uc3m | Universidad Carlos III de Madrid

Máster Universitario en Ciencias Actuariales y Financieras
2020-2021

Trabajo Fin de Máster

"Predicción de *Cross-selling* con técnicas de *Machine Learning*"

Joanna Lempicka

Tutores

José Miguel Rodríguez-Pardo del Castillo

Jesús Ramón Simón del Potro

Madrid, 2021

Firmado:

RESUMEN

El análisis de la tasa de *cross-selling* es un tema de gran interés para todas las compañías y, más concretamente, para las aseguradoras. Conseguir una venta cruzada requiere de una dificultad mayor y el tipo de cliente que se genera posee mayor lealdad y fidelidad hacia la compañía. La utilización de algoritmos de *Machine Learning* permite predecir la tasa de conversión de un cliente potencial a un cliente final, además de detectar qué tipo de factores son más determinantes para cerrar un cliente potencial. De esta forma, cuando se realiza una petición a un proveedor de bases de datos, en principio, se debería priorizar variables que aporten mayor importancia. Así, este Trabajo Fin de Máster se centra en el análisis de la tasa de *cross-selling* mediante la aplicación de algoritmos de *Machine Learning.*

SUMMARY

Analysis of cross-selling rate is a great topic of interest for all companies, specifically for insurances. Getting a cross-sale is very difficult but the type of customer generated from it obtains greater loyalty to the company. Using Machine Learning algorithms allows predicting the conversion rate from a potential customer to the final client. Allowing companies to detect which kind of features are most determining to conclude a sale. Therefore, when a insurance company does a request to a database provider, factors that provide greater importance should be prioritized. Master's final project focus on analysis of cross-selling rate through application of Machine Learning techniques.

ÍNDICE

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

CAPÍTULO 1. INTRODUCCIÓN

Este primer capítulo introduce el trabajo final de máster realizado por Joanna Lempicka titulado *"Predicción de Cross-selling con técnicas de Machine Learning"*.

1.1. Motivación

El Aprendizaje Automático es una de las disciplinas científicas de la Inteligencia Artificial con mayor crecimiento en los últimos años. La utilización de este tipo de algoritmos sobre problemas reales, como la predicción de la tasa de *cross-selling* en el sector seguros, hace que sea un trabajo ambicioso y con alto potencial de aplicación en el mundo real. La tasa de *cross-selling* es un aspecto fundamental dentro de la gestión de los canales de venta para cualquier compañía. Beneficios como el incremento de la rentabilidad, mejora de la fidelización, reducción del porcentaje de *churn*[1] y el aumento de la vida media de los clientes están estrechamente relacionados con la tasa de *cross-selling.*

Mi interés por el análisis de datos, por la tecnología del Big Data y, más concretamente, por el campo de *Machine Learning*, sumado a la potencial aplicación al mundo real y a la importancia de la gestión de los clientes potenciales son las razones fundamentales que me han llevado a escoger este tema.

1.2. Objetivo de estudio

En los últimos años gracias al desarrollo tecnológico, el término *Machine Learning* ha ido ganando terreno y el interés por esta disciplina científica ha incrementado de forma notable. Pero ¿qué es exactamente *Machine Learning*? El Aprendizaje Automático no se basa en el aprendizaje de memoria, sino en el reconocimiento de patrones complejos y en la toma de decisiones inteligentes basadas en datos. Por tanto, la dificultad radica tanto en la complejidad de la toma de decisiones como en la posible dimensionalidad de los datos.

[1] El porcentaje de *churn* es un término utilizado en el ámbito del Marketing que hace referencia al porcentaje de clientes que se dan de baja de un servicio.

Para abordar este problema ML desarrolla algoritmos que descubren el conocimiento a partir de datos y experiencias específicas, basados en principios estadísticos y computacionales sólidos. Integra múltiples fundamentos matemáticos como la lógica, teoría de la probabilidad, estadística, optimización, aprendizaje por refuerzo…etc. Estos algoritmos guardan una gran relación con la minería de datos, el mundo actuarial, los videojuegos, la predicción mediante algoritmos de inteligencia artificial...etc.

Así, el objetivo de este estudio es predecir la tasa de *cross-selling* mediante la utilización de técnicas avanzadas de *Machine Learning*. Es decir, partiendo de una base de clientes potenciales se trata de predecir el porcentaje de personas que contratarían un seguro de automóvil para la misma compañía aseguradora. Para materializar este objetivo, se aplicarán diferentes métodos de *Machine Learning* y se explicará cuáles son los métodos más acertados para la predicción llevada a cabo.

CAPÍTULO 2. LA VENTA CRUZADA EN EL SECTOR SEGUROS

El *cross-selling* o venta cruzada consiste en vender un producto o servicio que, en principio, guarda relación con el producto principal y está dirigido a un cliente en cartera o potencial. Es una de las estrategias de ventas más utilizadas. Cabe destacar que no solo se trata de vender un producto asociado, sino más bien de conocer realmente cuáles son las necesidades del cliente para, de esta forma, mejorar su vida media e incrementar su fidelización. Para llevar a cabo una correcta implementación de la venta cruzada es recomendable apoyarse de un CRM[2].

2.1. Importancia de un modelo de *Cross-selling*

Para las compañías aseguradoras es fundamental contar con un plan de generación y gestión de las ventas cruzadas. Entre las principales ventajas de una buena planificación y gestión de las ventas cruzadas para la compañía se encuentran:

- Incremento de la lealtad y fidelización del cliente: cuando un cliente posee varios productos o contrata varios servicios de la misma compañía se genera una mayor confianza y fidelidad a la marca. Es lo que se conoce, dentro del ámbito del Marketing, como cliente fiel o cliente integral.
- Ahorro de costes: el coste de vender un producto o servicio a un cliente que ya tenemos en la cartera es inferior al de un cliente nuevo. Esto se debe, fundamentalmente, al ahorro de costes de administración y gestión que vienen derivados de la generación de una nueva póliza. Por tanto, este ahorro de costes implica un incremento de la rentabilidad.
- La generación de valor al cliente: es esencial crear una propuesta de valor para lograr la satisfacción del cliente. El hecho de poder ofrecer una oferta integral al cliente hace que éste sienta la satisfacción y tranquilidad de tener todo bajo una misma compañía de seguros. Este hecho es un factor importante.

[2] CRM: "Customer Relationship Management". Hace referencia al total de acciones, estrategias y tecnologías orientadas a la relación con el consumidor.

- Crecimiento en mercados maduros: es complicado para una compañía crecer en un mercado que ya es maduro, por tanto, una de las formas más viables es hacerlo a través de los clientes en cartera.

2.2. Dificultades y potenciales inconvenientes de una mala gestión del *cross-selling*

En general, se puede creer que el principal obstáculo de la venta cruzada es la propia resistencia del asegurado. No obstante, son numerosas las situaciones donde el origen real del obstáculo viene por parte de la compañía o, concretamente, del agente de ventas. Es decir, la compañía aseguradora no debería esperar a que sea el cliente que el que solicite otro seguro, sino más bien es el propio agente de venta quien debe poseer la iniciativa. Los puntos débiles que se pueden encontrar cuando la estrategia de *cross-selling* no funciona adecuadamente son los siguientes:

- Ausencia de un plan estratégico guiado para llevar a cabo acciones de *cross-selling*.
- El agente debe tener un determinado perfil comercial, un gran conocimiento de ambos productos, orientación comercial, poseer destreza, entender la relación de forma clara y saber transmitir al cliente los beneficios potenciales derivados de la contratación del producto relacionado.
- Asociación que debe existir entre ambos productos donde, en caso contrario, estaríamos ante una disociación del producto principal que se vería afectado por la venta de otro producto que no tiene relación con el producto principal. Por tanto, se perdería el valor del producto porque podría confundir al cliente y éste sentiría rechazo, desorientación y posible desconfianza hacia la compañía aseguradora.

A pesar de estas dificultades, si la compañía posee un buen plan de *cross-selling* y consigue predecir de forma adecuada el problema que se aborda, podrá obtener no solo el porcentaje de clientes a los que realizar una venta cruzada para un determinado seguro, si no el tipo de cliente más propenso a aceptar este nuevo seguro.

Así, la compañía sabrá, con cierta confianza, donde deberá centrar sus esfuerzos, es decir, para qué segmentos de clientes deberá prestar mayor atención e invertir una mayor cuantía. Se podrá realizar un *scoring* con los clientes en cartera en función de la propensión a contratar un nuevo seguro. Realmente, se trata de una forma de optimizar el

gasto, los esfuerzos irán destinados en mayor proporción a aquellos clientes en cartera que tengan un mayor *scoring* para contratar un nuevo seguro. Esta misma idea podría aplicarse para nueva cartera también, es decir, una vez recibido un listado de personas que podrían estar interesadas en un nuevo seguro o en un cambio de compañía, podría realizarse un *scoring* en función de ciertas variables explicativas.

Por tanto, es una cuestión que afecta a varios departamentos de forma transversal. Por una parte, el departamento de Marketing mediante la gestión de las estrategias de *cross-selling*, así como al departamento Actuarial que será quien lleve a cabo toda la modelización y predicción concretamente en la línea de negocio de *pricing* o tarificación.

CAPITULO 3. ALGORITMOS DE *MACHINE LEARNING*

3.1. ¿Qué es *Machine Learning*?

Machine Learning es un campo que forma parte de la Inteligencia Artificial donde el objetivo fundamental es la predicción de un *target* o resultado a partir de una serie de datos o *imputs*. Así, tanto la cantidad como la calidad de los datos juegan un papel fundamental. Para poder predecir se necesitan tres elementos: los datos, los factores o variables y los algoritmos. En ML se pueden distinguir tres grandes campos: aprendizaje supervisado, no supervisado y reforzado.

En primer lugar, en el aprendizaje supervisado se conoce el resultado que se desea obtener antes de entrenar o poner en práctica el modelo. Así, la base de datos es dividida en *training* y *test*. En primer lugar, se entrena el modelo con los datos de *training*. Después, una vez entrenado el modelo, se prueba el modelo sobre el data *test*, éstos son datos que no han sido utilizados en el training y donde, a priori, se conoce el resultado. Finalmente, se compara la predicción realizada por el modelo con el valor real.

En segundo lugar, el aprendizaje no supervisado presenta una filosofía diferente porque el resultado no suele conocerse antes de entrenar el modelo, se trata de encontrar nuevos patrones o relaciones. La idea fundamental detrás de estos métodos es formar agrupaciones que sean similares. Concretamente, cuando hay pólizas que forman parte del mismo subgrupo se pueden tratar de forma similar, por ejemplo, aplicar el mismo precio o bien, un aumento o reducción de la prima para todas las pólizas que formen parte de ese subgrupo.

Por último, el aprendizaje reforzado es el tipo de aprendizaje más moderno e implica aprender qué hacer, cómo asignar situaciones a acciones con el objetivo de maximizar una señal de recompensa numérica. Son, de forma esencial, problemas de circuito cerrado porque en el sistema de aprendizaje las acciones influyen en sus entradas posteriores. Además, no se informa al *learner*[3] qué acciones tomar, como en otras formas de aprendizaje automático, sino que debe descubrir qué acciones producen la mayor recompensa al probarlas. Las acciones no solo afectan a la recompensa inmediata sino

[3] *Learner* hace referencia al aprendiz del modelo.

también podrían afectar a la siguiente situación y, a través de ella, todas las recompensas posteriores.

3.2. *Machine Learning* Supervisado

El desarrollo del Trabajo Final de Máster se centrará en *Machine Learning* supervisado, todos los algoritmos que se aplicarán forman parte de este subcampo específico. Uno de los libros de referencia de ML es el escrito por Alpaydin[4], la última versión es del año 2020. En esta obra unifica diferentes problemas y soluciones que pueden plantearse en este contexto. Kotsiantis[5], por su parte, realiza una revisión sobre los modelos de clasificación del tipo supervisados. Así, dentro de los métodos de generación del algoritmo de ML supervisado se encuentran:

- Método de regresión

  Es usado con el objetivo de predecir un *target*[6] continúo. Un claro ejemplo sería un modelo para predecir el volumen de precipitaciones en función de los datos disponibles de otros años.
- Método de clasificación

  Se utiliza cuando el *target* es una variable cualitativa o categórica. Un ejemplo podría ser para identificar un tumor cancerígeno (Respuesta: Sí / No) en función del tamaño, edad del paciente, hábitos... etc.
- Método de agrupación

  Es usado para clasificar cuando, *a priori*, no son conocidas las categorías de clasificación. De esta forma, se consigue obtener clústeres, es decir, grupos con similitudes por aproximación que podrían formar parte de una misma categoría.

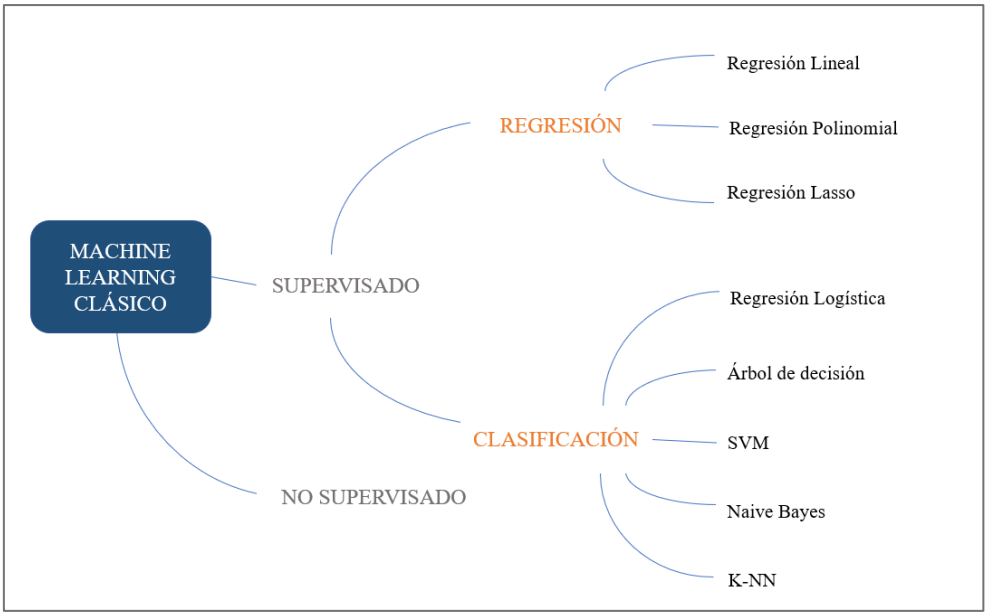
Es necesario tener en cuenta que en el aprendizaje supervisado no existe un orden explicito para ordenar las clases, de hecho, en ocasiones se utilizan etiquetas en vez de números para denotar las clases. Para ambos tipos de output (discreto o continuo) se pueden utilizar *imputs* discretos o continuos en función del método de algoritmo supervisado que vayamos a aplicar. Así, existe una convención general para etiquetas las

[4] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
[5] Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 26(3), 159-190.
[6] Se empleará este término para hacer referencia a la variable respuesta o variable dependiente del modelo, es decir, la variable que se tratará de modelizar.

predicciones: se denominan regresiones cuando el *output* es una variable continua y se denominan clasificaciones cuando estamos prediciendo un *output* de tipo categórico.

Para obtener una visión general de los algoritmos de *Machine Learning* Supervisado, se hará una clasificación de ML clásico y ML moderno donde se enumerarán los algoritmos más conocidos. A continuación, en la Ilustración 1 se muestran los algoritmos más conocidos de ML supervisado clásico.

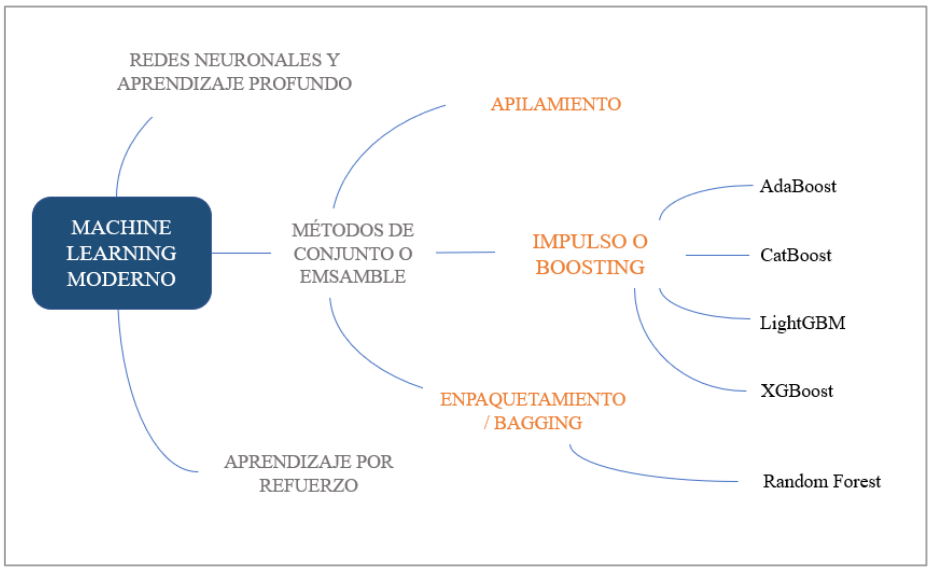
Ilustración 1 Esquema algoritmos de *Machine Learning* Clásico



Fuente: Elaboración propia

En la Ilustración 2 se detalla una clasificación de algoritmos ML moderno, concretamente se centra en los métodos de conjunto o *Emsable Tecniques*, utilizados para el caso práctico de estudio.

Ilustración 2 Esquema algoritmos de *Machine Learning* Moderno



Fuente: Elaboración propia

3.3. Explicación de los métodos de *Machine Learning* a utilizar

Existen multitud de modelos de ML, no obstante, se ha considerado escoger únicamente métodos de ML supervisado con el objetivo de observar el comportamiento de estos métodos para el caso de estudio y ver cuál que resultados dan los modelos.

A continuación, se enumeran los diferentes métodos a utilizar en el estudio práctico de la predicción de *cross-selling* en seguros de vida y se procede a explicar, de forma detallada, cada uno de ellos. La explicación de los métodos de ML supervisado se realizará de forma secuencial, primero los métodos de clasificación, después *bagging* y, por último, *boosting*.

1. GLM: método de clasificación que pertenece a ML clásico supervisado.
2. Árbol de decisión: método de clasificación que pertenece a ML clásico supervisado.
3. *Stochastic Gradient Descent*: método de clasificación que pertenece a ML clásico supervisado.
4. *Random Forest:* método de conjunto que pertenece a ML moderno supervisado. Método de *Bagging*.
5. *CatBoost*: método de conjunto que pertenece a ML moderno supervisado. Método de *Boosting*.

6. *XGBoost*: método de conjunto que pertenece a ML moderno supervisado. Método de *Boosting*.
7. *LGBM*: método de conjunto que pertenece a ML moderno supervisado. Método de *Boosting*.

Métodos de clasificación

A continuación, se detalla cada uno de los métodos de clasificación que se van a aplicar en la parte práctica para la predicción.

- ***Modelos GLM***

Se basa en la modelización de la variable objetivo generando una serie de coeficientes que explican las variables explicativas de una forma similar a la regresión lineal. No obstante, la principal diferencia es que mientras la regresión lineal supone que la variable dependiente sigue una Normal, en este caso la variable dependiente se realiza una transformación mediante una función no lineal.

Es el método clásico de referencia en las compañías aseguradoras y las principales ventajas que se obtienen de este método son: la gran facilidad para interpretar los coeficientes porque se puede observar cuales son los factores más significativos o con mayor poder explicativo sobre el *target*. Así mismo, se trata de un modelo clásico con fundamentos matemáticos que pueden ser explicados ante el regulador, esto es un aspecto de gran importancia.

Para especificar un modelo GLM es necesario tener en cuenta tres hipótesis o puntos de partida:

- Componente aleatorio: hace referencia a la distribución que sigue el target
- Componente sistemático: es el vector de los factores que explican el target.
- Función *link*: relaciona la media de la variable dependiente con el componente sistemático, es decir, con los factores que explican el target. Es importante señalar que la función link debe ser monótona y diferenciable.

En la Ilustración 3 se puede observar las combinaciones más utilizadas de variable dependiente y variables independientes con una función vínculo y una distribución del error particular para cada tipo de análisis.

Ilustración 3. GLM - Combinaciones variable respuesta y variables explicativas

Tipo de análisis	Variable respuesta	Variable explicativa	Función de vínculo	Distribución de errores
Regresión	Continua	Continua	Identidad	Normal
ANOVA	Continua	Factor	Identidad	Normal
Regresión	Continua	Continua	Recíproca	Gamma
Regresión	Conteo	Continua	Logarítmica	Poisson
Tabla de contingencia	Conteo	Factor	Logarítmica	Poisson
Proporciones	Proporción	Continua	Logit	Binomial
Regresión logística	Binaria	Continua	Logarítmica	Binomial
Análisis de supervivencia	Tiempo	Continua	Recíproca	Exponencial

Fuente: Cayuela, L. (2009). Modelos lineales generalizados (GLM)

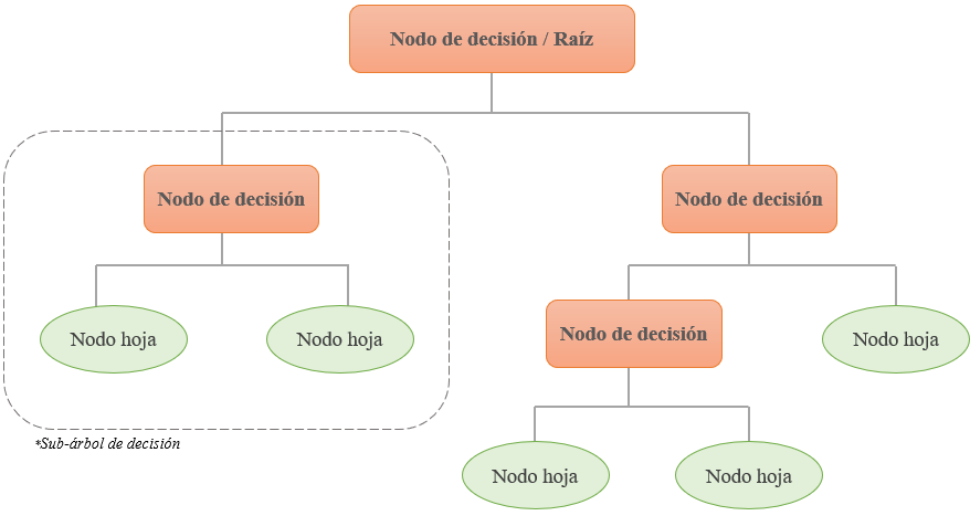
Uno de los criterios más utilizados para comparar entre modelos es el índice AIC[7]. Este índice evalúa como ajusta el modelo al *dataset* además de la propia complejidad del modelo. Para comparar entre modelos se escogerá aquel modelo que tenga un menor AIC, siempre y cuando se traten de modelos válidos, es decir, cuanto el *test* de significación para cada uno de los factores del modelo indique que se trata de una variable significativa.

- ***Decision Tree***

Un árbol de decisión es una estructura similar a un diagrama de flujo, donde los nodos internos representan las características, las ramas del árbol son las reglas de decisión y los nodos hojas, los resultados. El nodo más alto del árbol se denomina nodo raíz, este nodo aprende a dividir de forma recursiva en función del mejor atributo o característica utilizando una medida de selección de atributos que puede ser la ganancia de información o el Índice de Gini.

[7] Akaike Information Criterion.

Ilustración 4 *Representación de un árbol de decisión*



Fuente: Elaboración propia

La **ganancia de información** mide los cambios que se producen en la entropía (aleatoriedad de los datos) después de realizar la división de los datos a partir de una característica, es decir, mide cuánta información nos está dando ese atributico sobre la clase. Así, aquel atributo que maximice la ganancia de información será el elegido para realizar la partición, así continua hasta el final, que será cuando no exista mayor ganancia de información, los nodos sean puros o bien el propio usuario establezca un límite de generación de ramas. En índice de Gini mide la impureza que se usa al formar un árbol de decisión, por tanto, serán mejores aquellos atributos que proporcionen con la clasificación un índice de Gini más bajo.

A continuación, se detalla la fórmula de cálculo del índice de Gini donde pi es la probabilidad de que una muestra sea clasificada en una clase determinada.

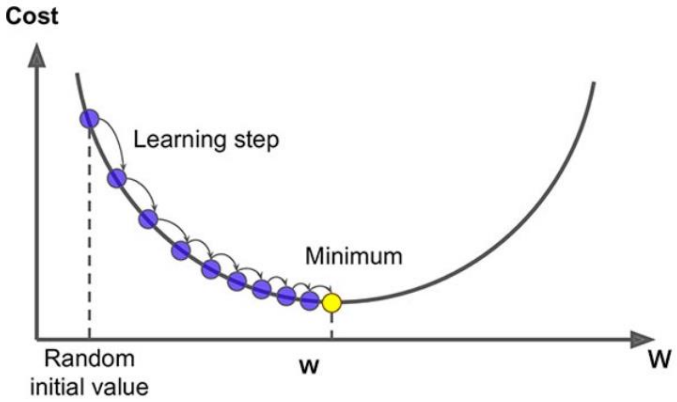
$$Índice\ de\ Gini = 1 - \sum_{k=1}^{n} (p_k)^2 \quad (1)$$

De esta forma, cuando el árbol debe elegir entre dos atributos preferirá elegir aquel con el menor índice de Gini como nodo raíz.

- ***SGD***

Stochactic Grandient Descent, algoritmo base de las redes neuronales, tiene como objetivo encontrar un valor de x de tal forma que minimice el valor de y, es decir, el valor de la pérdida.

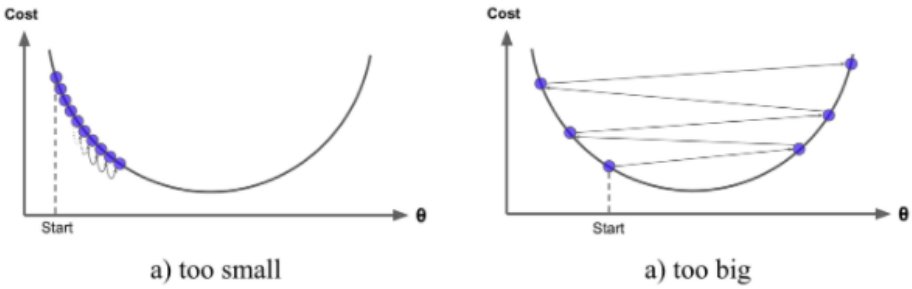
Ilustración 5 Función de pérdida y Learning rate



Fuente: Gradient descent (Geron, 2017)

A modo de ejemplo, si la siguiente función es la función de perdida, el gradiente es la derivada de la función de pérdida con respecto a los parámetros del modelo. La tasa de aprendizaje, *learning rate*, es el tamaño del paso que se da cada vez que se avanza hasta llegar al mínimo global. Es fundamental elegir bien este valor porque si se escoge un *learning rate* muy alto podría ocurrir que se salte el mínimo global y se obtuviera únicamente un mínimo local. No obstante, si el *learning rate* es muy pequeño podría ocurrir que nunca se llegue a ese mínimo global.

Ilustración 6 Comparación de Learnings rates



Fuente: Learning rate comparisons (Geron, 2017)

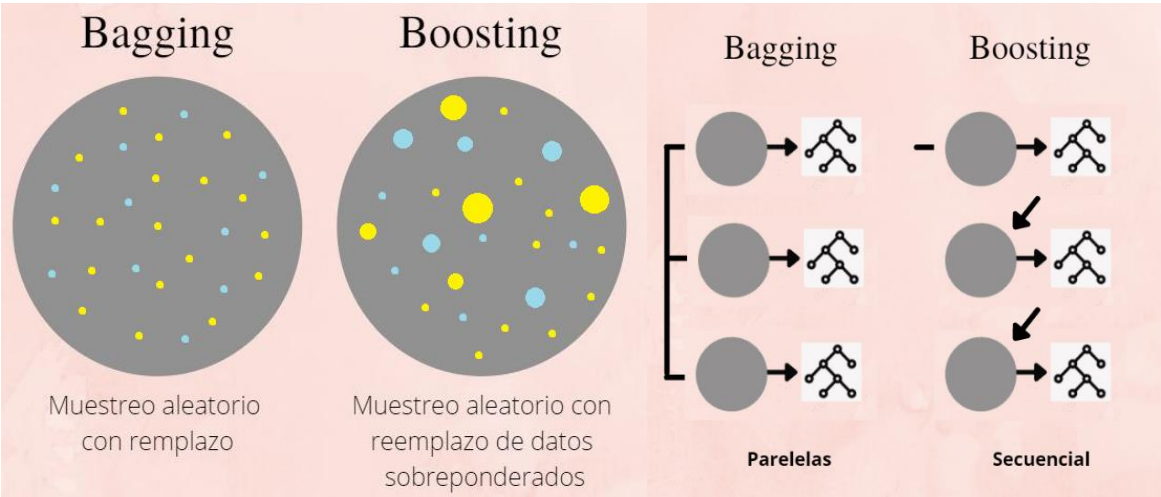
SGD funciona bien para problemas donde los problemas de aprendizaje son convexos, no obstante, muchos problemas de ML no son convexos, así las redes neuronales son no convexas, existen muchos valores mínimos. El gradiente estocástico hace referencia a que se calcula en un solo ejemplo y no en todo el conjunto de datos, puesto que esto requeriría un tiempo computacional muy grande. Así, aunque se requieran

una gran cantidad de pasos generales, el tiempo computacional se reduce y los resultados son buenos. El SGD puede resolver este problema de no convexidad porque muestrea de forma aleatoria una fracción del total de observaciones de entrenamiento y realiza el siguiente árbol utilizando esta submuestra. Así, aunque no se asegure que vaya a encontrar el mínimo global, podría ayudar a acercarse a él.

- **Métodos de Conjunto**

Los Métodos de Conjunto se basan en entrenar múltiples modelos utilizando el mismo algoritmo de aprendizaje. Si las principales causas de error en el aprendizaje se deben al ruido, el sesgo y la varianza, estas técnicas de conjunto ayudan a minimizar estos factores, están diseñadas para mejorar la estabilidad y precisión de los algoritmos de ML. Así, una combinación de múltiples clasificadores disminuye la varianza, especialmente en el caso de clasificadores inestables, y puede producir una clasificación más fiable.

Ilustración 7 Muestreo de datos y construcción de un nuevo *learner* en *Bagging* y *Boosting*



Fuente: Elaboración propia.

Como se observa en la Ilustración 7 los algoritmos de *Bagging* y *Boosting* consiguen *learners* creando datos adicionales en la etapa de entrenamiento mediante muestreo aleatorio con reemplazo del conjunto original. Al muestrear con reemplazo, algunas observaciones pueden repetirse en cada nuevo conjunto de datos de entrenamiento.

Para *Bagging*, cualquier elemento tiene la misma probabilidad de aparecer en un nuevo conjunto de datos, los modelos se construyen de forma paralela o independiente y el resultado de predicción se obtiene aplicando la media a las respuestas de los N alumnos.

En *Boosting* se construye al nuevo *learner* de forma secuencial, las observaciones se ponderan, tal y como se observa en la Ilustración 7, por tanto, algunas de ellas participarán en los nuevos conjuntos con más frecuencia. Después de cada paso de entrenamiento, los pesos se redistribuyen. Los datos que resultan mal clasificados aumentarán sus pesos para enfatizar los casos más difíciles. De esta forma, los *learners* posteriores se centrarán en ellos durante su formación.

Boosting no es mejor o peor que *Bagging*, no hay un ganador como tal porque depende de los datos, la simulación y las circunstancias. Ambos disminuyen la varianza de su estimación única, combinan varias estimaciones de diferentes modelos. Entonces, el resultado puede ser un modelo con mayor estabilidad. Si el problema es que el modelo único obtiene un rendimiento muy bajo, el *Bagging* rara vez tendrá un mejor sesgo. No obstante, *Boosting* podría generar un modelo combinado con menos errores, ya que optimiza las ventajas y reduce las trampas del modelo único. Por el contrario, si la dificultad del modelo único es sobreajuste, entonces el Bagging es la mejor opción. *Boosting*, por su parte, no ayuda a evitar un ajuste excesivo; de hecho, esta técnica se enfrenta a este problema en sí. Por esta razón, el *Bagging* es más efectivo que el *Boosting*.

A continuación, se procede a explicar los métodos de conjunto que serán aplicados en este estudio: *Random Forest, CatBoost, GBM y XGB.*

- ***Random Forest***

En Random Forest los árboles corren en paralelo, sin interacción entre ellos, construyendo varios árboles de decisión en el *training* para después generar una media de las clases que corresponde con la predicción. Es un modelo potente y preciso que, por lo general, funciona muy bien en muchos problemas, incluidos las funciones con relaciones no lineales. Sin embargo, entre las desventajas se encuentran: no hay interpretabilidad, el sobreajuste puede ocurrir fácilmente y es necesario elegir el número de árboles que se ejecutarán en el modelo.

Actualmente, los algoritmos de *boosting* son algoritmos de referencia y, en los últimos años, se han desarrollado varios tipos de algoritmos dentro de esta familia. A continuación, se analizarán las similitudes y diferencias entre los algoritmos de *boosting* que se van a aplicar en este trabajo: *XGBoost, CatBoost y LGBM.*

La diferencia entre los algoritmos mencionados se encuentra en la propia implementación, en la compatibilidad de los datos introducidos y en las limitaciones técnicas para cada uno de los modelos. En primer lugar, *XGBoost* trató de acelerar el tiempo de training de Gradient Boosting. Después LGBM y CatBoost tratararon de realizar lo mismo, con diferentes técnicas basadas en la forma de dividir. A continuación, se repasan las principales diferencias para cada uno de los modelos:

Respecto a las **divisiones**, LGBM utiliza por un lado un muestreo basado en gradientes y, por otro lado, una muestra aleatoria con gradientes pequeños. CatBoost utiliza una técnica llamada Minimal Variance Samplig (MVS), es un muestreo con ponderación del aumento de gradiente estocástico. XGboost, por su parte, no usa ninguna técnica de muestreo ponderado, esto provoca que el tiempo de decisión sea más lento en comparación con los dos métodos explicados anteriormente.

Respecto al **crecimiento del árbol**, CatBoost genera un árbol en equilibrio, es decir, en cada nodo de decisión se elige el par de división de atributos que minimiza la perdida. LGBM usa el método de hacer crecer a aquella hoja que minimiza la perdida, por tanto, se genera un árbol desequilibrado. Este árbol no aumenta por niveles, sino por hojas, aquí es importante tener muy en cuenta la profundidad del árbol. Por último, XGBoost divide el árbol hasta el parámetro especificado por el usuario y después va de abajo hacia arriba, es decir, suprime aquellas particiones que no hay una ganancia de información.

Respecto al método de importancia de factores (*feature importance*), *CatBoost* tiene dos: el primero llamado cambio en el valor predicho muestra la variación de la predicción en función del cambio en el valor del atributo, por tanto, aquellas características con mayor variación tendrán más importancia. El segundo método se denomina cambio en la Función de Perdida y el valor es la pérdida del modelo con el atributo y sin el atributo. Si la pérdida del modelo sin el atributo es inferior a con el atributo indicará que no es una característica de gran importancia.

LGBM y *XGBoost* comparten dos métodos parecidos, el primero de ellos es "*Gain*" y hace referencia al incremento de la precisión (*Accuracy*) que aporta un atributo a dicha

rama específica. El segundo método, denominado "Split" en *LGBM* y "*Frecuency*" o "*Weigth*" en *XGBoost*, determina la frecuencia con la que un atributo aparece en todas las particiones de los árboles. Además, *XGBoost* cuenta con un método llamado "Cobertura" que hace referencia a la frecuencia relativa de observaciones relacionadas con un atributo. Para cada atributo, se cuenta el total de observaciones empleadas para decidir el nodo hoja.

Finalmente, en cuanto a la gestión de los atributos categóricos, *CatBoost* emplea una combinación one-hot[8] para factores con un número pequeño de categorías. *LGBM* divide los atributos categóricos en 2 subconjuntos en función del objetivo de entrenamiento. *XGBoost*, por su parte, no posee un método para los factores categóricos y es el usuario quien debe realizarlo de forma previa.

3.4. Medidas de performance

Para determinar qué medidas de performance emplear para comparar los modelos es importante analizar el problema que se trata. Por ejemplo, cuando un modelo trata de predecir si se ha producido fraude o no, no es lo mismo el errores que se produce cuandola predicción es que una persona ha cometido fraude siendo esto falso a cuando el modelo predice que no se ha producido fraude cuando realmente si se ha producido. Por tanto, este último caso es más grave en ese caso de estudio del fraude. A continuación, se detallan ciertas métricas de valoración estudio de los modelos:

- **Curva ROC**

El análisis de la **curva ROC** (*Receiver Operating Characteristics*) es un método estadístico que permite ver la eficacia y exactitud de un modelo. Es un gráfico visual y sencillo de interpretar donde la variable ajustada es de tipo binario. Lo que nos muestra la curva ROC es la comparación entre los verdaderos positivos y los falsos positivos. Respecto a la forma de la curva ROC, cuanto más cerca este de una L invertida indica que más preciso es el modelo. Por el contrario, cuanto más próximo a la diagonal, disminuye la precisión del modelo. En ausencia de un modelo la diagonal representa la propia aleatoriedad, es decir, un área bajo la curva de 0.5, el resultado de predecir una variable de tipo binaria de forma aleatoria es 50% de probabilidad de predecirlo de forma

[8] One-hot: consiste en la transformación de los datos categóricos en numéricos. Uno de los grandes problemas de los algoritmos de ML es que ciertos algoritmos no pueden operar con variables categóricas. Por ello, es necesario realizar dicha transformación.

correcta. Por tanto, muestra cómo de bueno es el modelo al realizar las predicciones, Concretamente, el valor debajo de la curva denota la probabilidad que hay de clasificar correctamente a una muestra escogida al azar.

- **Matriz de confusión**

La matriz de confusión muestra, una vez aplicados los modelos, una comparación entre el valor real y el valor predicho por el modelo. De esta, se pueden obtener 4 posibles resultados:

- Verdadero positivo: el modelo predice que están interesados en contratar el seguro de automóvil y la respuesta real, es la misma.
- Falso positivo: el modelo predice que están interesados en contratar el seguro de automóvil y, realmente, no lo están. Por tanto, el modelo no predice correctamente.
- Verdadero negativo: el modelo predice que no están interesados en contratar el seguro de automóvil y la respuesta real, es la misma.
- Falso negativo: el modelo predice que no están interesados en contratar el seguro de automóvil cuando, realmente, si lo están.

- **Exactitud**

La exactitud es la proporción de los datos que el modelo ha clasificado de forma correcta. Así, si denotamos como Ac la proporción estimada correctamente, la tasa de error sería igual **a E = 1 – Ac.** En la matriz se puede observar las distintas clasificaciones en función de si se han clasificado correctamente, es decir, correspondería a los verdaderos positivos y a los verdaderos negativos, y los que se han clasificado de forma incorrecta serían los falsos negativos y los falsos positivos.

Tabla 1 Tipo de error de clasificación

		Prediction class	
		A	**B**
Actual class	**A**	True Positive	False Negative **Error type 2**
	B	False Positive **Error type 1**	True Negative

Fuente: Elaboración propia.

$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

- **Precisión**

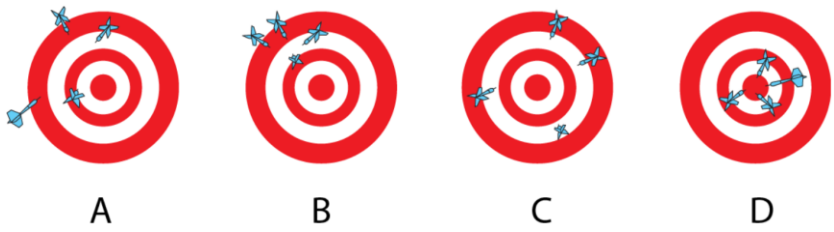
La precisión mide cuantas observaciones han sido clasificadas correctamente. Es importante destacar que se trata de una métrica que no es conveniente utilizar cuando existe una descompensación grande en la respuesta, porque el resultado estaría sesgado por esta descompensación de datos.

$$Precisión = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (3)$$

Así mismo, la precisión hace referencia a cuánto de cerca está un valor del verdadero o falso, mide los verdaderos positivos entre valor resultante de sumar los verdaderos positivos más los falsos positivos. La precisión y la exactitud son conceptos que difieren en gran medida en sí, un modelo puede ser preciso, pero no exacto y viceversa. El mejor modelo será aquel que sea preciso y exacto. En la Ilustración 8 se observan las siguientes dianas:

A. No es exacto ni preciso.
B. Es preciso porque los dardos están muy juntos, pero no exacto.
C. No hay precisión porque los dardos están separados, pero sí exactitud porque los dardos están equidistantes y si se calculara la media de la posición entre los dados, se daría en el centro de la diana.
D. Hay exactitud y precisión.

Ilustración 8. Precisión vs Exactitud



Fuente: Practices of Science: Precision vs. Accuracy[9]

[9] Practices of Science: Precision vs. Accuracy, Disponible en: https://manoa.hawaii.edu/exploringourfluidearth/physical/world-ocean/map-distortion/practices-science-precision-vs-accuracy

Existen otras medidas como el análisis de sensibilidad y especificidad. El análisis de sensibilidad nos muestra cuál es la proporción de verdaderos positivos respecto al total de positivos. La especificidad hace referencia a los verdaderos negativos respecto al total de negativos.

$$Sensibilidad = \frac{True\ Positive}{Positive} \quad (4)$$

$$Especificidad = \frac{True\ Negative}{Negative} \quad (5)$$

CAPITULO 4. CASO PRÁCTICO: APLICACIÓN DE LOS ALGORITMOS DE *MACHINE LEARNING*

4.1. Explicación del método de aplicación práctica

Después de realizar un análisis sobre la importancia del *cross-selling* en una compañía aseguradora y explicar el marco teórico que fundamenta los métodos de ML se procede a la aplicación práctica de cada uno de los algoritmos explicados en el Capítulo 2 del presente trabajo.

En primer lugar, es importante mencionar la fuente de origen de la base de datos. Ésta ha sido obtenida del repositorio *Kaggle*[10] que contiene millones de bases de datos referentes a distintas disciplinas. A continuación, se resume de forma breve cada uno de los pasos que se van a llevar a cabo para la aplicación de cada uno de los modelos:

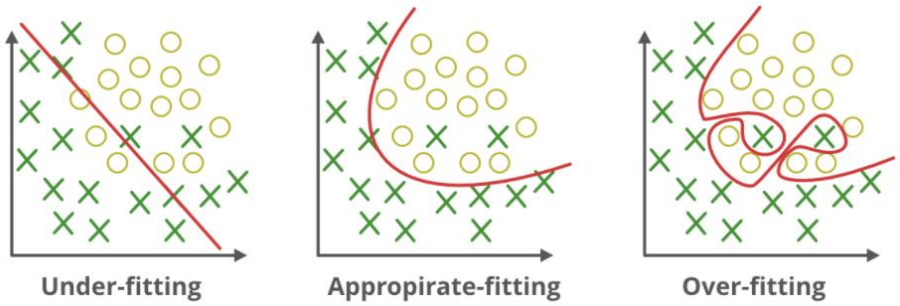
1. Exploración y limpieza de la base de datos
2. Análisis exploratorio de los factores mediante un análisis univariable y multivariable (análisis de correlación)
3. Análisis de la importancia de los factores
4. Aplicación de una técnica de balanceado de datos
5. Aplicación práctica de los modelos de *Machine Learning*
6. Obtención de los resultados y conclusiones

Para la aplicación de los métodos de *Machine Learning*, se ha utilizado el método de Train – Test. La primera etapa se denomina *train* y, la segunda, *test*. En primer lugar, los datos de la base de datos son divididos, en este caso se ha escogido un 80% para el *training* y el 20% restante para el *test*, En la primera etapa, los datos se utilizan para entrenar el modelo. En la segunda etapa, el objetivo es observar el poder de predicción de este algoritmo sobre unos datos que no han sido usados en la generación del algoritmo. De esta forma, se puede comparar si los resultados obtenidos son similares o distan en gran medida de los reales mediante técnicas de *performance*. En este último supuesto,

[10] Kaggle. "Health insurance cross sell prediction". https://www.kaggle.com/anmolkumar/health-insurance-cross-sell-prediction

deberíamos hacer un *overfitting*[11] o un *underfitting*[12]. En la siguiente ilustración se puede observar de forma visual estos conceptos.

Ilustración 7. Underfitting y Overfitting en *Machine Learning*



Fuente: Geeksforgeeks (Mayo, 2020)

4.2. Análisis exploratorio de los datos

A continuación, se procede a detallar cada una de las variables explicativas o independientes que pertenecen a la base de datos.

Tabla 2 Factores de riesgo

Nombre de la variable	Explicación de la variable
id	Variable que identifica al cliente
Gender	Género
Age	Edad
Driving_License	0: El cliente no dispone de Carnet de conducir 1: El cliente dispone de Carnet de conducir
Region_Code	Código único que identifica la región.
Previously_Insured	0: El cliente no tiene seguro de coche 1: El cliente ya tiene seguro de coche
Vehicule_Age	Años de antigüedad del vehículo
Vehicule_Damage	0: El cliente no ha sufrido daños en su coche en el pasado 1: El cliente ha sufrido un accidente en el pasado
Annual_Premium	Prima anual que paga el cliente.
PolicySalesChannel	Código anónimo que identifica el canal de venta (en persona, por teléfono, por correo…etc)
Vintage	Número de días que el cliente ha estado asociado en la compañía
Response	0: No interesado en la poliza 1: Interesado en la poliza

Fuente: Elaboración propia

[11] El concepto de *overfitting* hace referencia a las correcciones del modelo cuando está sobreajustado.
[12] El concepto de *underfitting* hace referencia a las correcciones del modelo cuando está infraajustado.

Cabe destacar ciertas particularidades de la base de datos. Para tener un orden lógico, se explican estas particularidades en orden:

- Id: es una variable que identifica al cliente con un código, no nos aporta información extra, por tanto, no será una de las variables explicativas a utilizar.
- *Gender*: el género del cliente se clasifica como *male* o *female*.
- *Age*: la edad del cliente está comprendida entre 20 y 85 años.
- *Driving_License*: la no posesión de la licencia de conducir implicará el no poder contratar el seguro del vehículo y, por tanto, será necesario eliminar aquellas pólizas donde el cliente no tenga carnet de conducir, se realizará en la limpieza de los datos.
- *Response:* sería nuestra Y, es decir, la variable dependiente o *target* que tratamos de predecir. Es una variable binaria que indica rechazo (valor 0) o la aceptación (valor 1) de la oferta de seguro de auto.

4.3. Análisis univariable y análisis de correlación

Antes de aplicar cualquier modelo, el primer paso, es realizar un EDA[13]. En primer lugar, se muestra un resumen de las principales características de cada una de las variables, se calculan las siguientes métricas: media, desviación típica, mínimo, máximo y cuartiles.

Tabla 3 Resumen estadístico de las variables

	Gender	Age	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
count	381109.00	381109.00	381109.00	381109.00	381109.00	381109.0	381109.00	381109.00	381109.00	381109.00
mean	0.54	38.82	26.39	0.46	0.61	0.5	30564.39	112.03	154.35	0.12
std	0.50	15.51	13.23	0.50	0.57	0.5	17213.16	54.20	83.67	0.33
min	0.00	20.00	0.00	0.00	0.00	0.0	2630.00	1.00	10.00	0.00
25%	0.00	25.00	15.00	0.00	0.00	0.0	24405.00	29.00	82.00	0.00
50%	1.00	36.00	28.00	0.00	1.00	0.0	31669.00	133.00	154.00	0.00
75%	1.00	49.00	35.00	1.00	1.00	1.0	39400.00	152.00	227.00	0.00
max	1.00	85.00	52.00	1.00	2.00	1.0	540165.00	163.00	299.00	1.00

Fuente: Elaboración propia

De la Tabla 3 los aspectos más relevantes que pueden extraerse son los siguientes:

- En el 54% de la base de datos el género corresponde a hombre.
- La edad media de los *leads* registrados en esta base de datos es 39 años.

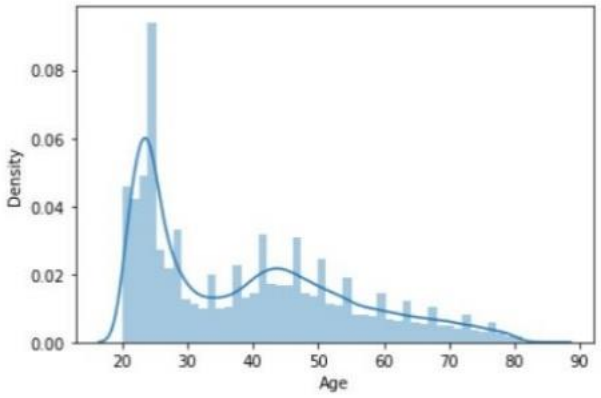
[13] *Exploration Data Analysis.*

- Respecto a la variable *Previuosly_insured*, se observa un comportamiento de un 46% no tiene seguro de auto, mientras que un 54% sí tiene.
- Respecto a los años del vehículo, la media es de 0.61 años.
- La variable *Vehicle_Age* indica que la mitad de los vehículos han sufrido algo tipo de daño con anterioridad.
- La prima anual media es de 30.564, no obstante, es importante destacar que esto solo es una media. Se encuentran valores atípicos donde el valor máximo es de 540.165. No se conoce a priori la unidad monetaria en la que está expresada dicha prima.
- Vintage hace referencia al número de días que el cliente ha estado asociado en la compañía.
- Finalmente, el *target* es la variable *response* e indica que un 12% del *dataset* está interesado en contratar el seguro de automóvil.

Mediante un análisis univariable para cada una de las variables explicativas, observar su distribución, los valores y las frecuencias que toman. Así, en la Ilustración 9 se observa como la distribución de las edades de se concentra entre los 20 y 30 años observándose un pequeño repunte de masa entre los 40 y 50 años. La razón de esta concentración podría deberse a:

- La compañía podría tener un mayor interés en vender un seguro de auto a este segmento de la población, personas jóvenes. Esto puede deberse al mayor perfil de riesgo, es decir, prima de riesgo y a la posible obtención de un margen mayor de rentabilidad.
- La compañía tiene en cartera un gran porcentaje de población joven versus adulta.

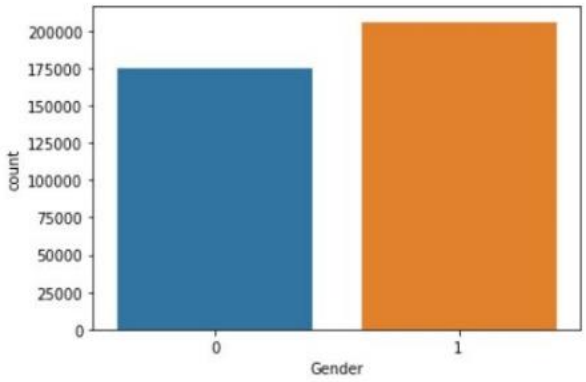
Ilustración 9 Distribución del factor *Age*



Fuente: Elaboración propia

La Ilustración 10 muestra la distribución del género de las pólizas donde se observan más pólizas pertenecientes a mujeres que a hombres.

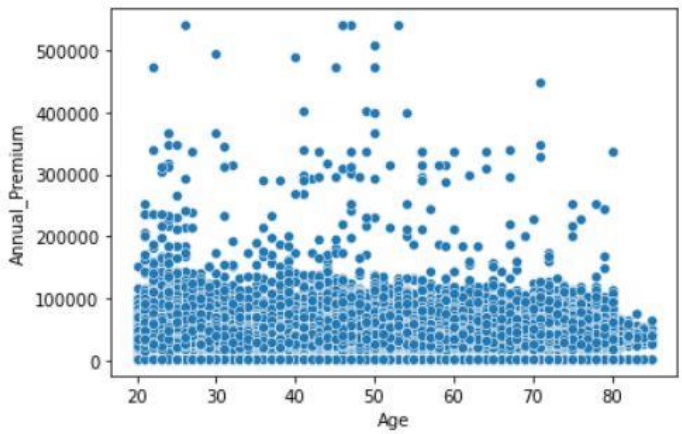
Ilustración 10 Distribución del factor gender



Fuente: Elaboración propia

En la Ilustración 11 se aprecia la distribución de la Prima anual del seguro de vida en función de la edad del asegurado, como se puede observar existe presencia de valores atípicos y también *outliers* que sobresalen de la parte inferior donde están concentrados la mayor parte de los valores. Estos outliers pueden proceder de un perfil de riesgo mayor y, por tanto, una prima de riesgo elevada.

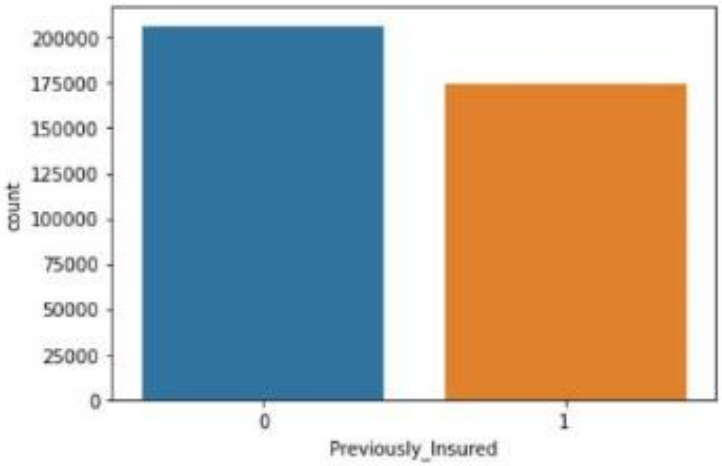
Ilustración 11 Distribución del factor Annual_Premium



Fuente: Elaboración propia

En la Ilustración 12 se enumera en número de pólizas que han estado aseguradas previamente. Hay un mayor porcentaje de pólizas que no han estado aseguradas previamente.

Ilustración 12 Distribución del factor Previously_Insured

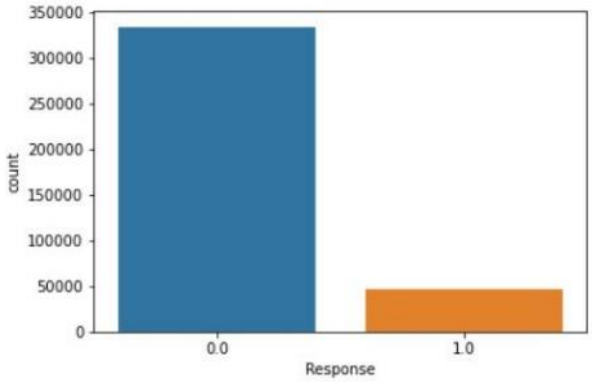


Fuente: Elaboración propia

En la Ilustración 13 se observa la variable objetivo o target, el valor 0 indica que no está interesa en contratar el seguro de automóvil y el valor 1 es el caso afirmativo. Se observa una gran descompensación y será necesario realizar un balanceado de datos porque existe el riesgo de que, al entrenar el modelo en la etapa de training, el modelo pueda tender hacia la parte con mayor proporción de forma errónea. La técnica que se

utilizará para balancear los datos se denomina *NearMiss*, después de realizar pruebas con diferentes técnicas de *Undersampling*[14], es la que mejores resultados ha proporcionado.

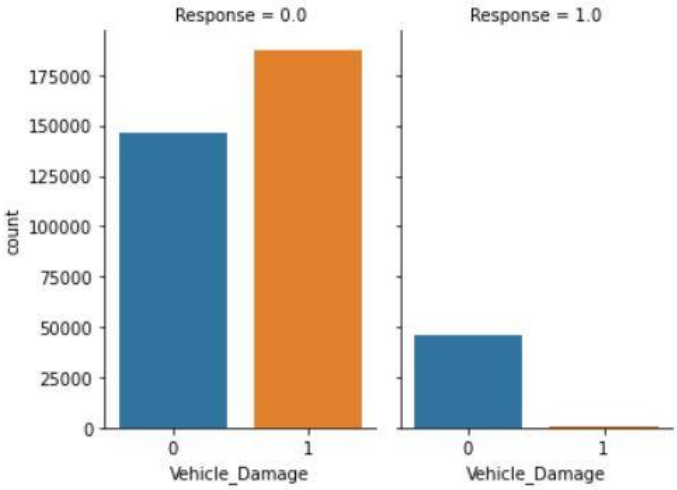
Ilustración 13. Distribución del factor Response



Fuente: Elaboración propia

En la Ilustración 14 se observa la distribución del factor vehículo dañado en función del *target*, es decir, en función de la respuesta: interesado o no en la contratación del seguro de auto. Como se observa, casi el 100 % de los clientes potenciales que podrían estar interesados en contratar el seguro no han sufrido daños, anteriormente, en el vehículo.

Ilustración 14. Distribución del factor Vehicle_Damage en función de Response

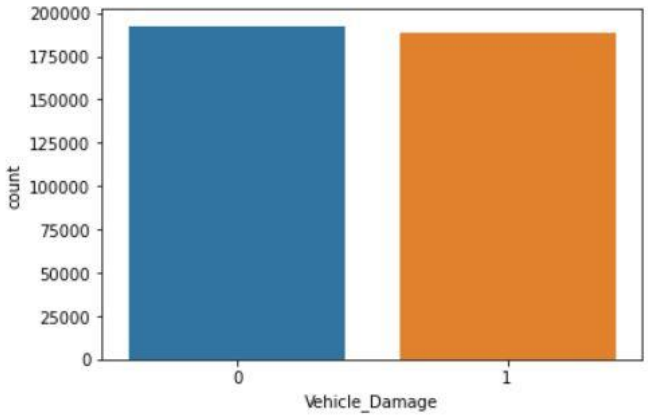


Fuente: Elaboración propia

[14] Las técnicas de *undersampling* son utilizadas para balancear los datos de forma que se reduce los datos de la proporción mayor en contraste con *oversampling* que genera nuevos datos, mediante diferentes técnicas, para incrementar la proporción menor.

En la Ilustración 15, la distribución de si el vehículo ha sido dañado o no, se observa prácticamente la mitad de los coches dañados y la otra mitad sin sufrir ningún tipo de daño.

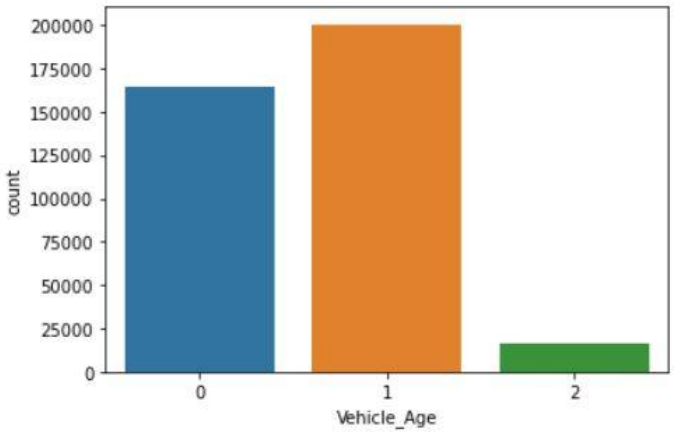
Ilustración 15. Distribución del factor Vehicle_Damage



Fuente: Elaboración propia

La Ilustración 16 muestra la variable años de antigüedad del vehículo, la mayor proporción son vehículos nuevos o con solo un año de antigüedad.

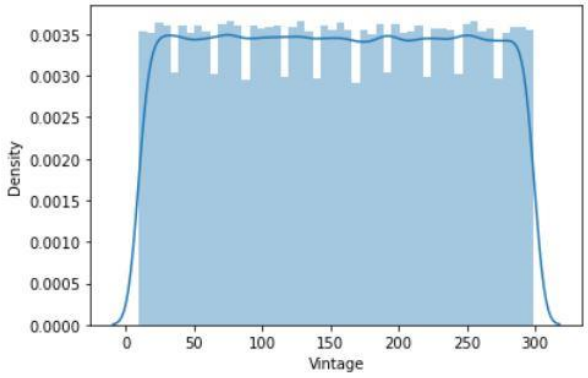
Ilustración 16. Distribución del factor Vehicle_Age



Fuente: Elaboración propia

En la Ilustración 17, la distribución del factor Vintage se observa una distribución uniforme a lo largo de todo el año.

Ilustración 17. Distribución del factor Vintage



Fuente: Elaboración propia

Así mismo, después de analizar cómo se comportan las variables de forma individual es importante realizar un análisis multivariable mediante un análisis de correlación y un análisis de *feature importance*. Se realiza el análisis de correlación entre las variables explicativas para tratar de entender mejor el comportamiento entre las variables y como interactúan entre ellas.

Ilustración 18. Matriz de correlaciones de las variables explicativas

Fuente: Elaboración propia

En la Ilustración 18 se presenta las correlaciones entre las variables explicativas. Para las correlaciones positivas destacan los años del vehículo y la edad, presentan una correlación fuerte positiva. Se puede encontrar sentido de negocio en esta correlación porque podríamos pensar que una persona joven tenderá a tener un vehículo menos antiguo en comparación con una persona de edad adulta.

La correlación entre *Previously_Insured* y *Vehicule_Damage* es muy fuerte, significa que hay una relación positiva entre aquellos clientes que han estado asegurados previamente y han sufrido daños en el vehículo y, por el contrario, aquellos clientes que no han estado asegurados previamente no han sufrido daños en el vehículo. Se puede intuir que este *dataset* procede de la realización previa de un *scoring*, principalmente porque la compañía podría estar interesada en ofrecer un seguro de auto a nuevos clientes que no hayan sufrido daños en el vehículo, porque a priori se entiende que es un perfil de riesgo menor, al no haber sufrido accidentes.

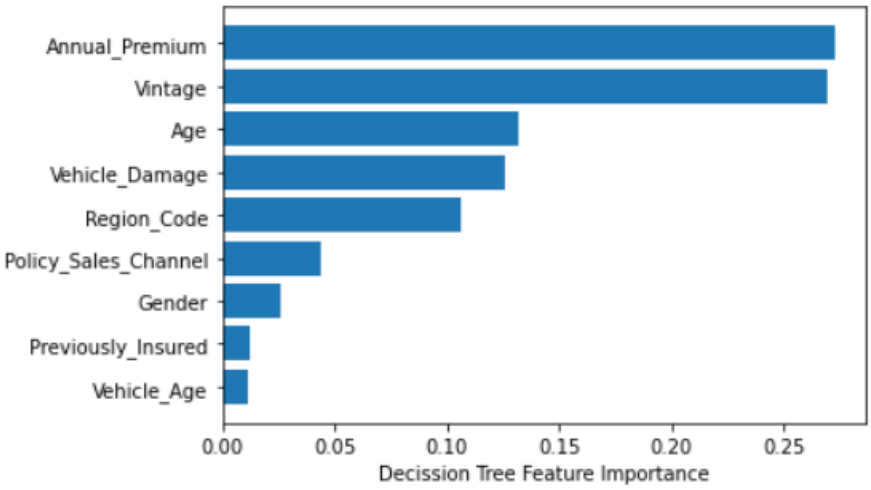
Respecto a las correlaciones negativas, destaca *Policy_Sales_Channel* y *Age*. Esto puede tener sentido de negocio porque en función del valor que tome la variable Policy_Sales_Channel, que se entiende como una variable categórica que denota el tipo de canal de venta, la edad puede influir en el sentido de que una persona joven quizá tiende a utilizar canales de venta que sean digitales en comparación con los canales tradicionales como puede ser un punto de venta físico.

4.4. Análisis importancia de los factores

Antes de aplicar las diferentes técnicas de *Machine Learning*, es importante entender qué variables poseen una mayor importancia sobre el target *Response*. De esta forma, se aplican diferentes métodos para analizar la importancia de los factores: *Decission Tree*, *Random Forest* y *XGBoost*. A continuación, se presentan los resultados obtenidos:

- ***Decission Tree***

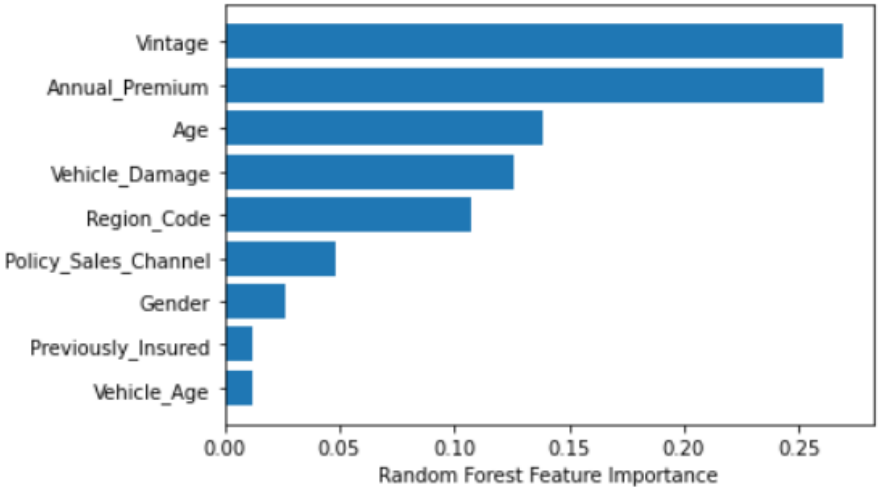
Ilustración 19. Decission Tree for future importance



Fuente: Elaboración propia

- ***Random Forest***

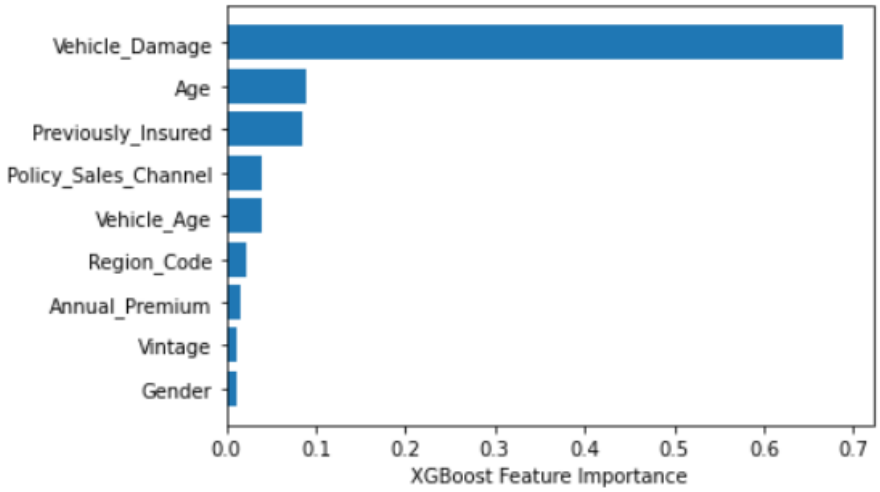
Ilustración 20. Random Forest for future importance



Fuente: Elaboración propia

- ***XGBoost***

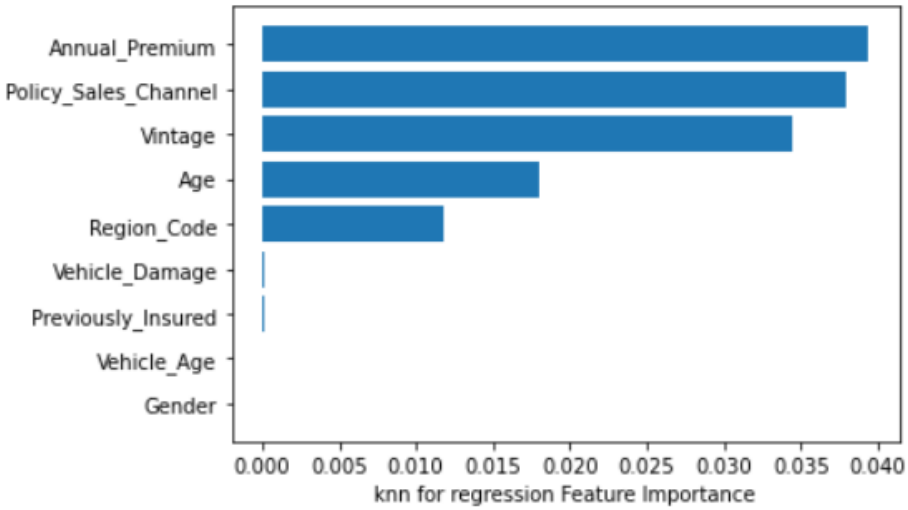
Ilustración 21. XGBoost for future importance



Fuente: Elaboración propia

- ***Knn for regression***

Ilustración 22. Knn for future importance



Fuente: Elaboración propia

Los resultados que se obtienen aplicando las dos primeras técnicas (*Decission Tree* y *Random Forest*) son similares, como era de esperar, debido a que *Random Forest* es una ampliación de *Decission Tree*. La diferencia radica en el valor de la importancia para los cuatro primeros factores más importantes. *Random Forest* genera múltiples árboles que

corren en paralelo para después sacar una media, es decir, *Decission Tree* de forma repetida. No obstante, *XGBoost* arroja unos resultados que difieren en gran medida, prioriza la variable *Vehicle_Damage* muy por encima del resto de variables. Por último, el método KNN[15], a diferencia de otros modelos, considera que la variable *Vehicule*_Damage no es un factor con gran importancia. No obstante, el resto de los factores guardan coherencia con los resultados obtenidos en los modelos previos.

Finalmente, el análisis *feature importance* resulta muy útil para observar, a priori, cuáles son las variables que aportan mayor valor o significancia al *target*. Cuando tenemos una gran cantidad de variables explicativas y, de acuerdo con el Principio de Parsimonia[16], el objetivo es crear un modelo simplificado que consiga explicar nuestro *target* simplemente con aquellas variables que sean significativas y no sean redundantes entre ellas, es decir, no exista correlación entre las variables.

4.5. Técnica de balanceado de datos

En el *dataset* original el *target* o variable dependiente presenta una descompensación. Concretamente, se observa que, de un total de 380.917 respuestas, hay 334.399 respuestas que son 0, no están interesados, y 46.710 que son 1, es decir, que sí están interesados en la contratación del seguro de automóvil. Por tanto, esta gran descompensación puede dar lugar a un riesgo de que el modelo tienda a la respuesta con mayor presencia. Por tanto, es necesario aplicar una técnica de balanceado de datos.

Ilustración 23. Número de respuestas para cada clase

```
0.0    334399
1.0     46710
Name: Response, dtype: int64
```

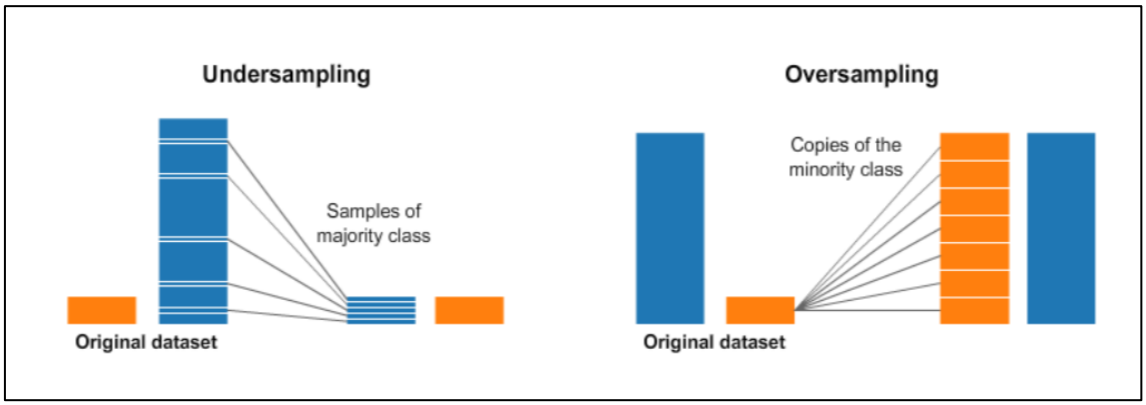
Fuente: Elaboración propia

[15] El método k-nearest neighbords (KNN) es un método supervisado de clasificación y regresión utilizado para reconocer patrones. Este método se centra en buscar las características o atributos de los k vecinos o datos más próximos para, posteriormente, obtener una salida. En clasificación, la salida es la propia pertenencia a una clase u otra. En regresión la salida es el propio valor obtenido.

[16] El principio de Parsimonia hace referencia a la simplicidad del modelo, es decir, se debe evitar incluir variables que sean redundantes.

Hay dos tipos de técnicas de balanceado: *Undersampling* y *Oversampling*. En la primera, se busca reducir la clase con mayor presencia y en la segunda incrementar la clase con menos presencia. En la Ilustración 24 se observa de forma más clara.

Ilustración 24. Técnicas de balanceado: Undersampling y Oversampling



Fuente: Resampling strategies for imbalanced datasets, Rafael Alencar, Kaggle

Para el caso de estudio, teniendo en cuenta la gran cantidad de datos disponibles, se ha considerado probar técnicas de *Undersampling* como *NearMiss*, *Tomek Links* y *Random Under Sample*. Se procede a realizar una breve explicación de estas técnicas. La técnica que mejor resultados ha dado a posteriori una vez aplicados los modelos de ML es NearMiss 1.

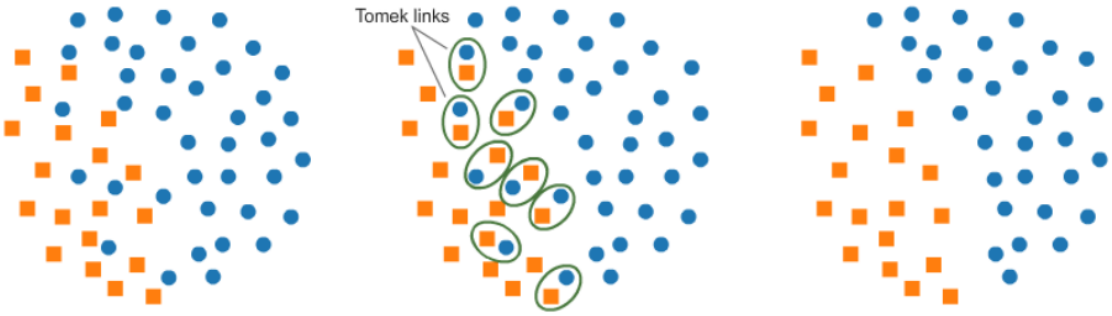
- ***Random Undersampling***

Es una técnica de submuestreo simple que de forma aleatoria elimina datos de la clase mayoritaria, esto puede llevar a la pérdida de información. No obstante, si para el caso de estudio los datos de la clase mayoritaria no presentan gran dispersión, podría dar buenos resultados.

- ***Tomek Links***

Esta técnica de submuestreo escoge dos puntos de diferentes clases que están muy próximos y elimina aquellos puntos de la clase mayoritaria que estén muy próximos a la clase contraria, incrementando de esta forma el espacio existente entre dos puntos pertenecientes a diferentes clases y facilitando, de esta forma, el proceso de clasificación, tal y como se muestra en la siguiente ilustración.

Ilustración 25. Técnica de submuestreo: Tomek Links



Fuente: Resampling strategies for imbalanced datasets, Rafael Alencar, Kaggle

- ***Near Miss 1***

De acuerdo con el enfoque realizado en el artículo "Knn Approach to Unbalanced Data Distribution: A Case Study involving Information Extraction" las técnicas de submuestreo *NearMiss* realizan el submuestreo de datos en la clase mayoritaria teniendo en cuenta la distancia entre los puntos. Hay 3 variantes. La variante escogida para el caso de estudio es *NearMiss 1* escoge aquellos puntos con la distancia media más cercana a otros k puntos cercanos, donde k es un hiperparámetro que se ajusta. En el "Anexo. Código Python" se muestra la aplicación de las 3 técnicas de balanceado.

CAPÍTULO 5. RESULTADOS

En el capítulo 5 se procede a explicar todos los resultados obtenidos para cada uno de los modelos de ML aplicados. Para valorar los resultados del modelo se utilizan aquellas métricas que mejor valoren el caso de estudio. De acuerdo con el objeto de estudio, se puede denotar la hipótesis nula y alternativa de la siguiente forma:

- H_0= 0. No interesado en la contratación del seguro
- $H_1 = 1$. Interesado en la contratación del seguro

Mientras el error tipo I se basa en rechazar la hipótesis nula cuando ésta es verdadera, el error tipo II consiste en aceptar la hipótesis nula cuando es falsa. Para el caso de estudio, el error tipo I se produce cuando el modelo predice que el cliente si está interesado en contratar el seguro cuando realmente no lo está. El error tipo II sería predecir que el cliente no está interesado en contratar el seguro cuando en realidad si está interesado.

Concretamente, el error más grave es el error tipo II debido a que es un cliente potencial que realmente sí contrataría el seguro, pero si el modelo predice que no, se perdería este posible cliente potencial o bien no se le daría la misma importancia respecto a los demás. Así, las métricas que se utilizarán para valorar el caso de estudio son: exactitud, precisión, el área bajo la curva ROC, la especificidad y la matriz de confusión.

5.1. Resultados modelo GLM

Un GLM posee tres componentes básicos, así, las características principales del modelo GLM escogido son:

1. El componente aleatorio: la variable dependiente o variable respuesta es binaria, el tipo de respuesta es no interesado (0) o interesado (1). Por tanto, se escoge la familia binomial.
2. El componente sistemático hace referencia a las variables predictoras del modelo, son las que se han detallado en el Capítulo 4.
3. La función de enlace para la familia binomial es una logit.

En cuanto al método utilizado que se denomina IRLS en la salida de Python hace referencia a estimaciones de máxima verosimilitud. Esta estimación está basada en la iteración, repetir tantas veces como sea necesario una operación hasta encontrar el máximo o mínimo de una función.

Los resultados que se obtienen en la regresión son satisfactorios porque todas las variables son significativas para un nivel de significación del 1%. Así mismo, cabe destacar que no existe correlación alta entre las variables incluidas en este modelo, por tanto, se trata de un modelo óptimo para realizar predicciones y observar el poder predictivo que presenta.

Ilustración 26. *Regresión logística con todas las variables*

Generalized Linear Model Regression Results

Dep. Variable:	Response	No. Observations:	74736
Model:	GLM	Df Residuals:	74727
Model Family:	Binomial	Df Model:	8
Link Function:	log	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	nan
Date:	Sat, 29 May 2021	Deviance:	2.1493e+05
Time:	10:54:27	Pearson chi2:	1.00e+11
No. Iterations:	100		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Gender	-3.985e-05	7.05e-09	-5655.089	0.000	-3.99e-05	-3.98e-05
Age	-1.097e-05	2.37e-10	-4.63e+04	0.000	-1.1e-05	-1.1e-05
Region_Code	-5.191e-05	4.61e-10	-1.13e+05	0.000	-5.19e-05	-5.19e-05
Previously_Insured	-3.5021	0.090	-38.708	0.000	-3.679	-3.325
Vehicle_Age	-0.0012	1.06e-08	-1.17e+05	0.000	-0.001	-0.001
Vehicle_Damage	-1.7622	0.034	-52.489	0.000	-1.828	-1.696
Annual_Premium	1.293e-08	4.92e-14	2.63e+05	0.000	1.29e-08	1.29e-08
Policy_Sales_Channel	-6.097e-06	8.02e-11	-7.6e+04	0.000	-6.1e-06	-6.1e-06
Vintage	7.431e-07	3.13e-11	2.37e+04	0.000	7.43e-07	7.43e-07

Fuente: Elaboración propia

Así mismo, el valor del criterio AIC que presenta será utilizado para comparar con otros modelos, aquel modelo GLM que presente el menor valor en el resultado de AIC será el escogido, siempre y cuando todas las variables del modelo sean significativas.

Ilustración 27. GLM - Resultado criterio AIC

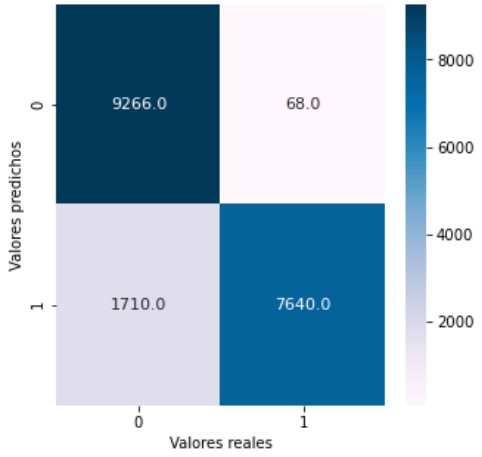
```
MSE GLM: 0.096
AIC GLM: -43849.764
```

Fuente: Elaboración propia

En la matriz de confusión se observan buenos resultados porque hay un porcentaje muy alto bien clasificado, no obstante, cabe destacar que se observa una descompensación en la contra diagonal. Es decir, dentro de los datos mal clasificados, entre el cuadrante inferior izquierdo y el cuadrante superior derecho hay una gran diferencia. Como se ha

mencionado anteriormente, el error tipo II es el más grave, se trata de reducir el valor del cuadrante superior derecho, el número de falsos negativos.

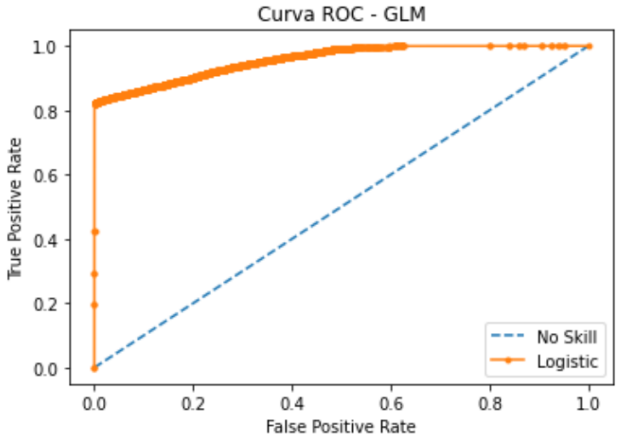
Ilustración 28. Matriz de confusión – GLM



Fuente: Elaboración propia

La curva ROC muestra que rápidamente se alcanzan valores altos de true positive rate, es una buena señal del modelo.

Ilustración 29. Curva ROC - GLM



Fuente: Elaboración propia

Tabla 4 Resultados GLM

Métrica	Valor obtenido
Exactitud	**0.9048**
Precisión	**0.9911**
ROC AUC	**0.9049**
Especificidad	**0.8171**

Fuente: Elaboración propia

En resumen, los resultados obtenidos en el GLM son buenos, se observa un alto valor de exactitud, precisión y área bajo la curva. Esto implica que este clasificador tiene una alta tasa de aciertos. Sin embargo, como el error más grave es el tipo II, es decir, una persona que si está interesada en el seguro de automóvil no será clasificada como una persona que no que está interesada, es más importante tener en consideración la especificidad. En este caso, el GLM presenta un valor de 0,8171.

5.2. Resultados modelo GLM reducido

Se ha probado realizar un GLM aplicando una reducción de variables y escogiendo únicamente las que mayor *feature importance* nos proporcionaban en el análisis. Así, se reduce el GLM a los factores que se observan en la Ilustración 30.

Ilustración 30 Salida GLM reducción de variables

Generalized Linear Model Regression Results

Dep. Variable:	Response	No. Observations:	74736
Model:	GLM	Df Residuals:	74730
Model Family:	Binomial	Df Model:	5
Link Function:	log	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	nan
Date:	Sat, 29 May 2021	Deviance:	75134.
Time:	11:11:27	Pearson chi2:	7.22e+04
No. Iterations:	77		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Age	-0.0025	0.000	-16.822	0.000	-0.003	-0.002
Region_Code	-0.0008	0.000	-3.730	0.000	-0.001	-0.000
Vehicle_Damage	-2.6510	0.035	-74.762	0.000	-2.721	-2.582
Annual_Premium	6.987e-07	1.06e-08	65.666	0.000	6.78e-07	7.2e-07
Policy_Sales_Channel	-0.0030	4.62e-05	-65.187	0.000	-0.003	-0.003
Vintage	-0.0002	3.23e-05	-7.114	0.000	-0.000	-0.000

Fuente: Elaboración propia

Los resultados que se obtienen en la regresión son satisfactorios porque todas las variables son significativas. Si comparamos los resultados obtenidos en el criterio AIC,

el modelo GLM con reducción de variables presenta un valor más bajo y, por tanto, es preferible este modelo.

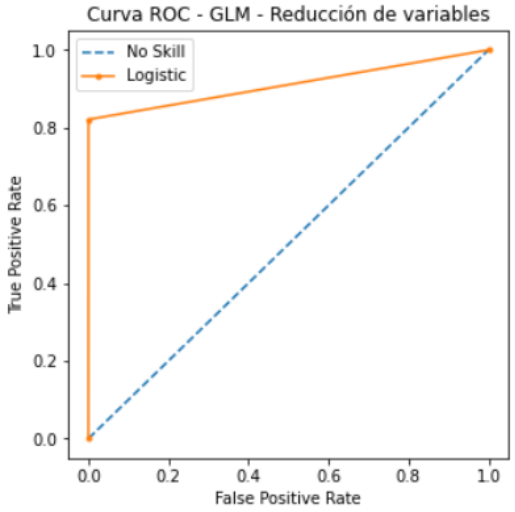
Ilustración 31. GLM reducido - Resultado criterio AIC

```
MSE GLM Reducción variables: 0.094
AIC GLM Reducción variables: -44251.359
```

Fuente: Elaboración propia

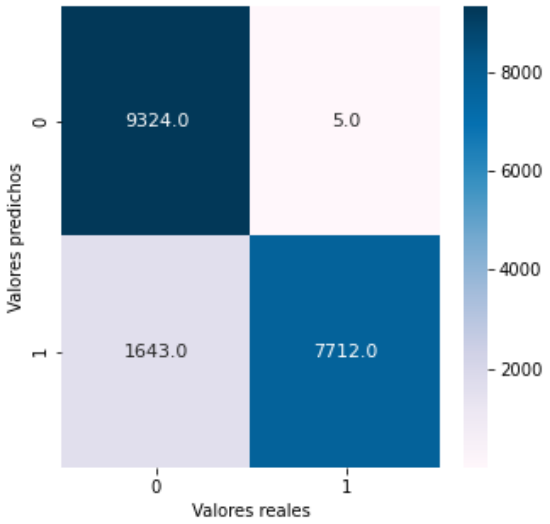
La curva ROC es similar a la obtenida en el modelo GLM completo. Alcanza valores altos de forma temprana.

Ilustración 32. Curva ROC – GLM – Reducción de variables



Fuente: Elaboración propia

Ilustración 33. Matriz de confusión – GLM reducción de variables



Fuente: Elaboración propia

Tabla 5 Resultados GLM reducido en métricas de performance

Métrica	Valor obtenido
Exactitud	**0.9117**
Precisión	**0.9993**
ROC AUC	**0.9119**
Especificidad	**0.8150**

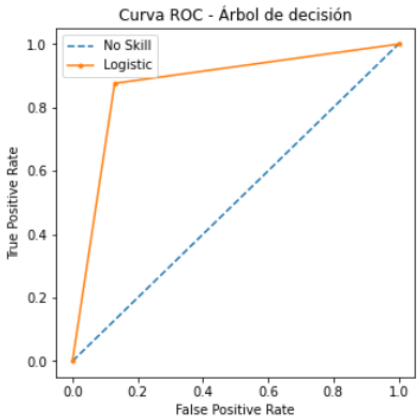
Fuente: Elaboración propia

Los resultados del GLM con la reducción de variables únicamente escogiendo aquellos factores con mayor importancia (*Age*, *Región_code*, *Vehicle_damage*, *Annual_premium*, *Policy_sales_channel* y *Vintage*) son muy similares a los resultados obtenidos en el GLM con la inclusión de todas las variables. En el GLM reducido se ganan centésimas en el área bajo la curva y se pierde algo de especificidad. No obstante, la cuantía pérdida no es significativa y de acuerdo con el criterio AIC y el Principio de Parsimonia, mencionado anteriormente, es preferible este modelo con reducción de variables.

5.2. Resultados árbol de decisión

En la curva ROC se observa que el modelo presenta una mayor lentitud para alcanzar valores altos de True Positive Rate, esto dará lugar a que el área bajo la curva sea inferior.

Ilustración 34. Curva ROC – Árbol de decisión



Fuente: Elaboración propia

Para el modelo Árbol de decisión, se obtiene unos valores altos en las diferentes métricas (Tabla 6), no obstante, en este modelo se observa que no se aproxima tanto a la coordenada (0,1) como lo hace el modelo GLM. Por tanto, el resultado del área bajo la curva es inferior. Así mismo, se observa una mayor lentitud en cuanto al avance del modelo, el clasificador presenta mayor dificultad para alcanzar una posición cercana al eje y=1.

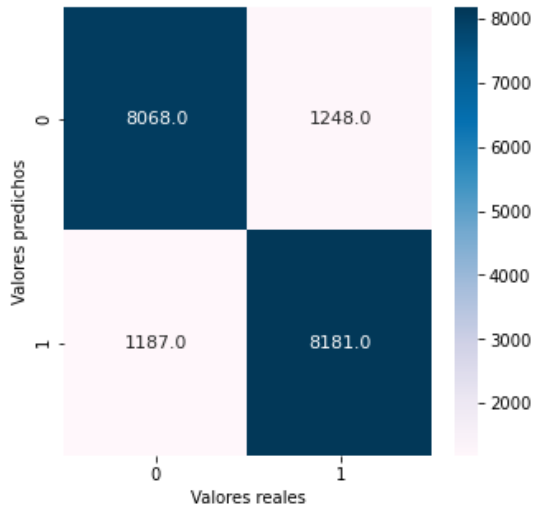
Tabla 6 Resultados Árbol de decisión en métricas de performance

Métrica	Valor obtenido
Exactitud	**0.8708**
Precisión	**0.8691**
ROC AUC	**0.8708**
Especificidad	**0.8171**

Fuente: Elaboración propia

En cuanto a la matriz de confusión, en general presenta una precisión de peor calidad al modelo GLM. Además, se observan valores elevados de falsos negativos y esto afecta negativamente al error tipo II.

Ilustración 35. Matriz de confusión - Árbol de decisión

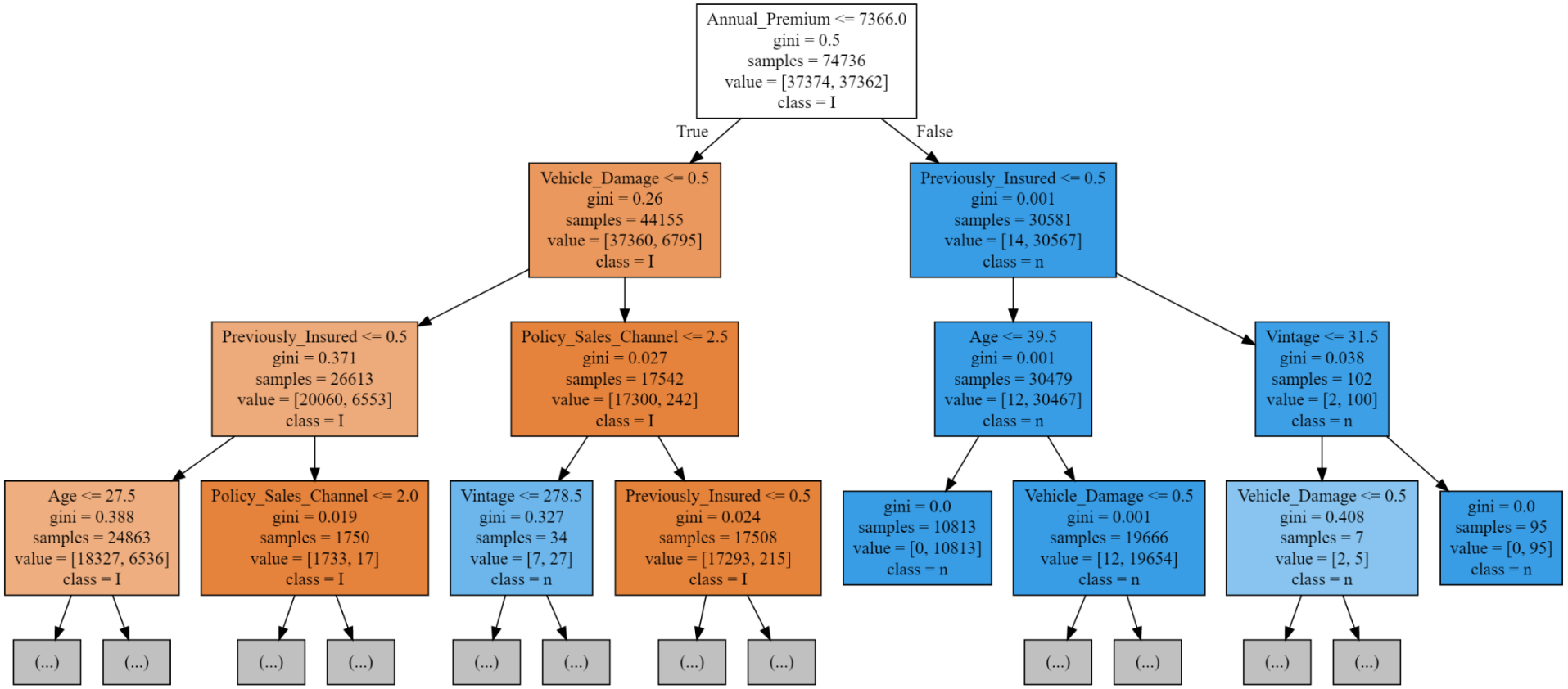


Fuente: Elaboración propia

A continuación, se presenta el árbol de decisión mostrando los 4 primeros nodos. De esta forma, se observa como el árbol realiza la toma de decisiones y los factores que escoge para realizar el corte en cada uno de los nodos. El método que se ha escogido es el indicador Gini que mide la impureza. Es importante señalar que los colores del árbol tienen un significado, a medida que aumenta la intensidad del color implica que el índice de Gini que se obtiene es bajo en comparación con el resto de los nodos. El color azul indica que la respuesta es que no está interesado en la contratación mientras que el color naranja es interesado en la contratación del seguro de auto.

El primer atributo que escoge para realizar la división del nodo raíz es la prima anual. Así, si la prima anual está por encima del valor 7366, parte derecha del árbol, implica que se consigue obtener unos índices de Gini relativamente bajos y la respuesta es no interesado en la contratación de la póliza. No obstante, si la prima anual está por debajo de ese valor, parte izquierda del árbol de decisión, se empiezan a tener en cuenta otro tipo de atributos como *Vehicle_Damage*, *Policy_Sales_Channel* y *Previously_Insured,* es decir, si el cliente ya cuenta con un seguro de auto.

Ilustración 36. Representación primeros nodos Árbol de decisión

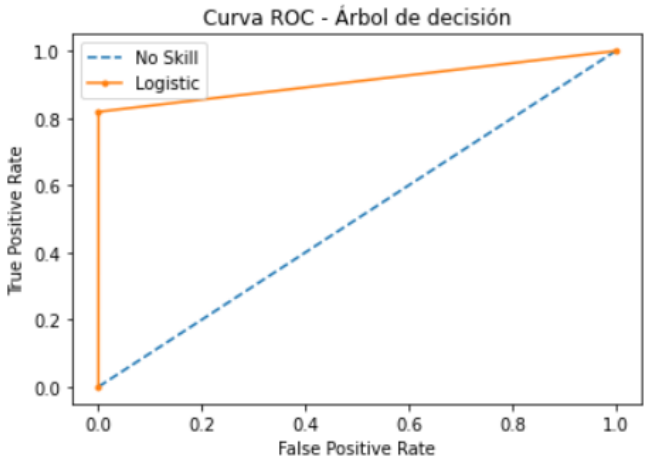


Fuente: Elaboración propia

5.3. Resultados Random Forest

Los resultados que se obtienen en la curva ROC para el modelo Random Forest es superior al Árbol de decisión, como era de esperar. Esto se debe a que Random Forest realiza una multitud de árboles para después obtener una media de estos.

Ilustración 37. Curva ROC – Random Forest



Fuente: Elaboración propia

Para el modelo Random Forest, se obtiene unos valores altos en las diferentes métricas (Tabla 7), este modelo se aproxima a la coordenada (0,1) como lo hace el modelo GLM. Por tanto, el resultado del área bajo la curva es inferior.

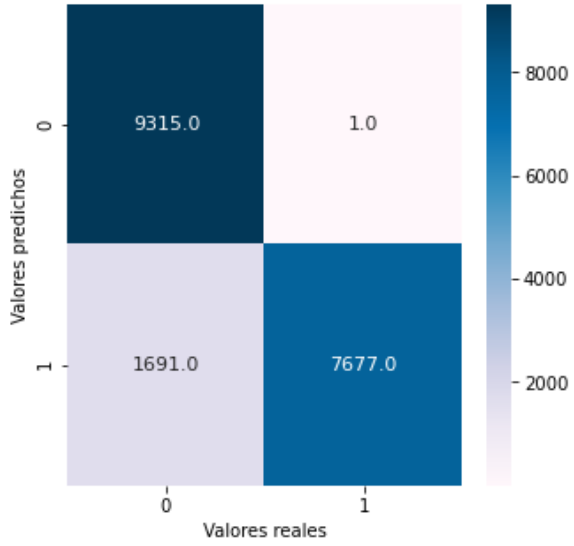
Tabla 7 Resultados Random Forest en métricas de performance

Métrica	Valor obtenido
Exactitud	**0.9121**
Precisión	**0.9927**
ROC AUC	**0.8708**
Especificidad	**0.8309**

Fuente: Elaboración propia

La matriz de confusión presenta una contradiagonal muy descompensada, no obstante, los falsos negativos es el valor más bajo de todos los modelos.

Ilustración 38. Matriz de confusión – Random Forest

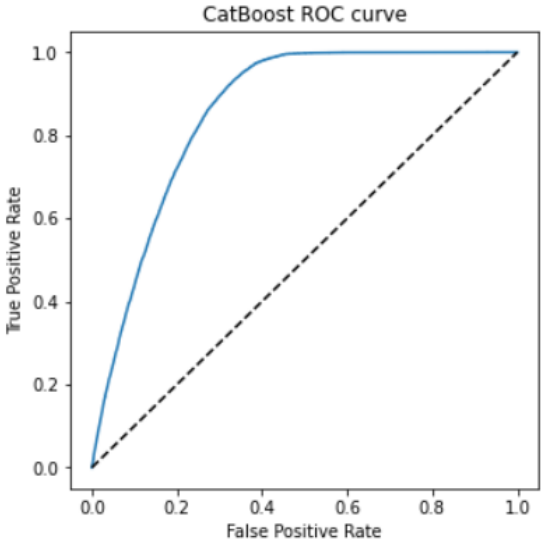


Fuente: Elaboración propia

5.4. Resultados CatBoost

El comportamiento que se observa en la curva ROC es un aprendizaje lento y no llega a alcanzar un valor alto en el eje y. Se considera que no es un buen modelo obteniendo un valor de 0.5043 en el área bajo la curva.

Ilustración 39. Curva ROC - CatBoost



Fuente: Elaboración propia

Así mismo, en la tabla 9 se observa como la especificidad es prácticamente 0, con lo cual se descarta este modelo.

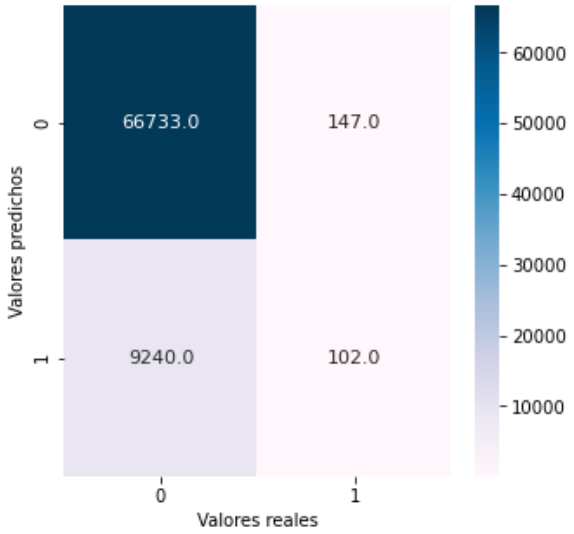
Tabla 8 Resultados CatBoost en métricas de performance

Métrica	Valor obtenido
Exactitud	**0.8768**
Precisión	**0.4096**
ROC AUC	**0.5043**
Especificidad	**0.0109**

Fuente: Elaboración propia

En la matriz de confusión se observa como los verdaderos positivos no se clasifican de forma correcta y, la mayor parte de los datos son clasificados como no interesados tomando el valor 0.

Ilustración 40 Matriz de confusión – CatBoost

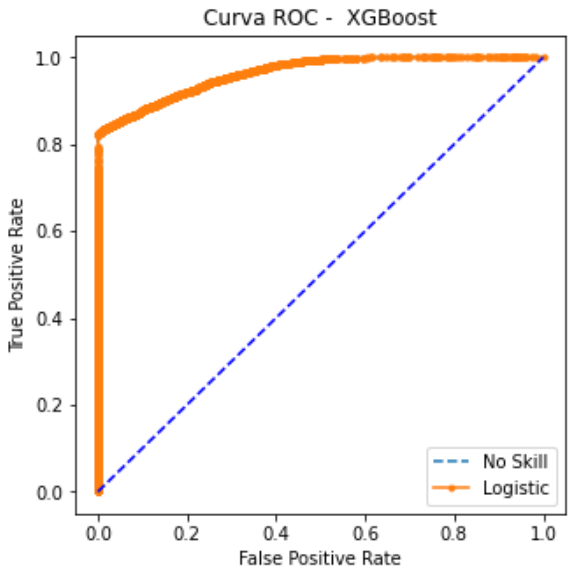


Fuente: Elaboración propia

5.5. Resultados XGBoost

Los resultados que se obtienen para el modelo XGBoost en la curva ROC son muy satisfactorios porque de forma rápida se alcanzan valores altos del ratio de verdaderos positivos.

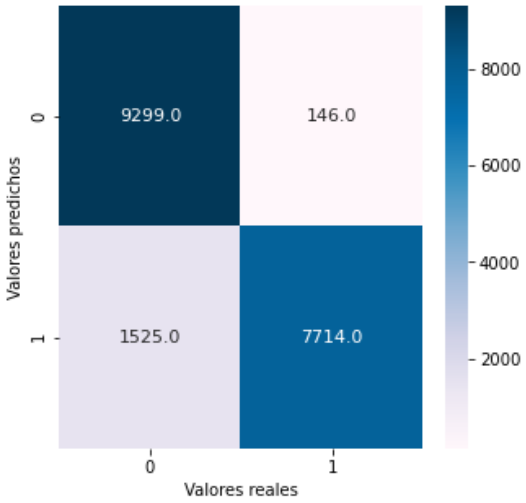
Ilustración 41 Curva ROC - XGBoost



Fuente: Elaboración propia

En cuanto a la matriz de confusión, se observa, nuevamente, una contradiagonal descompensada. No obstante, este aspecto no invalida el modelo y el número de falsos negativos es relativamente bajo.

Ilustración 42. Matriz de confusión - XGBoost



Fuente: Elaboración propia

Así, la tabla 9 presenta buenos valores para las diferentes métricas. Se considera un modelo válido y óptimo para el caso de estudio.

Tabla 9 Resultados XGBoost en métricas de performance

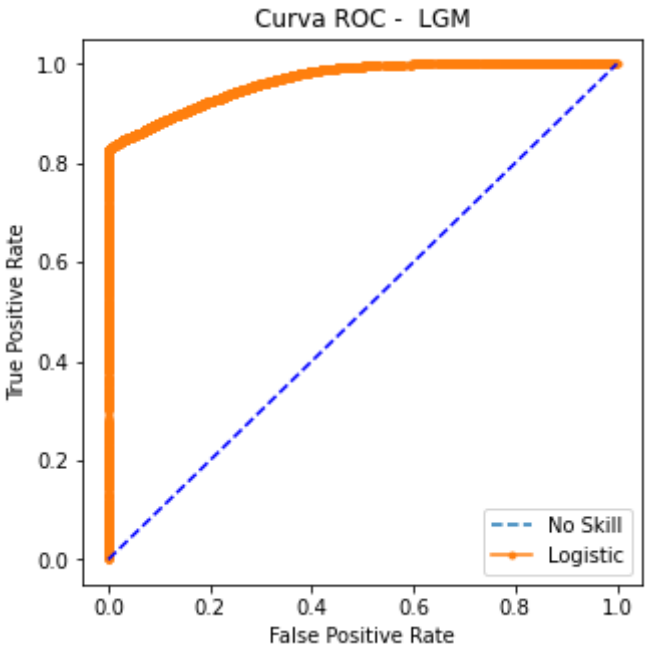
Métrica	Valor obtenido
Exactitud	**0.9105**
Precisión	**0.9814**
ROC AUC	**0.9097**
Especificidad	**0.8349**

Fuente: Elaboración propia

5.6. Resultados LGB

Para el modelo LGB la curva ROC presenta un buen comportamiento, el aprendizaje es veloz como en el caso anterior. Por tanto, se obtienen unos resultados similares al modelo XGBoost.

Ilustración 43. Curva ROC – LGB

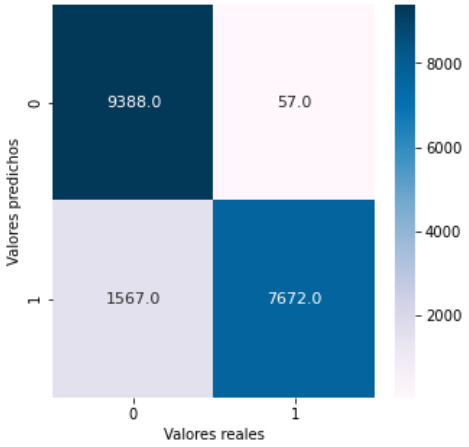


Fuente: Elaboración propia

En la matriz de confusión se observa, otra vez, la misma descompensación, no invalida el modelo. Se trata de un modelo óptimo y potente para realizar predicciones. Siguiendo la premisa detallada anteriormente, es importante que el modelo sea preciso,

exacto, alcance unos valores altos en la curva ROC y el número de falsos negativos sea bajo con el objetivo de reducir el error II comentado con anterioridad.

Ilustración 44. Matriz de confusión - LGB



Fuente: Elaboración propia

Tabla 10 Resultados LGB en métricas de performance

Métrica	Valor obtenido
Exactitud	**0.9130**
Precisión	**0.9926**
ROC AUC	**0.9121**
Especificidad	**0.8303**

Fuente: Elaboración propia

5.7. Comparación de los resultados entre los modelos

En la tabla 11 se observa una comparación de los resultados obtenidos para los diferentes modelos aplicados. Para el caso de estudio, los modelos que presentan buenos resultados son: GLM, GLM con la reducción de factores, *Random Forest, XGB* y *LGB*. Estos son los modelos que mejores resultados presentan, no obstante, entre ellos las diferencias que se observan en las métricas son de centésimas. Por tanto, modelos como el GLM con la reducción de variables resultan muy prácticos debido a la facilidad de interpretar los coeficientes del modelo mientras que en el caso de Random Forest, así como el resto de los modelos mencionados, es más difícil interpretar los resultados. No obstante, no por ello resultan no óptimos.

Así mismo, para el caso de estudio el modelo que proporciona peores resultados es CatBoost debido a que la precisión es muy baja así como el área bajo la curva. Se descarta este modelo por no considerarse óptimo.

Tabla 11 Resultados LGB en métricas de performance

Métrica	GLM	GLM reduc[17].	DT	RF	CATBoost	XGB	LGB
Exactitud	**0.9048**	**0.9117**	**0.8708**	**0.9121**	**0.8768**	**0.9105**	**0.9130**
Precisión	**0.9911**	**0.9993**	**0.8691**	**0.9927**	**0.4096**	**0.9814**	**0.9926**
ROC AUC	**0.9049**	**0.9119**	**0.8708**	**0.8708**	**0.5043**	**0.9097**	**0.9121**
Especificidad	**0.8171**	**0.8150**	**0.8171**	**0.8309**	**0.0109**	**0.8349**	**0.8303**

Fuente: Elaboración propia

[17] Resultados del GLM con reducción de variables.

CAPÍTULO 6. CONCLUSIONES

La utilización de métodos de *Machine Learning* es un fenómeno que ha ido *in crescendo* en la última década. Es importante mencionar que la aparición y utilización de este tipo de técnicas no implica la desaparición de los modelos clásicos como GLM. Las dos razones fundamentales por las que se ha escogido este tema son la importancia de la venta cruzada en el sector asegurador y la potencial aplicación práctica al mundo real.

En primer lugar, como se ha mencionado en el Capítulo 2, cada vez son más las aseguradoras que crean alianzas con compañías pertenecientes al sector bancario, sector energético u otro tipo de sectores. El objetivo que se busca cuando se cierra este tipo de alianzas suele ser incrementar la fidelización de los clientes, crear una oferta integral, aumentar el tiempo de vida medio del cliente y, para ambas partes, crear una sinergia *win-win*[18] donde ambas compañías salen beneficiosas con el acuerdo cerrado.

En segundo lugar, la posibilidad de aplicar este caso de estudio a la vida real genera mucho valor. En este sentido, cuando una compañía realiza una compra de una base de datos a un proveedor debe especificar el grado de enriquecimiento de la base de datos. Gracias a la aplicación de las técnicas de *Machine Learning*, la compañía podrá saber qué tipo de variables debe pedir en la base de datos y adquirir, de esta forma, una base de datos enriquecida con variables que realmente sí son importantes de cara a conseguir la venta cruzada. Así, con la base de datos, de clientes potenciales y enriquecida con información relevante, se podrá realizar un *scoring* o filtrado de aquellos clientes con mayor propensión a contratar el seguro.

En el Capítulo 3, se han explicado los diferentes modelos de *Machine Learning*. Cabe destacar que en este caso de estudio se han priorizado las técnicas de conjunto o *Ensembles Techniques*. Este tipo de modelos de *Machine Learning* se utilizan para reducir el error y mejorar la precisión del modelo en comparación con otro tipo de modelos más clásicos. No obstante, es importante señalar que en ciertas ocasiones estas técnicas no son elegidas porque se pierde la interpretabilidad del modelo a cambio de obtener un mejor rendimiento del modelo. Así mismo, además de aplicar técnicas de

[18] La estrategia de Marketing win-win hace referencia a aquella alianza o simplemente acuerdo puntual donde ambas partes, ya sean compañías, fuerza de ventas, comerciales, distribuidores, salen beneficiosos de la colaboración que han pactado.

conjunto, *bagging* y *boosting*, se ha aplicado el modelo clásico GLM para comparar un modelo clásico como es GLM con modelos de *Machine Learning*.

En el Capítulo 4 se ha procedido a aplicar cada uno de estos modelos. No obstante, es fundamental destacar el tratamiento de la base de datos. El procesamiento y tratamiento de la base de datos se puede considerar uno de los procesos más importantes antes de aplicar cualquier tipo de modelo. Es fundamental conocer cada una de las variables, los valores que toman, la posible existencia de valores atípicos y cómo gestionarlos, la eliminación de los valores erróneos...etc. Si esta esta del procesamiento de la base de datos no se realiza de forma correcta podría dar lugar a la invalidez del modelo.

Así mismo, es importante contar con una base de datos donde la variable, en este caso binaria, presente cierto equilibrio en la respuesta. En caso contrario, es importante aplicar una técnica de balanceado de datos porque si no el modelo que se aplicase en la práctica presentaría una ligera tendencia hacia la clase mayoritaria. Así, gracias a la aplicación de la técnica de *undersampling* aplicada se obtienen unos resultados más satisfactorios y el modelo no está sesgado hacia la clase mayoritaria.

En el Capítulo 5 se han presentado los resultados obtenidos. En general, son muy satisfactorios, se observan valores muy altos en las métricas estudiadas. Esto implica que la mayor parte de los modelos estudiados se consideran óptimos y dependerá de la utilización y juicio experto del analista. Más bien se trata de un avance multidisciplinar y será, bajo el criterio de la persona experta en el caso de estudio, quien opte por un tipo de modelo u otro.

Concretamente, como se ha mencionado, uno de los aspectos claves de las medidas de *performance* ha sido minimizar el error tipo II. Así, la métrica especificidad refleja el nivel de error tipo II que se comete. Los modelos que representan un mejor comportamiento en esta métrica son GLM reducido, *Random Forest, XGBoost* y *LGB*, presentan valores más altos en especificidad. Así mismo, en cuanto al área bajo la curva ROC que mide cuantas observaciones han sido clasificadas de forma correcta los modelos que presentan un comportamiento más acertado son GLM con reducción de variables y LGB.

Es importante destacar para el caso de estudio, el modelo *CatBoost* no presenta buenos resultados debido a que el área bajo la curva es prácticamente la misma área que se obtendría si se debiese a la propia aleatoriedad, es decir, un área de 0.5. Por tanto, este modelo se descarta, no es un modelo aceptable, es importante recalcar que se descarta para este caso concreto de estudio.

Es fundamental tener en cuenta que los modelos GLM son modelos clásicos fundamentados en ecuaciones matemáticas que pueden ser explicadas y justificadas frente al regulador. Así, los algoritmos de *Machine Learning* son modelos modernos que no presentan una justificación tan clara como los modelos GLM. Por tanto, será el analista quien determine el modelo o los modelos más convenientes de aplicación en función de su juicio experto y siempre velando por un sentido de negocio.

Para finalizar y de cara a realizar próximas investigaciones, es importante mencionar la matriz fuga-valor que poseen ciertas empresas. Esta matriz establece, por un lado, en un eje, la probabilidad de que un cliente abandone el contrato o también llamada probabilidad de fuga y, por otro lado, en el eje restante establece el valor que aporta ese cliente a la compañía, pudiendo cuantificar este valor con diferentes indicadores como puede ser la rentabilidad que se obtiene de dicho cliente.

Así, de esta forma la compañía a la hora de tomar decisiones no establecerá políticas comerciales de retención para aquellos clientes que aportan poco valor y tienen una probabilidad de fuga alta. Por el contrario, establecerá acciones comerciales dirigidas a aquellos clientes que aporten un valor alto y posean una probabilidad alta de fuga con el objetivo de retener estos clientes, por ejemplo, mediante descuentos, regalos u otras estrategias de fidelización.

Existe una tendencia actual de valorar de forma conjunta la probabilidad de fuga con el precio y la sensibilidad de precios de ese mercado para ese riesgo. Así, en mercados donde exista una alta sensibilidad al precio, podría ser suficiente con ofrecer un descuento comercial porque el cliente valorará en gran cantidad este descuento. No obstante, aquellos mercados donde la sensibilidad al precio sea baja, no será suficiente con ofrecer un mero descuento, será necesario realizar otro tipo de acciones de retención y fidelización.

Finalmente, cuando se ofrecen unos descuentos muy altos con el objetivo de ser competitivos en precios en el mercado, es importante tener en cuenta el incremento de riesgo que esto conlleva. Los niveles de Solvencia de la compañía podrían verse afectados y sería necesario dotar una mayor cantidad de capital en las provisiones técnicas, que implica una disminución de la rentabilidad ajustada al coste del capital del accionista. Por ello, es importante mantener el equilibrio óptimo entre rentabilidad del contrato, elasticidad al precio de mercado y carga de capital.

BIBLIOGRAFÍA

Ardabili, S., Mosavi, A., & Várkonyi-Kóczy, A. R. (2019). Advances in machine learning modeling reviewing hybrid and ensemble methods. Paper presented at the International Conference on Global Research and Education, 215-227.

Brijain, M., Patel, R., Kushik, M., & Rana, K. (2014). A survey on decision tree algorithm for classification.

Dietterich, T. G. (2000). Ensemble methods in machine learning. Paper presented at the International Workshop on Multiple Classifier Systems, 1-15.

Fürnkranz, J. (2002). Pairwise classification as an ensemble technique. Paper presented at the European Conference on Machine Learning, 97-110.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Machine learning basics. Deep Learning, 1, 98-164.

Harrison, T., & Ansell, J. (2002). Customer retention in the insurance industry: Using survival analysis to predict cross-selling opportunities. Journal of Financial Services Marketing, 6(3), 229-239.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science (New York, N.Y.), 349(6245), 255-260. doi:10.1126/science.aaa8415 [doi]

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 160(1), 3-24.

Mani, I., & Zhang, I. (2003). kNN approach to unbalanced data distributions: A case study involving information extraction. Paper presented at the Proceedings of Workshop on Learning from Imbalanced Datasets, , 126

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning MIT press.

Oza, N. C. (2005). Online bagging and boosting. Paper presented at the 2005 IEEE International Conference on Systems, Man and Cybernetics, , 3 2340-2345.

Polikar, R. (2006). Ensemble based systems in decision making. IEEE Circuits and Systems Magazine, 6(3), 21-45.

Roncancio Avila, M. N., Reina Moreno, D. K., Hualpa Zuniga, A. M., Felizzola Jimenez, H. A., & Arango Londono, C. A. (2017). Using learning curves and confidence intervals in a time study for the calculation of standard times. Inge Cuc, 13(2), 18-27.

Roy, R., & George, K. T. (2017). Detecting insurance claims fraud using machine learning techniques. Paper presented at the 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), 1-6.

Saugat Bhattarai.What is gradient descent in machine learning? (22 Junio, 2018 ed.)

Serrano, C. G. C. (2016). Fidelización y rentabilización de usuarios de seguros todo riesgo de vehículos por medio de la venta cruzada y la venta escalonada. un enfoque promocional para la industria aseguradora. Universidad & Empresa, 18(30), 143-157.

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

Soydaner, D. (2020). A comparison of optimization algorithms for deep learning. ArXiv Preprint arXiv:2007.14166,

Zboja, J. J., & Hartline, M. D. (2012). An examination of high-frequency cross-selling. Journal of Relationship Marketing, 11(1), 41-55.

Zhang, C., & Ma, Y. (2012). Ensemble machine learning: Methods and applications Springer.

ANEXO A. CÓDIGO GRÁFICO ÁRBOL DE DECISIÓN

```
digraph Tree {
node [shape=box, style="filled", color="black"] ;
0 [label="Annual_Premium <= 7366.0\ngini = 0.5\nsamples = 74736\nvalue = [37374, 37362]\nclass = I", fillcolor="#ffffff"] ;
1 [label="Vehicle_Damage <= 0.5\ngini = 0.26\nsamples = 44155\nvalue = [37360, 6795]\nclass = I", fillcolor="#ea985d"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
2 [label="Previously_Insured <= 0.5\ngini = 0.371\nsamples = 26613\nvalue = [20060, 6553]\nclass = I", fillcolor="#edaa7a"] ;
1 -> 2 ;
3 [label="Age <= 27.5\ngini = 0.388\nsamples = 24863\nvalue = [18327, 6536]\nclass = I", fillcolor="#eeae80"] ;
2 -> 3 ;
4 [label="(...)", fillcolor="#C0C0C0"] ;
3 -> 4 ;
1891 [label="(...)", fillcolor="#C0C0C0"] ;
3 -> 1891 ;
15812 [label="Policy_Sales_Channel <= 2.0\ngini = 0.019\nsamples = 1750\nvalue = [1733, 17]\nclass = I", fillcolor="#e5823b"] ;
2 -> 15812 ;
15813 [label="(...)", fillcolor="#C0C0C0"] ;
15812 -> 15813 ;
15814 [label="(...)", fillcolor="#C0C0C0"] ;
15812 -> 15814 ;
15913 [label="Policy_Sales_Channel <= 2.5\ngini = 0.027\nsamples = 17542\nvalue = [17300, 242]\nclass = I", fillcolor="#e5833c"] ;
1 -> 15913 ;
15914 [label="Vintage <= 278.5\ngini = 0.327\nsamples = 34\nvalue = [7, 27]\nclass = n", fillcolor="#6cb6ec"] ;
15913 -> 15914 ;
15915 [label="(...)", fillcolor="#C0C0C0"] ;
```

15914 -> 15915 ;
15926 [label="(...)", fillcolor="#C0C0C0"] ;
15914 -> 15926 ;
15929 [label="Previously_Insured <= 0.5\ngini = 0.024\nsamples = 17508\nvalue = [17293, 215]\nclass = I", fillcolor="#e5833b"] ;
15913 -> 15929 ;
15930 [label="(...)", fillcolor="#C0C0C0"] ;
15929 -> 15930 ;
16709 [label="(...)", fillcolor="#C0C0C0"] ;
15929 -> 16709 ;
16766 [label="Previously_Insured <= 0.5\ngini = 0.001\nsamples = 30581\nvalue = [14, 30567]\nclass = n", fillcolor="#399de5"] ;
0 -> 16766 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
16767 [label="Age <= 39.5\ngini = 0.001\nsamples = 30479\nvalue = [12, 30467]\nclass = n", fillcolor="#399de5"] ;
16766 -> 16767 ;
16768 [label="gini = 0.0\nsamples = 10813\nvalue = [0, 10813]\nclass = n", fillcolor="#399de5"] ;
16767 -> 16768 ;
16769 [label="Vehicle_Damage <= 0.5\ngini = 0.001\nsamples = 19666\nvalue = [12, 19654]\nclass = n", fillcolor="#399de5"] ;
16767 -> 16769 ;
16770 [label="(...)", fillcolor="#C0C0C0"] ;
16769 -> 16770 ;
16855 [label="(...)", fillcolor="#C0C0C0"] ;
16769 -> 16855 ;
16862 [label="Vintage <= 31.5\ngini = 0.038\nsamples = 102\nvalue = [2, 100]\nclass = n", fillcolor="#3d9fe6"] ;
16766 -> 16862 ;
16863 [label="Vehicle_Damage <= 0.5\ngini = 0.408\nsamples = 7\nvalue = [2, 5]\nclass = n", fillcolor="#88c4ef"] ;
16862 -> 16863 ;
16864 [label="(...)", fillcolor="#C0C0C0"] ;

16863 -> 16864 ;
16865 [label="(...)", fillcolor="#C0C0C0"] ;
16863 -> 16865 ;
16866 [label="gini = 0.0\nsamples = 95\nvalue = [0, 95]\nclass = n", fillcolor="#399de5"] ;
16862 -> 16866 ;
}

ANEXO B. CÓDIGO PYTHON

Script 1. Limpieza de datos y EDA

June 25, 2021

Nombre: Joanna Lempicka

Trabajo Final de Master: Predicción de Cross-selling con Machine Learning

Universidad Carlos III de Madrid

0.1 Librerias

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.pyplot import figure
```

1 LIMPIEZA DE DATOS

[ ]:
```
#Carga los datos
basecompleta=pd.read_csv('C:/Users/ASKA/Desktop/TFM/basecompleta.csv')

#Se eliminan los na
basecompleta = basecompleta.dropna()
#basecompleta.head(5)
```

[ ]:
```
# Se cambian los valores por variables binarias

Gender  = {'Male': 1,'Female': 0}
Vehicle_Damage  = {'Yes': 0,'No': 1}
Vehicle_Age = {'< 1 Year':0,'1-2 Year':1,'> 2 Years':2}

basecompleta.Gender = [Gender[item] for item in basecompleta.Gender]
basecompleta.Vehicle_Damage = [Vehicle_Damage[item] for item in basecompleta.
 ↪Vehicle_Damage]
basecompleta.Vehicle_Age = [Vehicle_Age[item] for item in basecompleta.
 ↪Vehicle_Age]

#basecompleta.head(5)
```

[ ]:
```
#basecompleta.columns
```

[ ]:
```
# Ver edades menores a 18 años, vemos que no hay

#basecompleta[basecompleta['Age'] <= 18]
```

[ ]:
```
# Ver personas que no tienen el carnet de conducir

#basecompleta[basecompleta['Driving_License'] < 1]
```

[ ]:
```
# Se eliminan las polizas que no tienen carnet de conducir

basecompleta = basecompleta[basecompleta['Age'] >= 18]
#basecompleta.head(5)
```

[ ]:
```
basecompleta.to_csv(r'C:/Users/ASKA/Desktop/TFM/basecompleta_1.csv', index =␣
↪False)
```

2 ANÁLISIS EXPLORATORIO DE LOS DATOS

[ ]:
```
figure(figsize=(4, 3), dpi=70)
sns.countplot(basecompleta.Response)
```

[ ]:
```
basecompleta.Response.value_counts()
```

[ ]:
```
figure(figsize=(4, 3), dpi=70)
sns.distplot(basecompleta.Age)
```

[ ]:
```
figure(figsize=(4, 3), dpi=70)
sns.scatterplot(x=basecompleta['Age'],y=basecompleta['Annual_Premium'])
```

[ ]:
```
figure(figsize=(4, 3), dpi=70)
sns.countplot(basecompleta.Gender)
```

[ ]:
```
figure(figsize=(4, 3), dpi=70)
df=basecompleta.groupby(['Gender','Response'])['id'].count().to_frame().
↪rename(columns={'id':'count'}).reset_index()
```

[ ]:
```
figure(figsize=(4, 3), dpi=70)
g = sns.catplot(x="Gender", y="count",col="Response",
                data=df, kind="bar",
                height=4, aspect=.7);
```

[ ]:
```
df=basecompleta.groupby(['Gender'])['Driving_License'].count().to_frame().
↪reset_index()
```

[ ]:
```
figure(figsize=(4, 3), dpi=70)
sns.catplot(x="Gender", y="Driving_License", data=df, kind="bar");
```