muestran tres partes, la primera de ellas son los conductores jóvenes con una mayor probabilidad de accidentes, con especial diferencia en aquellos menores de 20 años. Desde los 30 años hasta la edad de jubilación se mantiene constante y a partir de ese punto vuelve a incrementar, aunque en ningún momento llega a los niveles iniciales.

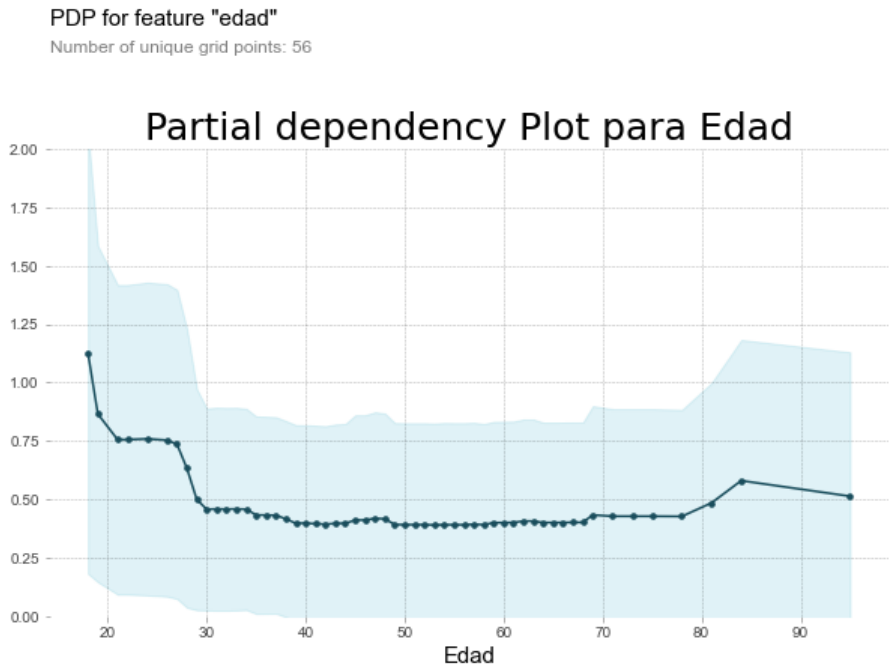
Finalmente, el índice de tráfico muestra una tendencia ascendente prácticamente lineal. A peor calidad de conducción del conductor más posibilidad de tener un accidente, con un incremento notorio en especial para los conductores con peor calidad.

Figura 54. *Partial dependency plot. Variable Credit Scoring*



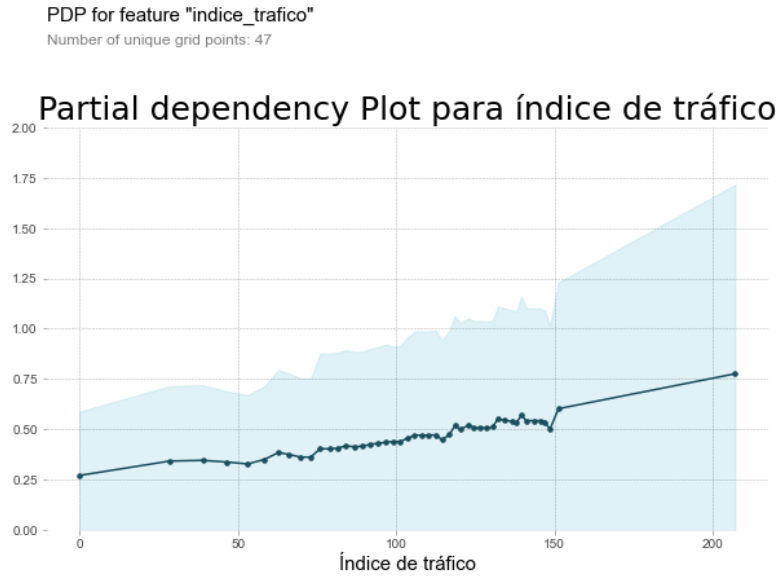
FUENTE: Elaboración propia

***Figura 55**. Partial dependency plot. Variable edad*



FUENTE: Elaboración propia

***Figura 56.** Partial dependency plot. Variable Índice de tráfico*



FUENTE: Elaboración propia

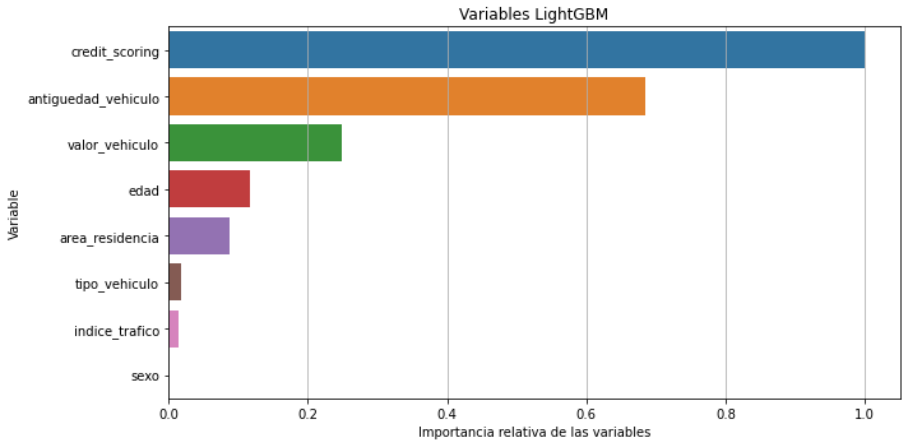
4.2.2 MODELO GBM PARA LA SEVERIDAD

Mediante este modelo pretendemos modelizar el coste de los siniestros, al igual que en el caso anterior utilizamos las mismas librerías y funciones.

Al igual que en la modelización GLM, en este caso debemos eliminar los registros o pólizas que no tienen siniestros ya que al tratarse de una distribución Gamma no podemos utilizar dichos valores. Por esta razón contamos con un número menor de datos, haciendo hincapié en que tanto para la frecuencia como para la severidad hemos eliminado los valores punta.

El primer árbol generado con el fin de obtener una primera idea general de cómo es la importancia de las variables tiene una profundidad de 5 nodos, con un total de 100 árboles y una tasa de aprendizaje de 0.02. Al igual que en la frecuencia la variable más importante es Credit Scoring, pero en este caso las variables de los vehículos toman importancia siendo la antigüedad y el valor la segunda y tercera variable más importante. Por el lado contrario el sexo y el índice de tráfico que en la frecuencia eran de las variables más transcendentales ahora son las menos relevantes.

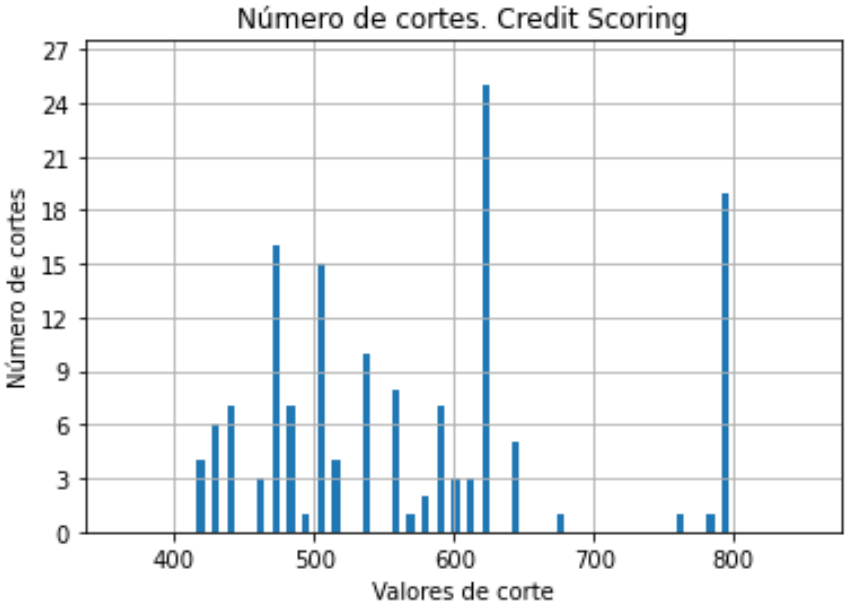
Figura 57. *Importancia relativa de las variables. Modelo severidad GBM*



FUENTE: Elaboración propia

El número de cortes es inferior al caso de la frecuencia al tratarse de un modelo con únicamente 100 árboles. Podemos comparar los resultados del Credit Scoring con la frecuencia, podemos ver ciertas diferencias, donde ocurre lo contrario, es decir, los cortes se llevan a cabo en los valores más bajos y apenas se generan cortes en la cola derecha.

***Figura 58**. Número de cortes acumulados. variable credit Scoring*



FUENTE: Elaboración propia

Tras observar cual puede ser la tendencia tanto en la importancia de las variables como en los cortes de los mismos debemos realizar la validación cruzada con el fin de obtener el mejor modelo. En este caso generamos un total de 60 modelos. Las características aplicadas son para el número de árboles un total de 3 posibilidades, 300, 500 y 750. Para la profundidad del árbol 5 posibilidades 1, 2 , 3, 4 y 5. Finalmente, para el ratio de aprendizaje 4 posibilidades 0.005, 0.01, 0.02 y 0.05.

En la siguiente tabla observamos los 10 mejores modelos. Los tres mejores cuentan con un ratio de aprendizaje igual a 0.05 y una profundidad del árbol de 2 y 3 nodos. Escogemos el primer modelo ya que es el que menor error generalizado tiene.

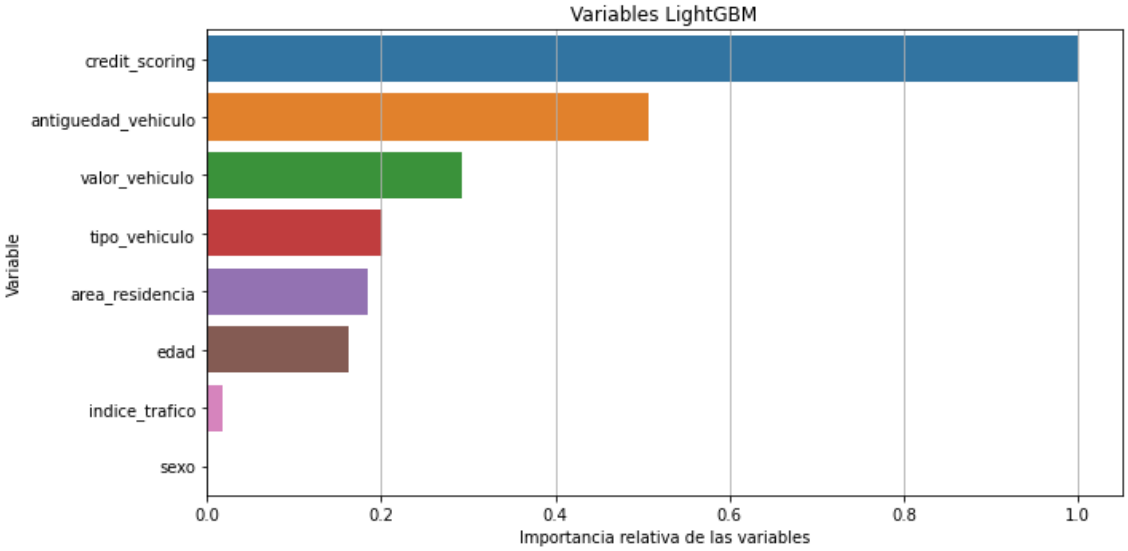
***Figura 59.** Comparativa de resultados. Modelos de severidad GBM*

Ratio aprendizaje	Nº de árboles	Profundidad	Error generalizado
0,05	400	4	0,50299
0,05	1.000	4	0,50299
0,05	500	4	0,50299
0,05	750	4	0,50299
0,02	750	4	0,50738
0,02	1.000	4	0,50738
0,10	500	4	0,50911
0,10	750	4	0,50911
0,10	1.000	4	0,50911
0,10	400	4	0,50911

FUENTE: Elaboración propia

La importancia de las variables en este nuevo árbol muestra una variación respecto al caso inicial, donde al igual que ocurrió con la frecuencia todas toman una mayor importancia. Únicamente el sexo apenas aporta información.

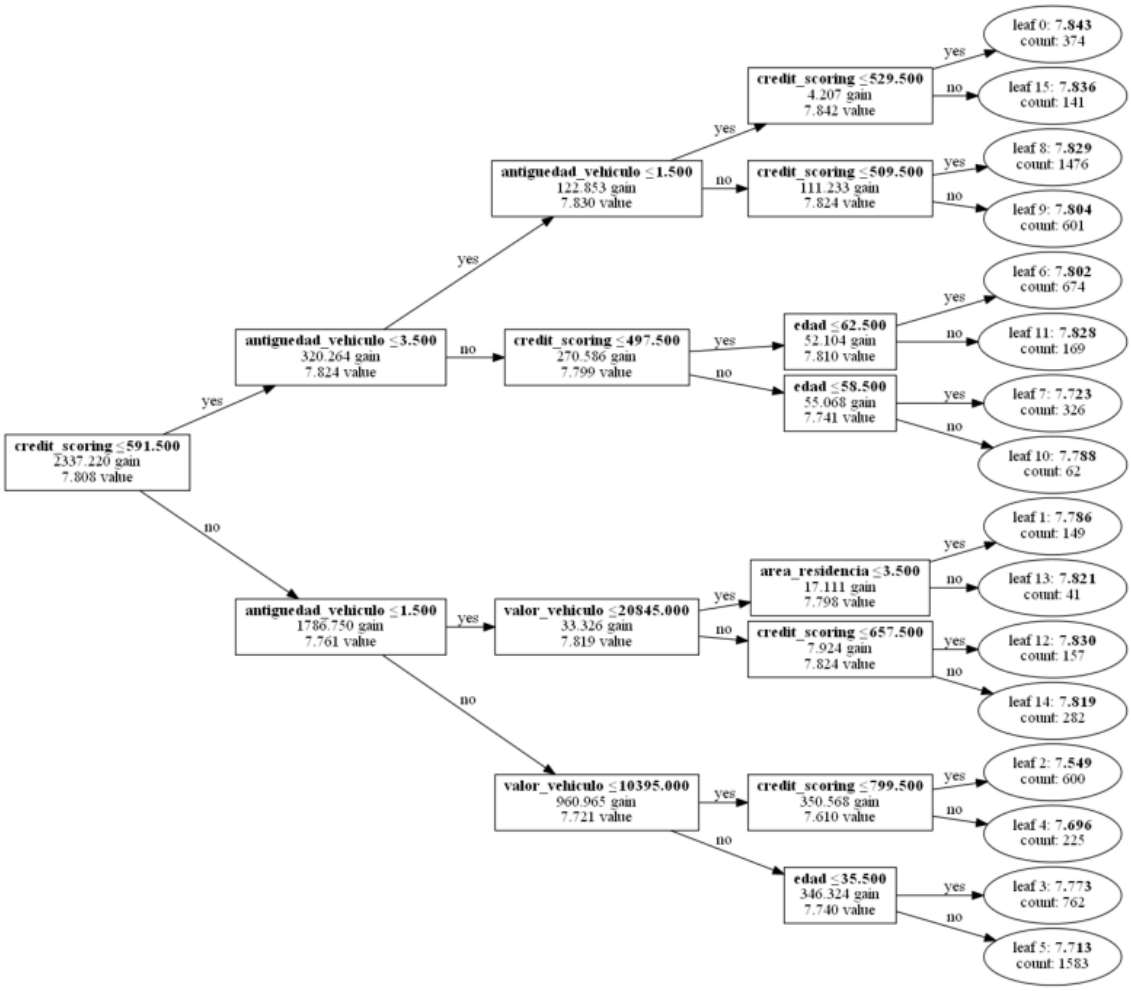
Figura 60. *Importancia relativa de las variables. Modelo de severidad GBM*



FUENTE: Elaboración propia

Mostramos a continuación el primer árbol. Al tratarse de un árbol con menor profundidad y por ende inferior número de hojas es más pequeño que el caso de la frecuencia, pero la idea de su funcionamiento es exactamente la misma que con la modelización previa. Pese a tener una estructura corta, el hecho de tener 400 árboles genera que sea un buen modelo ya que puede aprender de los árboles anteriores.

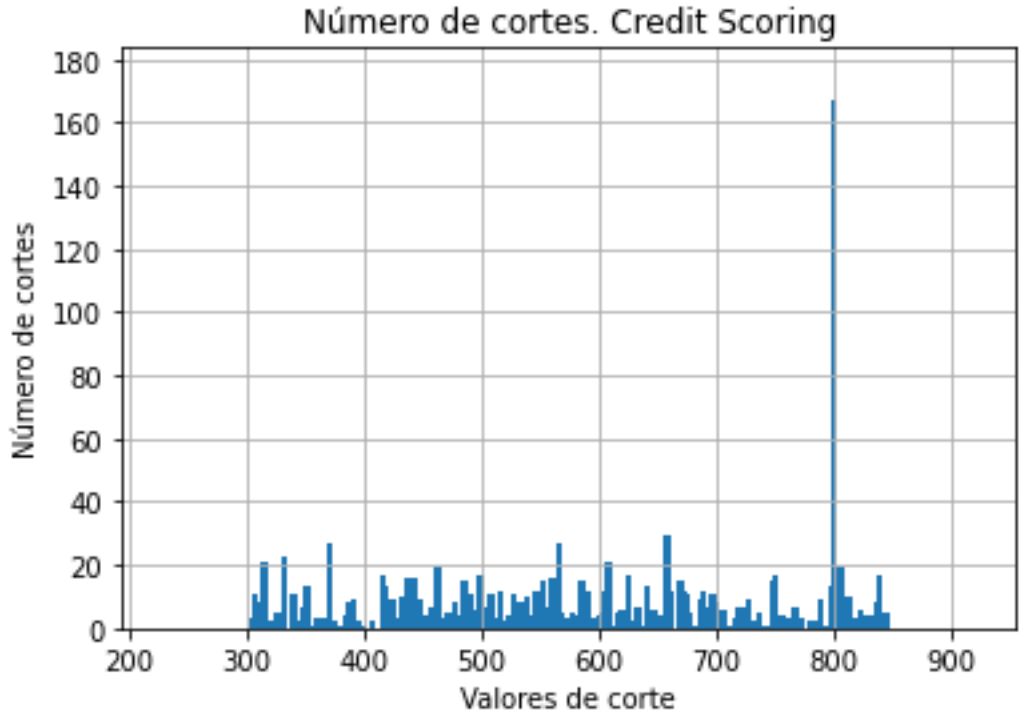
Figura 61. *Primer árbol del modelo de severidad GBM*



FUENTE: Elaboración propia

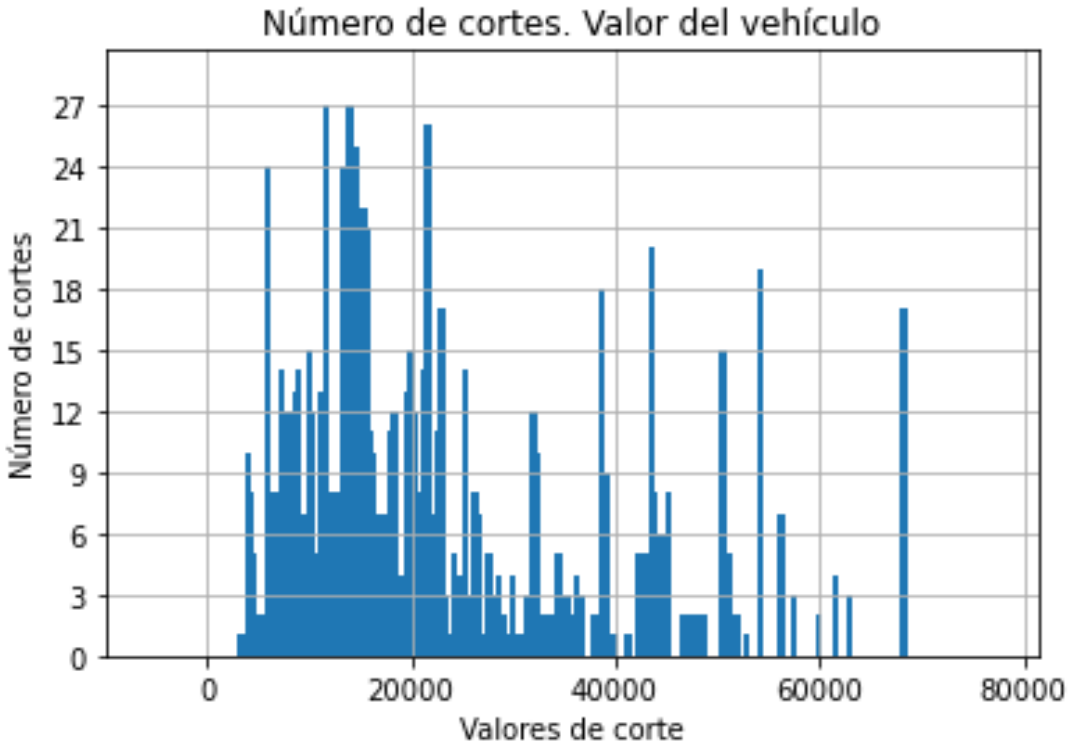
Respecto a los cortes para la creación de los árboles vemos como continúa destacando la mediana al igual que en el resto de los modelos previos para la variable del riesgo crediticio. El resto de los cortes se distribuyen de manera bastante uniforme cubriendo toda la muestra. La segunda variable más importante, el valor del vehículo vemos que los cortes se llevan a cabo en la parte izquierda del gráfico que es donde se encuentra la mayor parte de la muestra, aunque también hay cortes en los valores altos de los vehículos.

Figura 62. *Número acumulado de cortes. Variable Credit Scoring*



FUENTE: Elaboración propia

Figura 63. *Número acumulado de cortes. Variable Valor del vehículo*



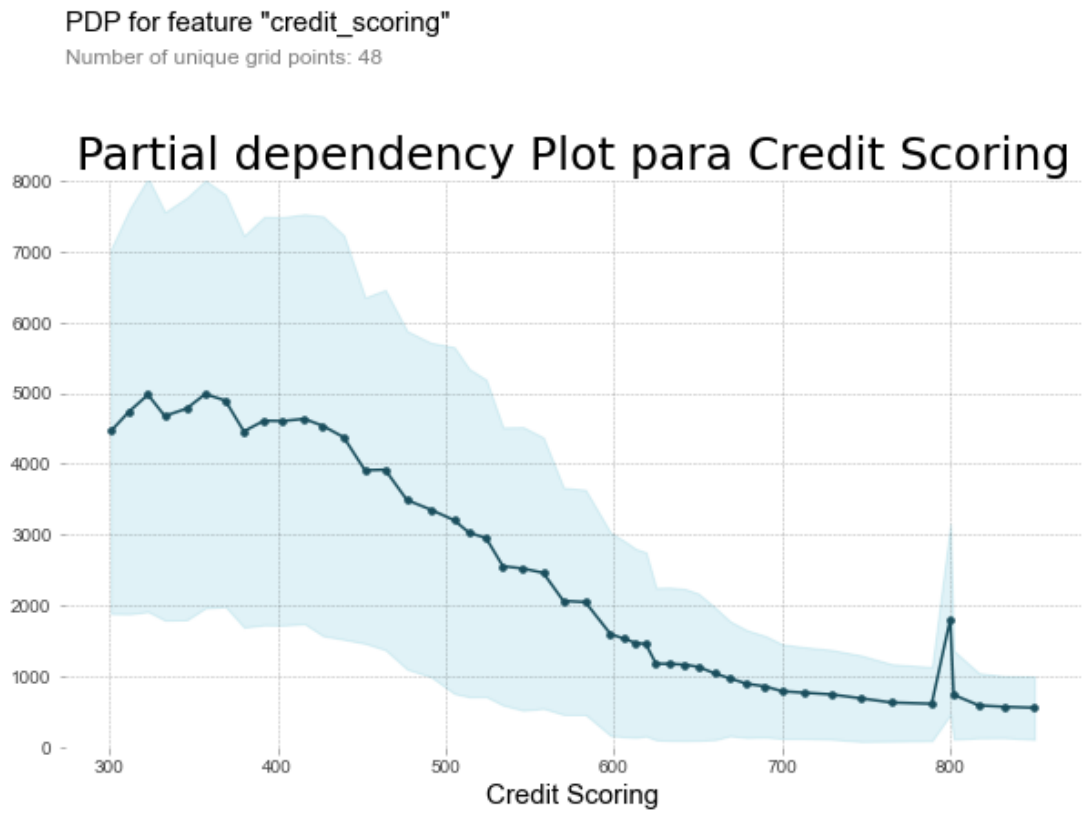
FUENTE: Elaboración propia

Para finalizar el estudio del modelo de la severidad debemos observar los gráficos de dependencia parcial para las distintas variables. Así seremos capaces de mostrar y analizar cómo afectan al coste del siniestro las distintas variables de manera individual.

Al igual que en el caso anterior analizaremos las tres variables más importantes. Comenzamos con Credit Scoring, que, al igual que antes mayor es el resultado cuanto mayor probabilidad de impago existe. Existe una tendencia negativa constante desde un comienzo. Respecto al valor atípico en el punto 800 posiblemente se deba a la inclusión de los datos que no tenían Credit Scoring en la mediana.

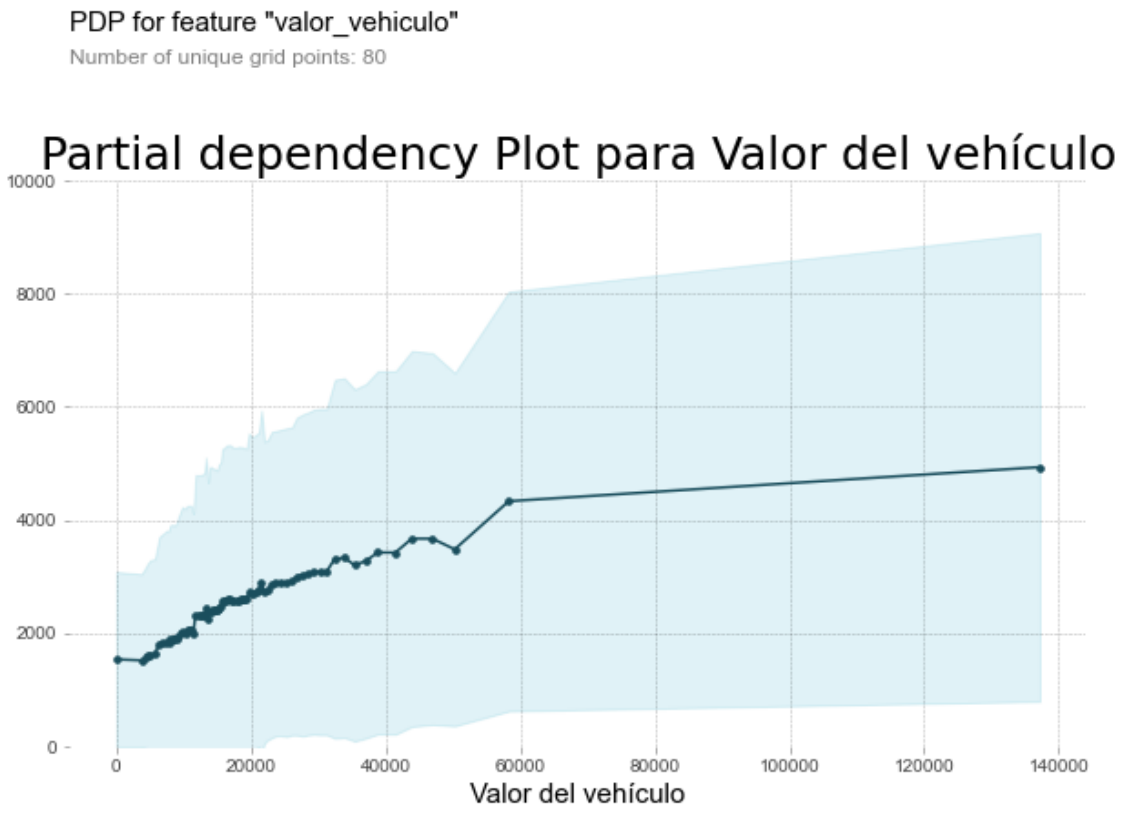
La segunda variable, el valor del vehículo, como es de esperar, la pendiente es positiva, mayor precio del vehículo más costosos son los siniestros. En último lugar, la variable de antigüedad del vehículo muestra un coste muy similar entre los cuatro valores que cotejamos, pero relativamente mayor para los más antiguos, los de nivel 4.

Figura 64. *Partial dependency Plot. Variable credit Scoring*



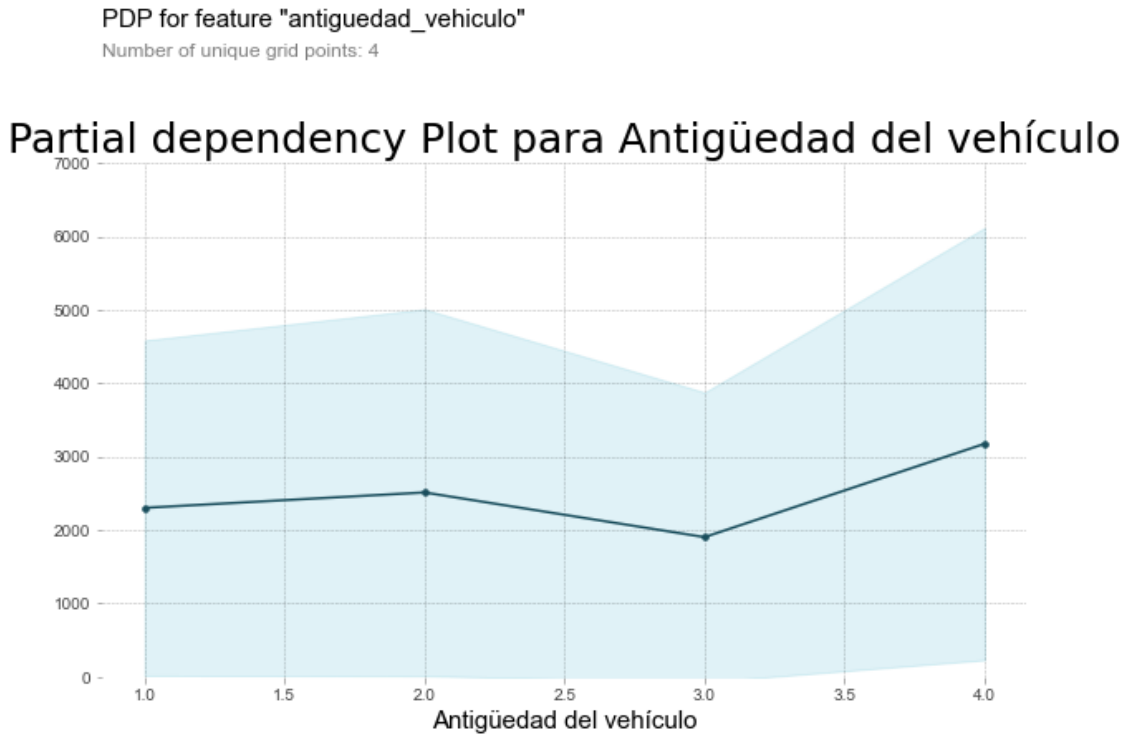
FUENTE: Elaboración propia

Figura 65. *Partial dependency plot. Variable Valor del vehículo*



FUENTE: Elaboración propia

Figura 66. *Partial dependency plot. variable Antigüedad del vehículo*



FUENTE: Elaboración propia

4.2.3 MODELO GBM. BURNING COST

Tras haber realizado las dos modelizaciones sobre la frecuencia y sobre la severidad debemos calcular la prima pura, también llamada Burning Cost. Al igual que en para los modelos lineales generalizados, la prima pura se obtiene de multiplicar la frecuencia modelada por el coste modelado.

Debemos recordar que ambas modelizaciones se han hecho con una muestra distinta, ya que en el caso de la severidad no se tienen en cuenta las pólizas sin siniestros. Para poder aplicar la fórmula descrita debemos otorgar valores a aquellas pólizas que no se pueden modelar. Al contrario que en los GLM, en este caso, al no tener coeficientes no podemos escoger como valor base la β_0, por ello, realizamos la media de la predicción.

En términos generales, los resultados obtenidos se pueden apreciar en la siguiente tabla.

***Tabla 16**. Resultados generales modelización GBM*

	Frecuencia modelada	Coste modelado	Burning Cost
Media	0,172	1374,731	255,130
Mediana	0,100	797,702	74,339

FUENTE: Elaboración propia

Además, mostramos una pequeña muestra de los resultados obtenidos.

***Tabla 17.** Ejemplo modelización de pólizas*

Póliza	Frecuencia Modelada	Coste modelado	Burning Cost (Prima Pura)
28	0,0737	824,52	60,77
29	0,0837	326,89	27,36
30	1,7126	9.935,13	17.014,90
31	0,1287	1.829,65	235,48
32	0,5603	1.030,59	577,44

FUENTE: Elaboración propia

5. COMPARATIVA Y MEJORA DE LOS RESULTADOS

5.1 COMPARATIVA DE LOS RESULTADOS

La comparativa entre ambas modelizaciones se va a llevar a cabo mediante distintos puntos, en primer lugar, compararemos los errores generalizados, donde cómo podemos ver en la tabla la modelización GBM es preferible tanto para la modelización de la frecuencia como para la severidad.

Tabla 18. *Comparativa error generalizado*

	Frecuencia	Severidad
GLM	0,722	0,758
GBM	0,703	0,503

FUENTE: Elaboración propia

La segunda manera para comparar será mediante el uso de clústeres. Un clúster es una agrupación o concentración de elementos, datos o variables con características similares donde se busca la máxima homogeneidad. En este caso, el primer clúster, el 0, está compuesto por los asegurados con menor riesgo y el último, el 8 será el peor valorado, ya sea por tener una mayor frecuencia, severidad o prima pura.

Realizaremos el análisis de conglomerados para la frecuencia modelada, la severidad modelada y finalmente, para la prima pura obtenida. Así pues, veremos las diferencias obtenidas por ambos modelos y podremos ver que modelización distribuye de una forma más homogénea los riesgos.

Antes de segmentar la muestra por clústeres eliminamos el 5% de cada extremo, ya que consideramos que estos riesgos no son representativos, es decir, las pólizas con menor frecuencia, severidad y prima pura no son algo común, al igual que aquellos con unos niveles elevados de estas tres características y pueden llegar a generar cierto sesgo. De esta manera, contamos con un 90% de la muestra que dividimos en 9 clústeres.

A parte de las variables que definen a cada asegurado, su prima pura también los define, ya que si es baja muestra que el riesgo a tener un siniestro es bajo y si es alta que la probabilidad de tenerlo es elevada. Como cada cartera contiene multitud de asegurados, es complejo y poco eficiente analizar a cada asegurado. Por ello, para analizar como se comporta la cartera se crean estas agrupaciones. El uso de clústeres es muy interesante por dos razones. La primera de ellas, como ya se ha explicado es que sirve para saber a priori la probabilidad de siniestros que puede llegar a tener un asegurado.

La segunda razón está relacionada con el negocio, tanto de nueva producción (nuevos asegurados para la compañía) como cartera (asegurados que ya están en la compañía). Una vez se ha creado el modelo de tarificación final y se crean los distintos clústeres sobre los asegurados debe controlarse que se mantiene la proporción en los clústeres, ya que

con la entrada de nuevos clientes puede surgir un cambio de tendencia que genere que únicamente contrates clientes que pertenecen a las peores agrupaciones o que se marchen asegurados de los mejores conjuntos.

Mediante esta segmentación puedes controlar como se distribuyen los asegurados, ya que, al estar en un mercado competitivo y cambiante, el resto de las empresas puede mejorar su modelos, bajar las primas para adquirir un mayor número de clientes, puede que los modelos utilizados dejen de capturar el riesgo correctamente, etc. Gracias a esta segmentación a medida que vaya evolucionando la cartera con el paso del tiempo podremos estudiar si realmente estamos analizando bien cada riesgo.

Comenzamos con la comparativa de la frecuencia. Las tablas muestran la frecuencia para cada clúster, la variación de la frecuencia respecto al clúster inicial y respecto al clúster previo. Como se puede apreciar, el crecimiento en ambas modelizaciones es muy similar, aunque en el caso del GBM (tabla azul) observamos como la variación respecto al clúster previo es más elevada, capturando de una manera más precisa a los asegurados con mayor frecuencia, ya que aquellos que son el clúster con mayor frecuencia para el modelo GLM, para el GBM se distribuyen entre los tres peores.

La variación respecto al nivel base tiene menor pendiente en el caso del GLM lo que muestra una menor granularidad de este modelo.

***Tabla 19**. Resultados Clústeres. Modelización GLM*

	Frecuencia	Variación respecto clúster 0	Variación respecto clúster previo
Clúster 0	0,030	1,00	-
Clúster 1	0,051	1,70	1,66
Clúster 2	0,072	2,40	1,43
Clúster 3	0,097	3,23	1,35
Clúster 4	0,128	4,27	1,31
Clúster 5	0,173	5,77	1,35
Clúster 6	0,240	8,00	1,39
Clúster 7	0,355	11,83	1,48
Clúster 8	0,671	22,37	1,89

FUENTE: Elaboración propia

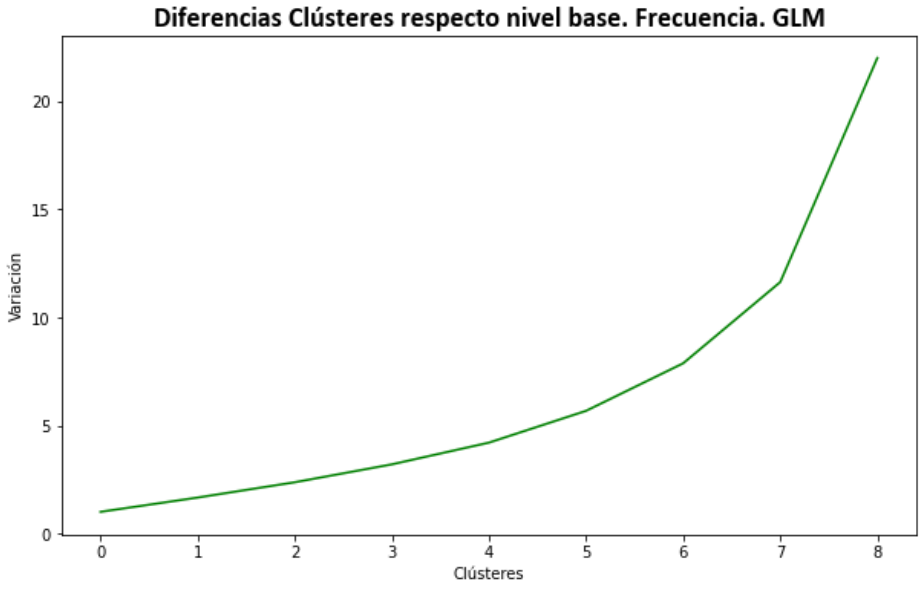
Tabla 20. *Resultados Clústeres. Modelización GBM*

	Frecuencia	Variación respecto clúster 0	Variación respecto clúster previo
Clúster 0	0,025	1,00	-
Clúster 1	0,041	1,66	1,66
Clúster 2	0,060	2,43	1,46
Clúster 3	0,084	3,40	1,40
Clúster 4	0,119	4,77	1,40
Clúster 5	0,168	6,78	1,42
Clúster 6	0,243	9,80	1,45
Clúster 7	0,375	15,09	1,54
Clúster 8	0,741	29,83	1,98

FUENTE: Elaboración propia

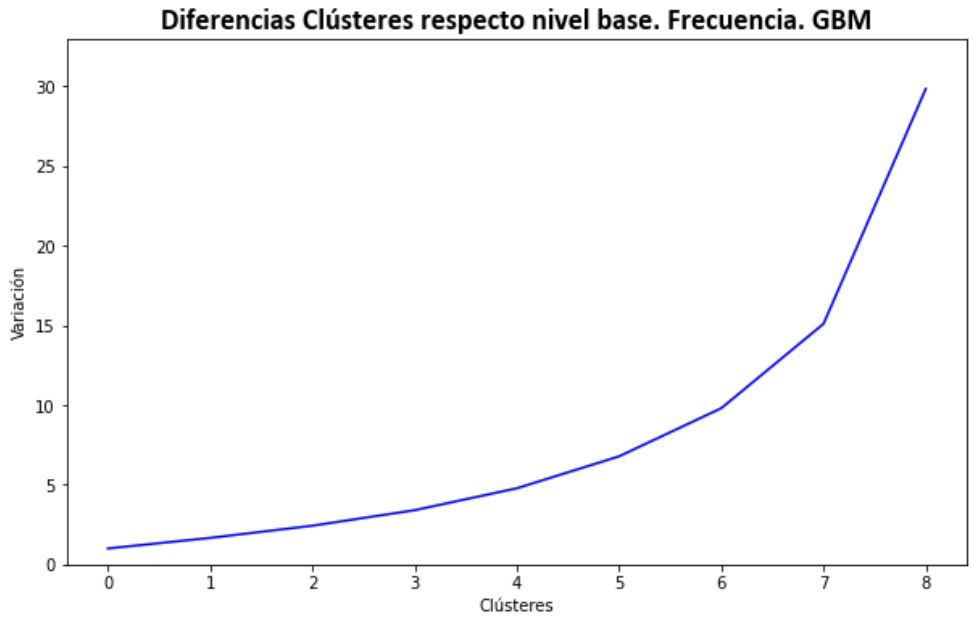
En las figuras 66 y 67 observamos como la modelización GBM respecto al nivel base parece tener una mayor progresión respecto al nivel inicial que la otra vía de estudio del riesgo. Por ello es importante analizar la primera y tercera columna de las tablas previas donde se muestra la variación respecto el nivel anterior y el alcance del clúster. Ambas parten desde un primer punto muy similar pero el último nivel captura una mayor frecuencia en el caso de GBM.

Figura 67. *Variaciones clústeres de la frecuencia respecto nivel base. Modelización GLM*



FUENTE: Elaboración propia

***Figura 68**. Variaciones clústeres de la frecuencia respecto nivel base. Modelización GBM*



FUENTE: Elaboración propia

El siguiente paso es comparar la severidad, en este caso sí que se muestran mayores diferencias. En el caso del GBM agrupa en 5 clústeres distintos a la misma media de siniestralidad que el GLM en 2. La primera modelización muestra un crecimiento de la siniestralidad por clúster cercano al 30%, inferior al casi 40% del GBM, pese a ello, al haber agrupado de manera distinta la siniestralidad otorga mayores diferencias a los siniestros más elevados.

***Tabla 21**. Resultados clústeres. Modelos de severidad GLM*

	Severidad	Variación respecto clúster 0	Variación respecto clúster previo
Clúster 0	718	1,00	-
Clúster 1	964	1,34	1,34
Clúster 2	1.194	1,66	1,24
Clúster 3	1.551	2,16	1,30
Clúster 4	1.971	2,75	1,27
Clúster 5	2.463	3,43	1,25
Clúster 6	3.282	4,57	1,33
Clúster 7	4.603	6,41	1,40
Clúster 8	7.351	10,24	1,60

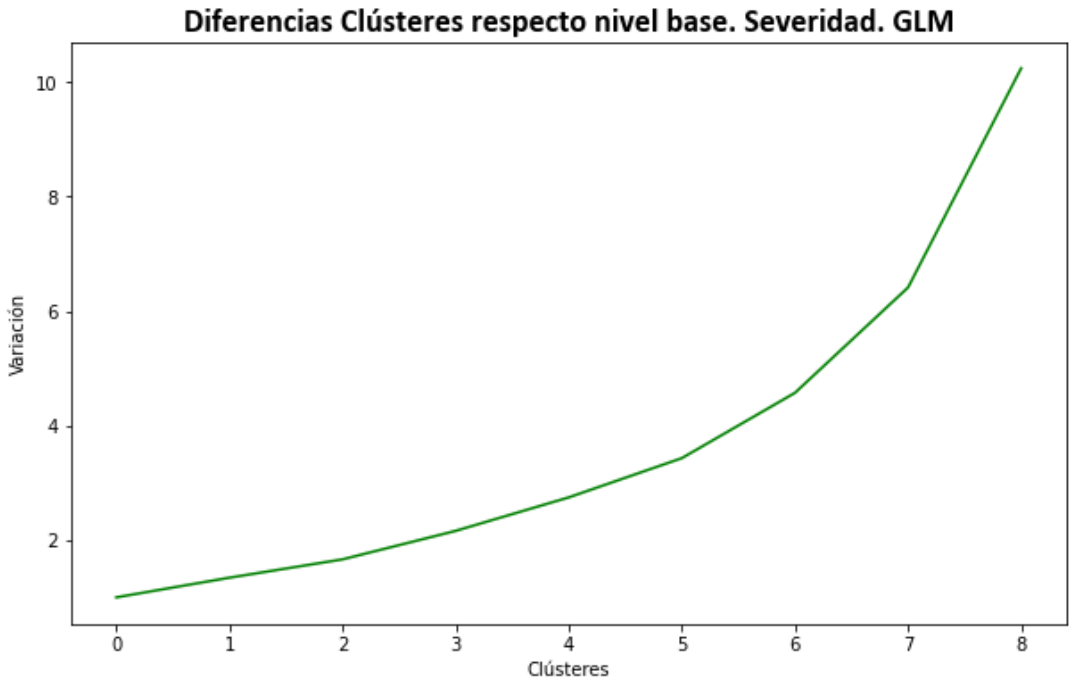
FUENTE: Elaboración propia

***Tabla 22**. Resultados clústeres. Modelos de severidad GBM*

	Severidad	Variación respecto clúster 0	Variación respecto clúster previo
Clúster 0	291	1,00	-
Clúster 1	391	1,34	1,34
Clúster 2	516	1,77	1,32
Clúster 3	687	2,36	1,33
Clúster 4	931	3,20	1,36
Clúster 5	1.290	4,43	1,39
Clúster 6	1.872	6,43	1,45
Clúster 7	2.923	10,04	1,56
Clúster 8	5.701	19,59	1,95

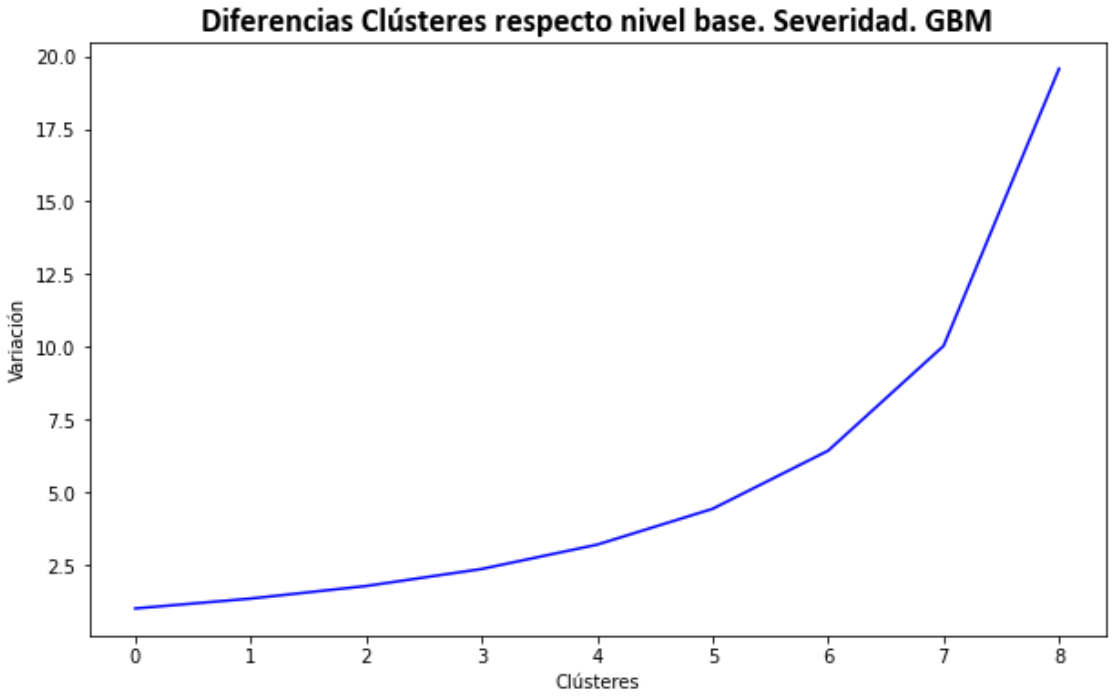
En el gráfico mostramos la variación respecto al nivel base, la segunda columna de las tablas previas. El GBM muestra gran importancia en los siniestros más leves, los más comunes y a medida que el coste empieza a aumentar, la pendiente también lo hace, alcanzando un incremento de casi el 200% entre el penúltimo y último clúster, por el contrario, la pendiente del modelo GLM es más paulatina.

***Figura 69**. Variaciones de la severidad respecto nivel base. Modelización GLM*



FUENTE: Elaboración propia

***Figura 70**. Variaciones de la severidad respecto nivel base. Modelización GBM*



FUENTE: Elaboración propia

Tras analizar los distintos segmentos o divisiones de la frecuencia y la severidad debemos analizar cómo se segmenta la prima pura. La prima pura al ser el resultado de la multiplicación de la frecuencia y la severidad engloba los resultados previos, pero al tratarse de un análisis académico se han mostrado los resultados previos con el fin de mostrar un comparativa completa entre ambos estudios.

La diferencia principal entre ambos casos se debe a como analizan los distintos riesgos, el GBM muestra una granularidad mucho mayor al que la modelización inicial, la prima pura del GLM tiene una distribución mucho más estrecha que la calculada mediante Machine Learning. En ambos casos, se aprecia un salto en el último clúster, que agrupa a los mayores riesgos, pero hay gran diferencia entre un resultado y otro. No hay un incremento constante respecto al clúster previo en ningún caso, pero si se aprecia una mayor pendiente en la segunda modelización, como se puede ver en la tercera columna de la tabla 24.

De esta manera podemos ver una clara diferencia donde la modelización más utilizada hasta el momento en el sector tiende a mostrar una prima pura menor a partir del quinto clúster, o cuatro si comenzamos por el 0. De esta manera, mediante la primera modelización se otorga un menor riesgo a los clientes de los últimos clústeres en comparación al GBM.

Por el contrario, la modelización GBM detecta que los riesgos más elevados pueden segmentarse en tres tramos por los dos de la primera predicción, obteniendo así una mayor segmentación de la muestra.

Tabla 23. *Resultados clústeres. Prima pura. Modelización GLM*

	Prima Pura	Variación respecto clúster 0	Variación respecto clúster previo
Clúster 0	18	1,00	-
Clúster 1	29	1,66	1,66
Clúster 2	42	2,37	1,43
Clúster 3	57	3,23	1,36
Clúster 4	76	4,27	1,32
Clúster 5	107	6,04	1,40
Clúster 6	155	8,76	1,45
Clúster 7	256	14,38	1,65
Clúster 8	984	55,63	3,85

FUENTE: Elaboración propia

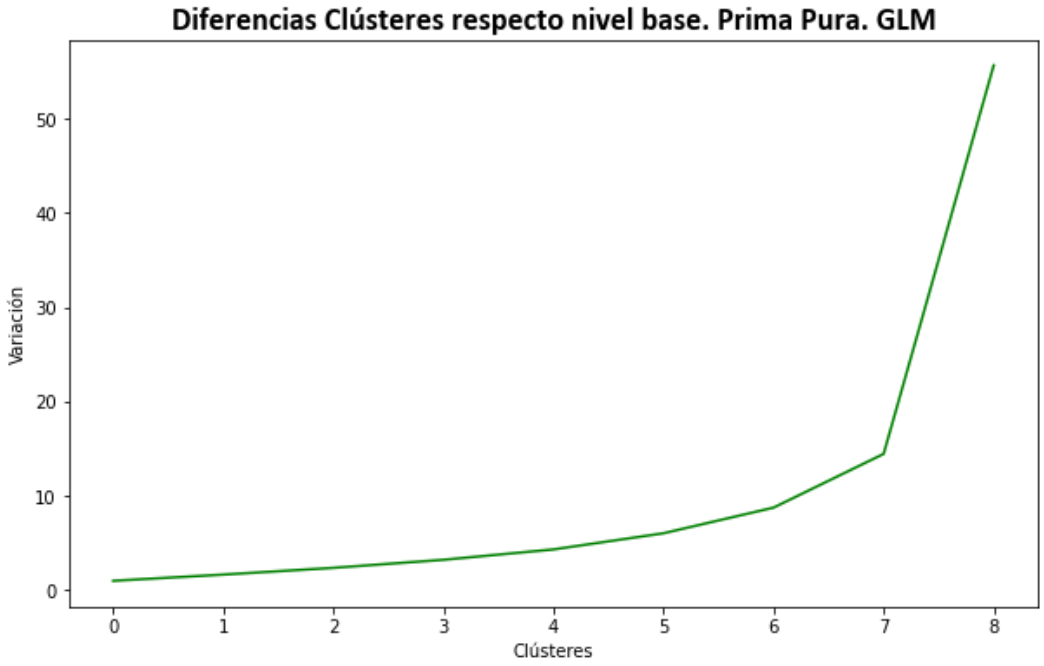
Tabla 24*. Resultados clústeres. Prima pura. Modelización GBM*

	Prima Pura	Variación respecto clúster 0	Variación respecto clúster previo
Clúster 0	12	1,00	-
Clúster 1	22	1,84	1,84
Clúster 2	37	3,07	1,66
Clúster 3	59	4,90	1,60
Clúster 4	95	7,97	1,63
Clúster 5	165	13,80	1,73
Clúster 6	318	26,62	1,93
Clúster 7	703	58,75	2,21
Clúster 8	2.313	193,33	3,29

FUENTE: Elaboración propia

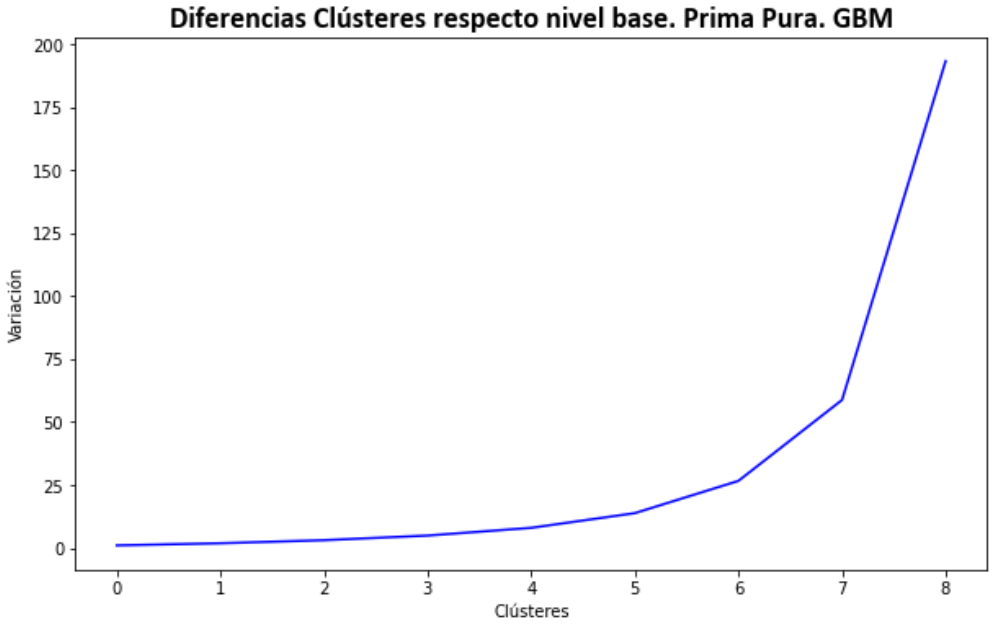
En las tablas donde hemos mostrado las variaciones de los clústeres incluíamos la segunda y tercera columna porque es importante tener ambas en cuenta. Una variación muy elevada respecto al nivel base no significa una mayor y mejor segmentación, ya que estas variaciones no significan que la distribución de los clústeres agrupe una mayor cantidad de información, puesto que depende del valor del primer segmento. Pese a ello, en este caso, la modelización GBM abarca un mayor rango tanto en la cola de la izquierda, en el clúster 0, como en la cola de la derecha, con el clúster 8.

Figura 71. *Variaciones de la prima pura respecto el nivel base. Modelización GLM*



FUENTE: Elaboración propia

Figura 72. *Variaciones de la prima pura respecto nivel base. Modelización GBM*

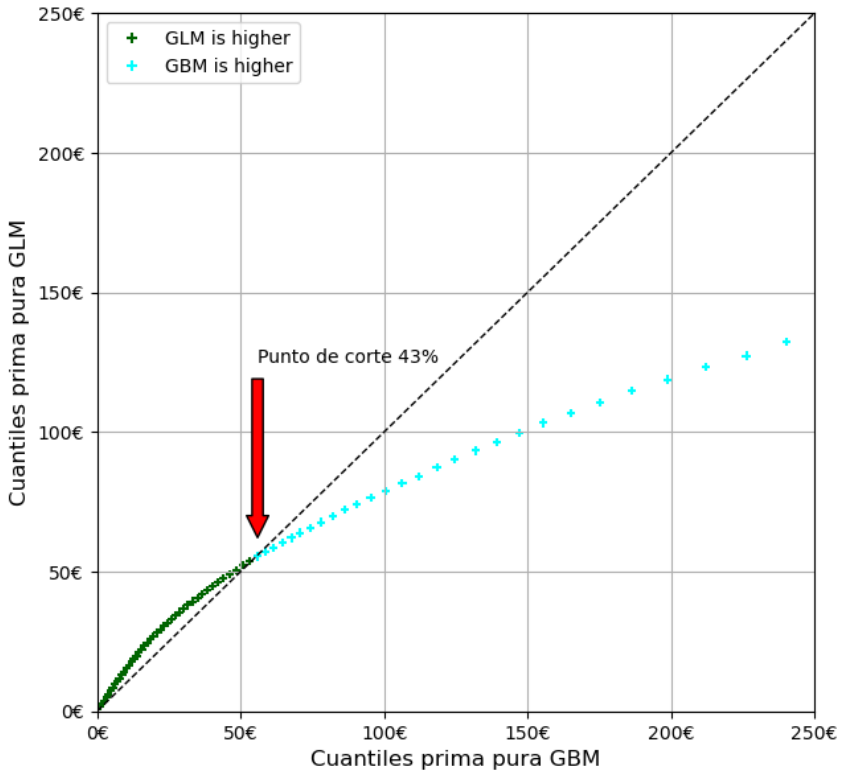


FUENTE: Elaboración propia

Tras analizar y comparar los tres aspectos, podemos ver una mayor partición en el caso de la modelización mediante Gradient Boosting con un mayor énfasis en los asegurados con mayor riesgo. Como podemos ver en la siguiente figura, en el primer 43% las primas GLM son más caras respecto a las obtenidas en el segundo estudio, a partir de ahí y como hemos visto, al poder segmentar mejor la muestra en los riesgos más elevados, la prima es más elevada según la modelización GBM. Es decir, mediante el segundo estudio a aquellos individuos con riesgos más elevados se le cobra una prima superior al caso del GLM y a aquellos con menor probabilidad de sufrir un siniestro tienen una menor prima.

Es decir, mediante el segundo estudio, aquellos individuos con menor riesgo tienen una menor prima y aquellos con mayor riesgo tienen un recargo en comparación al GLM

Figura 73. *Comparativa de la prima pura*



FUENTE: Elaboración propia

Tras haber analizado el comportamiento de la frecuencia, la severidad y la prima pura segmentado por distintos clústeres debemos analizar las variables para obtener una imagen más clara de las diferencias entre ambos casos. Para las variables cuantitativas utilizamos la media para cada conjunto y para las categóricas su moda.

Comenzamos la comparativa entre las variables cuantitativas. La edad se mantiene constante en los primeros tres clústeres en ambos casos y desde ese punto comienza a disminuir, siendo el intervalo mayor en el caso del GLM. Ambos estudios tienen una

tendencia muy similar y que concuerda con el análisis ex-ante, ya que los conductores más jóvenes tienden a tener un mayor riesgo ante la falta de experiencia.

El riesgo crediticio también es descendente, ya que aquellos con mayor probabilidad de impago tienden a tener más siniestros, en este caso el GBM muestra un mayor intervalo entre el menor y mayor clúster. El índice de tráfico es ascendente, es decir, a menor calidad de conducción te encuentras en un mayor clúster. Al contrario que en el caso anterior el rango es más amplio en la primera modelización.

La antigüedad del vehículo se muestra de manera contrapuesta en ambos casos, en el GLM los vehículos más jóvenes se encuentran en los primeros grupos mientras que en el GBM existe una pendiente negativa, lo vehículos más antiguos se encuentran en los clústeres de menor riesgo. Respecto a la última variable de vehículo la tendencia en ambos casos es la misma con vehículos más baratos en los conjuntos iniciales, pero existe una mayor homogeneidad en el GLM donde apenas hay variaciones.

Es importante mencionar dos cuestiones. En primer lugar, todas las variables los conjuntos muestran una continuidad de la tendencia, es decir, en el caso de ser una variable creciente ningún clúster muestra un dato inferior al anterior, salvo en el caso de la edad. En segundo lugar, únicamente mostramos la media de cada variable para cada clúster lo que significa que puede haber personas jóvenes en el primer clúster o personas con buen riesgo crediticio en el último. Simplemente se muestran las tablas para tener una idea más concisa de como éstos están formados.

***Tabla 25**. Media de variables cuantitativas por clúster. Modelización GLM*

	Pólizas	Edad	Credit Scoring	Índice de trafico	Antigüedad del vehiculo	Valor del vehiculo
Clúster 0	5.989	52	723,55	96,96	2,64	19.502,31
Clúster 1	5.989	52	725,16	97,39	2,62	19.583,62
Clúster 2	5.989	51	721,37	101,34	2,66	19.307,29
Clúster 3	5.989	50	710,69	106,42	2,64	19.095,96
Clúster 4	5.988	49	700,48	108,54	2,68	19.064,14
Clúster 5	5.989	47	679,86	108,75	2,73	19.122,91
Clúster 6	5.989	47	663,53	111,35	2,75	19.194,06
Clúster 7	5.989	46	637,82	114,29	2,77	19.227,00
Clúster 8	5.989	42	556,97	115,21	2,78	19.742,49

FUENTE: Elaboración propia

***Tabla 26**. Media de variables cuantitativas por clúster. Modelización GBM*

	Pólizas	Edad	Credit Scoring	Índice de trafico	Antigüedad del vehiculo	Valor del vehiculo
Clúster 0	5.989	51	750,72	102,93	3,10	14.630,63
Clúster 1	5.989	52	741,42	106,31	2,98	15.548,82
Clúster 2	5.989	51	730,41	106,39	2,86	17.059,01
Clúster 3	5.989	50	717,94	106,46	2,78	18.166,80
Clúster 4	5.988	49	705,99	107,59	2,67	19.506,16
Clúster 5	5.989	49	686,43	107,79	2,55	21.166,23
Clúster 6	5.989	46	655,56	107,73	2,43	22.847,80
Clúster 7	5.989	44	599,23	106,19	2,42	22.843,29
Clúster 8	5.989	42	515,97	107,37	2,39	23.112,58

FUENTE: Elaboración propia

En el caso de las variables categóricas observamos resultados muy similares en ambas modelizaciones para todas las variables. En el caso del sexo esto ocurre porque como ya hemos mostrado previamente hay grandes diferencias en la frecuencia de accidentes de mujeres y hombres y acaba generando una tendencia hacia que éstas acaben en clústeres con menor riesgo.

En las otras dos variables, al escoger la moda, aquellos valores que resaltan sobre los demás en la muestra inicial son los que acaban apareciendo, por ello en el área de residencia aparecen únicamente la primera y tercera ciudad, ya que la primera tiene unos índices de frecuencia y severidad muy bajitos y además tiene población suficiente en comparación con la tercera ciudad (la más poblada) que es la que mayor número de asegurados tiene en 6 y 8 clústeres. Respecto al tipo de vehículo, la berlina de 2 volúmenes es el coche más utilizado por gran diferencia y por ello es el que más aparece en todos los conjuntos, a pesar de ello, es interesante exponer que el todo terreno a medida que los clústeres aumentaban obtenía mayor importancia.

***Tabla 27.** Media de variables categóricas por clúster. Modelización GLM*

	Sexo	Área de residencia	Tipo de Vehículo
Clúster 0	Femenino	1	Berlina 2 vol
Clúster 1	Femenino	1	Berlina 2 vol
Clúster 2	Femenino	1	Berlina 2 vol
Clúster 3	Femenino	3	Berlina 2 vol
Clúster 4	Femenino	3	Berlina 2 vol
Clúster 5	Femenino	3	Berlina 2 vol
Clúster 6	Masculino	3	Berlina 2 vol
Clúster 7	Masculino	3	Berlina 2 vol
Clúster 8	Masculino	3	Berlina 2 vol

FUENTE: Elaboración propia

Tabla 28. *Media de variables categóricas por clúster. Modelización GBM*

	Sexo	Área de residencia	Tipo de Vehículo
Clúster 0	Femenino	1	Berlina 2 vol
Clúster 1	Femenino	3	Berlina 2 vol
Clúster 2	Femenino	3	Berlina 2 vol
Clúster 3	Femenino	3	Berlina 2 vol
Clúster 4	Femenino	3	Berlina 2 vol
Clúster 5	Masculino	3	Berlina 2 vol
Clúster 6	Masculino	3	Berlina 2 vol
Clúster 7	Masculino	3	Berlina 2 vol
Clúster 8	Masculino	3	Berlina 2 vol

FUENTE: Elaboración propia

Mostramos en la siguiente tabla los clústeres de la prima pura de las 5 pólizas que mostramos de ejemplo para ambas modelizaciones. Como podemos ver de las 5 únicamente coinciden 2, la póliza número 30, que no pertenece a ningún clúster porque tiene un riesgo muy elevado (se encuentra en el 5% superior eliminado en ambas modelizaciones) y la póliza número 32, que se encuentra en el peor clúster. La póliza 28 es peor considerada en el GLM, al contrario que la póliza número 29.

Con esta muestra de únicamente 5 pólizas podemos apreciar como ambos estudios conllevan una modelización del riesgo distinta aunque la diferencia de grupos tampoco es muy abultada.

Tabla 29. *Comparativa de Clústeres*

Póliza	Clúster GLM	Clúster GBM
28	Cluster 5	Cluster 4
29	Cluster 3	Cluster 5
30	-	-
31	Cluster 7	Cluster 8
32	Cluster 8	Cluster 8

FUENTE: Elaboración propia

Recapitulando toda la información de este apartado obtenemos unos resultados similares en ambos estudios siendo el GBM el más preciso y el que mejor segmenta. Esta segmentación es clave en el sector asegurador ya que puedes premiar al cliente que no genera costes porque no tiene siniestros y escoger una prima adecuada parta aquellos conductores con mayor riesgo. Esta diferencia en la segmentación del riesgo es realmente notoria en la distribución por clústeres de la prima pura donde hemos encontrado una gran amplitud entre los valores más bajos y los más elevados.

Respecto al análisis de las variables, destaca la ausencia de linealidad como hemos visto en los gráficos de dependencia parcial en el caso del GBM y las diferencias en algunas variables entre los clústeres como puede ser la edad media del vehículo o la diferencia de valores mostrados entre las dos modelizaciones como es el caso del valor del vehículo.

En definitiva, ambas modelizaciones muestran resultados similares, aunque la mejor segmentación, mostrando unas primas puras inferiores a los mejores riesgos y unas primas más elevadas a aquellos con mayor siniestralidad muestran que la mejor modelización es la más innovadora, el GBM.

5.2 MEJORA DE LOS RESULTADOS

Tras analizar los resultados obtenidos surgen vías de mejora en distintos aspectos. En primer lugar, la base de datos podría mejorarse ya que como hemos visto contamos con un número reducido de variables. En este caso al tratarse de un estudio académico es suficiente para mostrar cómo actúan ambas modelizaciones, pero si realmente necesitásemos utilizar estos modelos sería preferible contar con más factores de riesgo.

Algunos ejemplos podrían ser si el pago del seguro es con tarjeta o cuenta corriente, el periodo entre la entrada en vigor de la póliza y la ocurrencia del siniestro, la potencia y peso de los vehículos. Algunas variables relacionadas con el conductor que sería interesante conocer serían su nacionalidad, la fecha de obtención del carné de conducir o si hay un segundo conductor. También sería interesante contar con variables geográficas como el código postal.

El siguiente punto de mejora son los modelos. En este estudio únicamente hemos realizado modelo para la severidad y la frecuencia, para mejorar el nivel de estudio y conocimiento de los clientes se pueden incluir multitud de modelos, como pueden ser los modelos de retención, para evitar que los clientes se cambien a otra aseguradora, modelos de elasticidad, para averiguar la variación máxima que esta, dispuesta a soportar el cliente sobre la prima, modelos de ingeniería inversa, para estimar el precio de las primas de los clientes o modelos para calcular el margen de beneficio.

6. ANÁLISIS DE IMPACTO. PUNTO DE VISTA DEL NEGOCIO

Tras el análisis técnico donde comparamos entre dos alternativas de modelización, una de ellas asentada en el sector actuarial y otra, más novedosa, mediante la aplicación del *machine learning* debemos llevar a cabo el análisis de impacto en primer lugar y, en segundo lugar, dar un punto de vista de negocio con el fin de mejorar la cartera.

Comenzamos con el cálculo del ratio de siniestralidad, que se define como el porcentaje de las pérdidas cubierto por la prima. En el caso de que supere el 100% significa que con esa póliza no estamos cubriendo las pérdidas, en caso contrario, si el ratio se muestra por debajo del 100% significa que obtenemos beneficio con dicha prima.

$$Ratio\ siniestralidad = \frac{Siniestralidad}{Prima\ anual} * 100 \qquad (15)$$

En la tabla 30 observamos los resultados de nuestra cartera donde la prima total se encuentra por encima de la siniestralidad total. A pesar de ello, el peso de la siniestralidad total sobre las primas no es muy elevada lo que puede generar que en caso de que surjan siniestros esa diferencia disminuya y la cartera genere pérdidas.

Tabla 30. *Ratio de siniestralidad*

Prima total	43.272.679,76
Siniestralidad total	38.945.347,26
Ratio de siniestralidad	90,00%

FUENTE: Elaboración propia

De cara a la renovación de cartera debemos aplicar cambios sobre la prima a cada tomador con el fin de que exista una mayor holgura o diferencia entre las primas cobradas y los costes.

Para ambas modelizaciones hemos eliminado los siniestros con valores punta, menos del 1% de la muestra total, que son los siniestros con costes por encima de 16.500 €, para evitar una distorsión en el estudio, pero para la modificación de las nuevas primas deben ser tenidos en cuenta. Para tratar estos costes se pueden llevar a cabo dos enfoques, el primero de ellos y siguiendo la línea actual de la cartera, mediante la mutualización de los mismos, recargando a todos los asegurados una misma cuantía fija adicional. La segunda opción es pasar a una distribución segmentada de las primas donde aquellos con un menor número de siniestros tendrían una menor subida que aquellos con una mayor probabilidad de sufrir siniestros. Es decir, se hará una distribución del monto final que suponen los siniestros puntas en proporción a la valoración realizada por los clústeres.

Ambas vías tienen ventajas y desventajas. En el primer caso, la mutualización de los valores punta, la ventaja principal es que puedes trabajar con toda la muestra para los diversos análisis y estudios posteriores como estas subidas de primas. Por el lado contrario, las desventajas principales son la caída de los asegurados con menor riesgo, ya que la elasticidad de cada uno de los riesgos y perfiles de la cartera es diferente y una subida de la prima puede incurrir en que no acepten una subida tan elevada de la prima puesto que pese a no haber tenido siniestros su prima se ve aumentada y un incremento de la selección adversa, es decir, aquellos con multitud de siniestros únicamente sufren una subida leve de su prima en comparación al coste generado, por tanto, mediante la mutualización se atraen riesgos altos y se pierden asegurados con una buena siniestralidad.

El método de la asignación segmentada de los puntas permite identificar los niveles de riesgo que suponen. De esta forma, los individuos sin apenas incidentes sufren un incremento menor que aquellos que han sufrido múltiples siniestros. Se puede llevar a cabo de distintas maneras, en este caso, las variaciones en las primas se harán basándonos en distintos clústeres, que pueden ser por tipo de vehículo, área de residencia o por los clústeres calculados en el apartado anterior sobre el Burning Cost.

Comenzamos con el primer enfoque, mediante la mutualización de la prima. En este caso, como vemos en la tabla 30, las 506 pólizas excluidas para el cálculo de los modelos generan un coste por encima de los 15 millones de euros. Si dividimos ese coste entre el número total de pólizas obtenemos que cada póliza debe aumentar su precio en casi 254€. De esta manera, cada conductor pasaría de tener que pagar 716,53€ a 970,5€.

Tabla 31. *Mutualización del coste.*

Mutualización del coste	
Coste siniestros punta punta	15.337.557,68
Número de pólizas	60.392
Subida de prima	253,97

FUENTE: Elaboración propia

Con este incremento por encima del 35% la presión ejercida sobre la prima de renovación incurriría en un riesgo de aumento de caídas. Mediante la segmentación de precios se trata de premiar a aquellos que no han dado ningún siniestro y aquellos con una peor siniestralidad otorgarles una mayor subida de prima.

Esta segmentación se puede llevar a cabo mediante una visión diferente del riesgo, como puede ser por tipo de vehículo en el caso de que observáramos una relación directa o plausible que pudiera generar una alta siniestralidad. También se puede hacer por provincias, donde las provincias con ciudades más grandes muestran una mayor siniestralidad y aquellas más cercanas a la costa lo contrario. Una tercera opción sería por el nivel de clúster de Burning Cost, donde nos apoyaríamos en los cálculos de cada modelización que hemos explicado en el apartado anterior.

Para los dos primeros casos no es necesario observar los resultados obtenidos en las modelizaciones, pero surgen ciertos problemas. En el caso del tipo de vehículo contamos con una muestra muy diversa donde en general predominan las berlinas, los todo terreno y los derivados de turismo (también llamados SUVs). Respecto al resto, nos encontramos con una muestra realmente escasa, en especial para las autocaravanas o los Targa, por lo que habría que hacer agrupaciones y podríamos llegar a homogeneizar la muestra.

El caso de la segmentación por provincias es realmente útil pero también tiene aspectos negativos, como puede ser el hecho de las zonas limítrofes, es decir, clientes que vivan en una provincia, pero lleven a cabo el uso del coche en otra. Esto ocurre en provincias cercanas a las grandes capitales del país como puede ser Madrid, Barcelona o Sevilla. Además, en nuestro caso, la base de datos únicamente nos muestra 5 provincias, por tanto, la segmentación puede que no sea óptima.

Finalmente, nos decantamos por una segmentación que sí utilice información de la modelización. Debemos ajustar estos costes a cada clúster calculado, que como hemos definido previamente es una variable nominal ordinal que mide el riesgo de menor a mayor siendo el clúster 0 aquel con menor nivel de riesgo y el 8 los que mayor nivel tienen. Además, a aquellas pólizas que se encuentren fuera porque no se ha modelizado sobre ellas deberán tener un recargo adicional.

Tabla 32. *Variación segmentada de la prima por Clústeres.*

	Pólizas	Variación	Prima actual	Nueva Prima	PRIMA TOTAL
Clúster 0	8.984	-5%	716,53	680,70	6.115.440,24
Clúster 1	5.989	0%	716,53	716,53	4.291.298,17
Clúster 2	5.989	5%	716,53	752,36	4.505.863,08
Clúster 3	5.989	10%	716,53	788,18	4.720.427,99
Clúster 4	5.988	20%	716,53	859,84	5.148.697,97
Clúster 5	5.989	30%	716,53	931,49	5.578.687,62
Clúster 6	5.989	40%	716,53	1.003,14	6.007.817,44
Clúster 7	5.989	50%	716,53	1.074,80	6.436.947,26
Clúster 8	5.989	100%	716,53	1.433,06	8.582.596,34
EXTRA	3.497	200%	716,53	2.149,59	7.517.116,23
Total	60.392	-	-	-	58.904.892,33

FUENTE: Elaboración propia

Mostramos una posible solución donde las primas obtenidas con los cambios superan ligeramente a la mutualización. Como se aprecia, el clúster 0, es un mayor número de pólizas ya que incluimos al 5% que excluimos en los cálculos para la división de clústeres. A falta de un proceso de optimización con una calibración de techos y suelos, los descuentos y recargos propuestos siguen una distribución lineal.

Éstos, al ser los mejores asegurados obtienen un descuento del 5%. El clúster 1 se mantiene exactamente igual y desde ese punto se lleva a cabo un incremento paulatino hasta llegar al clúster 8 donde la póliza duplica el precio inicial. La fila denominada como

extra son el 5% extraído de las modelizaciones y las 502 pólizas consideradas como puntas, como se puede ver su nueva tarifa triplica el valor inicial, mediante esta nueva prima se espera que el cliente se marche a otra aseguradora o que por el contrario trate de generar una menor siniestralidad esperada durante el próximo periodo.

Comparando ambas posibilidades observamos cómo gracias a la segmentación por clústeres únicamente los tres últimos y el nivel extra superan los 970 € de la prima mutualizada, es decir, conseguimos que aquellos con un riesgo más elevado deban pagar una mayor cantidad respecto al resto de asegurados.

El análisis económico no acaba aquí, ya que como podemos ver el incremento de las pólizas es realmente abultado y por tanto es probable que multitud de clientes decidan marcharse. Para evitar una posible fuga masiva y pese a que pueden generarse pérdidas una solución es obtener una menor prima total y así mitigar la pérdida esperada, donde es preferible el cobro de una prima por debajo de su óptimo.

Para ello, existe la creación de diversos modelos sofisticados de retención de cartera, donde se analizan diversos aspectos como puede ser la elasticidad del cliente respecto la variación en la subida del precio de la prima, las anulaciones antes de la finalización del contrato o modelos de ingeniería inversa para hallar los modelos y ofertas de las empresas competidoras entre otros.

En nuestro caso, realizaremos un modelo de techos y suelos, es decir, pondremos límites de manera que la variación de la prima no sea lo suficientemente grande como para que los clientes decidan marcharse. Esta metodología se utiliza tanto para descuentos de prima como para incrementos, en nuestro caso, únicamente aplicaremos techos ya que al estar en una cartera con unos costes tan cercanos a las primas recaudadas no se bajarán los precios de manera general.

El incremento de la prima será progresivo manteniendo los primeros niveles como antes y únicamente variando los dos últimos poniendo un incremento de prima máximo de 60%. Mostramos las variaciones respecto la idea principal en la tabla 32, donde como vemos el ingreso disminuiría en torno a los 5 millones de euros.

Tabla 33. *Variación segmentada de la prima por Clústeres. Techos modificados.*

	Pólizas	Variación	Prima actual	Nueva Prima	PRIMA TOTAL
Clúster 0	8.984	-5%	716,53	680,70	6.115.440,24
Clúster 1	5.989	0%	716,53	716,53	4.291.298,17
Clúster 2	5.989	5%	716,53	752,36	4.505.863,08
Clúster 3	5.989	10%	716,53	788,18	4.720.427,99
Clúster 4	5.988	20%	716,53	859,84	5.148.697,97
Clúster 5	5.989	30%	716,53	931,49	5.578.687,62
Clúster 6	5.989	40%	716,53	1.003,14	6.007.817,44
Clúster 7	5.989	50%	716,53	1.074,80	6.436.947,26
Clúster 8	5.989	55%	716,53	1.110,62	6.651.512,16
EXTRA	3.497	60%	716,53	1.146,45	4.009.128,66
Total	60.392	-	-	-	53.465.820,58

FUENTE: Elaboración propia

Tras realizar estos cambios, debemos llevar a cabo un seguimiento de la cartera con el fin de saber si las medidas llevadas a cabo han tenido el efecto deseado analizando el ratio de retención de cartera. Este tipo de maduraciones tardan entre 3 y 6 meses.

Antes de la aplicación del cambio, debemos hacer un *back testing*, analizar a posteriori como se comporta el modelo, sobre renovaciones anteriores y analizar si la distribución del modelo de techos y suelos para cada clúster tiene la capacidad de segmentar de manera adecuada los recargos a los clientes. Además, se debería realizar un modelo de elasticidad, para saber cuánto le puedo subir la prima, uno de negociación, para saber si el cliente quiere negociar la subida de la prima, y por último, un modelo de retención, donde tras haber modificado la prima y haber negociado saber si el cliente se marcha de la compañía o no.

7. CONCLUSIONES

El sector asegurador se encuentra en una constante evolución en todos los ámbitos, principalmente gracias a los avances tecnológicos. En este estudio se lleva a cabo la comparativa entre la modelización clásica más utilizada en el sector (los modelos generalizados lineales o GLM) y una metodología basada en *Machine Learning* llamada *Gradient Boosting*.

Antes de comentar los resultados obtenidos es importante mencionar las ventajas y desventajas de cada modelización. Una de las principales ventajas de los modelos clásicos es la fácil interpretabilidad de los coeficientes. Además, son modelos robustos porque los errores se reparten mediante una distribución paramétrica conocida, en este caso, una Poisson y una Gamma. En contraposición, este método tiene desventajas, como puede ser la necesidad de un mayor trabajo previo de preparación de los datos, que implica a su vez un fuerte conocimiento técnico de los mismos, un mayor tiempo de creación del modelo, o como en este caso, un peor ajuste del riesgo.

La modelización GBM, por otra parte, presenta ventajas: un uso de memoria muy limitado, lo cual sumado a la creciente capacidad computacional moderna, abre la capacidad de procesamiento de datos de forma nunca antes vista; gran velocidad de aplicación, optimización del tiempo de creación del modelo gracias al mejor uso de los recursos informáticos, etc. En resumen, es una modelización muy eficiente y optimizada. Además, el nivel de ajuste es más elevado respecto a la primera modelización y muestra un gran rendimiento ante la falta de linealidad, puesto que no sigue una distribución paramétrica. Respecto a las desventajas, las más importantes son: la falta de interpretabilidad mediante coeficientes de las distintas variables (es complicado entender el modelo a simple vista sin entrar en detalles técnicos), que se trata de suplir mediante gráficos de dependencia parcial; o la gran facilidad para el sobreajuste (*overfitting*).

Los resultados obtenidos en este estudio muestran que la modelización más innovadora tiene mejores capacidades predictivas y una mayor capacidad para diferenciar el tipo de riesgo, como se ha visto en el análisis por clústeres, donde se otorgaba una mayor prima a aquellos con mayor probabilidad de sufrir un siniestro, y una menor prima a aquellos individuos con menor riesgo.

De esta manera, gracias a la información aportada por este estudio, se demuestra una aplicación directa de estos métodos en el sector asegurador de autos. Estas nuevas técnicas deben pasar una validación interna de trazabilidad, ya que deben estar en línea con Solvencia II y conforme a las peticiones de las auditorías internas y externas.

Esta nueva modelización requiere adaptación y entendimiento por parte de la empresa y el regulador, por ello surgen softwares actuariales que tratan de estandarizar los procedimientos. Se están desarrollando avances de distinta índole, como, por ejemplo, mostrar la importancia de la interacción entre las variables o variaciones descendentes en

el ratio de aprendizaje de los árboles de decisión para obtener mayor eficiencia y comprensión.

Respecto al análisis desde el punto de vista del negocio se muestra la comparativa entre dos posibles maneras de renovar la cartera: en primer lugar, mediante una mutualización, donde todos los clientes tienen la misma subida de prima, y, en segundo lugar, mediante una segmentación en precios que, dependiendo del clúster al que estén asociados los clientes, tienen un descuento o un recargo sobre la prima.

Sobre este último modelo se aplica un modelo básico de techos y suelos con el fin de que la prima de los asegurados con mayor riesgo no sea muy elevada. Con esto se pretende aumentar la retención de cartera.

En definitiva, nos encontramos ante dos tipos de modelización con características diferentes y en puntos de maduración distintos. Uno de ellos lleva multitud de años en funcionamiento, mientras que el segundo comienza a presentar su viabilidad a nivel empresarial. Ambos son válidos de cara al estudio académico y, observando este estudio, es el momento para comenzar a utilizar esta nueva metodología a nivel profesional.

8. BIBLIOGRAFÍA

Agresti, A. (2015). *Foundations of linear and generalized linear models*. John Wiley & Sons.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Denuit, M., Hainaut, D., & Trufin, J. (2019). *Effective Statistical Learning Methods for Actuaries I*. Springer International Publishing AG.

Denuit, M., Hainaut, D., & Trufin, J. (2020). *Effective Statistical Learning Methods for Actuaries II*. Springer International Publishing AG.

Denuit, M., Hainaut, D., & Trufin, J. (2019). *Effective Statistical Learning Methods for Actuaries II*. Springer International Publishing AG.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367-378.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Garrido, J., Genest, C., & Schulz, J. (2016). Generalized linear models for dependent frequency and severity of insurance claims. *Insurance: Mathematics and Economics*, *70*, 205-215.

Guelman, L. (2012). Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Systems with Applications*, *39*(3), 3659-3667.

Pons, J. P. (2006). El seguro de accidentes de trabajo en España: de la obligación al negocio (1900–1940). *Investigaciones de Historia Económica*, *2*(4), 77-100.

Pons, J. P., & Brías, M. Á. P. (Eds.). (2010). *Investigaciones históricas sobre el seguro español*. Fundación Mapfre.

Lempicka, J. (2021). *Predicción de Cross-selling con técnicas de Machine Learning.* (Trabajo fin de máster).

Martínez de Lizarduy Kostornichenko, V. (2021). *Comparative performance analysis between Grandient Boosting models and GLMs for non-life pricing* (Master's thesis).

Ng, S., Lestari, D., & Devila, S. (2019, November). Generalized linear model for deductible pricing in non-life insurance. In *AIP Conference Proceedings* (Vol. 2168, No. 1, p. 020038). AIP Publishing LLC.

Ridgeway, G. (2007). Generalized Boosted Models: A guide to the gbm package. *Update*, *1*(1), 2007.

Vidal-Llana, X., & Guillén, M. (2020). Advanced analytics pricing for the calculation of post-covid19 scenarios in automobile insurance. *Anales del Instituto de Actuarios Españoles, 2020, vol. 26, p. 157-179.*

Scikit-learn, Python. https://scikit-learn.org/stable/index.html

Microsoft, & Light GBM contributors. (2021). LightGBM.

https://lightgbm.readthedocs.io/en/latest/

9. ANEXO

```
####Librerias
import pandas as pd
import numpy as np

#Para fechas
from datetime import datetime
from datetime import date
from dateutil.relativedelta import relativedelta

#Para gráficos
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib import colors

################
#Carga de Datos#
################

path=r"C:\Users\Gonzalo\Documents\Universidad\Actuariales\2Curso\2cuatrimestre
      \TFM\ polizas_marketing_2010.xls"

bbdd= pd.read_excel(path, sheet_name= "Datos_mod")

#####################################
##Preparación y Creación de Variables##
#####################################


bbdd['Fecha_vencimiento'] = bbdd['Fecha_vencimiento'].dt.date
bbdd['fecha_nacimiento'] = bbdd['fecha_nacimiento'].dt.date

bbdd['fecha_ini_poliza'] = bbdd['Fecha_vencimiento'] - relativedelta(years=1)
bbdd['edad']    =    [relativedelta(ini,    fin).years    for    0ini,    fin    in
      zip(bbdd['fecha_ini_poliza'],bbdd['fecha_nacimiento'])]
```

```
#variable exposición
fecha_ini=date(2018,12,31)
bbdd['exposure'] = ((bbdd['Fecha_vencimiento']-fecha_ini).dt.days)/365
#Chequeo Exposición no puede haber valores negativos ni mayores a 1
min(bbdd['exposure'])
max(bbdd['exposure'])

##########################################################
##Exploración de la Base de datos. Análisis de variable##
##########################################################

#Parte 1. Análisis de cada variable

###Frecuencia
fig = plt.figure(figsize=(20,9))
ax1 = fig.add_subplot(1,2,1)
ax2= fig.add_subplot(1,2,2)

ax1.hist ((bbdd['num_siniestros']), color='#F2AB6D')
ax1.set_title("Escala lineal de número siniestros",
          fontdict={'family': 'Calibri',
                    'color' : 'black',
                    'weight': 'bold',
                    'size': 18})
ax1.set_xlabel("Número de siniestros del vehículo", size = 16,)
ax1.set_ylabel("Número de pólizas", size = 16,)

#escala logarítmica
ax2.hist ((bbdd['num_siniestros']), color='#F2AB6D')

plt.yscale( "log")
ax2.set_title("Escala logarítmica de número siniestros",
          fontdict={'family': 'Calibri',
                    'color' : 'black',
                    'weight': 'bold',
                    'size': 18})
ax2.set_xlabel("Número de siniestros del vehículo", size = 16,)
ax2.set_ylabel("Número de pólizas", size = 16,)
```

```
frecuencia=pd.DataFrame((bbdd['poliza']).groupby(bbdd['num_siniestros']).count
      ())



#Severidad
fig = plt.figure(figsize=(20,9))
ax1 = fig.add_subplot(1,2,1)
ax2= fig.add_subplot(1,2,2)

ax1.hist ((bbdd['coste_siniestro_total']), color='lightgreen')
ax1.set_title("Coste de los accidentes",
          fontdict={'family': 'Calibri',
                    'color' : 'black',
                    'weight': 'bold',
                    'size': 18})
ax1.set_xlabel("Coste de los siniestros del vehículo", size = 16,)
ax1.set_ylabel("Número de pólizas", size = 16,)

#Sin 0
sev_0 = bbdd
sev_0  =  sev_0.sort_values(  'coste_siniestro_total')    #ordenamos  por  coste
      siniestro
x = sev_0[sev_0["coste_siniestro_total"]== 0].index #seleccionamos los valores
      igual a 0
sev_0=  sev_0.drop(x)
ax2.hist ((sev_0['coste_siniestro_total']), color='lightgreen')
ax2.set_title("Coste de los siniestros mayores de 0",
          fontdict={'family': 'Calibri',
                    'color' : 'black',
                    'weight': 'bold',
                    'size': 18})
ax2.set_xlabel("Coste de los siniestros del vehículo", size = 16,)
ax2.set_ylabel("Número de pólizas", size = 16,)

x= sev_0['coste_siniestro_total'].sort_values()

# Sexo
sexo= pd.DataFrame((bbdd['poliza']).groupby(bbdd['sexo']).count())
#43% son hombre y 57% son mujeres
```

```
# Edad conductor agrupada por décadas
edad_agrupada = pd.DataFrame((bbdd['poliza']).groupby(bbdd['edad_conductor
    _agrupada']).count())

n= len(bbdd['poliza'])

#Creación de variable Edad, a partir de fecha de nacimiento
bbdd['edad'] = [relativedelta(ini, fin).years for ini, fin in
    zip(bbdd['fecha_ini_poliza'],bbdd['fecha_nacimiento'])]

#Credit scoring. Hay que segmentar
moda_cs= float((bbdd['credit_scoring'].mode()))
bbdd['credit_scoring'] = bbdd.credit_scoring.fillna(moda_cs)

Cuantil_cs10 =(bbdd['credit_scoring']).quantile(0.1)   #452
Cuantil_cs25 =(bbdd['credit_scoring']).quantile(0.25)  #606
Cuantil_cs50 =(bbdd['credit_scoring']).quantile(0.5)   #675
Cuantil_cs75 =(bbdd['credit_scoring']).quantile(0.75)  #767
Cuantil_cs90 =(bbdd['credit_scoring']).quantile(0.9)   #819


#Area residencia
area_residencia =
    pd.DataFrame((bbdd['poliza']).groupby(bbdd['area_residencia']).count())

#indice tráfico. Hay que segmentar

indice_trafico=pd.DataFrame((bbdd['poliza']).groupby(bbdd['indice_trafico']).
    count())
#Asumimos que donde no teníamos información pertenecen a la moda de la variable
moda_it= float((bbdd['indice_trafico'].mode()))
bbdd['indice_trafico'] = bbdd.indice_trafico.fillna(moda_it)

#Segmentamos por cuantiles
Cuantil_it10 =(bbdd['indice_trafico']).quantile(0.1)   #57.8
Cuantil_it25 =(bbdd['indice_trafico']).quantile(0.25)  #82.1
Cuantil_it50 =(bbdd['indice_trafico']).quantile(0.5)   #111.4
Cuantil_it75 =(bbdd['indice_trafico']).quantile(0.75)  #135.7
Cuantil_it90 =(bbdd['indice_trafico']).quantile(0.9)   #148.5

#Antigüedad vehículo
```

```
antiguedad_vehiculo=pd.DataFrame((bbdd['poliza']).groupby(bbdd['antiguedad_veh
      iculo']).count())


#Tipo de vehículo
B7_tipo_vehiculo=pd.DataFrame((bbdd['poliza']).groupby(bbdd['B7_tipo_vehiculo'
      ]).count())


#Valor de vehículo. Hay que segmentar


Cuantil_v25 =(bbdd['valor_vehiculo']).quantile(0.25) #11.110
Cuantil_v50 =(bbdd['valor_vehiculo']).quantile(0.5) #16.500
Cuantil_v75 =(bbdd['valor_vehiculo']).quantile(0.75) #23650
Cuantil_v90 =(bbdd['valor_vehiculo']).quantile(0.9) #35.739


#Segmentamos este último tramo para diferenciar los coches de mayor gama.
# Oficina. Muchos NA.
oficina = pd.DataFrame((bbdd['poliza']).groupby(bbdd['oficina']).count())
#Faltan muchos datos añadimos un nuevo canal, lo llamamos 5. Entendemos que
      puede ser un canal de venta vía online (rastreator, pej)
bbdd['oficina'] = bbdd.oficina.fillna(5)
oficina = pd.DataFrame((bbdd['poliza']).groupby(bbdd['oficina']).count())
# numero siniestros
num_siniestros=pd.DataFrame((bbdd['poliza']).groupby(bbdd['num_siniestros']).
      count())


#######Gráficos
#Grafico de variables 1
fig = plt.figure(figsize=(24,48))
ax1 = fig.add_subplot(6,2,1)
ax2 = fig.add_subplot(6,2,2)
ax3 = fig.add_subplot(6,2,3)
ax4= fig.add_subplot(6,2,4)
ax5= fig.add_subplot(6,2,5)
ax6= fig.add_subplot(6,2,6)


#sexo
ax1.hist ((bbdd['sexo']), color='#F2AB6D')
ax1.set_xlabel("Sexo", size = 24,)
ax1.set_ylabel("Número de pólizas", size = 16,)


#edad
```

```
ax2.hist ((bbdd['edad']), color='LightBlue')
ax2.set_xlabel("Edad", size = 24,)
ax2.set_ylabel("Número de pólizas", size = 16,)

#credit Scoring

ax3.hist ((bbdd['credit_scoring']), color='LightGreen')
ax3.set_xlabel("Credit Scoring", size = 24,)
ax3.set_ylabel("Número de pólizas", size = 16,)

#Area residencia
ax4.hist ((bbdd['area_residencia']), color='Purple')
ax4.set_xlabel("Área de residencia", size = 24,)
ax4.set_ylabel("Número de pólizas", size = 16,)

#Índice tráfico
ax5.hist ((bbdd['indice_trafico']), color='Red')
ax5.set_xlabel("Índice Tráfico", size = 24,)
ax5.set_ylabel("Número de pólizas", size = 16,)

#Oficina
ax6.hist ((bbdd['oficina']), color='Gray')
ax6.set_xlabel("Oficinas", size = 24,)
ax6.set_ylabel("Número de pólizas", size = 16,)

plt.subplots_adjust(left=0.125,bottom=0.05,  right=0.9,  top=0.7,  wspace=0.25,
      hspace=0.5)
#Gráfico de variables 2

fig = plt.figure(figsize=(24, 35))
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,2)

#Antigüedad del vehículo
ax1.hist ((bbdd['antiguedad_vehiculo']), color='#F2AB6D')
ax1.set_xlabel("Antigüedad del vehículo", size = 24,)
ax1.set_ylabel("Número de pólizas", size = 24,)



#Coste del siniestro. Hay que segmentar
```

```
        #Por cuantiles? --> No puede ser, el 83% de individuos no tiene
      siniestros.


plt.scatter(bbdd['coste_siniestro_total'], bbdd['num_siniestros'])
plt.xlabel("Coste de los siniestros")
plt.ylabel("Número de siniestros")
plt.title("Correlación entre número de siniestros y su coste")


# Prima. Vemos que es una mutualización


#Parte 2.1 Se va a realizar un analisis de las variables explicativas para el
modelo en relación con la exposición


Exposicion_total = (bbdd['exposure']).sum()


#### Sexo
filtro_hombre = bbdd['sexo'] == 'M'
bbdd_filtro_hombre = bbdd[filtro_hombre]
exposicion_hombre = (bbdd_filtro_hombre['exposure']).sum()
exposicion_mujer = Exposicion_total - exposicion_hombre
exposicion_sexo = [exposicion_hombre, exposicion_mujer ]


#Gráfico 1
nombre = [ "Masculino", "Femenino"]
colores= ["#AAF683","#FFD97D"]
plt.figure(figsize=(4, 3))
plt.pie(exposicion_sexo, labels = nombre , autopct="%0.1f %%", colors=colores)
plt.title("Exposición asegurados por Sexo", fontdict={'family': 'Calibri',
'color' : 'black', 'weight': 'bold',: 18})


### Edad conductor agrupada, por décadas.


filtro_90 = bbdd['edad_conductor _agrupada'] == 1
bbdd_filtro_90 = bbdd[filtro_90]
exposicion_90 = (bbdd_filtro_90['exposure']).sum()


filtro_80 = bbdd['edad_conductor _agrupada'] == 2
bbdd_filtro_80 = bbdd[filtro_80]
exposicion_80 = (bbdd_filtro_80['exposure']).sum()
```

```
filtro_70 = bbdd['edad_conductor _agrupada'] == 3

bbdd_filtro_70 = bbdd[filtro_70]

exposicion_70 = ((bbdd_filtro_70['exposure']).sum())


filtro_60 = bbdd['edad_conductor _agrupada'] == 4

bbdd_filtro_60 = bbdd[filtro_60]

exposicion_60 = ((bbdd_filtro_60['exposure']).sum())


filtro_50 = bbdd['edad_conductor _agrupada'] == 5

bbdd_filtro_50 = bbdd[filtro_50]

exposicion_50 = ((bbdd_filtro_50['exposure']).sum())


exposicion_40= Exposicion_total - (exposicion_90 + exposicion_80 + exposicion_70
+ exposicion_60 + exposicion_50)


exposicion_edad = [exposicion_90, exposicion_80, exposicion_70,  exposicion_60,
exposicion_50, exposicion_40]


# Gráfico 2.

decadas = [ "Años 90", "Años 80", "Años 70", "Años 60", "Años 50", "Años 40"]

edad=    pd.DataFrame(list(zip(exposicion_edad,    decadas)),    columns    =
['exposicion_edad', 'decadas'])



plt.bar (edad['decadas'], edad['exposicion_edad'])

plt.title("Exposición asegurados por Tramos de edad", ={'family': 'Calibri',
'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Décadas", size = 16,)

plt.ylabel("Exposición total", size = 16)



### Credit Scoring


filtro_cs10 = bbdd['credit_scoring'] < Cuantil_cs10

bbdd_filtro_cs10 = bbdd[filtro_cs10]

exposicion_cs10 = ((bbdd_filtro_cs10['exposure']).sum())


filtro_cs25 = (bbdd['credit_scoring'] > Cuantil_cs10) & (bbdd['credit_scoring']
< Cuantil_cs25)

bbdd_filtro_cs25 = bbdd[filtro_cs25]

exposicion_cs25 = ((bbdd_filtro_cs25['exposure']).sum())
```

```
filtro_cs50 = (bbdd['credit_scoring'] > Cuantil_cs25) & (bbdd['credit_scoring'] < Cuantil_cs50)

bbdd_filtro_cs50 = bbdd[filtro_cs50]

exposicion_cs50 = ((bbdd_filtro_cs50['exposure']).sum())


filtro_cs75 = (bbdd['credit_scoring'] > Cuantil_cs50) & (bbdd['credit_scoring'] < Cuantil_cs75)

bbdd_filtro_cs75 = bbdd[filtro_cs75]

exposicion_cs75 = ((bbdd_filtro_cs75['exposure']).sum())


filtro_cs90= (bbdd['credit_scoring'] > Cuantil_cs75) & (bbdd['credit_scoring'] < Cuantil_cs90)

bbdd_filtro_cs90 = bbdd[filtro_cs90]

exposicion_cs90 = ((bbdd_filtro_cs90['exposure']).sum())


exposicion_csfinal = Exposicion_total - (exposicion_cs10 + exposicion_cs25 + exposicion_cs50 + exposicion_cs75 + exposicion_cs90)


exposicion_cs=     [exposicion_cs10,     exposicion_cs25,     exposicion_cs50, exposicion_cs75, exposicion_cs90, exposicion_csfinal]


#Gráfico 3

cuantil_cs = [ '(301; 405)', '(405; 602)', '(602; 682)', '(682; 772)', '(772; 821)', '(821, 850)']

Exposicion_cs=  pd.DataFrame(list(zip(exposicion_cs,  cuantil_cs)),  columns  = ['exposicion_cs', 'cuantil_cs'])


plt.figure(figsize=(14,9))

plt.bar (Exposicion_cs['cuantil_cs'], Exposicion_cs['exposicion_cs'])

plt.title("Exposición  asegurados  por  Credit  Scoring",  fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Credit Scoring", size = 16,)

plt.ylabel("Exposición total", size = 16)



####Área Residencia

filtro_1 = bbdd['area_residencia'] == 1

bbdd_filtro_1 = bbdd[filtro_1]

exposicion_1 = ((bbdd_filtro_1['exposure']).sum())


filtro_2 = bbdd['area_residencia'] == 2
```

```
bbdd_filtro_2 = bbdd[filtro_2]
exposicion_2 = ((bbdd_filtro_2['exposure']).sum())
filtro_3 = bbdd['area_residencia'] == 3
bbdd_filtro_3 = bbdd[filtro_3]
exposicion_3 = ((bbdd_filtro_3['exposure']).sum())

filtro_4 = bbdd['area_residencia'] == 4
bbdd_filtro_4 = bbdd[filtro_4]
exposicion_4 = ((bbdd_filtro_4['exposure']).sum())

filtro_5 = bbdd['area_residencia'] == 5
bbdd_filtro_5 = bbdd[filtro_5]
exposicion_5 = ((bbdd_filtro_5['exposure']).sum())

exposicion_6 = Exposicion_total - (exposicion_1 + exposicion_2 + exposicion_3 +
exposicion_4 + exposicion_5)

exposicion_area = [exposicion_1, exposicion_2, exposicion_3, exposicion_4,
exposicion_5, exposicion_6]

#Gráfico 4
zonas = [ "Área 1", "Área 2", "Área 3", "Área 4", "Área 5", "Área 6"]
area= pd.DataFrame(list(zip(exposicion_area, zonas)), columns =
['exposicion_area', 'zonas'])

plt.bar (area['zonas'], area['exposicion_area'])
plt.title("Exposición asegurados por Áreas de residencia", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})
plt.xlabel("Áreas de residencia", size = 16,)
plt.ylabel("Exposición total", size = 16)



### Indice de tráfico

filtro_it10 = bbdd['indice_trafico'] < Cuantil_it10
bbdd_filtro_it10 = bbdd[filtro_it10]
exposicion_it10 = (bbdd_filtro_it10['exposure']).sum()

filtro_it25 = (bbdd['indice_trafico'] > Cuantil_it10) & (bbdd['indice_trafico']
< Cuantil_it25)
bbdd_filtro_it25 = bbdd[filtro_it25]
```

```
exposicion_it25 = (bbdd_filtro_it25['exposure']).sum()


filtro_it50 = (bbdd['indice_trafico'] > Cuantil_it25) & (bbdd['indice_trafico']
< Cuantil_it50)

bbdd_filtro_it50 = bbdd[filtro_it50]

exposicion_it50 = (bbdd_filtro_it50['exposure']).sum()


filtro_it75 = (bbdd['indice_trafico'] > Cuantil_it50) & (bbdd['indice_trafico']
< Cuantil_it75)

bbdd_filtro_it75 = bbdd[filtro_it75]

exposicion_it75 = (bbdd_filtro_it75['exposure']).sum()


filtro_it90 = (bbdd['indice_trafico'] > Cuantil_it75) & (bbdd['indice_trafico']
< Cuantil_it90)

bbdd_filtro_it90 = bbdd[filtro_it90]

exposicion_it90 = (bbdd_filtro_it90['exposure']).sum()


exposicion_it100 = Exposicion_total - (exposicion_it10 + exposicion_it25 +
exposicion_it50 + exposicion_it75 + exposicion_it90)


exposicion_it=     [exposicion_it10,     exposicion_it25,     exposicion_it50,
exposicion_it75, exposicion_it90, exposicion_it100]


#Gráfico 5

Indice_trafico = [ "(0; 57,8)", "(57,8; 82,1)", "(82,1; 111,4)", "(111,4;
135,7)", "(135,7; 148,5)", "(148,5; 207) "]

indice_trafico= pd.DataFrame(list(zip(exposicion_it, Indice_trafico)), columns
= ['exposicion_it', 'Indice_trafico'])


plt.figure(figsize=(9,4))

plt.bar (indice_trafico['Indice_trafico'], indice_trafico['exposicion_it'])

plt.title("Exposición asegurados por Índice de tráfico",

          fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold',
'size': 18})

plt.xlabel("Indice de tráfico", size = 16,)

plt.ylabel("Exposición total", size = 16)


### Antigüedad del vehículo


filtro_a1 = bbdd['antiguedad_vehiculo'] == 1

bbdd_filtro_a1 = bbdd[filtro_a1]

exposicion_a1 = (bbdd_filtro_a1['exposure']).sum()
```

```
filtro_a2 = bbdd['antiguedad_vehiculo'] == 2

bbdd_filtro_a2 = bbdd[filtro_a2]

exposicion_a2 = (bbdd_filtro_a2['exposure']).sum()


filtro_a3 = bbdd['antiguedad_vehiculo'] == 3

bbdd_filtro_a3 = bbdd[filtro_a3]

exposicion_a3 = (bbdd_filtro_a3['exposure']).sum()


exposicion_a4 = Exposicion_total - (exposicion_a1 + exposicion_a2 + exposicion_a3)


exposicion_a= [exposicion_a1, exposicion_a2, exposicion_a3, exposicion_a4]


#Gráfico 6

Edad_vehiculo = [ "1 año", "2 año", "3 año", "4 año"]

edad_vehiculo= pd.DataFrame(list(zip(exposicion_a, Edad_vehiculo)), columns = ['exposicion_a', 'Edad_vehiculo'])


plt.bar (edad_vehiculo['Edad_vehiculo'], edad_vehiculo['exposicion_a'])

plt.title("Exposición asegurados por Edad del vehículo", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Edad del vehículo", size = 16,)

plt.ylabel("Exposición total", size = 16)



### Tipo de vehículo


filtro_car = bbdd['B7_tipo_vehiculo'] == "Autocaravana"

bbdd_filtro_car = bbdd[filtro_car]

exposicion_car = ((bbdd_filtro_car['exposure']).sum())


filtro_b2 = bbdd['B7_tipo_vehiculo'] == "Berlina 2 vol"

bbdd_filtro_b2 = bbdd[filtro_b2]

exposicion_b2 = ((bbdd_filtro_b2['exposure']).sum())


filtro_b3 = bbdd['B7_tipo_vehiculo'] == "Berlina 3 vol"

bbdd_filtro_b3 = bbdd[filtro_b3]

exposicion_b3 = ((bbdd_filtro_b3['exposure']).sum())
```

```
filtro_cab = bbdd['B7_tipo_vehiculo'] == "Cabrio"
bbdd_filtro_cab = bbdd[filtro_cab]
exposicion_cab = ((bbdd_filtro_cab['exposure']).sum())

filtro_cou = bbdd['B7_tipo_vehiculo'] == "Coupe"
bbdd_filtro_cou = bbdd[filtro_cou]
exposicion_cou = ((bbdd_filtro_cou['exposure']).sum())

filtro_ddt = bbdd['B7_tipo_vehiculo'] == "Derivado de Turismo"
bbdd_filtro_ddt = bbdd[filtro_ddt]
exposicion_ddt = ((bbdd_filtro_ddt['exposure']).sum())

filtro_f = bbdd['B7_tipo_vehiculo'] == "Familiar"
bbdd_filtro_f = bbdd[filtro_f]
exposicion_f = ((bbdd_filtro_f['exposure']).sum())

filtro_fu = bbdd['B7_tipo_vehiculo'] == "Furgoneta"
bbdd_filtro_fu = bbdd[filtro_fu]
exposicion_fu = ((bbdd_filtro_fu['exposure']).sum())

filtro_m = bbdd['B7_tipo_vehiculo'] == "Monovolumen"
bbdd_filtro_m = bbdd[filtro_m]
exposicion_m = ((bbdd_filtro_m['exposure']).sum())

filtro_p = bbdd['B7_tipo_vehiculo'] == "Pick-up"
bbdd_filtro_p = bbdd[filtro_p]
exposicion_p = ((bbdd_filtro_p['exposure']).sum())

filtro_t = bbdd['B7_tipo_vehiculo'] == "Targa"
bbdd_filtro_t = bbdd[filtro_t]
exposicion_t = ((bbdd_filtro_t['exposure']).sum())

filtro_tt = bbdd['B7_tipo_vehiculo'] == "Todo Terreno"
bbdd_filtro_tt = bbdd[filtro_tt]
exposicion_tt = ((bbdd_filtro_tt['exposure']).sum())

exposicion_desc = Exposicion_total - (exposicion_car + exposicion_b2 +
exposicion_b3 + exposicion_cab + exposicion_cou + exposicion_ddt + exposicion_f
+exposicion_fu + exposicion_m + exposicion_p + exposicion_t + exposicion_tt)
```

```
exposicion_tv = [exposicion_car, exposicion_b2, exposicion_b3, exposicion_cab, exposicion_cou, exposicion_ddt, exposicion_f, exposicion_fu, exposicion_m, exposicion_p, exposicion_t, exposicion_tt, exposicion_desc ]



#Gráfico 7

tipo_vehiculo = [ "Autocaravana", "Berlina 2 vol", "Berlina 3 vol", "Cabrio", "Coupe", "Der. de Turismo", "Familiar", "Furgoneta", "Monovolumen", "Pick-up", "Targa", "Todoterreno", "Desconocido"]

tipo_vehiculo= pd.DataFrame(list(zip(exposicion_tv, tipo_vehiculo)), columns = ['exposicion_tv', 'tipo_vehiculo'])


plt.figure(figsize=(18,9))

plt.bar (tipo_vehiculo['tipo_vehiculo'], tipo_vehiculo['exposicion_tv'])

plt.title("Exposición asegurados por tipo del vehículo", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Tipo del vehículo", size = 16,)

plt.ylabel("Exposición total", size = 16)



###Valor del vehículo (por cuantiles)


filtro_v1 = bbdd['valor_vehiculo'] < Cuantil_v25

bbdd_filtro_v1 = bbdd[filtro_v1]

exposicion_v1 = ((bbdd_filtro_v1['exposure']).sum())


filtro_v2 = (bbdd['valor_vehiculo'] > Cuantil_v25) &  (bbdd['valor_vehiculo'] < Cuantil_v50)

bbdd_filtro_v2 = bbdd[filtro_v2]

exposicion_v2 = ((bbdd_filtro_v2['exposure']).sum())


filtro_v3 = (bbdd['valor_vehiculo'] > Cuantil_v50) &  (bbdd['valor_vehiculo'] < Cuantil_v75)

bbdd_filtro_v3 = bbdd[filtro_v3]

exposicion_v3 = ((bbdd_filtro_v3['exposure']).sum())


filtro_v4 = (bbdd['valor_vehiculo'] > Cuantil_v75) &  (bbdd['valor_vehiculo'] < Cuantil_v90)

bbdd_filtro_v4 = bbdd[filtro_v4]

exposicion_v4 = ((bbdd_filtro_v4['exposure']).sum())


exposicion_v5  =  Exposicion_total  -  (exposicion_v1  +  exposicion_v2  + exposicion_v3 + exposicion_v4)
```

```
exposicion_v= [exposicion_v1, exposicion_v2, exposicion_v3, exposicion_v4, exposicion_v5]


#Grafico 8

valor_vehiculo = [ "[0,11.100€)", "[11.100€, 16.500€)", "[16.500€, 23.650€)", "[23.650€, 35.739€)", "[35.739€,380.160€]"]

valor_vehiculo= pd.DataFrame(list(zip(exposicion_v, valor_vehiculo)), columns = ['exposicion_v', 'valor_vehiculo'])


plt.figure(figsize=(12,5))

plt.bar (valor_vehiculo['valor_vehiculo'], valor_vehiculo['exposicion_v'])

plt.title("Exposición asegurados por Valor del vehículo", fontdict={'family': 'Calibri',: 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Valor del vehículo", size = 16,)

plt.ylabel("Exposición total", size = 16)


#Número de siniestros


filtro_ns0 = bbdd['num_siniestros'] == 0

bbdd_filtro_ns0 = bbdd[filtro_ns0]

exposicion_ns0 = ((bbdd_filtro_ns0['exposure']).sum())


filtro_ns1 = bbdd['num_siniestros'] == 1

bbdd_filtro_ns1 = bbdd[filtro_ns1]

exposicion_ns1 = ((bbdd_filtro_ns1['exposure']).sum())


filtro_ns2 = bbdd['num_siniestros'] == 2

bbdd_filtro_ns2 = bbdd[filtro_ns2]

exposicion_ns2 = ((bbdd_filtro_ns2['exposure']).sum())


filtro_ns3 = bbdd['num_siniestros'] == 3

bbdd_filtro_ns3 = bbdd[filtro_ns3]

exposicion_ns3 = ((bbdd_filtro_ns3['exposure']).sum())


filtro_ns4 = bbdd['num_siniestros'] == 4

bbdd_filtro_ns4 = bbdd[filtro_ns4]

exposicion_ns4 = ((bbdd_filtro_ns4['exposure']).sum())


exposicion_ns5= Exposicion_total - (exposicion_ns0 + exposicion_ns1 + exposicion_ns2 + exposicion_ns3 + exposicion_ns4)
```

```
exposicion_ns= [exposicion_ns0, exposicion_ns1, exposicion_ns2, exposicion_ns3,
exposicion_ns4, exposicion_ns5]


#Grafico 9

siniestros_vehiculo = [ "0 siniestros", "1 siniestros", "2 siniestros", "3
siniestros", "4 siniestros", "5 siniestros"]

siniestros_vehiculo=                    pd.DataFrame(list(zip(exposicion_ns,
siniestros_vehiculo)), columns = ['exposicion_ns', 'siniestros_vehiculo'])


plt.figure(figsize=(8,4))


plt.bar                          (siniestros_vehiculo['siniestros_vehiculo'],
siniestros_vehiculo['exposicion_ns'])

plt.title("Exposición    asegurados    por    siniestros    del    vehículo",
fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Siniestros del vehículo", size = 16,)

plt.ylabel("Exposición total", size = 16)


#Parte 2.2. Se va a realizar un analisis de las variables explicativas para el
modelo en relación con el Coste


###COSTE MEDIO

Coste_medio=((bbdd['coste_siniestro_total']).sum())/

((bbdd['num_siniestros']) sum())


###COSTE TOTAL

Coste_total = ((bbdd['coste_siniestro_total']).sum())


#Debemos analizar como afecta el coste  a las distintas variables de la base de
datos


#### Sexo

#Coste medio

filtro_hombre = bbdd['sexo'] == 'M'

bbdd_filtro_hombre = bbdd[filtro_hombre]

cm_hombre     =     ((bbdd_filtro_hombre['coste_siniestro_total']).sum())     /
((bbdd_filtro_hombre['num_siniestros']).sum())


filtro_mujer = bbdd['sexo'] == 'F'

bbdd_filtro_mujer = bbdd[filtro_mujer]

cm_mujer     =     ((bbdd_filtro_mujer['coste_siniestro_total']).sum())     /
((bbdd_filtro_mujer['num_siniestros']).sum())
```

```
cm_sexo = [cm_hombre, cm_mujer ]


#Gráfico 10
nombre = [ "Masculino", "Femenino"]
colores= ["#AAF683","#FFD97D"]
plt.figure(figsize=(12, 6))
plt.pie(cm_sexo, labels = nombre , autopct="%0.1f %%", colors=colores)
plt.title("Coste medio asegurados por Sexo", fontdict={'family': 'Calibri',
'color' : 'black', 'weight': 'bold', 'size': 18})
plt.legend(loc='lower right')
#Coste total


filtro_hombre = bbdd['sexo'] == 'M'
bbdd_filtro_hombre = bbdd[filtro_hombre]
ct_hombre = ((bbdd_filtro_hombre['coste_siniestro_total']).sum())


filtro_mujer = bbdd['sexo'] == 'F'
bbdd_filtro_mujer = bbdd[filtro_mujer]
ct_mujer = ((bbdd_filtro_mujer['coste_siniestro_total']).sum())
ct_sexo = [ct_hombre, ct_mujer ]


#Gráfico 11
nombre = [ "Masculino", "Femenino"]
colores= ["#AAF683","#FFD97D"]
plt.figure(figsize=(12, 6))
plt.pie(ct_sexo, labels = nombre , autopct="%0.1f %%", colors=colores)
plt.title("Coste total asegurados por Sexo", fontdict={'family': 'Calibri',
'color' : 'black', 'weight': 'bold', 'size': 18})
plt.legend(loc='lower left')


###Edad agrupada


#Coste medio
filtro_90 = bbdd['edad_conductor _agrupada'] == 1
bbdd_filtro_90 = bbdd[filtro_90]
cm_90       =       ((bbdd_filtro_90['coste_siniestro_total']).sum())       /
((bbdd_filtro_90['num_siniestros']).sum())


filtro_80 = bbdd['edad_conductor _agrupada'] == 2
bbdd_filtro_80 = bbdd[filtro_80]
```

```
cm_80       =       ((bbdd_filtro_80['coste_siniestro_total']).sum())       /
((bbdd_filtro_80['num_siniestros']).sum())


filtro_70 = bbdd['edad_conductor _agrupada'] == 3

bbdd_filtro_70 = bbdd[filtro_70]

cm_70       =       ((bbdd_filtro_70['coste_siniestro_total']).sum())       /
((bbdd_filtro_70['num_siniestros']).sum())


filtro_60 = bbdd['edad_conductor _agrupada'] == 4

bbdd_filtro_60 = bbdd[filtro_60]

cm_60       =       ((bbdd_filtro_60['coste_siniestro_total']).sum())       /
((bbdd_filtro_60['num_siniestros']).sum())


filtro_50 = bbdd['edad_conductor _agrupada'] == 5

bbdd_filtro_50 = bbdd[filtro_50]

cm_50       =       ((bbdd_filtro_50['coste_siniestro_total']).sum())       /
((bbdd_filtro_50['num_siniestros']).sum())


filtro_40 = bbdd['edad_conductor _agrupada'] == 6

bbdd_filtro_40 = bbdd[filtro_40]

cm_40       =       ((bbdd_filtro_40['coste_siniestro_total']).sum())       /
((bbdd_filtro_40['num_siniestros']).sum())


cm_edad = [cm_90, cm_80, cm_70, cm_60, cm_50, cm_40]


# Gráfico 12.

decadas = [ "Años 90", "Años 80", "Años 70", "Años 60", "Años 50", "Años 40"]

edad=   pd.DataFrame(list(zip(cm_edad,   decadas)),   columns   =   ['cm_edad',
'decadas'])



plt.bar (edad['decadas'], edad['cm_edad'])

plt.title("Coste  medio  asegurados  por  Tramos  de  edad",  fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Décadas", size = 16,)

plt.ylabel("Coste medio", size = 16)


#Coste total

filtro_90 = bbdd['edad_conductor _agrupada'] == 1

bbdd_filtro_90 = bbdd[filtro_90]

ct_90 = ((bbdd_filtro_90['coste_siniestro_total']).sum())
```

```
filtro_80 = bbdd['edad_conductor _agrupada'] == 2
bbdd_filtro_80 = bbdd[filtro_80]
ct_80 = ((bbdd_filtro_80['coste_siniestro_total']).sum())
filtro_70 = bbdd['edad_conductor _agrupada'] == 3
bbdd_filtro_70 = bbdd[filtro_70]
ct_70 = ((bbdd_filtro_70['coste_siniestro_total']).sum())

filtro_60 = bbdd['edad_conductor _agrupada'] == 4
bbdd_filtro_60 = bbdd[filtro_60]
ct_60 = ((bbdd_filtro_60['coste_siniestro_total']).sum())

filtro_50 = bbdd['edad_conductor _agrupada'] == 5
bbdd_filtro_50 = bbdd[filtro_50]
ct_50 = ((bbdd_filtro_50['coste_siniestro_total']).sum())

filtro_40 = bbdd['edad_conductor _agrupada'] == 6
bbdd_filtro_40 = bbdd[filtro_40]
ct_40 = ((bbdd_filtro_40['coste_siniestro_total']).sum())

ct_edad = [ct_90, ct_80, ct_70, ct_60, ct_50, ct_40]

# Gráfico 13.

nombre = [ "Años 90", "Años 80", "Años 70", "Años 60", "Años 50", "Años 40"]
normdata = colors.Normalize(min(ct_edad), max(ct_edad))
colormap = cm.get_cmap("Reds")
colores =colormap(normdata(ct_edad))
plt.figure(figsize=(12, 11))
plt.pie(ct_edad, labels = nombre , autopct="%0.1f %%", colors = colores)
plt.title("Coste total asegurados por franjas de edad",
          fontdict={'family': 'Calibri',
                    'color' : 'black',
                    'weight': 'bold',
                    'size': 18})
plt.legend(loc='upper left')

###Credit Scoring

#Coste medio
```

```
filtro_cs10 = bbdd['credit_scoring'] < Cuantil_cs10

bbdd_filtro_cs10 = bbdd[filtro_cs10]

cm_cs10      =      ((bbdd_filtro_cs10['coste_siniestro_total']).sum())      /
((bbdd_filtro_cs10['num_siniestros']).sum())


filtro_cs25 = (bbdd['credit_scoring'] > Cuantil_cs10) & (bbdd['credit_scoring']
< Cuantil_cs25)

bbdd_filtro_cs25 = bbdd[filtro_cs25]

cm_cs25      =      ((bbdd_filtro_cs25['coste_siniestro_total']).sum())      /
((bbdd_filtro_cs25['num_siniestros']).sum())


filtro_cs50 = (bbdd['credit_scoring'] > Cuantil_cs25) & (bbdd['credit_scoring']
< Cuantil_cs50)

bbdd_filtro_cs50 = bbdd[filtro_cs50]

cm_cs50      =      ((bbdd_filtro_cs50['coste_siniestro_total']).sum())      /
((bbdd_filtro_cs50['num_siniestros']).sum())


filtro_cs75 = (bbdd['credit_scoring'] > Cuantil_cs50) & (bbdd['credit_scoring']
< Cuantil_cs75)

bbdd_filtro_cs75 = bbdd[filtro_cs75]

cm_cs75      =      ((bbdd_filtro_cs75['coste_siniestro_total']).sum())      /
((bbdd_filtro_cs75['num_siniestros']).sum())


filtro_cs90= (bbdd['credit_scoring'] > Cuantil_cs75) & (bbdd['credit_scoring']
< Cuantil_cs90)

bbdd_filtro_cs90 = bbdd[filtro_cs90]

cm_cs90      =      ((bbdd_filtro_cs90['coste_siniestro_total']).sum())      /
((bbdd_filtro_cs90['num_siniestros']).sum())


filtro_cs100= (bbdd['credit_scoring'] > Cuantil_cs90) & (bbdd['credit_scoring']
< max(bbdd['credit_scoring']))

bbdd_filtro_cs100 = bbdd[filtro_cs90]

cm_cs100     =     ((bbdd_filtro_cs100['coste_siniestro_total']).sum())     /
((bbdd_filtro_cs100['num_siniestros']).sum())




cm_cs= [cm_cs10, cm_cs25, cm_cs50, cm_cs75, cm_cs90, cm_cs100]


#Gráfico 14

cuantil_cs = [ '(301; 405)', '(405; 602)', '(602; 682)', '(682; 772)', '(772;
821)', '(821, 850)']

cm_cs=   pd.DataFrame(list(zip(cm_cs,   cuantil_cs)),   columns   =   ['cm_cs',
'cuantil_cs'])
```

```
plt.figure(figsize=(14,9))

plt.bar (cm_cs['cuantil_cs'], cm_cs['cm_cs'])

plt.title("Coste medio asegurados por Credit Scoring", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Credit Scoring", size = 16,)

plt.ylabel("Coste medio", size = 16)


#Coste total

filtro_cs10 = bbdd['credit_scoring'] < Cuantil_cs10

bbdd_filtro_cs10 = bbdd[filtro_cs10]

ct_cs10 = ((bbdd_filtro_cs10['coste_siniestro_total']).sum())


filtro_ct25 = (bbdd['credit_scoring'] > Cuantil_cs10) & (bbdd['credit_scoring']
< Cuantil_cs25)

bbdd_filtro_cs25 = bbdd[filtro_cs25]

ct_cs25 = ((bbdd_filtro_cs25['coste_siniestro_total']).sum())


filtro_cs50 = (bbdd['credit_scoring'] > Cuantil_cs25) & (bbdd['credit_scoring']
< Cuantil_cs50)

bbdd_filtro_cs50 = bbdd[filtro_cs50]

ct_cs50 = ((bbdd_filtro_cs50['coste_siniestro_total']).sum())


filtro_cs75 = (bbdd['credit_scoring'] > Cuantil_cs50) & (bbdd['credit_scoring']
< Cuantil_cs75)

bbdd_filtro_cs75 = bbdd[filtro_cs75]

ct_cs75 = ((bbdd_filtro_cs75['coste_siniestro_total']).sum())


filtro_cs90= (bbdd['credit_scoring'] > Cuantil_cs75) & (bbdd['credit_scoring']
< Cuantil_cs90)

bbdd_filtro_cs90 = bbdd[filtro_cs90]

ct_cs90 = ((bbdd_filtro_cs90['coste_siniestro_total']).sum())


filtro_cs100= (bbdd['credit_scoring'] > Cuantil_cs90) & (bbdd['credit_scoring']
< max(bbdd['credit_scoring']))

bbdd_filtro_cs100 = bbdd[filtro_cs90]

ct_cs100 = ((bbdd_filtro_cs100['coste_siniestro_total']).sum())



ct_cs= [ct_cs10, ct_cs25, ct_cs50, ct_cs75, ct_cs90, ct_cs100]


#Gráfico 15
```

```
nombre = [  '(301; 405)', '(405; 602)', '(602; 682)', '(682; 772)', '(772; 821)',
'(821, 850)']

normdata = colors.Normalize(min(ct_cs), max(ct_cs))

colormap = cm.get_cmap("Reds")

colores =colormap(normdata(ct_cs))

plt.figure(figsize=(20, 12))

plt.pie(ct_edad, labels = nombre , autopct="%1.2f %%")

plt.title("Coste total asegurados por Credit Scoring",

          fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold',
'size': 18})


plt.legend(loc='upper left')



####Área Residencia

filtro_1 = bbdd['area_residencia'] == 1

bbdd_filtro_1 = bbdd[filtro_1]

cm_1        =        ((bbdd_filtro_1['coste_siniestro_total']).sum())        /
((bbdd_filtro_1['num_siniestros']).sum())


filtro_2 = bbdd['area_residencia'] == 2

bbdd_filtro_2 = bbdd[filtro_2]

cm_2        =        ((bbdd_filtro_2['coste_siniestro_total']).sum())        /
((bbdd_filtro_2['num_siniestros']).sum())


filtro_3 = bbdd['area_residencia'] == 3

bbdd_filtro_3 = bbdd[filtro_3]

cm_3        =        ((bbdd_filtro_3['coste_siniestro_total']).sum())        /
((bbdd_filtro_3['num_siniestros']).sum())


filtro_4 = bbdd['area_residencia'] == 4

bbdd_filtro_4 = bbdd[filtro_4]

cm_4        =        ((bbdd_filtro_4['coste_siniestro_total']).sum())        /
((bbdd_filtro_4['num_siniestros']).sum())


filtro_5 = bbdd['area_residencia'] == 5

bbdd_filtro_5 = bbdd[filtro_5]

cm_5        =        ((bbdd_filtro_5['coste_siniestro_total']).sum())        /
((bbdd_filtro_5['num_siniestros']).sum())


filtro_6 = bbdd['area_residencia'] == 6

bbdd_filtro_6 = bbdd[filtro_6]
```

```
cm_6        =        ((bbdd_filtro_6['coste_siniestro_total']).sum())        /
((bbdd_filtro_6['num_siniestros']).sum())


cm_area = [cm_1, cm_2, cm_3, cm_4, cm_5, cm_6]


#Gráfico 16
zonas = [ "Area 1", "Area 2", "Area 3", "Area 4", "Area 5", "Area 6"]
area= pd.DataFrame(list(zip(cm_area, zonas)), columns = ['cm_area', 'zonas'])


plt.bar (area['zonas'], area['cm_area'])
plt.title("Coste medio asegurados por Áreas de residencia", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold' 'size': 18})
plt.xlabel("Áreas de residencia", size = 16,)
plt.ylabel("Coste medio", size = 16)



#Coste total
filtro_1 = bbdd['area_residencia'] == 1
bbdd_filtro_1 = bbdd[filtro_1]
ct_1 = ((bbdd_filtro_1['coste_siniestro_total']).sum())

filtro_2 = bbdd['area_residencia'] == 2
bbdd_filtro_2 = bbdd[filtro_2]
ct_2 = ((bbdd_filtro_2['coste_siniestro_total']).sum())

filtro_3 = bbdd['area_residencia'] == 3
bbdd_filtro_3 = bbdd[filtro_3]
ct_3 = ((bbdd_filtro_3['coste_siniestro_total']).sum())

filtro_4 = bbdd['area_residencia'] == 4
bbdd_filtro_4 = bbdd[filtro_4]
ct_4 = ((bbdd_filtro_4['coste_siniestro_total']).sum())

filtro_5 = bbdd['area_residencia'] == 5
bbdd_filtro_5 = bbdd[filtro_5]
ct_5 = ((bbdd_filtro_5['coste_siniestro_total']).sum())

filtro_6 = bbdd['area_residencia'] == 6
bbdd_filtro_6 = bbdd[filtro_6]
ct_6 = ((bbdd_filtro_6['coste_siniestro_total']).sum())
```

```
ct_area = [ct_1, ct_2, ct_3, ct_4, ct_5, ct_6]


#Gráfico 16
nombre = [ "Area 1", "Area 2", "Area 3", "Area 4", "Area 5", "Area 6"]
normdata = colors.Normalize(min(ct_area), max(ct_area))


plt.figure(figsize=(15, 10))
plt.pie(ct_area, labels = nombre , autopct="%0.1f %%")
plt.title("Coste total asegurados por Área de residencia", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})
plt.legend(loc='upper left')


### Indice de tráfico


#Coste medio
filtro_it10 = bbdd['indice_trafico'] < Cuantil_it10
bbdd_filtro_it10 = bbdd[filtro_it10]
cm_it10     =          ((bbdd_filtro_it10['coste_siniestro_total']).sum())     /
((bbdd_filtro_it10['num_siniestros']).sum())


filtro_it25 = (bbdd['indice_trafico'] > Cuantil_it10) & (bbdd['indice_trafico']
< Cuantil_it25)
bbdd_filtro_it25 = bbdd[filtro_it25]
cm_it25     =          ((bbdd_filtro_it25['coste_siniestro_total']).sum())     /
((bbdd_filtro_it25['num_siniestros']).sum())


filtro_it50 = (bbdd['indice_trafico'] > Cuantil_it25) & (bbdd['indice_trafico']
< Cuantil_it50)
bbdd_filtro_it50 = bbdd[filtro_it50]
cm_it50     =          ((bbdd_filtro_it50['coste_siniestro_total']).sum())     /
((bbdd_filtro_it50['num_siniestros']).sum())




filtro_it75 = (bbdd['indice_trafico'] > Cuantil_it50) & (bbdd['indice_trafico']
< Cuantil_it75)
bbdd_filtro_it75 = bbdd[filtro_it75]
cm_it75     =          ((bbdd_filtro_it75['coste_siniestro_total']).sum())     /
((bbdd_filtro_it75['num_siniestros']).sum())


filtro_it90 = (bbdd['indice_trafico'] > Cuantil_it75) & (bbdd['indice_trafico']
< Cuantil_it90)
```

```
bbdd_filtro_it90 = bbdd[filtro_it90]

cm_it90     =          ((bbdd_filtro_it90['coste_siniestro_total']).sum())     /
((bbdd_filtro_it90['num_siniestros']).sum())


filtro_it100 = (bbdd['indice_trafico'] > Cuantil_it90) & (bbdd['indice_trafico']
< max(bbdd['indice_trafico']))

bbdd_filtro_it100 = bbdd[filtro_it100]

cm_it100    =         ((bbdd_filtro_it100['coste_siniestro_total']).sum())    /
((bbdd_filtro_it100['num_siniestros']).sum())


cm_it= [cm_it10, cm_it25, cm_it50, cm_it75, cm_it90, cm_it100]


#Gráfico 17

Indice_trafico  =  [  "(0;  57,8)",  "(57,8;  82,1)",  "(82,1;  111,4)",  "(111,4;
135,7)", "(135,7; 148,5)", "(148,5; 207) "]

indice_trafico=   pd.DataFrame(list(zip(cm_it,   Indice_trafico)),   columns   =
['cm_it', 'Indice_trafico'])


plt.figure(figsize=(9,4))

plt.bar (indice_trafico['Indice_trafico'], indice_trafico['cm_it'])

plt.title("Coste medio asegurados por Índice de tráfico", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Indice de tráfico", size = 16,)

plt.ylabel("Coste medio total", size = 16)


#Coste total

filtro_it10 = bbdd['indice_trafico'] < Cuantil_it10

bbdd_filtro_it10 = bbdd[filtro_it10]

ct_it10 =  ((bbdd_filtro_it10['coste_siniestro_total']).sum())


filtro_it25 = (bbdd['indice_trafico'] > Cuantil_it10) & (bbdd['indice_trafico']
< Cuantil_it25)

bbdd_filtro_it25 = bbdd[filtro_it25]

ct_it25 =  ((bbdd_filtro_it25['coste_siniestro_total']).sum())


filtro_it50 = (bbdd['indice_trafico'] > Cuantil_it25) & (bbdd['indice_trafico']
< Cuantil_it50)

bbdd_filtro_it50 = bbdd[filtro_it50]

ct_it50 =  ((bbdd_filtro_it50['coste_siniestro_total']).sum())


filtro_it75 = (bbdd['indice_trafico'] > Cuantil_it50) & (bbdd['indice_trafico']
< Cuantil_it75)
```

```
bbdd_filtro_it75 = bbdd[filtro_it75]

ct_it75 =  ((bbdd_filtro_it75['coste_siniestro_total']).sum())


filtro_it90 = (bbdd['indice_trafico'] > Cuantil_it75) & (bbdd['indice_trafico']
< Cuantil_it90)

bbdd_filtro_it90 = bbdd[filtro_it90]

ct_it90 =  ((bbdd_filtro_it90['coste_siniestro_total']).sum())


filtro_it100 = (bbdd['indice_trafico'] > Cuantil_it90) & (bbdd['indice_trafico']
< max(bbdd['indice_trafico']))

bbdd_filtro_it100 = bbdd[filtro_it100]

ct_it100 =  ((bbdd_filtro_it100['coste_siniestro_total']).sum())


ct_it= [ct_it10, ct_it25, ct_it50, ct_it75, ct_it90, ct_it100]


#Gráfico 18

nombre =  [ "(0; 57,8)", "(57,8; 82,1)", "(82,1; 111,4)", "(111,4; 135,7)",
"(135,7; 148,5)", "(148,5; 207) "]

normdata = colors.Normalize(min(ct_it), max(ct_it))

colormap = cm.get_cmap("Reds")

colores =colormap(normdata(ct_it))


plt.figure(figsize=(15, 10))

plt.pie(ct_it, labels = nombre , autopct="%0.1f %%", colors = colores)

plt.title("Coste total asegurados por Índice de tráfico", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.legend(loc='upper left')


###Antigüedad del vehículo


#Coste medio

filtro_a1 = bbdd['antiguedad_vehiculo'] == 1

bbdd_filtro_a1 = bbdd[filtro_a1]

cm_a1       =       ((bbdd_filtro_a1['coste_siniestro_total']).sum())       /
((bbdd_filtro_a1['num_siniestros']).sum())


filtro_a2 = bbdd['antiguedad_vehiculo'] == 2

bbdd_filtro_a2 = bbdd[filtro_a2]

cm_a2       =       ((bbdd_filtro_a2['coste_siniestro_total']).sum())       /
((bbdd_filtro_a2['num_siniestros']).sum())
```

```
filtro_a3 = bbdd['antiguedad_vehiculo'] == 3

bbdd_filtro_a3 = bbdd[filtro_a3]

cm_a3       =       ((bbdd_filtro_a3['coste_siniestro_total']).sum())       /
((bbdd_filtro_a3['num_siniestros']).sum())


filtro_a4 = bbdd['antiguedad_vehiculo'] == 4

bbdd_filtro_a4 = bbdd[filtro_a4]

cm_a4       =       ((bbdd_filtro_a4['coste_siniestro_total']).sum())       /
((bbdd_filtro_a4['num_siniestros']).sum())


cm_a= [cm_a1, cm_a2, cm_a3, cm_a4]


#Gráfico 19

Edad_vehiculo = [ "1 año", "2 año", "3 año", "4 año"]

edad_vehiculo= pd.DataFrame(list(zip(cm_a, Edad_vehiculo)), columns = ['cm_a',
'Edad_vehiculo'])


plt.bar (edad_vehiculo['Edad_vehiculo'], edad_vehiculo['cm_a'])

plt.title("Coste medio asegurados por Edad del vehículo", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Edad del vehículo", size = 16,)

plt.ylabel("Coste medio", size = 16)


#Coste total

filtro_a1 = bbdd['antiguedad_vehiculo'] == 1

bbdd_filtro_a1 = bbdd[filtro_a1]

ct_a1 = ((bbdd_filtro_a1['coste_siniestro_total']).sum())


filtro_a2 = bbdd['antiguedad_vehiculo'] == 2

bbdd_filtro_a2 = bbdd[filtro_a2]

ct_a2 = ((bbdd_filtro_a2['coste_siniestro_total']).sum())


filtro_a3 = bbdd['antiguedad_vehiculo'] == 3

bbdd_filtro_a3 = bbdd[filtro_a3]

ct_a3 = ((bbdd_filtro_a3['coste_siniestro_total']).sum())


filtro_a4 = bbdd['antiguedad_vehiculo'] == 4

bbdd_filtro_a4 = bbdd[filtro_a4]

ct_a4 = ((bbdd_filtro_a4['coste_siniestro_total']).sum())


ct_a= [ct_a1, ct_a2, ct_a3, ct_a4]
```

```
#Gráfico 20

nombre =  [ "1 año", "2 año", "3 año", "4 año"]

normdata = colors.Normalize(min(ct_a), max(ct_a))

colormap = cm.get_cmap("Blues")

colores =colormap(normdata(ct_a))


plt.figure(figsize=(14, 9))

plt.pie(ct_a, labels = nombre , autopct="%0.1f %%", colors = colores)

plt.title("Coste total asegurados por Edad del vehículo", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.legend(loc='upper left')


### Tipo de vehículo


#Coste Medio


filtro_car = bbdd['B7_tipo_vehiculo'] == "Autocaravana"

bbdd_filtro_car = bbdd[filtro_car]

cm_car       =       ((bbdd_filtro_car['coste_siniestro_total']).sum())       /
((bbdd_filtro_car['num_siniestros']).sum())


filtro_b2 = bbdd['B7_tipo_vehiculo'] == "Berlina 2 vol"

bbdd_filtro_b2 = bbdd[filtro_b2]

cm_b2        =       ((bbdd_filtro_b2['coste_siniestro_total']).sum())       /
((bbdd_filtro_b2['num_siniestros']).sum())


filtro_b3 = bbdd['B7_tipo_vehiculo'] == "Berlina 3 vol"

bbdd_filtro_b3 = bbdd[filtro_b3]

cm_b3        =       ((bbdd_filtro_b3['coste_siniestro_total']).sum())       /
((bbdd_filtro_b3['num_siniestros']).sum())


filtro_cab = bbdd['B7_tipo_vehiculo'] == "Cabrio"

bbdd_filtro_cab = bbdd[filtro_cab]

cm_cab       =       ((bbdd_filtro_cab['coste_siniestro_total']).sum())       /
((bbdd_filtro_cab['num_siniestros']).sum())


filtro_cou = bbdd['B7_tipo_vehiculo'] == "Coupe"

bbdd_filtro_cou = bbdd[filtro_cou]

cm_cou       =       ((bbdd_filtro_cou['coste_siniestro_total']).sum())       /
((bbdd_filtro_cou['num_siniestros']).sum())
```

```
filtro_ddt = bbdd['B7_tipo_vehiculo'] == "Derivado de Turismo"

bbdd_filtro_ddt = bbdd[filtro_ddt]

cm_ddt       =       ((bbdd_filtro_ddt['coste_siniestro_total']).sum())       /
((bbdd_filtro_ddt['num_siniestros']).sum())


filtro_f = bbdd['B7_tipo_vehiculo'] == "Familiar"

bbdd_filtro_f = bbdd[filtro_f]

cm_f        =        ((bbdd_filtro_f['coste_siniestro_total']).sum())        /
((bbdd_filtro_f['num_siniestros']).sum())


filtro_fu = bbdd['B7_tipo_vehiculo'] == "Furgoneta"

bbdd_filtro_fu = bbdd[filtro_fu]

cm_fu       =       ((bbdd_filtro_fu['coste_siniestro_total']).sum())       /
((bbdd_filtro_fu['num_siniestros']).sum())



filtro_m = bbdd['B7_tipo_vehiculo'] == "Monovolumen"

bbdd_filtro_m = bbdd[filtro_m]

cm_m        =        ((bbdd_filtro_m['coste_siniestro_total']).sum())        /
((bbdd_filtro_m['num_siniestros']).sum())



filtro_p = bbdd['B7_tipo_vehiculo'] == "Pick-up"

bbdd_filtro_p = bbdd[filtro_p]

cm_p        =        ((bbdd_filtro_p['coste_siniestro_total']).sum())        /
((bbdd_filtro_p['num_siniestros']).sum())



filtro_t = bbdd['B7_tipo_vehiculo'] == "Targa"

bbdd_filtro_t = bbdd[filtro_t]

cm_t        =        ((bbdd_filtro_t['coste_siniestro_total']).sum())        /
((bbdd_filtro_t['num_siniestros']).sum())



filtro_tt = bbdd['B7_tipo_vehiculo'] == "Todo Terreno"

bbdd_filtro_tt = bbdd[filtro_tt]

cm_tt       =       ((bbdd_filtro_tt['coste_siniestro_total']).sum())       /
((bbdd_filtro_tt['num_siniestros']).sum())


filtro_desc = bbdd['B7_tipo_vehiculo'] == "Desconocido"

bbdd_filtro_desc = bbdd[filtro_desc]
```

```
cm_desc      =      ((bbdd_filtro_desc['coste_siniestro_total']).sum())      /
((bbdd_filtro_desc['num_siniestros']).sum())


cm_tv = [cm_car, cm_b2, cm_b3, cm_cab, cm_cou, cm_ddt, cm_f, cm_fu, cm_m, cm_p,
cm_t, cm_tt, cm_desc ]


#Gráfico 21

tipo_vehiculo = [ "Autocaravana", "Berlina 2 vol", "Berlina 3 vol", "Cabrio",
"Coupe", "Der. de Turismo", "Familiar", "Furgoneta", "Monovolumen", "Pick-up",
"Targa", "Todoterreno", "Desconocido"]

tipo_vehiculo=   pd.DataFrame(list(zip(cm_tv,   tipo_vehiculo)),   columns   =
['cm_tv', 'tipo_vehiculo'])


plt.figure(figsize=(18,9))

plt.bar (tipo_vehiculo['tipo_vehiculo'], tipo_vehiculo['cm_tv'])

plt.title("Coste medio asegurados por tipo del vehículo", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 25})

plt.xlabel("Tipo del vehículo", size = 16,)

plt.ylabel("Coste medio", size = 16)


#Coste total


filtro_car = bbdd['B7_tipo_vehiculo'] == "Autocaravana"

bbdd_filtro_car = bbdd[filtro_car]

ct_car = ((bbdd_filtro_car['coste_siniestro_total']).sum())


filtro_b2 = bbdd['B7_tipo_vehiculo'] == "Berlina 2 vol"

bbdd_filtro_b2 = bbdd[filtro_b2]

ct_b2 = ((bbdd_filtro_b2['coste_siniestro_total']).sum())


filtro_b3 = bbdd['B7_tipo_vehiculo'] == "Berlina 3 vol"

bbdd_filtro_b3 = bbdd[filtro_b3]

ct_b3 = ((bbdd_filtro_b3['coste_siniestro_total']).sum())


filtro_cab = bbdd['B7_tipo_vehiculo'] == "Cabrio"

bbdd_filtro_cab = bbdd[filtro_cab]

ct_cab = ((bbdd_filtro_cab['coste_siniestro_total']).sum())


filtro_cou = bbdd['B7_tipo_vehiculo'] == "Coupe"

bbdd_filtro_cou = bbdd[filtro_cou]

ct_cou = ((bbdd_filtro_cou['coste_siniestro_total']).sum())
```

```
filtro_ddt = bbdd['B7_tipo_vehiculo'] == "Derivado de Turismo"
bbdd_filtro_ddt = bbdd[filtro_ddt]
ct_ddt = ((bbdd_filtro_ddt['coste_siniestro_total']).sum())

filtro_f = bbdd['B7_tipo_vehiculo'] == "Familiar"
bbdd_filtro_f = bbdd[filtro_f]
ct_f = ((bbdd_filtro_f['coste_siniestro_total']).sum())

filtro_fu = bbdd['B7_tipo_vehiculo'] == "Furgoneta"
bbdd_filtro_fu = bbdd[filtro_fu]
ct_fu = ((bbdd_filtro_fu['coste_siniestro_total']).sum())

filtro_m = bbdd['B7_tipo_vehiculo'] == "Monovolumen"
bbdd_filtro_m = bbdd[filtro_m]
ct_m = ((bbdd_filtro_m['coste_siniestro_total']).sum())

filtro_p = bbdd['B7_tipo_vehiculo'] == "Pick-up"
bbdd_filtro_p = bbdd[filtro_p]
ct_p = ((bbdd_filtro_p['coste_siniestro_total']).sum())

filtro_t = bbdd['B7_tipo_vehiculo'] == "Targa"
bbdd_filtro_t = bbdd[filtro_t]
ct_t = ((bbdd_filtro_t['coste_siniestro_total']).sum())

filtro_tt = bbdd['B7_tipo_vehiculo'] == "Todo Terreno"
bbdd_filtro_tt = bbdd[filtro_tt]
ct_tt = ((bbdd_filtro_tt['coste_siniestro_total']).sum())

filtro_desc = bbdd['B7_tipo_vehiculo'] == "Desconocido"
bbdd_filtro_desc = bbdd[filtro_desc]
ct_desc = ((bbdd_filtro_desc['coste_siniestro_total']).sum())

ct_tv = [ct_car, ct_b2, ct_b3, ct_cab, ct_cou, ct_ddt, ct_f, ct_fu, ct_m, ct_p,
ct_t, ct_tt, ct_desc ]

#Gráfico 22
nombre = [ "Autocaravana", "Berlina 2 vol", "Berlina 3 vol", "Cabrio", "Coupe",
"Der. de Turismo", "Familiar", "Furgoneta", "Monovolumen", "Pick-up", "Targa",
"Todoterreno", "Desconocido"]
```

```
normdata = colors.Normalize(min(ct_tv), max(ct_tv))

colormap = cm.get_cmap("Blues")

colores =colormap(normdata(ct_tv))


plt.figure(figsize=(30, 20))

plt.pie(ct_tv, labels = nombre , autopct="%0.1f %%")

plt.title("Coste total asegurados por Tipo de vehículo", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.legend(loc='upper left')


### Valor vehículo


filtro_v1 = bbdd['valor_vehiculo'] < Cuantil_v25

bbdd_filtro_v1 = bbdd[filtro_v1]

cm_v1       =       ((bbdd_filtro_v1['coste_siniestro_total']).sum())       /
((bbdd_filtro_v1['num_siniestros']).sum())


filtro_v2 = (bbdd['valor_vehiculo'] > Cuantil_v25) &  (bbdd['valor_vehiculo'] <
Cuantil_v50)

bbdd_filtro_v2 = bbdd[filtro_v2]

cm_v2       =       ((bbdd_filtro_v2['coste_siniestro_total']).sum())       /
((bbdd_filtro_v2['num_siniestros']).sum())


filtro_v3 = (bbdd['valor_vehiculo'] > Cuantil_v50) &  (bbdd['valor_vehiculo'] <
Cuantil_v75)

bbdd_filtro_v3 = bbdd[filtro_v3]

cm_v3       =       ((bbdd_filtro_v3['coste_siniestro_total']).sum())       /
((bbdd_filtro_v3['num_siniestros']).sum())


filtro_v4 = (bbdd['valor_vehiculo'] > Cuantil_v75) &  (bbdd['valor_vehiculo'] <
Cuantil_v90)

bbdd_filtro_v4 = bbdd[filtro_v4]

cm_v4       =       ((bbdd_filtro_v4['coste_siniestro_total']).sum())       /
((bbdd_filtro_v4['num_siniestros']).sum())


filtro_v5 = (bbdd['valor_vehiculo'] > Cuantil_v90) &  (bbdd['valor_vehiculo'] <
max(bbdd['valor_vehiculo']))

bbdd_filtro_v5 = bbdd[filtro_v5]

cm_v5       =       ((bbdd_filtro_v5['coste_siniestro_total']).sum())       /
((bbdd_filtro_v5['num_siniestros']).sum())


cm_v= [cm_v1, cm_v2, cm_v3, cm_v4, cm_v5]
```

```
#Grafico 23

valor_vehiculo = [ "[0,11.100€)", "[11.100€, 16.500€)", "[16.500€, 23.650€)",
"[23.650€, 35.739€)", "[35.739€, 380.160]"]

valor_vehiculo=   pd.DataFrame(list(zip(cm_v,   valor_vehiculo)),   columns   =
['cm_v', 'valor_vehiculo'])

plt.figure(figsize=(10, 7))

plt.bar (valor_vehiculo['valor_vehiculo'], valor_vehiculo['cm_v'])

plt.title("Coste medio asegurados por Valor del vehículo",

          fontdict={'family': 'Calibri',

                    'color' : 'black',

                    'weight': 'bold',

                    'size': 25})

plt.xlabel("Valor del vehículo", size = 16,)

plt.ylabel("Coste medio", size = 16)


#Coste total

filtro_v1 = bbdd['valor_vehiculo'] < Cuantil_v25

bbdd_filtro_v1 = bbdd[filtro_v1]

ct_v1 = ((bbdd_filtro_v1['coste_siniestro_total']).sum())


filtro_v2 = (bbdd['valor_vehiculo'] > Cuantil_v25) &  (bbdd['valor_vehiculo'] <
Cuantil_v50)

bbdd_filtro_v2 = bbdd[filtro_v2]

ct_v2 = ((bbdd_filtro_v2['coste_siniestro_total']).sum())


filtro_v3 = (bbdd['valor_vehiculo'] > Cuantil_v50) &  (bbdd['valor_vehiculo'] <
Cuantil_v75)

bbdd_filtro_v3 = bbdd[filtro_v3]

ct_v3 = ((bbdd_filtro_v3['coste_siniestro_total']).sum())


filtro_v4 = (bbdd['valor_vehiculo'] > Cuantil_v75) &  (bbdd['valor_vehiculo'] <
Cuantil_v90)

bbdd_filtro_v4 = bbdd[filtro_v4]

ct_v4 = ((bbdd_filtro_v4['coste_siniestro_total']).sum())


filtro_v5 = (bbdd['valor_vehiculo'] > Cuantil_v90) &  (bbdd['valor_vehiculo'] <
max(bbdd['valor_vehiculo']))

bbdd_filtro_v5 = bbdd[filtro_v5]

ct_v5 = ((bbdd_filtro_v5['coste_siniestro_total']).sum())


ct_v= [ct_v1, ct_v2, ct_v3, ct_v4, ct_v5]
```

```
#Gráfico 24

nombre  =      ["[0,11.100€)",  "[11.100€,  16.500€)",  "[16.500€,  23.650€)",
"[23.650€, 35.739€)", "[35.739€, 380.160]"]


normdata = colors.Normalize(min(ct_v), max(ct_v))

colormap = cm.get_cmap("Reds")

colores =colormap(normdata(ct_v))


plt.figure(figsize=(20, 11))

plt.pie(ct_v, labels = nombre , autopct="%0.1f %%", colors = colores)

plt.title("Coste total asegurados por Valor del vehículo", fontdict={'family':
'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 25})

plt.legend(loc='upper left')



###Número de siniestros


#Coste medio

filtro_ns0 = bbdd['num_siniestros'] == 0

bbdd_filtro_ns0 = bbdd[filtro_ns0]

cm_ns0       =       ((bbdd_filtro_ns0['coste_siniestro_total']).sum())       /
((bbdd_filtro_ns0['num_siniestros']).sum())


filtro_ns1 = bbdd['num_siniestros'] == 1

bbdd_filtro_ns1 = bbdd[filtro_ns1]

cm_ns1       =       ((bbdd_filtro_ns1['coste_siniestro_total']).sum())       /
((bbdd_filtro_ns1['num_siniestros']).sum())


filtro_ns2 = bbdd['num_siniestros'] == 2

bbdd_filtro_ns2 = bbdd[filtro_ns2]

cm_ns2       =       ((bbdd_filtro_ns2['coste_siniestro_total']).sum())       /
((bbdd_filtro_ns2['num_siniestros']).sum())



filtro_ns3 = bbdd['num_siniestros'] == 3

bbdd_filtro_ns3 = bbdd[filtro_ns3]

cm_ns3       =       ((bbdd_filtro_ns3['coste_siniestro_total']).sum())       /
((bbdd_filtro_ns3['num_siniestros']).sum())
```

```
filtro_ns4 = bbdd['num_siniestros'] == 4

bbdd_filtro_ns4 = bbdd[filtro_ns4]

cm_ns4       =       ((bbdd_filtro_ns4['coste_siniestro_total']).sum())       /
((bbdd_filtro_ns4['num_siniestros']).sum())


filtro_ns5 = bbdd['num_siniestros'] == 5

bbdd_filtro_ns5 = bbdd[filtro_ns5]

cm_ns5       =       ((bbdd_filtro_ns5['coste_siniestro_total']).sum())       /
((bbdd_filtro_ns5['num_siniestros']).sum())



cm_ns= [cm_ns0, cm_ns1, cm_ns2, cm_ns3, cm_ns4, cm_ns5]


#Grafico 25

siniestros_vehiculo = [ "0 siniestros", "1 siniestros", "2 siniestros", "3
siniestros", "4 siniestros", "5 siniestros"]

siniestros_vehiculo=    pd.DataFrame(list(zip(cm_ns,    siniestros_vehiculo)),
columns = ['cm_ns', 'siniestros_vehiculo'])


plt.figure(figsize=(8,4))

plt.bar                          (siniestros_vehiculo['siniestros_vehiculo'],
siniestros_vehiculo['cm_ns'])

plt.title("Coste    medio    asegurados    por    siniestros    del    vehículo",
fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Siniestros del vehículo", size = 16,)

plt.ylabel("Coste medio", size = 16)


#Coste total

filtro_ns0 = bbdd['num_siniestros'] == 0

bbdd_filtro_ns0 = bbdd[filtro_ns0]

ct_ns0 = ((bbdd_filtro_ns0['coste_siniestro_total']).sum())


filtro_ns1 = bbdd['num_siniestros'] == 1

bbdd_filtro_ns1 = bbdd[filtro_ns1]

ct_ns1 = ((bbdd_filtro_ns1['coste_siniestro_total']).sum())


filtro_ns2 = bbdd['num_siniestros'] == 2

bbdd_filtro_ns2 = bbdd[filtro_ns2]

ct_ns2 = ((bbdd_filtro_ns2['coste_siniestro_total']).sum())
```

```
filtro_ns3 = bbdd['num_siniestros'] == 3

bbdd_filtro_ns3 = bbdd[filtro_ns3]

ct_ns3 = ((bbdd_filtro_ns3['coste_siniestro_total']).sum())



filtro_ns4 = bbdd['num_siniestros'] == 4

bbdd_filtro_ns4 = bbdd[filtro_ns4]

ct_ns4 = ((bbdd_filtro_ns4['coste_siniestro_total']).sum())


filtro_ns5 = bbdd['num_siniestros'] == 5

bbdd_filtro_ns5 = bbdd[filtro_ns5]

ct_ns5 = ((bbdd_filtro_ns5['coste_siniestro_total']).sum())



ct_ns= [ct_ns0, ct_ns1, ct_ns2, ct_ns3, ct_ns4, ct_ns5]


#Gráfico 26

nombre =  [  "1 siniestros", "2 siniestros", "3 siniestros", "4 siniestros", "5
siniestros"]

normdata = colors.Normalize(min(ct_v), max(ct_ns))

colormap = cm.get_cmap("Blues")

colores =colormap(normdata(ct_ns))


plt.figure(figsize=(18, 9))

plt.pie(ct_v, labels = nombre , autopct="%0.1f %%", colors = colores)

plt.title("Coste total asegurados por Número de siniestros", fontdict={'family':
'Calibri', 'color': 'black', 'weight': 'bold', 'size': 18})

plt.legend(loc='upper left')


'''La variable tipos de vehículo debemos modificarla,

Tras realizar el estudio, los vehículos desconocidos no se comportan como la
moda'''


#respecto exposición, como se aprecia son todos muy similares.


exposicion_tv  =  (np.array([exposicion_car,  exposicion_b2,  exposicion_b3,
exposicion_cab, exposicion_cou, exposicion_ddt, exposicion_desc, exposicion_f,
exposicion_fu, exposicion_m, exposicion_p, exposicion_t, exposicion_tt]))

B7_tipo_vehiculo                                                              =
pd.DataFrame((bbdd['poliza']).groupby(bbdd['B7_tipo_vehiculo']).count()).to_nu
mpy()

B7_tipo_vehiculo= (B7_tipo_vehiculo.T)
```

```
exposicion_media_tv = exposicion_tv / B7_tipo_vehiculo

print(exposicion_media_tv)

#Respecto al coste medio se comporta entre Pick-up y Targa

print(cm_desc)

print(cm_t)

print(cm_p)


#Respecto al valor del vehículo se encuentra entre Pick-up y Targa

B7_valor_vehiculo                                                    =
pd.DataFrame((bbdd['valor_vehiculo']).groupby(bbdd['B7_tipo_vehiculo']).mean()
)

bbdd['B7_tipo_vehiculo_mod']  =  bbdd.B7_tipo_vehiculo.replace({"Desconocido":
"Berlina 2 vol"})

pd.DataFrame((bbdd['poliza']).groupby(bbdd['B7_tipo_vehiculo_mod']).count())


############

###EXTRA####

############

bbdd= bbdd.drop([ 'Unnamed: 16', 'Unnamed: 17'], axis=1)

corr = bbdd.corr()

import seaborn as sns

plt.figure(figsize=(18, 15))

# Definir tipo de matriz de corr

tipo_mask = np.triu(np.ones_like(bbdd.corr(), dtype=np.bool))

heatmap = sns.heatmap(bbdd.corr(), mask=tipo_mask, vmin=-1, vmax=1, annot=True,
cmap='coolwarm')



###############

##MODELO GLM##

###############


####Eliminamos valores punta que puedan a afectar al modelo de severidad.

####Los eliminamos antes de los modelos de frecuencia para mantener la
concordancia


#Severidad.

# Como es la variable.

#deciles

bbdd_j = bbdd
```

```
bbdd_j = bbdd_j.sort_values( 'coste_siniestro_total')   #ordenamos por coste siniestro

x = bbdd_j[bbdd_j["coste_siniestro_total"]== 0].index

bbdd_j=  bbdd_j.drop(x)

sev_0 = bbdd_j['coste_siniestro_total'].to_numpy()


Sev_deciles = np.percentile(sev_0, np.arange(0,100,10))

sev_percentiles = np.percentile(sev_0, np.arange(0,100,1))


### Gráfico

fig = plt.figure(figsize=(20,9))

ax1 = fig.add_subplot(1,2,1)

ax2= fig.add_subplot(1,2,2)


ax1.plot (Sev_deciles, color='#F2AB6D')

ax1.set_title("Distribución acumulada de la severidad. Deciles", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 25})

ax1.set_xlabel("Decil", size = 16,)


ax2.plot (sev_percentiles, color='Blue')


ax2.set_title("Distribución acumulada de la severidad. Percentiles",

          fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 25})

ax2.set_xlabel("Percentil", size = 16,)


variación_percentil = np.zeros(100)

for i in range (0, len(sev_percentiles)-1):

    variación_percentil[i+1] = (sev_percentiles[i+1]- sev_percentiles[i]) / sev_percentiles[i]

    i= i+1


plt.figure(figsize=(20,9))


plt.plot (variación_percentil, color='#F2AB6D')

plt.title("Incremento. Porcentiles", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 25})

ax1.set_xlabel("Porcentiles", size = 16,)


#Valores punta

variación_percentil = np.delete (variación_percentil, (95, 96, 97,98,99))
```

```
plt.figure(figsize=(20,9))


plt.plot (variación_percentil, color='#F2AB6D')

plt.title("Incremento. Porcentiles", fontdict={'family': 'Calibri', 'color' :
'black', 'weight': 'bold', 'size': 25})

ax1.set_xlabel("Porcentiles", size = 16,)


valor_limite = np.percentile(bbdd_j['coste_siniestro_total'], 95)

coste_siniestro = bbdd['coste_siniestro_total'].to_numpy()

n= len(bbdd['coste_siniestro_total'])

valores_punta= np.empty(n)

valores_normales = np.empty(n)

valores_0= np.empty(n)


for i in range (0, n):

    if coste_siniestro[i] ==0:

        valores_0[i] = coste_siniestro[i]

    elif coste_siniestro[i] == valor_limite:

        valores_normales[i] =coste_siniestro[i]

    else:

        valores_normales[i] =coste_siniestro[i]

        valores_punta[i] =coste_siniestro[i]


valores_punta = np.delete(valores_punta, valores_punta<valor_limite)

min(valores_punta)

max(valores_punta)

len(valores_punta)/n


bbdd_2 = bbdd

bbdd_2  =  bbdd_2.sort_values(  'coste_siniestro_total')   #ordenamos  por  coste
siniestro

x = bbdd_2[bbdd_2["coste_siniestro_total"]>= valor_limite].index #seleccionamos
los valores punta

bbdd_2=  bbdd_2.drop(x)

len(bbdd_2)   #comprobamos que se haya eliminado correctamente

max(bbdd_2['coste_siniestro_total'])

bbdd_2['coste_siniestro_total'] = bbdd_2['coste_siniestro_total'].astype(int)

bbdd_2 = bbdd_2.sort_values( 'poliza')
```

```
#Variable Credit Scoring
moda_cs= float((bbdd_2['credit_scoring'].mode()))
bbdd_2['credit_scoring'] = bbdd_2.credit_scoring.fillna(moda_cs)

Cuantil_cs10 =(bbdd_2['credit_scoring']).quantile(0.1)   #452
Cuantil_cs25 =(bbdd_2['credit_scoring']).quantile(0.25)  #606
Cuantil_cs50 =(bbdd_2['credit_scoring']).quantile(0.5)   #675
Cuantil_cs75 =(bbdd_2['credit_scoring']).quantile(0.75)  #767
Cuantil_cs90 =(bbdd_2['credit_scoring']).quantile(0.9)   #819

# Variable indice de trafico
indice_trafico                                                              =
pd.DataFrame((bbdd_2['poliza']).groupby(bbdd_2['indice_trafico']).count())
#Asumimos que donde no teníamos información pertenecen a la moda de la variable
moda_it= float((bbdd_2['indice_trafico'].mode()))
bbdd_2['indice_trafico'] = bbdd_2.indice_trafico.fillna(moda_it)

    #Segmentamos por cuantiles
Cuantil_it10 =(bbdd_2['indice_trafico']).quantile(0.1)
Cuantil_it25 =(bbdd_2['indice_trafico']).quantile(0.25)
Cuantil_it50 =(bbdd_2['indice_trafico']).quantile(0.5)
Cuantil_it75 =(bbdd_2['indice_trafico']).quantile(0.75)
Cuantil_it90 =(bbdd_2['indice_trafico']).quantile(0.9)

#Variable tipo de vehículo

bbdd_2['B7_tipo_vehiculo_mod']                                              =
bbdd_2.B7_tipo_vehiculo.replace({"Desconocido": "Berlina 2 vol"})
pd.DataFrame((bbdd_2['poliza']).groupby(bbdd_2['B7_tipo_vehiculo_mod']).count(
))

#Variable valor del vehículo
Cuantil_v25 =(bbdd_2['valor_vehiculo']).quantile(0.25
Cuantil_v50 =(bbdd_2['valor_vehiculo']).quantile(0.5)
Cuantil_v75 =(bbdd_2['valor_vehiculo']).quantile(0.75)
Cuantil_v90 =(bbdd_2['valor_vehiculo']).quantile(0.9)

#######Creación de variables. Modelo GLM

bbdd_2['credit_scoring_mod'] = np.where(bbdd_2['credit_scoring'] < 351, 350,
np.where(bbdd_2['credit_scoring'] < 401, 400, np.where(bbdd_2['credit_scoring']
```

```
<     451,     450,     np.where(bbdd_2['credit_scoring']     <     501,     500,
np.where(bbdd_2['credit_scoring'] < 551, 550, np.where(bbdd_2['credit_scoring']
<     601,     600,     np.where(bbdd_2['credit_scoring']     <     651,     650,
np.where(bbdd_2['credit_scoring'] < 701, 700,750)))))))


bbdd_2['credit_scoring_mod2']  =  np.where(bbdd_2['credit_scoring']  <  485,  0,
np.where(bbdd_2['credit_scoring'] < 650, 1, 2))

bbdd_2['area_residencia_46']  =  np.where(bbdd_2['area_residencia'].isin  ([4,
6]),4, bbdd_2['area_residencia'])

bbdd_2['area_residencia_23_46']  =  np.where(bbdd_2['area_residencia_46']==2,3,
bbdd_2['area_residencia_46'])

bbdd_2['area_residencia_36']  =  np.where(bbdd_2['area_residencia'].isin  ([3,
6]),3, bbdd_2['area_residencia'])


bbdd_2['indice_trafico_mod'] = np.where(bbdd_2['indice_trafico'] <

np.where(bbdd_2['indice_trafico']      <      Cuantil_it25+1,      Cuantil_it25,
np.where(bbdd_2['indice_trafico']      <      Cuantil_it50+1,      Cuantil_it50,
np.where(bbdd_2['indice_trafico']      <      Cuantil_it75+1,      Cuantil_it75,
np.where(bbdd_2['indice_trafico']      <      Cuantil_it90+1,      Cuantil_it90,
max(bbdd_2['indice_trafico'])))))


bbdd_2['edad_mayores70'] = np.where(bbdd_2['edad'] > 71, 1, 0)

bbdd_2['valor_vehículo_tramos']     =     np.where(bbdd_2['valor_vehiculo']     <
(bbdd_2['valor_vehiculo']).quantile(0.25),  0,np.where(bbdd_2['valor_vehiculo']
< (bbdd_2['valor_vehiculo']).quantile(0.75), 1, 2))


bbdd_2['edad_tramos']=np.where(bbdd_2['edad']<(bbdd_2['edad']).quantile(0.25),
0,np.where(bbdd_2['edad'] < (bbdd_2['edad']).quantile(0.75), 1, 2))


bbdd_2['antiguedad_vehiculo_23'] = np.where(bbdd_2['antiguedad_vehiculo'].isin
([3]) , 2, bbdd_2['antiguedad_vehiculo'])


######################

##Modelo de Frecuencia##

######################

seed = 45

freq_train, freq_test = train_test_split(bbdd_2, test_size=0.2, random_state =
seed)


len(freq_train)

len(freq_test)

freq_train['num_siniestros'].value_counts(normalize=True)

freq_test['num_siniestros'].value_counts(normalize=True)
```

```
#Variables a utilizar
num_siniestros =            freq_train['num_siniestros']
coste_siniestro_total=      freq_train['coste_siniestro_total']

edad =                      freq_train['edad']
edad_mayores70 =              freq_train['edad_mayores70']

exposicion =                freq_train['exposure']
sexo =                      freq_train['sexo']
credit_scoring_mod =       freq_train['credit_scoring_mod']
credit_scoring_mod2 =       freq_train['credit_scoring_mod2']

area_residencia_23_46=      freq_train['area_residencia_23_46']

valor_vehiculo =            freq_train['valor_vehiculo']
indice_trafico_mod =        freq_train['indice_trafico_mod']



####MODELO PRUEBA####
modelo_prueba = 'num_siniestros ~ C(sexo, Treatment (reference = "F" ))'


y_train, x_train = dmatrices( modelo_prueba, data = freq_train, return_type=
'dataframe' )
y_test, x_test = dmatrices( modelo_prueba, data = freq_test, return_type=
'dataframe' )


modelo_prueba = GLM(y_train, x_train, exposure = exposicion,   family =
sm.families.Poisson(link = sm.families.links.log)).fit()


print(modelo_prueba.summary())
print(modelo_prueba.aic)
#Modelo 1. Todo significativo. Univariante


modelo_1 = 'num_siniestros ~ C(sexo, Treatment (reference = "F" )) + edad  +
credit_scoring_mod  +C(edad_mayores70, Treatment (reference = 0 ))    +
credit_scoring_mod2 + C(area_residencia_23_46, Treatment (reference = 3 ))
+indice_trafico_mod   + valor_vehiculo'


y_train, x_train = dmatrices( modelo_1, data = freq_train, return_type=
'dataframe' )
y_test, x_test = dmatrices( modelo_1, data = freq_test, return_type= 'dataframe'
)
```

```
modelo_1 = GLM(y_train, x_train, exposure = exposicion, family = sm.families.Poisson(link = sm.families.links.log)).fit()


print(modelo_1.summary())

NLL = (modelo_1.llf) #loglikelyhood

print(modelo_1.aic)

print(modelo_1.bic)

#GE = deviance/n° obs

GE = 34622/47912

##Comprobación con test

seed = 45

freq_train, freq_test = train_test_split(bbdd_2, test_size=0.2, random_state = seed)


len(freq_train)

len(freq_test)

freq_train['num_siniestros'].value_counts(normalize=True)

freq_test['num_siniestros'].value_counts(normalize=True)



#Variables a utilizar

num_siniestros =            freq_test['num_siniestros']

coste_siniestro_total=      freq_test['coste_siniestro_total']


edad =                      freq_test['edad']

edad_mayores70 =             freq_test['edad_mayores70']

exposicion =                freq_test['exposure']

sexo =                      freq_test['sexo']

credit_scoring_mod2 =       freq_test['credit_scoring_mod2']

credit_scoring_mod =       freq_test['credit_scoring_mod']

area_residencia_23_46=      freq_train['area_residencia_23_46']

valor_vehiculo =            freq_test['valor_vehiculo']

indice_trafico_mod =        freq_test['indice_trafico_mod']



#Modelo 1. Todo significativo. Univariante

modelo_1t = 'num_siniestros ~ C(sexo, Treatment (reference = "F" )) + edad  + credit_scoring_mod  +C(edad_mayores70,  Treatment  (reference  =  0  ))    +
```

```
credit_scoring_mod2 + C(area_residencia_23_46, Treatment (reference = 3 ))
+indice_trafico_mod   + valor_vehiculo'



modelo_1t = glm( modelo_1t, freq_test, offset = np.log(freq_test['exposure']),
family=sm.families.Poisson(link=sm.families.links.log)).fit()


print(modelo_1t.summary())

print(modelo_1t.aic)

print(modelo_1t.bic)


#Otra posibilidad

modelo_6 = 'num_siniestros ~ edad_jovenes +edad_jovenes2 + edad_mayores+
credit_scoring_2 + jovenes_scoring + mayores_scoring + C(area_residencia_2,
Treatment (reference = 2 )) + indice_trafico_mod + C(antiguedad_vehiculo_2,
Treatment(reference=2))                    +          C(B7_tipo_vehiculo_mod_2,
Treatment(reference="Berlinas")) + valor_vehiculo_bajo + valor_vehiculo_alto '


modelo_6 = glm( modelo_6, freq_train, offset = np.log(freq_train['exposure']),
family=sm.families.Poisson(link=sm.families.links.log)).fit()


print(modelo_6.summary())

print(modelo_6.aic)

print(modelo_6.bic)


#No significativo en el test

modelo_6t = glm( modelo_6, freq_test, offset = np.log(freq_test['exposure']),
family=sm.families.Poisson(link=sm.families.links.log)).fit()


print(modelo_6t.summary())

print(modelo_6t.aic)

print(modelo_6t.bic)



#########################

###### GLM SEVERIDAD#####

#########################

bbdd_3 = bbdd_2

bbdd_3 = bbdd_3.sort_values( 'coste_siniestro_total')   #ordenamos por coste
siniestro

x = bbdd_3[bbdd_3["coste_siniestro_total"]== 0].index #seleccionamos los valores
punta

bbdd_3=  bbdd_3.drop(x)
```

```
len(bbdd_3)   #comprobamos que se haya eliminado correctamente
max(bbdd_3['coste_siniestro_total'])


seed = 45
coste_train, coste_test = train_test_split(bbdd_3, test_size=0.2, random_state
= seed)


len(coste_train)
len(coste_test)
coste_train['num_siniestros'].value_counts(normalize=True)
coste_test['num_siniestros'].value_counts(normalize=True)



#Variables a utilizar


coste_siniestro_total=      coste_train['coste_siniestro_total']


edad_tramos =              coste_train ['edad_tramos']


exposicion =                coste_train ['exposure']


credit_scoring=             coste_train['credit_scoring']


credit_scoring_mod2 =       coste_train['credit_scoring_mod2']


area_residencia_36 =         coste_train['area_residencia_36']


antiguedad_vehiculo_23 =    coste_train['antiguedad_vehiculo_23']
valor_vehículo_tramos = coste_train['valor_vehículo_tramos']



modelo_1sg      =      'coste_siniestro_total      ~      C(credit_scoring_mod2,
Treatment(reference=2)) + C(valor_vehículo_tramos, Treatment (reference = 1 ))
+ C(area_residencia_36, Treatment (reference = 3 )) + C(edad_tramos, Treatment
(reference = 1 )) +  C(antiguedad_vehiculo_23, Treatment (reference = 2 )) '



y_train, x_train = dmatrices( modelo_1sg, data = coste_train, return_type=
'dataframe' )
y_test, x_test = dmatrices( modelo_1sg, data = coste_test, return_type=
'dataframe' )
```

```
modelo_1sg = GLM (y_train, x_train, family = sm.families.Gamma(link = sm.families.links.log)).fit()


print(modelo_1sg.summary())

print(modelo_1sg.aic)

print(modelo_1sg.bic)


#####MODELO SEVERIDAD TEST

coste_siniestro_total=      coste_test['coste_siniestro_total']


edad_tramos =               coste_test ['edad_tramos']


credit_scoring_mod2 =         coste_test['credit_scoring_mod2']


area_residencia_36 =          coste_test['area_residencia_36']


antiguedad_vehiculo_23 =      coste_test['antiguedad_vehiculo_23']

valor_vehículo_tramos =        coste_test['valor_vehículo_tramos']


modelo_1sgt    =    'coste_siniestro_total    ~        C(credit_scoring_mod2, Treatment(reference=2)) + C(valor_vehículo_tramos, Treatment (reference = 1 )) +C(area_residencia_36, Treatment (reference = 3 )) + C(edad_tramos, Treatment (reference = 1 )) +  C(antiguedad_vehiculo_23, Treatment (reference = 2 ))  '


y_train, x_train = dmatrices( modelo_1sgt, data = coste_train, return_type= 'dataframe' )

y_test, x_test = dmatrices( modelo_1sgt, data = coste_test, return_type= 'dataframe' )



modelo_1sgt = GLM (y_test, x_test, family = sm.families.Gamma(link = sm.families.links.log)).fit()


print(modelo_1sgt.summary())


######################

##MODELO BURNING COST##

######################

#Debemos unir ambos modelos. Frecuencia y severidad. Para luego multiplicar ambos resultados.
```

```
freq_train['freq_modelada']= modelo_1.predict()
freq_test['freq_modelada']= modelo_1t.predict()

Frecuencia_final= freq_train.append(freq_test)

Resultado_final = Frecuencia_final.sort_values('poliza')
Resultado_final['freq_modelada'] = Resultado_final['freq_modelada']

coste_train['coste_modelado']= modelo_1sg.predict()
coste_test['coste_modelado']= modelo_1sgt.predict()

Severidad= coste_train.append(coste_test)
Severidad = Severidad.sort_values('poliza')

Resultado_final_GLM = pd.merge( Resultado_final, Severidad[['poliza', 'coste_modelado']], on='poliza', how='left')
Resultado_final_2_GLM = pd.merge( Severidad[['poliza', 'coste_modelado']], Resultado_final, on='poliza', how='left')

Resultado_final_GLM['coste_modelado']
#Incluimos el val valor base para aquellos sin siniestro = Resultado_final_GLM.coste_modelado.fillna(math.exp(6.3273))
Resultado_final_GLM['Burning_cost'] = Resultado_final_GLM['coste_modelado'] * Resultado_final_GLM['freq_modelada']

Resultado_final_GLM = Resultado_final_GLM[['poliza', 'freq_modelada', 'coste_modelado', 'Burning_cost']]

np.mean(Resultado_final_GLM)
Resultado_final_GLM.iloc[27:32]   ##Ejemplo de pólizas



################
###Modelo GBM###
################

bbdd['sexo']= pd.get_dummies(bbdd['sexo']) #Mujer es 1 y Hombre 0

bbdd['tipo_vehiculo'] = ""
B7_tipo_vehiculo = list( bbdd['B7_tipo_vehiculo'])
```

```
tipo_vehiculo = list (np.zeros(len(bbdd['tipo_vehiculo'])))

for i in range (0, len(bbdd['tipo_vehiculo'])):
    if B7_tipo_vehiculo [i] == "Autocaravana":
        tipo_vehiculo [i] = 1
    elif B7_tipo_vehiculo [i] == "Berlina 2 vol":
        tipo_vehiculo [i] = 2
    elif B7_tipo_vehiculo [i] == "Berlina 3 vol":
        tipo_vehiculo [i] = 3
    elif B7_tipo_vehiculo[i] == "Cabrio":
        tipo_vehiculo [i] = 4
    elif B7_tipo_vehiculo [i]== "Coupe":
        tipo_vehiculo [i] = 5
    elif B7_tipo_vehiculo[i] == "Derivado de Turismo":
        tipo_vehiculo [i] = 6
    elif B7_tipo_vehiculo [i] == "Desconocido":
        tipo_vehiculo [i] = 7
    elif B7_tipo_vehiculo [i] == "Familiar":
       tipo_vehiculo [i] = 8
    elif B7_tipo_vehiculo [i] == "Furgoneta":
        tipo_vehiculo [i] = 9
    elif B7_tipo_vehiculo [i] == "'Monovolumen'":
       tipo_vehiculo [i] = 10
    elif B7_tipo_vehiculo [i] == "Pick-up":
        tipo_vehiculo [i] = 11
    elif B7_tipo_vehiculo [i] == "Targa":
        tipo_vehiculo [i] = 12
    elif B7_tipo_vehiculo [i] == "Todo Terreno":
        tipo_vehiculo [i] = 13

bbdd['tipo_vehiculo'] = tipo_vehiculo

pd.DataFrame((bbdd['poliza']).groupby(bbdd['tipo_vehiculo']).count())
pd.DataFrame((bbdd['poliza']).groupby(bbdd['B7_tipo_vehiculo']).count())

#variables con resultados vacíos

#########################
###GBM SOBRE FRECUENCIA###
```

```
#########################

seed=45


variables = ['sexo', 'edad', 'credit_scoring', 'indice_trafico',
'antiguedad_vehiculo','area_residencia','tipo_vehiculo','valor_vehiculo','expo
sure','num_siniestros']

freq_train, freq_test = train_test_split(bbdd_2, test_size=0.20, random_state =
seed)


###Entrenamiento

freq_train_2 = freq_train [variables]

freq_train_2_y = freq_train_2['num_siniestros'].values /
freq_train_2['exposure'].values

freq_train_2_w = freq_train_2['exposure'].values

freq_train_2_X = freq_train_2.drop(['num_siniestros', 'exposure'], axis = 1)

###Test

freq_test_2 = freq_test[variables]

freq_test_2_y = freq_test_2['num_siniestros'].values

freq_test_2_w = freq_test_2['exposure'].values

freq_test_2_X = freq_test_2.drop(['num_siniestros', 'exposure'], axis = 1)


GBM_freq = lgb.LGBMRegressor(max_depth = 3, learning_rate = 0.1, n_estimators =
200, objective = "poisson", min_child_samples = 200, importance_type = "gain",
random_state = seed)


GBM_freq.fit(freq_train_2_X, freq_train_2_y, sample_weight = freq_train_2_w)

GBM_freq.feature_importances_


importances= GBM_freq.feature_importances_ / max(GBM_freq.feature_importances_)

feature_imp = pd.DataFrame(sorted(zip(importances, GBM_freq.feature_name_)),

columns=['Valor','Variable'])

plt.figure(figsize=(10, 5))

plt.grid(True)

sns.barplot(x="Valor", y="Variable", data=feature_imp.sort_values(by="Valor",
ascending=False))

plt.title('Variables LightGBM')

plt.xlabel('Importancia relativa de las variables')

plt.tight_layout()


import os

os.environ["PATH"]+=os.pathsep+'C:/Users/Gonzalo/anaconda3/Library/bin/graphvi
z'
```

```
#Primer arbol

lgb.plot_tree(GBM_freq, tree_index = 0, show_info = ["internal_value",
"leaf_count", "split_gain"], figsize=(20, 10), dpi = 150, orientation =
'horizontal')

#Último árbol

lgb.plot_tree(GBM_freq, tree_index = 199, show_info = ["internal_value",
"leaf_count", "split_gain"], figsize=(20, 10), dpi = 150, orientation =
'horizontal')



lgb.plot_split_value_histogram(GBM_freq, feature = "credit_scoring", width_coef
= 2)

plt.title('Cortes acumulados de Credit Scoring')

plt.xlabel ('Valores de corte')

plt.ylabel ('Número de cortes')

lgb.plot_split_value_histogram(GBM_freq, feature = "edad", width_coef = 0.8)

plt.title('Cortes acumulados de la Edad del conductor')


#Predicción LGBM

fitted_y = GBM_freq.predict(freq_train_2_X)

pred_y = GBM_freq.predict(freq_test_2_X)


#Entrenamiento

freq_train_results = freq_train_2[['num_siniestros', 'exposure']]

freq_train_results['Predicción GBM entrenamiento'] = fitted_y * freq_train_2_w

media_freq_GBM_train = sum(freq_train_results['num_siniestros']) /
sum(freq_train_results['exposure'])


#Test

freq_test_results = freq_test_2[['num_siniestros', 'exposure']]

#freq_test_results['GLM prediction'] = freq_fitted

freq_test_results['Predicción GBM test'] = pred_y * freq_test_2_w

media_freq_GBM_test=sum(freq_test_results['num_siniestros'])/
sum(freq_test_results['exposure'])


      ####################

      ##CROSS VALIDATION##

      ####################

def DevianceP(y_i, mu_i):

 D = np.empty(shape = y_i.shape[0])

 for i in range(y_i.shape[0]):
```

```
    if y_i[i] == 0:
        D[i] = mu_i[i] - y_i[i]
    else:
       D[i] = y_i[i] * np.log(y_i[i] / mu_i[i]) - (y_i[i] - mu_i[i])

 return(2 * sum(D))

parametros = { 'ratio_aprendizaje': [0.005, 0.01, 0.02, 0.05], 'n_arboles': [500, 750, 1000], 'profundidad': [3,4,5]}

def param_GBM_frec(nfolds, X, y, pesos, ratio_apren , n_arboles, profundidad ):

 seed = 45
 GBM_frec = lgb.LGBMRegressor(max_depth = profundidad, learning_rate = ratio_apren, n_estimators = n_arboles, objective = 'poisson', random_state = seed, importance_type = "gain")

 kf = KFold(n_splits=nfolds, shuffle=False)
 GEpoissonCV = []

 for train_index, test_index in kf.split(X, y):

    X_train, X_val = X.iloc[train_index], X.iloc[test_index]
    y_train, y_val = y[train_index], y[test_index]
    w_train, w_val = pesos[train_index], pesos[test_index]

    GBM_frec.fit(X_train, y_train, sample_weight=w_train, eval_set=[(X_val, y_val)], early_stopping_rounds=20)
    eval_pred = GBM_frec.predict(X_val) * w_val
    y_val = y_val*w_val
    Dpois = DevianceP(y_val, eval_pred)
    GEpoissonCV.append(Dpois / y_val.shape[0])

    GEpoissonCVav = np.mean(GEpoissonCV)

 return GEpoissonCVav

GEs = []
ratio_apren = []
n_arboles = []
profundidad = []
```

```
for i in parametros['ratio_aprendizaje']:
    for j in parametros['n_arboles']:
        for k in parametros['profundidad']:
            GE = param_GBM_frec(nfolds = 5, X = freq_train_2_X, y = freq_train_2_y, pesos=freq_train_2_w, ratio_apren = i, n_arboles = j, profundidad = k)
            GEs.append(GE)

            ratio_apren.append(i)
            n_arboles.append(j)
            profundidad.append(k)

Resultados_generales = pd.DataFrame({'Ratio aprendizaje':ratio_apren, 'N° de árboles':n_arboles, 'Profundidad':profundidad, 'Error generalizado':GEs})
Resultados_generales = Resultados_generales.set_index(['Ratio aprendizaje', 'N° de árboles', 'Profundidad'])

Resultados_generales.sort_values(by = "Ratio aprendizaje")
Resultados_generales.sort_values(by = "Error generalizado").head(10)

###creamos el modelo óptimo

GBM_freq_final = lgb.LGBMRegressor(max_depth = 3, learning_rate = 0.05, n_estimators = 750, objective = "poisson", importance_type = "gain", random_state = seed)

GBM_freq_final.fit(freq_train_2_X, freq_train_2_y, sample_weight = freq_train_2_w)
GBM_freq_final.feature_importances_

importances = GBM_freq_final.feature_importances_ / max(GBM_freq_final.feature_importances_)
feature_imp = pd.DataFrame(sorted(zip(importances, GBM_freq_final.feature_name_)),
columns=['Valor','Variable'])
plt.figure(figsize=(10, 5))
plt.grid(True)
sns.barplot(x="Valor", y="Variable", data=feature_imp.sort_values(by="Valor", ascending=False))
plt.title('Variable LightGBM')
plt.xlabel('Importancia relativa de las variables')
plt.tight_layout()
```

```
import os

os.environ["PATH"]+=os.pathsep+'C:/Users/Gonzalo/anaconda3/Library/bin/graphviz'


#Primer arbol

lgb.plot_tree(GBM_freq_final, tree_index = 0, show_info = ["internal_value", "leaf_count", "split_gain"], figsize=(20, 10), dpi = 150, orientation = 'horizontal')

#Último árbol

lgb.plot_tree(GBM_freq_final, tree_index = 749, show_info = ["internal_value", "leaf_count", "split_gain"], figsize=(20, 10), dpi = 150, orientation = 'horizontal')



lgb.plot_split_value_histogram(GBM_freq_final, feature = "credit_scoring", width_coef = 2)

plt.title('Cortes acumulados de Credit Scoring')

plt.xlabel ('Valores de corte')

plt.ylabel ('Número de cortes')

lgb.plot_split_value_histogram(GBM_freq_final, feature = "edad", width_coef = 0.8)

plt.title('Cortes acumulados de la Edad del conductor')

plt.xlabel ('Valores de corte')

plt.ylabel ('Número de cortes')


#Predicción LGBM

fitted_y = GBM_freq_final.predict(freq_train_2_X)

pred_y = GBM_freq_final.predict(freq_test_2_X)


#Entrenamiento

freq_train_results = freq_train_2[['num_siniestros', 'exposure']]

#freq_train_results['GLM prediction'] = freq_fitted

freq_train_results['Predicción GBM entrenamiento'] = fitted_y * freq_train_2_w


#Test

freq_test_results = freq_test_2[['num_siniestros', 'exposure']]

#freq_test_results['GLM prediction'] = freq_fitted

freq_test_results['Predicción GBM test'] = pred_y * freq_test_2_w


      ###############

      ###PDP PLOTS###
```

```
      ###############


variables_pdp = ['sexo', 'edad', 'credit_scoring', 'indice_trafico',
'area_residencia','antiguedad_vehiculo', 'tipo_vehiculo', 'valor_vehiculo']


#Credit Scoring

pdp_dist = pdp.pdp_isolate(model=GBM_freq_final, dataset=freq_train_2_X ,
model_features=variables_pdp, num_grid_points = 50, feature='credit_scoring')

fig, axes = pdp.pdp_plot(pdp_dist, 'credit_scoring', center = False, figsize =
(10, 8))

axes['pdp_ax'].set_ylim([0,2])

axes['pdp_ax'].set_xlabel("Credit Scoring", fontsize=15)

axes['pdp_ax'].set_title("Partial dependency Plot para Credit Scoring",
fontsize=25)


#Edad

pdp_dist = pdp.pdp_isolate(model=GBM_freq_final, dataset=freq_train_2_X ,
model_features=variables_pdp, num_grid_points = 80, feature='edad')

fig, axes = pdp.pdp_plot(pdp_dist, 'edad', center = False, figsize = (10, 8))

axes['pdp_ax'].set_ylim([0,2])

axes['pdp_ax'].set_xlabel("Edad", fontsize=15)

axes['pdp_ax'].set_title("Partial dependency Plot para Edad", fontsize=25)


#Índice tráfico

pdp_dist = pdp.pdp_isolate(model=GBM_freq_final, dataset=freq_train_2_X ,
model_features=variables_pdp, num_grid_points = 50, feature='indice_trafico')

fig, axes = pdp.pdp_plot(pdp_dist, 'indice_trafico', center = False, figsize =
(10, 8))

axes['pdp_ax'].set_ylim([0,2])

axes['pdp_ax'].set_xlabel("Índice de tráfico", fontsize=15)

axes['pdp_ax'].set_title("Partial dependency Plot para índice de tráfico",
fontsize=25)


#########################

###GBM SOBRE SEVERIDAD###

#########################


bbdd_sev= bbdd_3


seed=45
```

```
variables = ['sexo', 'edad', 'credit_scoring', 'indice_trafico',
'antiguedad_vehiculo', 'area_residencia','tipo_vehiculo', 'valor_vehiculo',
'coste_siniestro_total', 'num_siniestros']


coste_train, coste_test = train_test_split(bbdd_sev, test_size=0.20,
random_state = seed)


#Entrenamiento

coste_train_2 = coste_train[variables]

coste_train_2_y = coste_train_2['coste_siniestro_total'].values

coste_train_2_X=coste_train_2.drop(['coste_siniestro_total',
'num_siniestros'], axis = 1)


###Test

coste_test_2 = coste_test[variables]


coste_test_2_y = coste_test_2['coste_siniestro_total'].values

coste_test_2_X = coste_test_2.drop(['coste_siniestro_total'], axis = 1)


GBM_sev = lgb.LGBMRegressor(max_depth = 5, learning_rate = 0.02, n_estimators =
100, min_child_samples = 1000, objective = "gamma", importance_type = "gain",
random_state = seed)


GBM_sev.fit(coste_train_2_X, coste_train_2_y)

GBM_sev.feature_importances_


importances = GBM_sev.feature_importances_ / max(GBM_sev.feature_importances_)

feature_imp = pd.DataFrame(sorted(zip(importances, GBM_sev.feature_name_)),

columns=['Valor','Variable'])

plt.figure(figsize=(10, 5))

plt.grid(True)

sns.barplot(x="Valor", y="Variable", data=feature_imp.sort_values(by="Valor",
ascending=False))

plt.title('Variables LightGBM')

plt.xlabel('Importancia relativa de las variables')

plt.tight_layout()


import os

os.environ["PATH"]+=os.pathsep+'C:/Users/Gonzalo/anaconda3/Library/bin/graphvi
z'

lgb.plot_tree(GBM_sev, tree_index = 0, show_info = ["internal_value",
"leaf_count", "split_gain"], figsize=(10, 6), dpi = 150, orientation =
'horizontal')
```

```
lgb.plot_tree(GBM_sev, tree_index = 99, show_info = ["internal_value", "leaf_count", "split_gain"], figsize=(10, 6), dpi = 150, orientation = 'horizontal')


lgb.plot_split_value_histogram(GBM_sev, feature = "credit_scoring", width_coef = 0.5)
plt.title('Número de cortes. Credit Scoring')
plt.xlabel ('Valores de corte')
plt.ylabel ('Número de cortes')
lgb.plot_split_value_histogram(GBM_sev, feature = "antiguedad_vehiculo", width_coef = 0.5)
plt.title('Número de cortes. Antigüedad del vehículo')
plt.xlabel ('Valores de corte')
plt.ylabel ('Número de cortes')


#Predicción LGBM
fitted_y = GBM_sev.predict(coste_train_2_X)
pred_y = GBM_sev.predict(coste_test_2_X)


#Entrenamiento
coste_train_results = coste_train_2['coste_siniestro_total']
coste_train_results['Predicción GBM entrenamiento'] = fitted_y


#Test
coste_test_results = coste_test_2[['coste_siniestro_total']]
coste_test_results ['Predicción GBM test'] = pred_y


print(np.mean(coste_train_results['Predicción GBM entrenamiento']))
print(np.mean(coste_test_results ['Predicción GBM test']))


       ####################
       ##CROSS VALIDATION##
       ####################


#Para saber cual es el mejor modelo debemos crear esta fórmula


def DevianceG(y_i, mu_i):
 D = np.empty(shape = y_i.shape[0])
 for i in range(y_i.shape[0]):
     D[i] = - np.log(y_i[i] / mu_i[i]) + (y_i[i] - mu_i[i]) / mu_i[i]
```

```
 return(2 * sum(D))



#Debemos saber que parámetros ajustan mejor nuetsro modelo.


parametros = { 'ratio_aprendizaje': [0.02, 0.05, 0.1], 'n_arboles':
[400,500,750,1000], 'profundidad': [4,5,6,7]}


def param_GBM_sev(nfolds, X, y, ratio_apren , n_arboles, profundidad ):


 seed = 45

 GBM_sev = lgb.LGBMRegressor(max_depth = profundidad, learning_rate =
ratio_apren, n_estimators = n_arboles, objective = 'gamma', random_state = seed,
importance_type = "gain")


 kf = KFold(n_splits=nfolds, shuffle=False)

 GEgammaCV = []



 for train_index, test_index in kf.split(X, y):


     X_train, X_val = X.iloc[train_index], X.iloc[test_index]

     y_train, y_val = y[train_index], y[test_index]


     GBM_sev.fit(X_train,       y_train,       eval_set=[(X_val,       y_val)],
early_stopping_rounds=20)

     eval_pred = GBM_sev.predict(X_val)

     Dpois = DevianceG(y_val, eval_pred)

     GEgammaCV.append(Dpois / y_val.shape[0])


     GEgammaCVav = np.mean(GEgammaCV)


     return GEgammaCVav


GEs = []

ratio_apren = []

n_arboles = []

profundidad = []


for i in parametros['ratio_aprendizaje']:

    for j in parametros['n_arboles']:
```

```
        for k in parametros['profundidad']:
            GE = param_GBM_sev(nfolds = 5, X = coste_train_2_X, y = coste_train_2_y, ratio_apren = i, n_arboles = j, profundidad = k)
            GEs.append(GE)

            ratio_apren.append(i)
            n_arboles.append(j)
            profundidad.append(k)

Resultados_generales = pd.DataFrame({'Ratio aprendizaje':ratio_apren, 'N° de árboles':n_arboles, 'Profundidad':profundidad, 'Error generalizado':GEs})
Resultados_generales = Resultados_generales.set_index(['Ratio aprendizaje', 'N° de árboles', 'Profundidad'])

Resultados_generales.sort_values(by = "Ratio aprendizaje")
Resultados_generales.sort_values(by = "Error generalizado").head(10)
Mejor = Resultados_generales['Error generalizado'].idxmin(axis = 1)
Peor = Resultados_generales['Error generalizado'].idxmax(axis = 1)

       ####Creamos el mejor modelo

GBM_sev_def = lgb.LGBMRegressor(max_depth = 4, learning_rate = 0.05, n_estimators = 400, objective = "gamma", importance_type = "gain", random_state = seed)

GBM_sev_def.fit(coste_train_2_X, coste_train_2_y)
GBM_sev_def.feature_importances_

importances=GBM_sev_def.feature_importances_/max(GBM_sev_def.feature_importances_)
feature_imp = pd.DataFrame(sorted(zip(importances, GBM_sev_def.feature_name_)),
columns=['Valor','Variable'])
plt.figure(figsize=(10, 5))
plt.grid(True)
sns.barplot(x="Valor", y="Variable", data=feature_imp.sort_values(by="Valor", ascending=False))
plt.title('Variables LightGBM')
plt.xlabel('Importancia relativa de las variables')
plt.tight_layout()

import os
```

```
os.environ["PATH"] += os.pathsep + 'C:/Users/Gonzalo/anaconda3/Library/bin/graphviz'

lgb.plot_tree(GBM_sev_def, tree_index = 0, show_info = ["internal_value", "leaf_count", "split_gain"], figsize=(10, 6), dpi = 800, orientation = 'horizontal')


lgb.plot_tree(GBM_sev_def, tree_index = 99, show_info = ["internal_value", "leaf_count", "split_gain"], figsize=(10, 6), dpi = 150, orientation = 'horizontal')


lgb.plot_split_value_histogram(GBM_sev_def, feature = "credit_scoring", width_coef = 2)

plt.title('Número de cortes. Credit Scoring')

plt.xlabel ('Valores de corte')

plt.ylabel ('Número de cortes')

lgb.plot_split_value_histogram(GBM_sev_def, feature = "valor_vehiculo", width_coef = 2)

plt.title('Número de cortes. Valor del vehículo')

plt.xlabel ('Valores de corte')

plt.ylabel ('Número de cortes')


#Predicción LGBM


bbdd_k = bbdd_2[variables].drop(['coste_siniestro_total', 'num_siniestros'], axis = 1)


fitted_y = GBM_sev_def.predict(bbdd_k)


np.mean(fitted_y)

pred_y = GBM_sev_def.predict(coste_test_2_X)


#Entrenamiento

coste_train_results = bbdd_2[['poliza' , 'coste_siniestro_total']]

coste_train_results['coste_modelado'] = fitted_y


      #########

      ###PDP###

      #########


variables_pdp = ['sexo', 'edad', 'credit_scoring', 'indice_trafico','area_residencia', 'antiguedad_vehiculo', 'tipo_vehiculo', 'valor_vehiculo']

#Credit Scoring
```

```
pdp_dist = pdp.pdp_isolate(model=GBM_sev_def, dataset=coste_train_2_X ,
model_features=variables_pdp, num_grid_points = 50, feature='credit_scoring')

fig, axes = pdp.pdp_plot(pdp_dist, 'credit_scoring', center = False, figsize =
(10, 8))

axes['pdp_ax'].set_ylim([0, 8000])

axes['pdp_ax'].set_xlabel("Credit Scoring", fontsize=15)

axes['pdp_ax'].set_title("Partial dependency Plot para Credit Scoring",
fontsize=25)


#Antigüedad vehiculo

pdp_dist = pdp.pdp_isolate(model=GBM_sev_def, dataset=coste_train_2_X ,
model_features=variables_pdp, num_grid_points = 4,
feature='antiguedad_vehiculo')

fig, axes = pdp.pdp_plot(pdp_dist, 'antiguedad_vehiculo', center = False,
figsize = (10, 8))

axes['pdp_ax'].set_ylim([0, 7000])

axes['pdp_ax'].set_xlabel("Antigüedad del vehículo", fontsize=15)

axes['pdp_ax'].set_title("Partial dependency Plot para Antigüedad del vehículo",
fontsize=25)


#Edad

pdp_dist = pdp.pdp_isolate(model=GBM_sev_def, dataset=coste_train_2_X ,
model_features=variables_pdp, num_grid_points = 80, feature='valor_vehiculo')

fig, axes = pdp.pdp_plot(pdp_dist, 'valor_vehiculo', center = False, figsize =
(10, 8))

axes['pdp_ax'].set_ylim([0, 10000])

axes['pdp_ax'].set_xlabel("Valor del vehículo", fontsize=15)

axes['pdp_ax'].set_title("Partial dependency Plot para Valor del vehículo",
fontsize=25)


######################

##Modelo Burning Cost##

######################


#mi método. Revisar los nombres!


#Frecuencia

testF = freq_test[GBM_freq_final.feature_name_]

trainF = freq_train[GBM_freq_final.feature_name_]

frec_GBM_train = freq_train[['poliza' ,'num_siniestros', 'exposure']]

frec_GBM_train ['freq_modelada'] = GBM_freq.predict(trainF) * freq_train_2_w

frec_GBM_test = freq_test[['poliza' ,'num_siniestros', 'exposure']]

frec_GBM_test ['freq_modelada'] = GBM_freq.predict(testF) * freq_test_2_w
```

```
frec_GBM_total = frec_GBM_train.append(frec_GBM_test).sort_values('poliza')


#Severidad
sev_GBM_total = coste_train_results


Resultado_final = frec_GBM_total
Resultado_final=    pd.merge(    Resultado_final,    sev_GBM_total[['poliza',
'coste_siniestro_total', 'coste_modelado']], on='poliza', how='left')


Resultado_final['Burning    Cost']    =    Resultado_final['freq_modelada']    *
Resultado_final['coste_modelado']


Resultado_final_GBM=    Resultado_final.drop(['num_siniestros',    'exposure',
'coste_siniestro_total' ], axis=1)


np.mean(Resultado_final_GBM)
Resultado_final_GBM.iloc[27:32]




#################
###COMPARATIVA###
#################



##Comparativa del Burning Cost
import statsmodels.api as sm
import pylab as py
quants = np.array(range(1,100,1))/100
GLMquants  =  [np.quantile(Resultado_final_GLM['Burning_cost'],  i)  for  i  in
quants]
GBMquants  =  [np.quantile(Resultado_final_GBM['Burning  Cost'],  i)  for  i  in
quants]


higher = [1 if gbm > glm else 0 for glm, gbm in zip(GLMquants, GBMquants)]
from matplotlib.pyplot import figure
from matplotlib.colors import ListedColormap
colormap = ListedColormap(['darkgreen', 'cyan'])
classes = ['GLM is higher', 'GBM is higher']
fig, ax = plt.subplots(figsize=(7, 7), dpi=100)
```

```
scat = ax.scatter(x = GBMquants, y = GLMquants, s = 20, c = higher, cmap = colormap, marker = "+")

ax.yaxis.set_major_formatter('{x:1.0f}€')

ax.xaxis.set_major_formatter('{x:1.0f}€')

ax.grid(True)

ax.set_xlim([0, 250])

ax.set_ylim([0, 250])

ax.plot([0,500], [0, 500], '--', lw=1, color = 'black')

ax.set_xlabel("Cuantiles prima pura GBM", fontsize=12)

ax.set_ylabel("Cuantiles prima pura GLM", fontsize=12)

ax.annotate('Punto de corte {}%'.format(len(GLMquants)-sum(higher)), xy=(55.94,58.94), xytext=(55.94, 125), arrowprops=dict(facecolor='red', shrink=0.05, width = 6 ))

ax.legend(handles=scat.legend_elements()[0], labels=classes)

GBMquants[len(GLMquants)-sum(higher)]


####CREACIón DE CLUSTERS


#############CASO GLM


###Frecuencia


#Eliminamos el primer y ultimo 5 por ciento

menor= np.percentile(Resultado_final_GLM['freq_modelada'],5)

mayor = np.percentile(Resultado_final_GLM['freq_modelada'],95)


bbdd_4 = Resultado_final_GLM

x1 = bbdd_4[bbdd_4['freq_modelada'] <= menor].index

x2 = bbdd_4[bbdd_4['freq_modelada'] >= mayor].index

bbdd_4 = bbdd_4.drop(x1)

bbdd_4 = bbdd_4.drop(x2)


Niveles_Cluster_GLM_frec = pd.qcut(bbdd_4['freq_modelada'], 9, retbins=True)[1]

bbdd_4['cluster'] = pd.qcut(bbdd_4['freq_modelada'], 9, labels=( 'Cluster 1','Cluster 2', 'Cluster 3', 'Cluster 4','Cluster 5', 'Cluster 6', 'Cluster 7', 'Cluster 8', 'Cluster 9'))


Niveles_Cluster_GLM_frec_100 = Niveles_Cluster_GLM_frec/Niveles_Cluster_GLM_frec[1]

Niveles_Cluster_GLM_frec_100 =np.delete(Niveles_Cluster_GLM_frec_100, 0)
```

```
plt.figure(figsize=(10, 6))
plt.plot(Niveles_Cluster_GLM_frec_100, color = 'green')

plt.title("Diferencias Clústeres respecto nivel base. Frecuencia. GLM", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})
plt.xlabel("Clústeres")
plt.ylabel("Variación")

###Severidad

#Eliminamos el primer y ultimo 5 por ciento
menor= np.percentile(Resultado_final_2_GLM['coste_modelado'],5)
mayor = np.percentile(Resultado_final_2_GLM['coste_modelado'],95)

bbdd_4 = Resultado_final_2_GLM
x1 = bbdd_4[bbdd_4['coste_modelado'] <= menor].index
x2 = bbdd_4[bbdd_4['coste_modelado'] >= mayor].index
bbdd_4 = bbdd_4.drop(x1)
bbdd_4 = bbdd_4.drop(x2)

Niveles_Cluster_GLM_sev = pd.qcut(bbdd_4['coste_modelado'], 9, retbins=True)[1]
bbdd_4['cluster'] = pd.qcut(bbdd_4['coste_modelado'], 9 labels=( 'Cluster 1','Cluster 2', 'Cluster 3', 'Cluster 4','Cluster 5', 'Cluster 6', 'Cluster 7', 'Cluster 8', 'Cluster 9'))
Niveles_Cluster_GLM_sev_100 = Niveles_Cluster_GLM_sev/Niveles_Cluster_GLM_sev[1]
Niveles_Cluster_GLM_sev_100 =np.delete(Niveles_Cluster_GLM_sev_100, 0)
plt.figure(figsize=(10, 6))
plt.plot(Niveles_Cluster_GLM_sev_100, color = 'green')

plt.title("Diferencias Clústeres respecto nivel base. Severidad. GLM", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})
plt.xlabel("Clústeres")
plt.ylabel("Variación")

###Burning Cost

#Eliminamos el primer y ultimo 5 por ciento
menor= np.percentile(Resultado_final_GLM['Burning_cost'],5)
```

```
mayor = np.percentile(Resultado_final_GLM['Burning_cost'],95)


bbdd_4 = Resultado_final_GLM
x1 = bbdd_4[bbdd_4['Burning_cost'] <= menor].index
x2 = bbdd_4[bbdd_4['Burning_cost'] >= mayor].index
bbdd_4 = bbdd_4.drop(x1)
bbdd_4 = bbdd_4.drop(x2)


Niveles_Cluster_GLM_bc = pd.qcut(bbdd_4['Burning_cost'], 9, retbins=True)[1]
bbdd_4['cluster'] = pd.qcut(bbdd_4['Burning_cost'], 9 labels=( 'Cluster 1','Cluster 2', 'Cluster 3', 'Cluster 4','Cluster 5', 'Cluster 6', 'Cluster 7', 'Cluster 8', 'Cluster 9'))
Niveles_Cluster_GLM_bc_100 = Niveles_Cluster_GLM_bc/Niveles_Cluster_GLM_bc[1]
Niveles_Cluster_GLM_bc_100 =np.delete(Niveles_Cluster_GLM_bc_100, 0)


plt.figure(figsize=(10, 6))
plt.plot(Niveles_Cluster_GLM_bc_100, color = 'green')


plt.title("Diferencias Clústeres respecto nivel base. Prima Pura. GLM", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})
plt.xlabel("Clústeres")
plt.ylabel("Variación")



####CASO GBM


###Frecuencia


#Eliminamos el primer y ultimo 5 por ciento
menor_GBM= np.percentile(Resultado_final_GBM['freq_modelada'],5)
mayor_GBM = np.percentile(Resultado_final_GBM['freq_modelada'],95)


bbdd_4_GBM = Resultado_final_GBM
x1 = bbdd_4_GBM[bbdd_4_GBM['freq_modelada'] <= menor_GBM].index
x2 = bbdd_4_GBM[bbdd_4_GBM['freq_modelada'] >= mayor_GBM].index
bbdd_4_GBM = bbdd_4_GBM.drop(x1)
bbdd_4_GBM = bbdd_4_GBM.drop(x2)


Niveles_Cluster_GBM_frec = pd.qcut(bbdd_4_GBM['freq_modelada'], 9, retbins=True)[1]
```

```
bbdd_4_GBM['cluster'] = pd.qcut(bbdd_4_GBM['freq_modelada'], 9, labels=(
'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5', 'Cluster 6',
'Cluster 7', 'Cluster 8', 'Cluster 9'))

Niveles_Cluster_frec_GBM_100 =
Niveles_Cluster_GBM_frec/Niveles_Cluster_GBM_frec[1]

Niveles_Cluster_frec_GBM_100 =np.delete(Niveles_Cluster_frec_GBM_100, 0)


plt.figure(figsize=(10, 6))

plt.plot(Niveles_Cluster_frec_GBM_100, color = 'blue')

plt.ylim(0,33)

plt.title("Diferencias Clústeres respecto nivel base. Frecuencia. GBM",
fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})

plt.xlabel("Clústeres")

plt.ylabel("Variación")


###Severidad


#Eliminamos el primer y ultimo 5 por ciento


menor_GBM= np.percentile(Resultado_final['coste_modelado'],5)

mayor_GBM = np.percentile(Resultado_final['coste_modelado'],95)


bbdd_4_GBM = Resultado_final

x1 = bbdd_4_GBM[bbdd_4_GBM['coste_modelado'] <= menor_GBM].index

x2 = bbdd_4_GBM[bbdd_4_GBM['coste_modelado'] >= mayor_GBM].index

bbdd_4_GBM = bbdd_4_GBM.drop(x1)

bbdd_4_GBM = bbdd_4_GBM.drop(x2)


Niveles_Cluster_sev_GBM = pd.qcut(bbdd_4_GBM['coste_modelado'], 9,
retbins=True)[1]

bbdd_4_GBM['cluster'] = pd.qcut(bbdd_4_GBM['coste_modelado'], 9, labels=(
'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5', 'Cluster 6',
'Cluster 7', 'Cluster 8', 'Cluster 9'))

Niveles_Cluster_sev_GBM_100 =
Niveles_Cluster_sev_GBM/Niveles_Cluster_sev_GBM[1]

Niveles_Cluster_sev_GBM_100 =np.delete(Niveles_Cluster_sev_GBM_100, 0)


plt.figure(figsize=(10, 6))

plt.plot(Niveles_Cluster_sev_GBM_100, color = 'blue')


plt.title("Diferencias Clústeres respecto nivel base. Severidad. GBM",
fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})
```

```
plt.xlabel("Clústeres")
plt.ylabel("Variación")

###Burning COST

#Eliminamos el primer y ultimo 5 por ciento
menor_GBM= np.percentile(Resultado_final_GBM['Burning Cost'],5)
mayor_GBM = np.percentile(Resultado_final_GBM['Burning Cost'],95)

bbdd_4_GBM = Resultado_final_GBM
x1 = bbdd_4_GBM[bbdd_4_GBM['Burning Cost'] <= menor_GBM].index
x2 = bbdd_4_GBM[bbdd_4_GBM['Burning Cost'] >= mayor_GBM].index
bbdd_4_GBM = bbdd_4_GBM.drop(x1)
bbdd_4_GBM = bbdd_4_GBM.drop(x2)

Niveles_Cluster_bc_GBM = pd.qcut(bbdd_4_GBM['Burning Cost'], 9, retbins=True)[1]
bbdd_4_GBM['cluster'] = pd.qcut(bbdd_4_GBM['Burning Cost'], 9, labels=( 'Cluster 1', 'Cluster 2', 'Cluster 3', 'Cluster 4', 'Cluster 5', 'Cluster 6', 'Cluster 7', 'Cluster 8', 'Cluster 9'))

Niveles_Cluster_bc_GBM_100 = Niveles_Cluster_bc_GBM/Niveles_Cluster_bc_GBM[1]
Niveles_Cluster_bc_GBM_100 =np.delete(Niveles_Cluster_bc_GBM_100, 0)

plt.figure(figsize=(10, 6))
plt.plot(Niveles_Cluster_bc_GBM_100, color = 'blue')

plt.title("Diferencias Clústeres respecto nivel base. Prima Pura. GBM", fontdict={'family': 'Calibri', 'color' : 'black', 'weight': 'bold', 'size': 18})
plt.xlabel("Clústeres")
plt.ylabel("Variación")


##Algunas de las tablas mostradas se han llevado a cabo en el entorno EXCEL.
```