Modelo Predictivo de Siniestralidad para Seguros de Asistencia en Viajes

Juan Pablo Botero Suaza
Máster Universitario Ciencia de Datos
Análisis y predicción de la siniestralidad en seguros

Jorge Segura Gisbert
Albert Solé Ribalta

01/2021

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Modelo Predictivo de Siniestralidad para Seguros de Asistencia en Viajes*
Nombre del autor:	*Juan Pablo Botero Suaza*
Nombre del consultor/a:	*Jorge Segura Gisbert*
Nombre del PRA:	*Albert Solé Ribalta*
Fecha de entrega (mm/aaaa):	01/2021
Titulación:	*Máster Universitario Ciencia de Datos*
Área del Trabajo Final:	*Análisis y predicción de la siniestralidad en seguros*
Idioma del trabajo:	*Español*
Palabras clave	*aprendizaje automático, siniestralidad en seguros, ciencia de datos*

Resumen del Trabajo:

La siniestralidad, entendida como la frecuencia en la que se materializan los riesgos cubiertos por una determinada póliza, es una variable clave en la industria aseguradora, pues le permite estudiar diversos factores como: la capacidad de cobertura asistencial en diferentes zonas, el desembolso de indemnizaciones, el valor de las primas del seguro, el cálculo de reservas, entre otros. Es por ello, que el habilitar capacidades de análisis de la información a través de modelos predictivos, se convierte en una herramienta de gran valor para la organización en la toma de decisiones bajo este contexto.

El presente trabajo tiene como principal objetivo, desarrollar un modelo analítico predictivo que permita correlacionar las diferentes variables asociadas a la siniestralidad de los seguros de asistencia en viajes, utilizando métodos de clasificación en el marco del aprendizaje automático supervisado. En específico, el modelo pronosticará a partir de dichas variables, la ocurrencia o no de una reclamación sobre el seguro por parte de los clientes, ante la materialización de riesgos amparados dentro de las coberturas de este tipo de seguros, que pueden ir desde la pérdida o retraso de equipaje, la cancelación de vuelo/ viaje, siniestros sobre al alquiler de vehículos, hasta la atención médica presencial/remota como producto de un accidente o enfermedad durante el viaje.

Abstract:

The loss ratio, understood as the frequency in which the risks covered by a given policy are materialized, is a key variable in the insurance industry, as it allows it to study various factors such as: the capacity to cover healthcare in different areas, the disbursement of compensation, the value of insurance premiums, the calculation of reserves, among others. That is why enabling information analysis capabilities through predictive models becomes a tool of great value for the organization in making decisions in this context.

The main objective of this work is to develop a predictive analytical model that allows correlating the different variables associated with the accident rate of travel assistance insurance, using classification methods within the framework of supervised machine learning. Specifically, the model will predict from these variables, the occurrence or not of a claim on the policy by the clients, before the materialization of risks covered within the coverage of this type of policy, which can range from loss or luggage delay, flight / trip cancellation, car rental claims, even face-to-face / remote healthcare as a result of an accident or illness during the trip.

Índice

Lista de figuras

Lista de tablas

1. Introducción

1.1 Contexto y justificación

En la actualidad, la industria aseguradora, y en especial, las soluciones de negocio correspondientes a seguros de asistencia en viajes se han visto seriamente comprometidas en cuanto a su rentabilidad y sostenibilidad se refiere, como producto de la emergencia sanitaria derivada de la Covid-19. La cancelación de una gran cantidad de vuelos en las aerolíneas, la cancelación de reservas hoteleras, las restricciones de movilidad entre zonas geográficas debido a la declaración de cuarentenas y el temor al contagio por parte de los viajeros han llevado a este sector a sufrir innumerables pérdidas económicas ocasionadas por el desembolso de indemnizaciones y una reducción considerable sobre las ventas para este tipo de seguros.

Si bien la apertura económica global se ha dado de forma parcial y gradual en diferentes regiones del mundo, la reactivación de este tipo de soluciones de seguro asistencial, tomará un tiempo considerable, los viajes de negocio y turismo requerirán de estrategias disruptivas por parte de las aseguradoras, incorporar soluciones como la telemedicina, la ampliación de la cobertura médica en sitio y el diseño de soluciones asistenciales a la medida de acuerdo al lugar de destino, serán retos por afrontar en los próximos meses. El cómo abordar estas situaciones no previstas de forma oportuna y adaptarse rápidamente a las nuevas necesidades del mercado, será un diferenciador para las aseguradoras, donde sólo la experiencia en el sector y el conocimiento de la dinámica del mercado será suficiente para resolverlos. Es allí donde las soluciones orientadas al estudio holístico y sistémico de los datos, como el aprendizaje automático, se convierten en una gran herramienta para el desarrollo conjunto de las estrategias de negocio y la toma de decisiones en ecosistemas sometidos a constantes cambios.

Con el fin de relacionar un caso práctico acorde a los escenarios expuestos anteriormente, en el contexto del aprendizaje automático, el presente estudio desarrolla un modelo predictivo utilizando técnicas de analítica y ciencia de datos basadas en el lenguaje de programación *python*, una plataforma ampliamente utilizada junto con *R* en el mundo empresarial y académico afín a este tipo de tareas. Su enfoque principal se centra en el entendimiento y estudio de la siniestralidad para el seguro de asistencia en viajes, bajo la óptica de la ocurrencia o no de una reclamación efectuada sobre el seguro.

Como fuente de información para el análisis se utiliza la base de datos abierta “Travel Insurance” provista por *kaggle*, la cual puede descargarse desde el siguiente enlace:

https://www.kaggle.com/mhdzahier/travel-insurance

En ella se relacionan características como el tipo de producto adquirido por el cliente para su viaje, el género del cliente, el lugar de destino del viaje, la venta neta por parte de la agencia que ofrece el seguro con su respectiva comisión, el canal de distribución utilizado por la agencia, entre otras…, y por supuesto la variable objeto del análisis, la ocurrencia o no de una reclamación sobre el seguro.

Adicionalmente, como parte de la investigación se realiza un análisis comparativo de la precisión de la predicción del modelo analítico obtenido utilizando técnicas formales del aprendizaje automático en contraste a las nuevas técnicas orientadas a Auto ML, un concepto que ha tomado un gran auge en los últimos años dentro la comunidad analítica mundial sobre el cual se entra en detalle en los próximos apartados.

1.2 Objetivos

- Identificar los rasgos característicos que dan lugar a la reclamación de un seguro de asistencia en viajes, a partir de un análisis exploratorio cuantitativo y visual de la fuente de datos “Travel Insurance”.

- Desarrollar un modelo analítico predictivo empleando técnicas del aprendizaje automático supervisado que permita correlacionar las variables más significativas de la fuente de datos “Travel Insurance”, tomada como caso práctico de referencia en el estudio de la siniestralidad para seguros de asistencia en viajes.

- Refinar el modelo analítico aplicando técnicas de optimización de hiper parámetros que permitan mejorar su capacidad de predicción.

- Evaluar la calidad para hacer pronósticos del modelo analítico refinado con el fin de valorar su idoneidad sobre el contexto de aplicación aquí expuesto.

- Comparar la calidad de predicción del modelo analítico construido utilizando las técnicas estándar de aprendizaje automático en contraste con los resultados obtenidos al aplicar técnicas de Auto ML sobre el mismo conjunto de datos.

1.3 Enfoque y metodología planteada

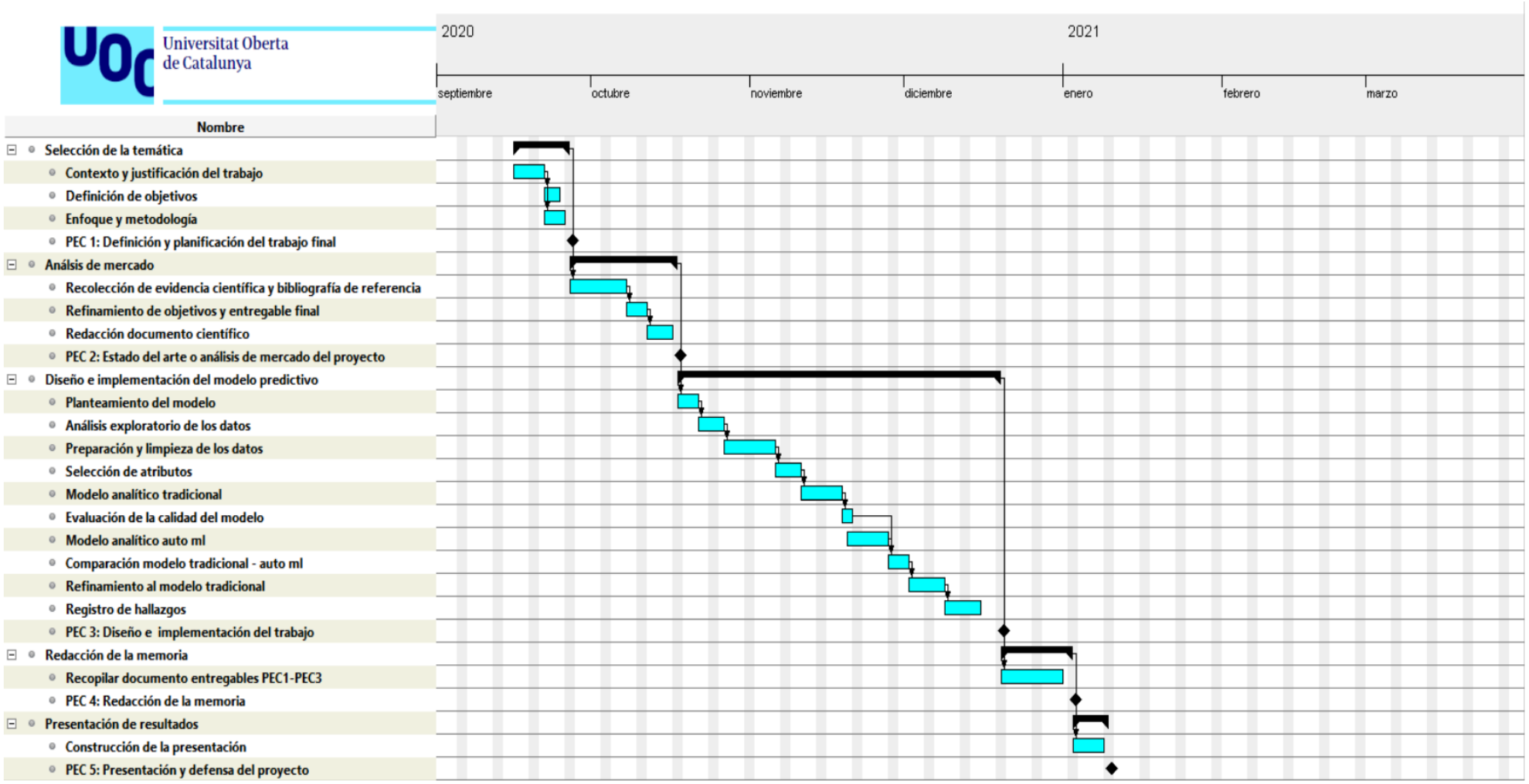
El estudio de la siniestralidad para seguros de asistencia en viajes propuesto está sustentado en la implementación de un modelo analítico predictivo que sigue las fases de la metodología clásica del aprendizaje automático supervisado sobre un conjunto de datos previamente definido:

- Análisis exploratorio.
- Preprocesamiento y limpieza.
- Selección de atributos.
- Elección del modelo a utilizar
- Ajustar el modelo sobre los datos de entrenamiento.
- Evaluar la calidad de las predicciones del modelo sobre los datos de prueba.
- Optimizar el rendimiento del modelo y evaluación de calidad
- Selección del modelo óptimo.

La elección de la estrategia de análisis a través de aprendizaje automático supervisado corresponde a la naturaleza misma del caso de estudio, es decir, a partir de un conjunto de características conocidas del conjunto de datos se quiere inferir el resultado de una variable objetivo o de salida, que también es conocida, en este caso la variable objetivo corresponde a si una reclamación se hace o no efectiva sobre el seguro de asistencia en viajes utilizando las demás variables existentes, y sobre la cual el modelo analítico hace sus pronósticos con datos no conocidos.

Dada la naturaleza binaria no continua de la variable objetivo (existe o no reclamación), su correspondencia es intrínseca a una variable categórica, en cuyo caso los modelos de predicción supervisados son conducidos por algoritmos y técnicas de *clasificación* para su diseño e implementación.

1.4 Planificación



1.5 Sumario de productos obtenidos

Como parte de la investigación realizada en el presente trabajo se obtienen los siguientes productos o entregables:

- Un modelo predictivo de clasificación, que tiene por variable objetivo, la ocurrencia de reclamación (SI/NO) sobre un seguro de asistencia en viajes.

- Un modelo predictivo de clasificación alterno empleando técnicas de Auto ML, con propósitos ilustrativos y académicos, introductorio a las nuevas tendencias en el campo del aprendizaje automático.

1.6 Descripción de los capítulos

En los siguientes capítulos de la investigación se hace referencia a aspectos importantes para el desarrollo de la propuesta, de una forma integral y coherente, correspondiente a la formalidad de los trabajos académicos de orden científico:

Estado del arte o análisis de mercado, para el caso práctico introducido en la sección 1.1, con referencias a bibliografía científica o trabajos previos sobre la aplicación del aprendizaje automático a la industria aseguradora y las nuevas tendencias en la comunidad analítica como Auto ML y MLaaS.

Diseño e implementación del modelo predictivo, haciendo especial énfasis en las etapas de la metodología propuesta en la sección 1.3 incorporando el tratamiento de datos desbalanceados a través de técnicas de re muestreo y la parametrización de algunos de los algoritmos de clasificación, para ello, se toma como base de referencia para el análisis, un modelo de regresión logística, el cual se contrasta con los resultados obtenidos de la aplicación de otras técnicas como la combinación de clasificadores y algunos *framework* open source construidos bajo el concepto de AutoML para python. Además, se presenta un refinamiento sobre el modelo inicial de regresión logística empleando optimización por hiper parámetros.

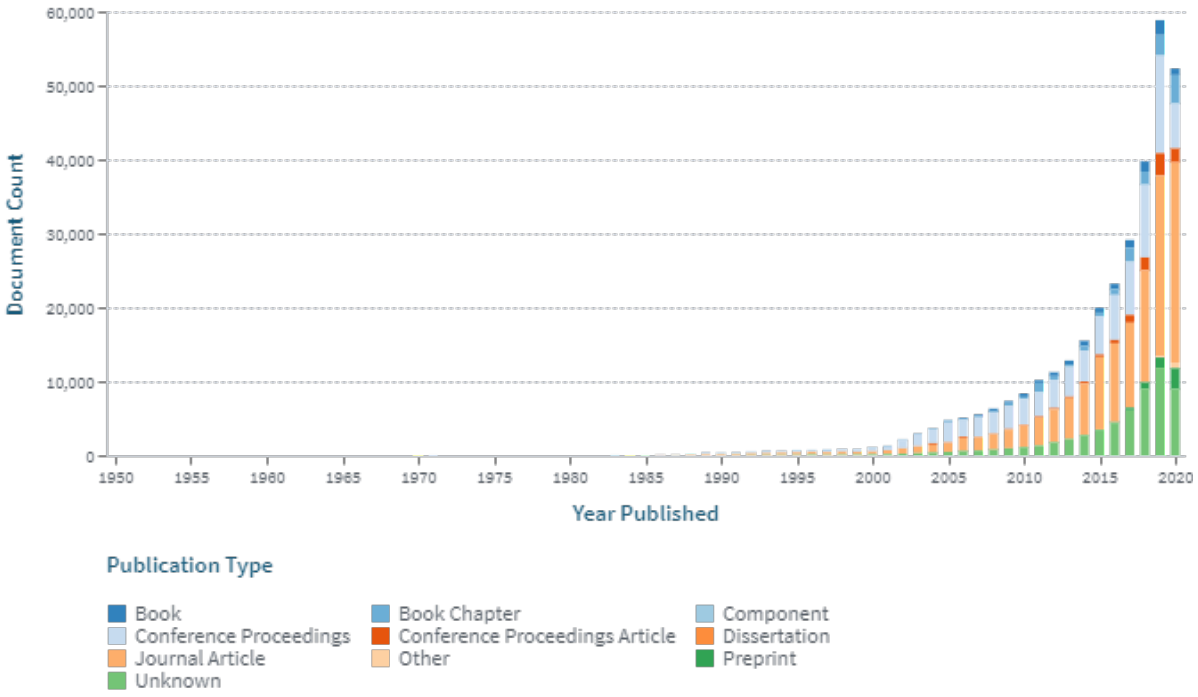
Por último, se presentan las conclusiones del trabajo de investigación y posibles trabajos futuros afines a la temática desarrollada.

2. Estado del arte

Es común escuchar, hoy en día, un sinnúmero de conceptos generalmente expresados como extranjerismos; esto es aún más común en el contexto de ejes temáticos cercanos a la tecnología y la innovación.

Uno de estos conceptos, conocido como Aprendizaje Automático o "Machine Learning", ha atraído durante las últimas décadas, la atención de académicos y ejecutivos alrededor del mundo entorno a las diversas formas de entender un fenómeno que se encuentra en constante evolución y desarrollo: la capacidad de un sistema para aprender una serie de tareas de forma automática.

Figura 1: Número de publicaciones asociadas a Machine Learning



Fuente: Motor de búsqueda de Lens.Org

Para introducir el concepto se plasmará un recuento breve de los inicios del aprendizaje automático, los cuales se encuentran en los años 50s, cuando Arthur Samuel, pionero en el campo de los juegos informáticos y la Inteligencia Artificial, escribió el primer programa de aprendizaje informático (IT, 2018). Con anterioridad, y tal vez el primer concepto alrededor del aprendizaje automático, apareció con la determinación del llamado "Test de Turing" que fue pensado para determinar si, efectivamente, una entidad no humana es realmente inteligente. La prueba fue construida alrededor de la observación: la entidad (computador en este caso) debe hacer que una persona no pueda distinguir a ciencia cierta si se está comunicando con otro ser humano o con una máquina. (Todo BI, 2019)

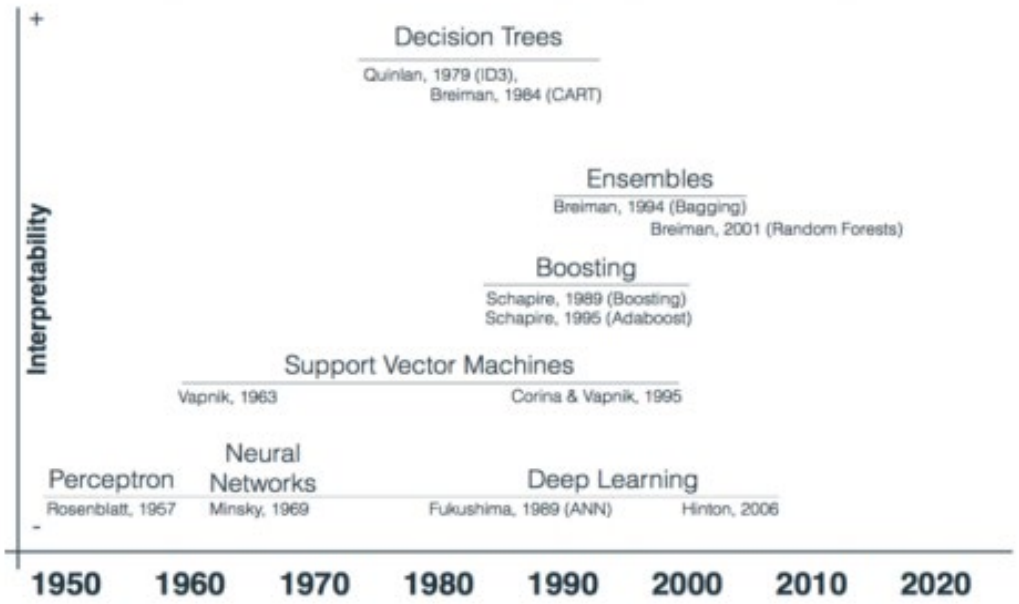
El aprendizaje automático, cabe mencionar, se desprende del concepto anteriormente mencionado conocido como Inteligencia Artificial (también llamada inteligencia computacional). La inteligencia artificial es la inteligencia exhibida por máquinas y algoritmos que emulan las características o capacidades de la inteligencia humana, como aprender y resolver problemas (Universidad de Alcalá, 2019) y tiene base en dos características primordiales de la capacidad cognitiva humana: el razonamiento y la conducta.

Aún en la década de los 50s, Frank Rosenblatt, ideó y acuño el concepto de "Perceptron", tecnología que asemeja al cerebro humano, ya que se trataba de una clase de red neuronal. En sus principios, el Perceptron conectaba una red de puntos o nodos donde se toman decisiones simples que se unen luego mediante un análisis global para resolver problemas complejos.

El aprendizaje automático, entonces, nació como una idea, una aplicación ambiciosa de la Inteligencia Artificial. Para ser más exactos, fue una subdisciplina de esta última, nacida de las ciencias de la computación y las neurociencias. (Kent, 2019).

A comienzos de la década 80s, Gerald Dejong plantea el concepto "Aprendizaje Basado en Explicación" (EBL, por sus siglas en inglés). Se trata de un proceso en el que la computadora analiza datos de entrenamiento (suministrados para esta función) y crea luego una regla general que se puede seguir para descartar datos que no se ajusten a la muestra inicialmente usada para el entrenamiento. Fue entonces en esta década que el Aprendizaje Automático se separó de la disciplina de la estadística y se convirtió en un campo de estudio aparte (Cetinsoy, 2016)

Figura 2: Evolución de los algortimos de Aprendizaje Automático



Fuente: Cetinsoy 2016

Lo que esta rama pretendía estudiar era el reconocimiento de patrones (en los procesos de ingeniería, matemáticas, ciencias de la salud, computación, etc.) y el cómo podría darse el aprendizaje por parte de las computadoras.

Sucedió entonces, con el paso de los años que, el Aprendizaje Automático comenzó a enfocarse en diferentes asuntos, tales como el razonamiento probabilístico, la investigación basada en la estadística, la recuperación de información, etc. y continuó profundizando cada vez más en el reconocimiento de patrones, tema bajo el cual se suelen situar hoy en día las aplicaciones de la mencionada tecnología.

En manera resumida, el Aprendizaje Automático es un campo de las ciencias de la computación que, de acuerdo a Arthur Samuel (pionero en el campo de los juegos informáticos y la inteligencia artificial), le da a las computadoras la habilidad de aprender sin ser explícitamente programadas, es la idea de que existen algoritmos que pueden entregar hallazgos o conclusiones relevantes obtenidas a partir de un conjunto de datos, sin que el ser humano tenga que escribir una gran cantidad de instrucciones o algoritmos para llegar a estos.

Por lo general, hoy en día, un proceso de implementación de Aprendizaje Automático consta de los siguientes pasos: (1) adquisición de datos, (2) preprocesamiento de estos datos, (3) extracción de características, (4) selección de características significativas y (5) materialización del aprendizaje supervisado, no supervisado o reforzado.

Las técnicas de Aprendizaje Automático se utilizan comúnmente para aprender de los datos y lograr una Inteligencia Artificial débil (IA débil se define como la inteligencia artificial racional que se centra típicamente en una tarea estrecha o limitada). El Aprendizaje Automático implica el estudio científico de modelos estadísticos y algoritmos que pueden aprender progresivamente de los datos y lograr un rendimiento adecuada en una tarea específica. (Rahul Kumar Sevakula, 2020).

Este tipo de aprendizaje puede clasificarse en tres grandes grupos:

1. **Aprendizaje supervisado**: depende de datos previamente etiquetados, como podría ser el que una computadora logre distinguir (separar) imágenes de automóviles, de las de aviones. Para esto, lo normal es que estas etiquetas o rótulos sean colocadas por seres humanos para asegurar efectividad y calidad de los datos.

2. **Aprendizaje no supervisado**: en esta categoría lo que sucede es que al algoritmo se le despoja de cualquier etiqueta, de modo que no cuenta con ninguna indicación previa. En cambio, se le provee de una enorme cantidad de datos con las características propias de un objeto para que pueda determinar qué es, a partir de la información recopilada.

3. **Aprendizaje reforzado**: en este caso particular, el aprendizaje se basa en el refuerzo. La máquina es capaz de aprender con base a pruebas y errores en un número de diversas situaciones.

Es importante también, para nuestro caso, mencionar que no se debe confundir el Aprendizaje Automático con otras ramas o subramas de la Inteligencia Artificial. En años recientes se produjo una explosión en el uso del Aprendizaje Automático, debido a que Geoffrey Hinton crea el término “Deep Learning” o Aprendizaje Profundo, con el que se explican nuevas arquitecturas (formas de ordenar o construir) sistemas de Redes Neuronales que permiten a las computadoras “ver” y distinguir objetos y texto dentro de imágenes y videos.

Es también importante acotar la definición de Inteligencia Artificial en aras de separarla del concepto en nuestro caso estudiado. Esta se define como el estudio de agentes inteligentes, que pueden percibir el entorno y actuar de forma inteligente como lo hacen los humanos. La Inteligencia Artificial puede clasificarse como fuerte o débil (ya explicada con anterioridad). Se dice que las máquinas que pueden actuar de una manera inteligente (pensamiento simulado) poseen una Inteligencia Artificial débil, y se dice que las máquinas que realmente pueden pensar poseen una IA fuerte. (Russell SJ, 2016). En las aplicaciones actuales, la mayoría de los investigadores en torno a la Inteligencia Artificial se dedican a implementar una de características “débiles” para automatizar tareas específicas, generalmente de baja complejidad.

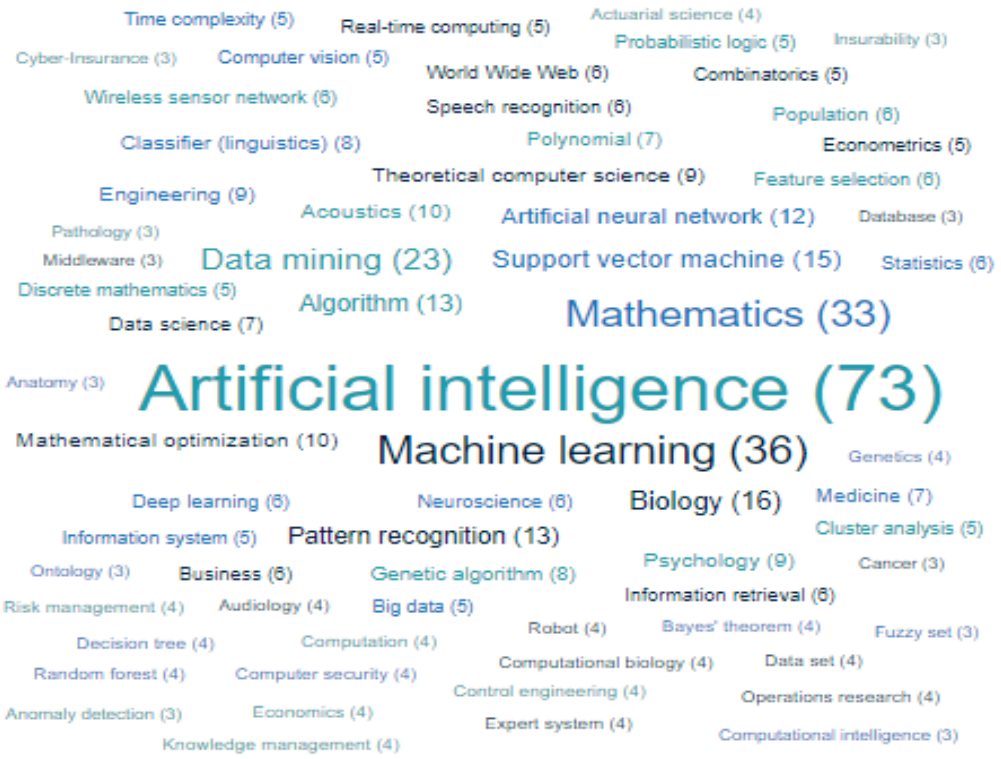
Llegamos entonces al cierre de este punto con la intención de mostrar el cómo se vuelve relevante una tecnología a partir de su apropiación y uso. McKinsey Global Institute, una de las compañías más influyentes en el mundo, sugiere que el 45% de las actividades en el lugar de trabajo en las empresas podrían automatizarse con las tecnologías actuales de Aprendizaje Automático y que el 80% de éste se puede atribuir a las capacidades computacionales ya existentes y los avances en el procesamiento del lenguaje natural (campo especializado del Aprendizaje Automático) podrían aumentar el impacto. (Henke N, 2016).

2.1 Machine learning en la industria de seguros

Las menciones relacionadas en motores de búsqueda (Google, Carrot2) y en bases de datos especializadas (Scopus, Lens.Org) son extensas al relacionar la aplicación específica del Aprendizaje Automático a la Industria de seguros, generalmente, los trabajos se desenvuelven en el estudio de puntos bastante acotados alrededor de los seguros de vida, automóviles y salud, y en particular, en la prevención de reclamaciones según el perfil de riesgo de los usuarios; de igual manera, hay un enfoque más amplio, no exclusivo, del aprendizaje automático, sino desde su “ciencia madre”: la Inteligencia Artificial.

Al realizar una búsqueda, por ejemplo, en el motor de búsqueda de Lens.Org para el Query “Machine Learning” e “Insurance” se obtiene un resultado de 241 documentos especializados, los cuales son difíciles de agrupar en categorías específicas de aplicación a la industria de seguros.

Figura 3: Nube de conceptos entorno a la busqueda "Machine Learning" e "Insurance"



Fuente: Construcción propia a partir del motor de búsqueda de la plataforma Lens.Org

Algunos generadores de contenido especializados han desarrollado material enfocado en los temas que nos traen a colación, estos generadores de contenido se enfocan en hallar información de fuentes primarias (entrevistas y encuestas) utilizando su amplia red de contactos. En este orden de ideas, la investigación se enfoca en la generación de hallazgos relevantes para el sector empresarial de cara a el tratamiento de información de los clientes con fines de ajustar modelos cuantitativos de diferente índole y uso en la industria de seguros.

Como se ha mencionado, el uso del aprendizaje automático en el sector asegurador, desde el mundo corporativo, se comienza a expandir con velocidad; desde la academia hay también diversidad de enfoques, sin embargo, existen algunos trabajos que recurrentemente se intentan resolver con la ayuda del aprendizaje automático: La administración de reclamaciones, la prevención de pérdidas en la suscripción de pólizas, la optimización de fuerza de ventas, predicción de ingresos, y la detección y prevención de fraude (Kuriyak, 2017)

El uso más usual para el aprendizaje automático se da en un entorno corporativo, especialmente en el cómo atraer-fidelizar más clientes y aumentar los ingresos minimizando la siniestralidad, no se escapa su uso en la función de Mercadeo y Ventas, en este espacio se han construido con ayuda de algoritmos de aprendizaje automático e Inteligencia Artificial, motores de recomendación los cuales ayudan a generar nuevas ventas, por ejemplo, basándose en datos de redes sociales y comportamiento de búsquedas en la web.

Adicionalmente, han sido desarrollados bajo el marco del aprendizaje automático aplicado al sector asegurador diversos trabajos con énfasis en los procesos de reclamaciones y/o siniestros, a continuación, se citan algunos de ellos:

- Se han realizado modelos de regresión basado en arboles de decisión multivariados aplicado a datos de reclamaciones de seguros. Se analizan diferentes tipos de coberturas en las líneas de propiedad, vehículos y equipos de contratistas y su correlación con las variables de respuesta propuestas en el estudio. Además, se muestra un comparativo sobre la mejora obtenida en la precesión de las predicciones utilizando arboles multivariados respecto a los univariados. (Quan & Valdez, 2018)

- También, se realizan predicciones del valor de pérdida en reclamacions de seguros mediante la utilización de diverses metodologías: Regresión Lineal, Random Forest Regression, Support Vector Regression y Feed Forward Neural Network. Se hace un énfasis especial en el uso de la regularización Lasso para evitar el sobreajuste en algunos de los modelos de regresión presentados. (Ogunnaike & Si, 2017)

- A partir de datos históricos de reclamaciones de siniestros para Automóviles, se ha analizado la precisión del método XGBoost y su idoneidad para el tratamiento de grandes volúmenes de datos. También hacen una comparación del rendimiento de XGBoost sobre otras técnicas de aprendizaje combinado como AdaBoost, Stochastic GB, Random Forest and Neural Network. (Fauzan & Murfi, 2018)

- La modelización del número de siniestros, es una forma de ejemplificar el comportamiento de las reclamaciones de seguros desde el punto de vista de su ocurrencia en el tiempo. Se muestra la aplicación de modelos lineales generalizados, arboles de decisión, agregación por boostrap, random forest, gradient boosting machine, redes neuronales, además de las técnicas de regularización para modelos de regresión como: Lasso, Ridge y Elastic Net. (Mendes et. al, 2017)

- Desarrollo de un modelo predictivo empleando arboles de decisión que pronostica la probabilidad de reclamación, dado algunos posibles factores de riesgo sobre la ocurrencia de siniestros de automóviles en la industria aseguradora. El modelo desarrollado tuvo en cuenta la clase de portafolios a nivel individual y colectivo, también considera el uso de vehículos como privados o comerciales. La edad del vehículo y la edad de asegurado fueron los principales factores de riesgo contribuyentes que predicen la ocurrencia de siniestros de automóviles motor tanto a nivel individual como colectivo. También concluye que los asegurados corporativos con vehículos de hasta 8 años tienen una mayor probabilidad de reclamación, mientras que los asegurados individuales entre las edades de 18 a 48 años tienen una alta probabilidad de presentar una reclamación en comparación con los asegurados mayores de 48 años cuando otras condiciones siguen siendo las mismas. (Frempong et al., 2017).

Por otra parte, no se identificaron citas bibliográficas, artículos científicos, revistas o trabajos académicos en los cuales se desarrolle de forma explícita el uso del aprendizaje automático aplicado a los seguros de asistencia en viajes, en ese sentido, la mayor afinidad corresponde a productos de salud y automóviles, donde algunas de sus coberturas pueden asociarse de forma homologa a las comprendidas en los seguros de tipo asistencial.

2.2 ¿Hacia dónde se encamina el aprendizaje automático?

Es importante recalcar que el aprendizaje automático no funciona como una ficha desagregada. Así, esta tecnología está embebida en un sistema de herramientas que, en conjunto, conforman un ecosistema que funciona a manera de simbiosis entre cada una de las partes. El desarrollo en campos del procesamiento de datos (Big data) y la capacidad de cómputo, apalancan el propio desarrollo del aprendizaje automático y potencian los resultados que pueden derivarse de su aplicación.

Dado que el objetivo final del Aprendizaje Automático es resolver problemas del mundo real, el sector asegurador debe concentrarse en la aplicación de los algoritmos de aprendizaje automático en los distintos procesos operativos para la búsqueda de economías de escala y mejora de la eficiencia. Uno de los factores impulsores en esta nueva dirección es la avalancha de datos generados impulsada por tecnologías de computación y almacenamiento en la nube cada vez más poderosas, que son capaces de procesar de manera rentable magnitudes relevantes de datos de mayor variedad, con mayor veracidad y velocidad (Cetinsoy, 2016).

Un enfoque prometedor para superar las fuentes de fricción que detienen la adopción del Aprendizaje Automático es la exploración de la automatización de la selección de algoritmos y el ajuste de parámetros y, al mismo tiempo, intentar controlar cualquier impacto negativo en el rendimiento final del modelo generado.

De esta manera, al establecer paralelismos con la posible evolución futura de las herramientas de Aprendizaje Automático, no es descabellado predecir que las APIs asumirán un papel de liderazgo en la definición del futuro de la mencionada tecnología. Con el telón de fondo de la publicidad masiva en los medios tecnológicos, se reduce al aprovisionamiento de una “rampa de acceso” más potente que admite canales de datos más sofisticados que conducen a aplicaciones de aprendizaje automático basadas en la nube más capaces. De manera bastante optimista, se piensa que esto es cuestión de tiempo, ya que un flujo constante de nuevas inversiones sigue fluyendo hacia el desarrollo de Aprendizaje Automático generando nuevas soluciones de servicio como el “Machine Learning as a Service” (MLaaS) en la nube, evidenciado más recientemente por los lanzamientos de Azure ML (2013) y AWS ML (2014).

La confluencia del cloud computing un nuevo significado. En ese sentido, podemos destacar los siguientes ítems:

- Seguir abstrayendo la complejidad de los algoritmos de Aprendizaje Automático, administrar sin problemas la infraestructura necesaria para aprender a partir de datos y para realizar predicciones a escala, es decir, sin servidores adicionales para aprovisionar o administrar.

- Descentralizar/democratizar el conocimiento especializado que conllevan las tareas de Aprendizaje Automático para dar mayor cobertura y usabilidad para todo tipo de público interesado en la tecnología.

- Cerrar fácilmente la brecha entre la fase de entrenamiento y la precisión sobre los pronósticos (scoring) entregados por los modelos de Aprendizaje Automático

- Capacitar a los científicos de datos con la automatización completa del flujo de trabajo para llevar a cabo proyecto de Aprendizaje Automático.

- Agregar trazabilidad y repetibilidad en todas las tareas de aprendizaje automático para su aplicación a nivel empresarial.

2.2.1 AutoML

La investigación del Aprendizaje Automático ha avanzado en múltiples aspectos, incluidas las estructuras usadas en los modelos y los métodos de aprendizaje. El esfuerzo por automatizar dicha investigación, conocido como “AutoML”, también ha logrado un progreso significativo. Sin embargo, este progreso se ha basado en gran medida en la arquitectura de las Redes Neuronales, donde se ha construido capas sofisticadas diseñadas por expertos como bloques o espacios de búsqueda igualmente restrictivos en términos de conocimiento (Real, 2020).

La demanda resultante de soluciones de fácil implementación para el Aprendizaje Automático ha dado lugar recientemente al mencionado campo del “Aprendizaje Automático Sistematizado” (AutoML)

Si bien la automatización completa puede motivar la investigación científica y proporcionar un objetivo de ingeniería a largo plazo, en la práctica probablemente lo que se debería buscar es la simple semi-automatización de la mayoría de estos y eliminar gradualmente la intervención humana en el ciclo, según sea necesario.

El campo del aprendizaje Automático Sistematizado (AutoML) tiene como objetivo tratar estas decisiones basándose en datos, objetiva y automáticamente: en el AutoML, el usuario simplemente proporciona datos y el sistema determina automáticamente el enfoque que funciona mejor para la aplicación en particular. Por lo tanto, hace que los enfoques de aprendizaje automático de vanguardia sean

accesibles para los investigadores y empresarios interesados en aplicar el aprendizaje automático pero que no tienen los recursos para aprender sobre las tecnologías detrás de él en detalle. Esto puede verse como una democratización del Aprendizaje Automático: con AutoML, el Aprendizaje Automático está al alcance de todos. (Gharamani, 2018).

2.2.2 Machine Learning as a Service (MLaaS)

El MLaaS es un conjunto de herramientas y servicios que diferentes proveedores ofrecen a los clientes para realizar diversas tareas de Aprendizaje Automático, por ejemplo, clasificación, regresión, deep learning, etc. Las herramientas y tecnologías pueden incluir servicios como: visualización de datos, preprocesamiento, entrenamiento y evaluación de modelos, predicciones, etc. Grandes empresas de tecnología (Goolge, Amazon, Microsoft, IBM, etc.) y una gran cantidad de empresas emergentes brindan diferentes tipos de servicios MLaaS. (Kuznetsov, s.f.)

Las plataformas de “Machine Learning as a Service” (MLaaS) también permiten a los analistas incorporar funcionalidades "inteligentes", como reconocimiento de imágenes, transcripción de voz y comprensión del lenguaje natural, en sus aplicaciones. Cuando una aplicación necesita invocar una de estas funcionalidades, se realiza una solicitud utilizando una API de MLaaS, simplemente entregando la responsabilidad del cálculo a la nube. Este paradigma proporciona un modelo atractivo para proveedores de servicios en la nube, lo que motiva a importantes empresas de tecnología a implementar y operar sus propias plataformas.

Las arquitecturas de servicios en la nube de MLaaS actuales siguen una estrategia de implementación de "talla única o marca blanca" en la cual varias instancias de la misma versión de servicio se escalan horizontalmente en la infraestructura informática del servicio para manejar a todos sus usuarios. Este diseño es problemático porque el MLaaS se basa en cálculos que son de naturaleza estadística: una exploración más profunda produce resultados más precisos, pero también requiere más tiempo de procesamiento para realizarse. Como resultado, los proveedores de MLaaS se ven obligados a hacer un sacrificio explícito entre la precisión de los resultados del servicio y la capacidad de respuesta.

3. Diseño e implementación del modelo predictivo

3.1 Planteamiento del modelo

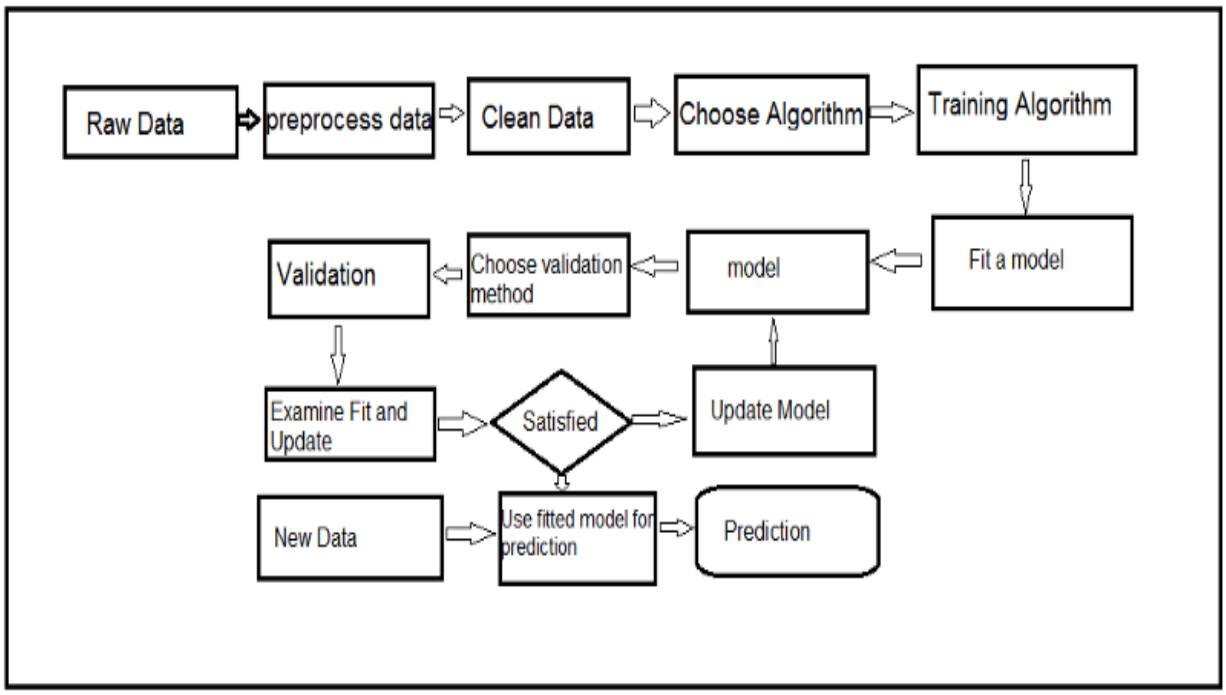
Retomando el caso de estudio planteado, el objetivo principal de este trabajo consiste en determinar a través de un modelo de *clasificación predictivo*, la ocurrencia o no de una reclamación sobre un seguro de asistencia en viaje, utilizando un conjunto de datos suministrado por una empresa anónima de seguros de viajes con sede en Singapur a través Kaggle. Para efectos de nombramiento, la base de datos se denotará con el nombre *travel insurance.csv* siendo 63.326 el total de instancias.

Las variables del conjunto de datos son las siguientes:

1. *Variable Objetivo: (Claim.Status)*
2. *Nombre de la agencia: (Agency)*
3. *Tipo de agencia de seguros de viajes (Agency.Type)*
4. *Canal de distribución de la agencia de seguros de viajes (Distribution.Channel)*
5. *Nombre del producto del seguro de viaje (Product.Name)*
6. *Duración del viaje (Duration)*
7. *Destino del viaje (Destination)*
8. *Ventas netas de pólizas de seguros de viajes (Net.Sales)*
9. *Comisión recibida por la agencia de seguros de viajes (Commission)*
10. *Genero del asegurado (Gender)*
11. *Edad del asegurado (Age)*

La metodología que se utiliza para diseñar y construir dicho modelo es la referenciada en la mayor parte de la literatura académica asociada a la ejecución de proyectos de aprendizaje automático, que a grandes rasgos, se sintetiza en el siguiente flujo de trabajo:

Figura 4: Machine Learning Model



Fuente: Analytics Vidhya, Machine Learning Model – Serverless Deployment, Diciembre 12, 2020

El ecosistema tecnológico sobre el cual se desarrolla el proceso de implementación de la metodología propuesta y, que igualmente es ampliamente utilizado en la comunidad analítica y en los entornos empresariales, en cuanto a capacidad de procesamiento moderada de datos se refiere, comprende los siguientes componentes:

Hardware

Servidor de cómputo virtual tipo VM.GPU3.1, con acceso público ubicado en Oracle Cloud Infrastructure:

CPU: 6 OCPU, Intel Xeon Platinum 8167M. Base frequency 2.0 GHz, max turbo frequency 2.4 GHz.
GPU: 1 NVIDIA® Tesla® V100
Memoria: 90 GB
Almacenamiento: Block Storage 47 GB

Software

Sistema operativo:
Ubuntu 16.04.7 LTS, 64 bits
Entorno de ejecución:
Anaconda3-2020.11
Python 3.8.5
Jupyter Notebook 6.1.4

Las principales librerías de python utilizadas durante la etapa de análisis exploratorio de los datos y la construcción del modelo predictivo son las siguientes: matplotlib, seaborn, pandas, numpy, sckit-learn e imblearn.

La configuración elegida cubre 2 propósitos principales: suministrar capacidades de cómputo dedicadas al análisis exploratorio de los datos y construcción del modelo analítico, y, adicionalmente, dar soporte a algunos algoritmos del framework AutoML que solo son compatibles bajo sistemas operativos basados en Unix.

3.2 Análisis exploratorio de los datos

Un primer acercamiento al caso de estudio propuesto es el entendimiento de las principales características que identifican el conjunto de datos a analizar. En ese sentido, se emplean métodos numéricos, estadísticos y representaciones gráficas que permiten acotar dicho entendimiento y obtener una información básica de las variables de interés así como la fase de preparación de los datos.

A partir del análisis realizado sobre el conjunto de datos “travel insurance.csv”, podemos destacar los siguientes aspectos:

Figura 5: Atributos conjunto de datos "travel insurance"

```
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   agency                63326 non-null  object
 1   agency_type           63326 non-null  object
 2   distribution_channel  63326 non-null  object
 3   product_name          63326 non-null  object
 4   claim                 63326 non-null  object
 5   duration              63326 non-null  int64
 6   destination           63326 non-null  object
 7   net_sales             63326 non-null  float64
 8   commision             63326 non-null  float64
 9   gender                18219 non-null  object
 10  age                   63326 non-null  int64
```

Fuente: Elaboración propia

- Tipos de variables:

  Categóricas: agency, agency_type, distribution_channel, product_name, claim, destination, gender
  Numéricas continuas: duration, age, net_sales, commission

- **Identificación de valores negativos para la variable net_sales.** Por su naturaleza, no deberían admitir valores inferiores o iguales a cero, lo cual hace pensar en posibles inconsistencias en la calidad de los datos.

- **Valores atípicos en la variable age.** La distribución de datos se ubica entre 0 y 118 años. La densidad poblacional mundial con edades superiores a los 100 años es pequeña respecto a la población mundial total. En este sentido, se debe analizar esta situación con mayor nivel de detalle.

- Si bien la unidad de medida para cada variable no es explícita en la base de datos, para la variable duration, se identificaron valores atípicos. Partiendo del supuesto que la duración sea en días, estos valores extremos parecerían ser bastante anormales.

- El 50% de las observaciones no presenta valores superiores a 0 en la variable comisión. En ese sentido, podría ser una restricción no conocida sobre las ventas netas (net_sales). Por ejemplo, si la comisión solo se asigna a partir de un umbral establecido sobre las ventas netas.

- Identificación de variables categóricas con cardinalidad media-alta:

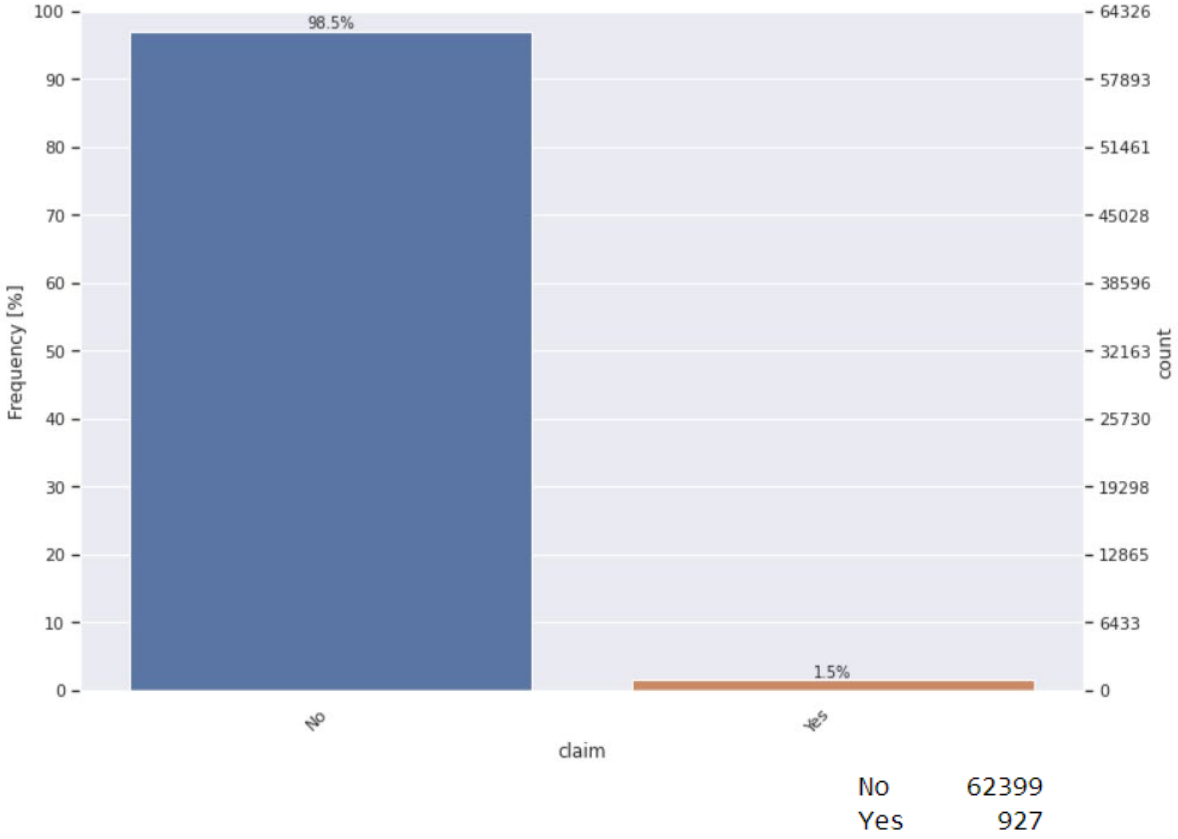
  - agency: 16 valores únicos
  - product_name: 26 valores únicos
  - destination: 149 valores únicos

  En el planteamiento del modelo es necesario considerar el tratamiento de las variables categóricas. Así, una codificación tipo One Hot Encoding puede derivar en una gran cantidad de variables "dummy", lo cual puede afectar la fase de entrenamiento y el performance del modelo.

- La variable gender tiene una cantidad elevada de valores NaN, más del 70% de las observaciones no tienen el género del asegurado.

- **Variable objetivo *Claim* desbalanceada.** Hemos de destacar que, más del 95% de las observaciones corresponden al nivel 'No'. Es una restricción importante dado que la mayoría de los algoritmos de aprendizaje automático parten del hecho que la proporción de los valores de clasificación objetivo son similares, es decir, las clases están balanceadas.

Figura 6: Distribución de frecuencias variable "claim"



Fuente: Elaboración Propia

Así, existe un sinfín de argumentos que inciden en el análisis de los datos desequilibrados y su impacto en la precisión de los algoritmos de aprendizaje automático. Se distinguen los siguientes motivos:

- La distribución dispar de la variable objetivo. La presencia de observaciones sin indicios de reclamación cuenta con una elevada presencia de registros, formando un grupo más uniforme y con menor variabilidad que las observaciones con reclamación positiva.

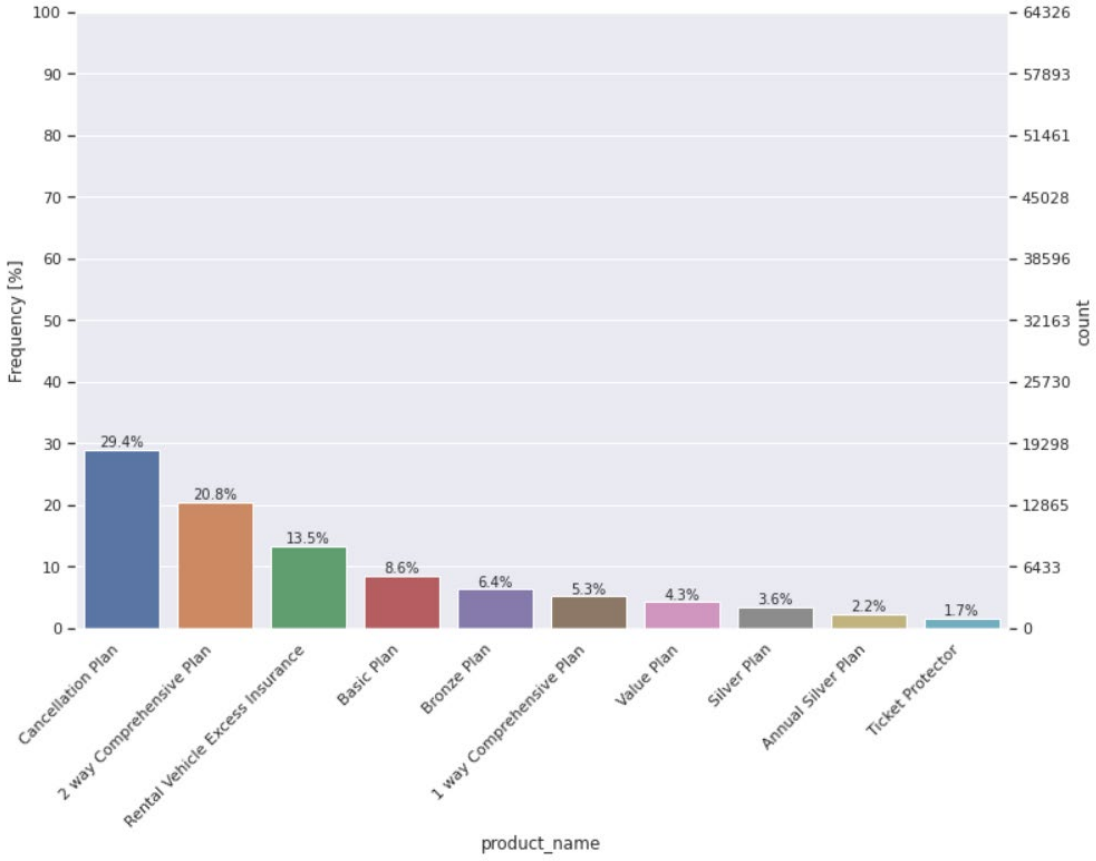
- Los algoritmos tienen como uno de sus principales objetivos disminuir el error general, al que la clase minoritaria aporta muy poco y donde existe una tendencia de la clasificación hacia la clase mayoritaria. Hemos de tener en cuenta que estas metodologías y algoritmos parten del supuesto de clases equilibradas.

- Los errores tienen la misma ponderación para las distintas clases y no tiene porqué ocurrir. De este modo, los errores tienen que ser valorados en función de su relevancia.

Análisis de las principales variables

A continuación analizamos el comportamiento de algunas de las variables más relevantes de forma individual. Todo ello permite obtener una primera aproximación de la tendencia de los resultados

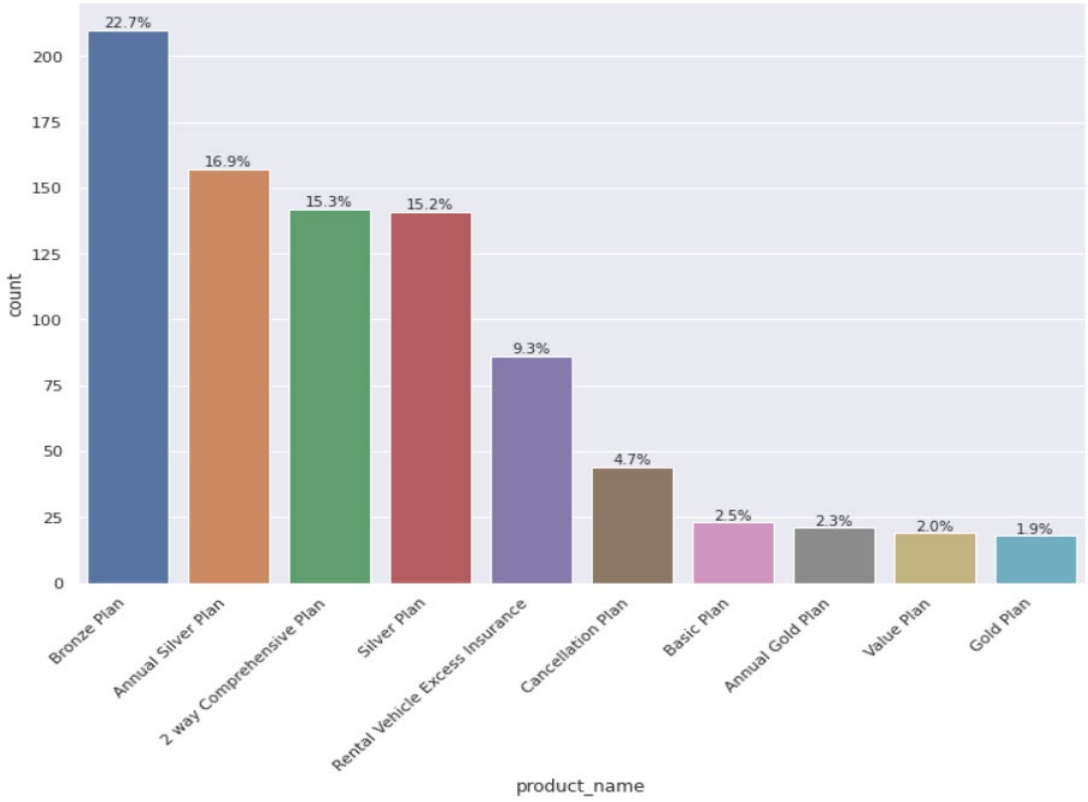
Figura 7: Productos más vendidos



Fuente: Elaboración propia

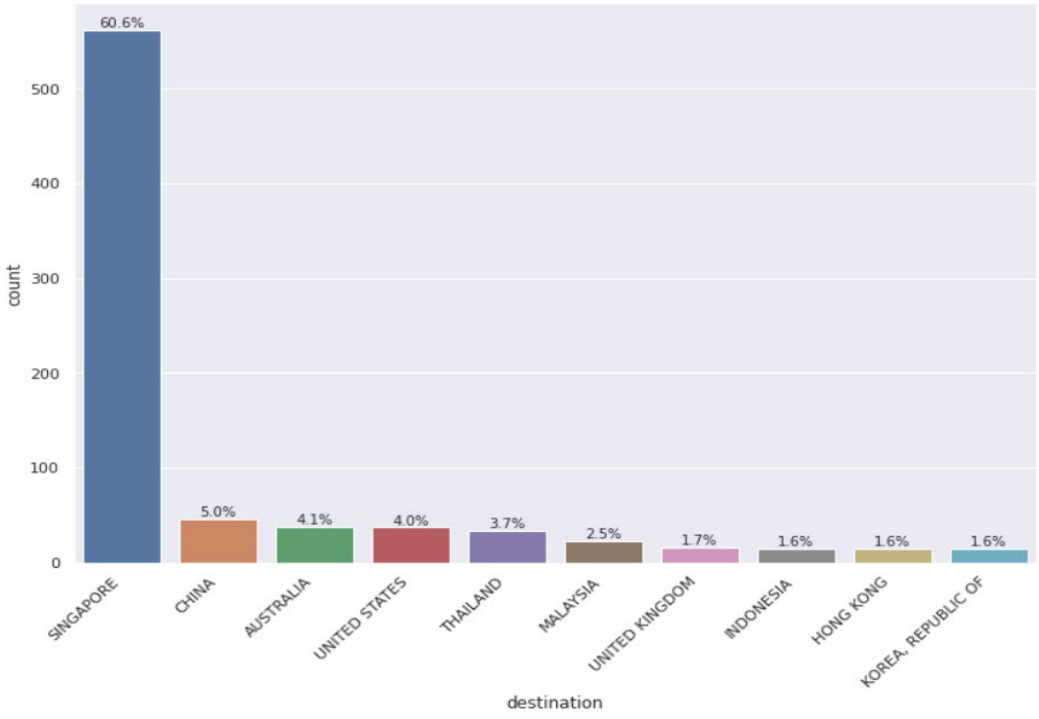
Los 3 productos más vendidos son: Cancellation Plan, 2 way Comprehensive Plan y Rental Vehicle Excess Insurance, en contraste con los productos que presentan más reclamaciones: Bronze Plan, Annual Silver Plan y 2 way Comprehensive Plan. Así, no necesariamente los productos que más se venden son los que presentan más reclamaciones.

Figura 8: Productos con mayor cantidad de reclamaciones



Fuente: Elaboración propia

Figura 9: Lugares de destino con mayor cantidad de reclamaciones



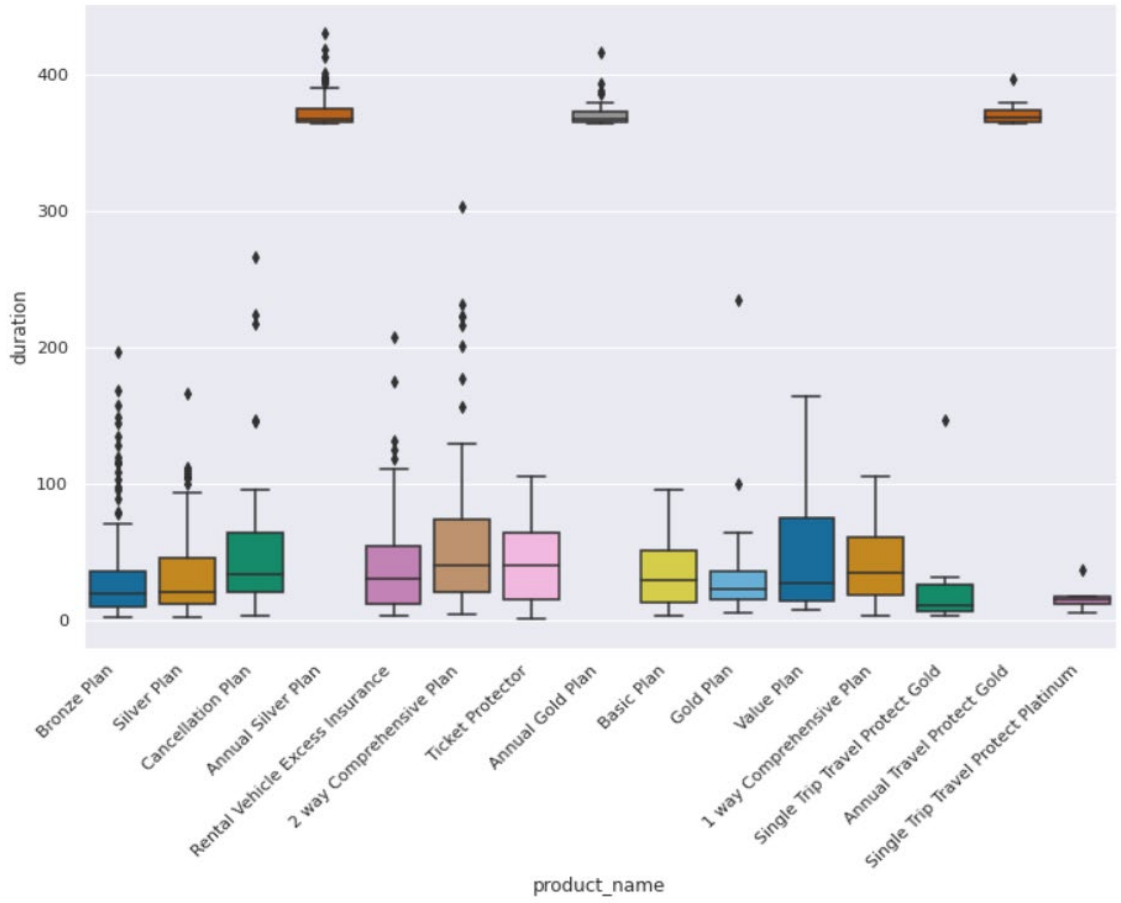
Fuente: Elaboración propia

Aproximadamente el 60% de las reclamaciones (Claim=Yes), tienen como lugar de destino Singapur (la sede de la aseguradora está en dicho país. A éste le prosiguen los países de destino China, Australia, Estados Unidos, Tailandia y Malasia.

La mayor proporción de las reclamaciones (Claim=Yes), provienen de asegurados con edades entre los 30 y 60 años. De igual forma se observa que los valores de las edades atípicas superiores a 80 años, en su mayoría, tienen como lugar de destino INDIA.

A simple vista, la suposición de la unidad de medida para la variable duration, parece ser cierta, y estaría dada en días, por su correspondencia con los planes anuales donde la duración es superior a 300 días.

Figura 10: Productos de seguros de viajes respecto a la duración (claim=Yes)



Fuente: Elaboración Propia

Las agencias que más pólizas de viajes gestionan son: EPX, CWT, C2B y JZI, con una participación acumulada superior al 90%. También se aprecia que los canales de distribución con mayor participación son los Online , con un valor superior al 95%; además sobre el canal Offline no se identifican reclamaciones efectivas (Claim=Yes). El análisis exploratorio detallado se encuentra en la sección 3 del notebook de Jupyter anexo.

3.3 Preparación de los datos

Las técnicas de preprocesamiento o preparación de los datos generalmente se refieren a la adición, eliminación o transformación de datos para que el modelo analítico los pueda procesar adecuadamente. Sin lugar a duda, la preparación de los datos es una de las etapas cruciales del aprendizaje automático siendo en muchas ocasiones una tarea compleja y farragosa. Así, durante el proceso se puede omitir, eliminar o sobrevalorar información que mejor describe la realidad que se quiere modelar.

En un proyecto de aprendizaje automático, como este caso, un problema de clasificación, no pueden utilizarse los algoritmos de modelamiento directamente sobre los datos originales por las siguientes razones (Brownlee, 2020)

Para solventar este tipo de restricciones, en el transcurso de los años se han desarrollado por parte de la comunidad académica y científica, un conjunto de técnicas estandarizadas o comunes que permiten adecuar la información requerida por los modelos predictivos y garantizar así su correcto funcionamiento. Podemos destacar los siguientes ítems:

- **Limpieza de los datos**: consiste en identificar y corregir inconsistencias o errores en los datos.
- **Selección de características**: identificar las variables de entrada o predictoras más relevantes para el modelo predictivo.
- **Transformaciones sobre los datos**: consiste en cambiar la escala o distribución de las variables.
- **Ingeniería de características**: consiste en construir nuevas variables derivadas a partir de los datos existentes.
- **Reducción de la dimensionalidad**: consiste en crear proyecciones compactas de los datos.

En los siguientes apartados aplicaremos algunas de estas técnicas sobre el conjunto de datos “travel insurance”, considerando además algunos de los resultados y observaciones obtenidas del análisis exploratorio.

3.3.1 Limpieza de los datos

Se eliminan o imputan algunos valores sobre datos de las observaciones que así lo requieran con relación a lo observado en el análisis exploratorio o se omiten algunas de las variables que inicialmente no son representativas para la construcción del modelo.

A continuación, se recopilan los hallazgos más relevantes del apartado anterior que están estrechamente relacionados al proceso de limpieza:

- **Cantidad considerable de valores NaN en la variable gender.** El porcentaje de valores nulos sobre esta variable es bastante alto, superior al 70% del total, no se considerará para la construcción del modelo. Además de los valores que

no son NaN, visualmente no se aprecia una diferencia significativa con respecto a la variable objetivo, la distribucion de datos respecto al genero es muy similar.

- **Valores negativos y atípicos en las variables net_sales y duration**. Para el caso de *net_sales* tenemos 2562 valores inferiores o iguales a cero. De este modo, se opta por eliminar este subconjunto de observaciones negativas, representan menos del 1% de la población total de observaciones. En el caso de la variable *duration* tenemos 14 observaciones con una duración superior a 1000 días. De este modo, se eliminan del conjunto de datos dado que generan ruido en la distribución de los datos. Además, se identifican 64 observaciones con una duración menor o igual 0 días, para este caso se opta por imputar los valores sobre este conjunto de observaciones tomando el valor promedio de la duración respecto al lugar de destino, es decir, la variable destination.

- **Valores atípicos en la variable age.** La distribución de datos se ubica entre 0 y 118 años: 953 observaciones con edades superiores a 100 años y 1 observación con edad igual a 0 años. Para las edades superiores a 100 años o iguales a cero, se imputan valores con la mediana de la variable age.

- **El canal de distribución con mayor participación es el Online**, con un valor superior al 95% del total. Sobre el canal Offline no se identifican reclamaciones efectivas (Claim=Yes) y, bajo esta premisa, la variable distribution_channel no se tendrá en cuenta para la construcción del modelo.

3.3.2 Transformaciones sobre los datos

Esta fase del proceso de preparación de los datos tiene 3 objetivos principales:

1. Convertir las variables categóricas predictoras en variables numéricas
2. Estandarizar las magnitudes de las variables predictoras
3. Codificar la variable categórica objetivo

Para llevar a cabo el proceso de conversión de las variables categóricas, primero se aplica una función que transforma las cadenas de caracteres contenidas en cada variable a letras minúsculas, por ejemplo, en la variable agency_type, los valores Airlines y Travel Agency, se convierten en ‘airlines’ y ‘travel agency’. Como segundo paso se aplica la estrategia *one hot encoding*, la cual crea una columna para cada valor distinto que exista en la característica que estamos codificando y, para cada registro, marca con un 1 la columna a la que pertenezca dicho registro y deja las demás con 0. (Interactive Chaos, s.f.)

Para demostrar la estrategia partamos del siguiente escenario:

# observación	agency_type
1	airlines
2	travel agency
3	airlines

Luego de aplicar one hot encoding:

# observación	airlines	travel agency
1	1	0
2	0	1
3	1	0

Con el fin de optimizar la cantidad de variables que se crea por cada categoría, vamos a utilizar una versión equivalente de la técnica, en la cual se utilizan n -1 categorías, dando como resultado la siguiente distribución:

# observación	travel agency
1	0
2	1
3	0

En este sentido, los valores 0 en travel agency corresponden intrínsicamente a la categoría ‘airlines’.

En cuanto a la estandarización de las magnitudes en las variables predictoras, debemos recordar que contamos con diferentes unidades de medida como años, días, valores monetarios, ...etc. Para ajustar este escenario, se aplica una función que resta a cada observación de la variable su valor medio y lo divide por su desviación estándar, de esta forma evitamos utilizar diferentes magnitudes sobre el conjunto de datos que pueden generar ruido en la fase de construcción del modelo.

La codificación de la variable objetivo (Yes/No) se hace con el fin de que los modelos de clasificación pueden realizar operaciones y cálculos sobre estos datos, obteniendo como resultado la siguiente asignación: Yes = ‘1’ , y No= ‘0’.

Como resultado final del proceso de preparación y limpieza de los datos se obtiene un conjunto de datos con las siguientes características:

- 191 variables numéricas predictoras y 1 variable categórica objetivo (claim_label).

- Un total de observaciones de 60.750, de las cuales, 59.826 observaciones no presentan reclamaciones (claim_label=0) y 924 si las presentan (claim_label=1).

3.3.2 Selección de atributos o características

El conjunto de datos obtenido de la etapa anterior cuenta con una total de 191 variables predictoras, lo cual representa cierto nivel de complejidad para la construcción del modelo desde el punto de vista computacional y puede llegar a generar sobre entrenamiento en el modelo predictivo. En este sentido, la selección de atributos es una técnica que se utiliza para reducir el número de variables predictoras a utilizar para la construcción del modelo. Así son las más significativas, desde un punto de vista cuantitativo, en relación con la variable objetivo.

Las principales razones por las cuales se aplica este tipo de técnicas previo a la construcción del modelo son:

- Se puede llegar a entrenar más rápidamente el algoritmo de clasificación.
- Facilita la interpretación de los resultados.
- Puede mejorar la precisión y reducir la sobre estimación.

Para efectos de este trabajo, se introducen un par de métodos que usualmente son empleados cuando el modelo representa una tarea de clasificación:

Métodos indirectos (filter)

Los métodos de selección de características indirectos utilizan técnicas estadísticas para evaluar la relación entre cada variable de entrada y la variable de destino, y estas puntuaciones se utilizan como base para elegir (filtrar) las variables de entrada que se utilizarán en el modelo. Un buen ejemplo de esta familia de métodos es la prueba estadística ANOVA.

El método ANOVA es utilizado cuando una variable es numérica y la otra es categórica, como es el caso de la tarea de clasificación acá planteada. Los resultados de esta prueba pueden ser utilizados para la selección de características, donde aquellas variables que son independientes de la variable objetivo pueden ser removidas del conjunto de datos.

Métodos directos (wrapper)

Los métodos de selección de características directos crean muchos modelos con diferentes subconjuntos de características de entrada y seleccionan aquellas características que dan como resultado el modelo de mejor rendimiento según una métrica dada. Pueden ser costosos computacionalmente. Un buen ejemplo de esta familia de métodos es el algoritmo Boruta.

Boruta

El algoritmo Boruta se desarrolló como un paquete en R en 2010. (Kursa & Rudnicki, 2010). Está diseñado como un wrapper para cualquier algoritmo de clasificación que pueda devolver puntuaciones de importancia para todas las características de un

conjunto de datos. Las características que se consideran menos relevantes para la construcción de modelos se obtienen mediante una prueba estadística y se eliminan en cada iteración. (Siddhant, 2020).

Como funciona el algoritmo:

- En primer lugar, agrega aleatoriedad al conjunto de datos dado mediante la creación de copias mezcladas de todas las características (que se denominan características sombra, shadow features).

- Luego, entrena un clasificador Random Forest en el conjunto de datos extendido y aplica una medida de importancia de la característica (el valor predeterminado es precisión de disminución media) para evaluar la importancia de cada característica donde más alto significa más importante.

- En cada iteración, comprueba si una característica real tiene una importancia mayor que la mejor de sus características sombra (es decir, si la característica tiene una puntuación Z más alta que la puntuación Z máxima de sus características sombra) y elimina constantemente las características que se consideran muy poco importante.

- Finalmente, el algoritmo se detiene cuando todas las características se confirman o rechazan o cuando alcanza un límite especificado de ejecuciones de Random Forest. (Dutta, 2016)

Veamos a continuación los aspectos más relevantes de la exploración de ambas técnicas sobre el conjunto de datos obtenido en la fase de preparación:

Resultados obtenidos al aplicar el método Boruta sobre el conjunto de datos de 191 características

Se obtiene una total de 13 variables significativas:

['duration', 'net_sales', 'commision', 'age', 'c2b', 'epx', 'jzi', 'travel_agency', '2_way_comprehensive_plan', 'annual_silver_plan', 'bronze_plan', 'cancellation_plan', 'singapore']

Resultados obtenidos al aplicar el método ANOVA f-test sobre el conjunto de datos de 191 características

Las 13 variables más significativas son:

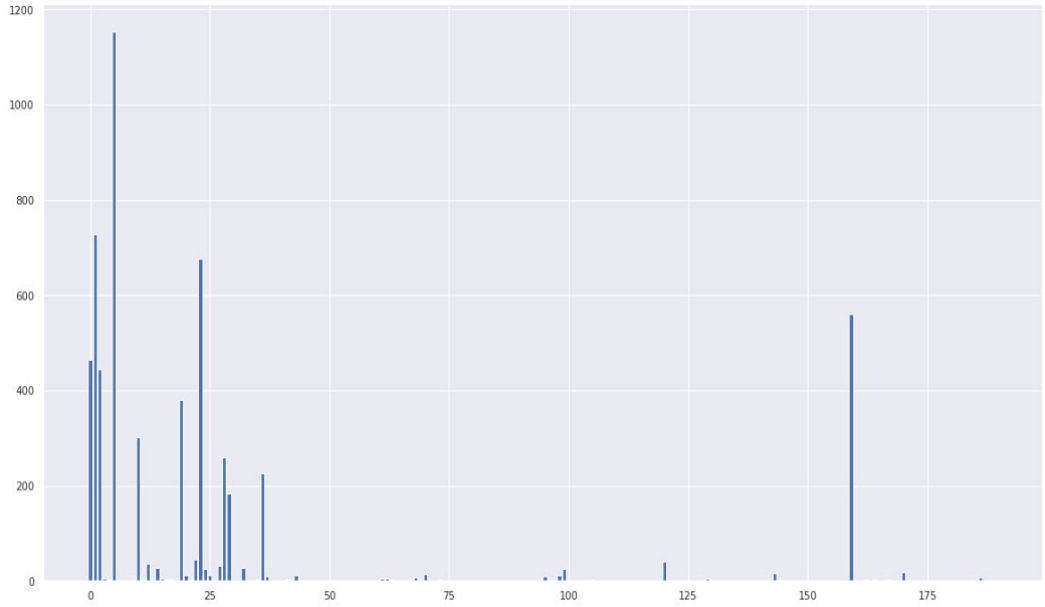
['duration', 'net_sales', 'commision', 'c2b', 'epx', 'travel_agency', 'annual_gold_plan', 'annual_silver_plan', 'bronze_plan', 'cancellation_plan', 'silver_plan', 'malaysia', 'singapore']

Los valores c2b, epx, jzi corresponden a identificadores de las agencias de seguros de viajes obtenidos a partir de la variable ‘agency’.

El valor travel_agency, corresponde a uno de los identificadores obtenidos a partir de la variable ‘agency_type’, la categoría restante corresponde al valor airlines.

Los valores 'annual_gold_plan', 'annual_silver_plan', 'bronze_plan', 'cancellation_plan', 'silver_plan', '2_way_comprehensive_plan' corresponden a identificadores de los productos de seguros de viajes obtenidos a partir de la variable product_name.
Los valores 'malaysia', 'singapore' corresponden a identificadores de lugares de destino obtenidos a partir de la variable destination.

Figura 11: Valores ANOVA f-test para cada una de las variables



Fuente: Elaboración propia

Se aprecia que ambos métodos convergen en la mayoría de los atributos seleccionados y es una buena métrica de contraste para identificar las características más importantes. Bajo esta premisa, la construcción del modelo predictivo se llevará a cabo utilizando las variables seleccionadas a partir del algoritmo Boruta, al ser este método más robusto en la estimación los atributos más relevantes.

Es evidente la bondad de aplicar este tipo de técnicas sobre conjuntos de datos que presentan alta dimensionalidad.

3.3.3 Reducción de la dimensionalidad

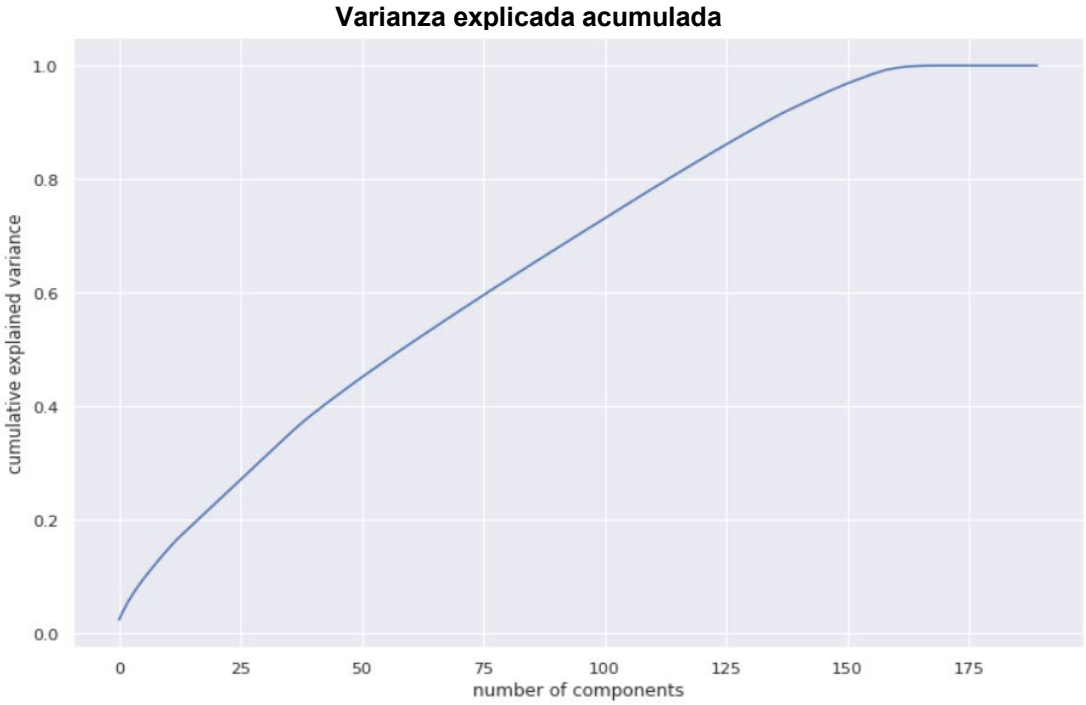
Las técnicas de reducción de la dimensionalidad tienen como objetivo principal proyectar un espacio vectorial de *n* dimensiones sobre un nuevo espacio vectorial de *m* dimensiones (m<n) que mejor captura la esencia de los datos originales. Las variables de este nuevo espacio de m dimensiones, generalmente, son representaciones numéricas obtenidas a partir de la aplicación de métodos y operaciones del algebra vectorial sobre el conjunto de datos de n dimensiones.
PCA es una técnica que busca extraer un nuevo conjunto de variables de un gran conjunto de variables existente. Estas variables extraídas se denominan componentes principales. (Jolliffe, I. T., & Cadima, J. , 2016).

Algunos de los puntos clave de esta técnica son los siguientes:

- Un componente principal es una combinación lineal de las variables originales.
- Los componentes principales se extraen de tal manera que el primer componente principal explica la varianza máxima en el conjunto de datos.
- El segundo componente principal intenta explicar la varianza restante en el conjunto de datos y no está correlacionado con el primer componente principal.
- El tercer componente principal trata de explicar la varianza que no se explica por los dos primeros componentes principales y así sucesivamente. (SHARMA, 2018)

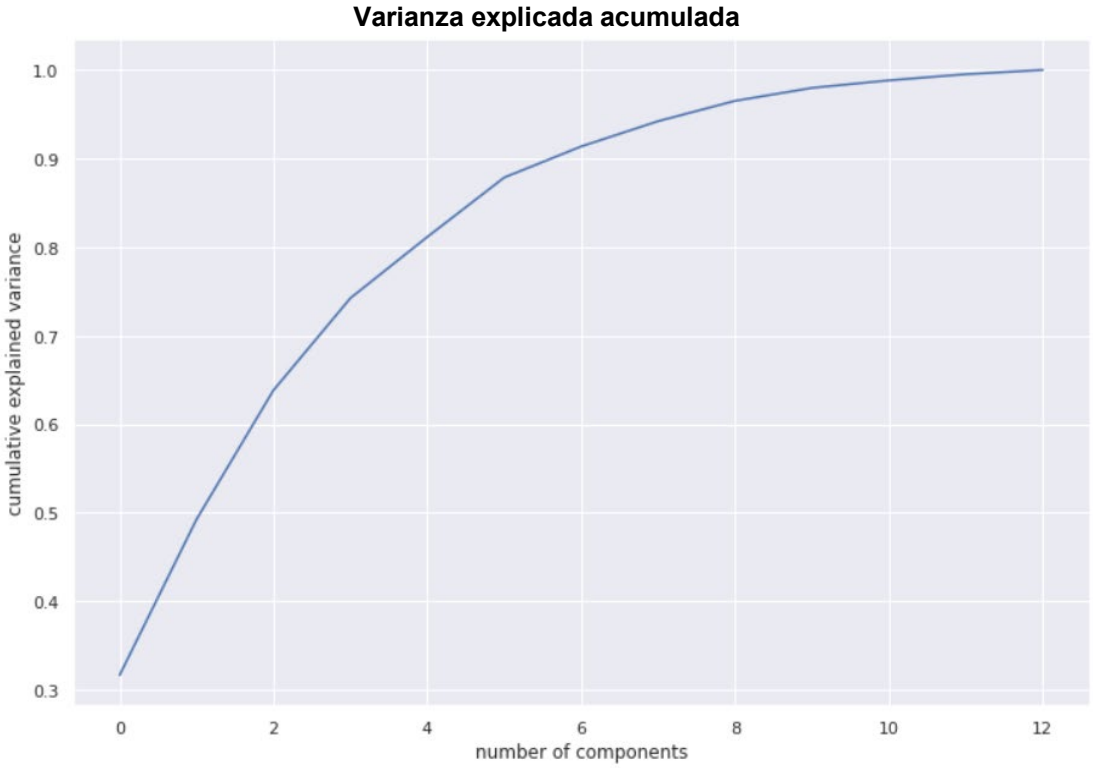
Para efectos ilustrativos, utilizaremos la técnica PCA sobre el conjunto de datos original de 191 variables y el obtenido a partir de la aplicación del algoritmo Boruta

Figura 12: Aplicación de la técnica PCA en conjunto de datos de 191 variables



Fuente: Elaboración propia

Figura 13: Aplicación de la técnica PCA en conjunto de datos de 13 variables



Fuente: Elaboración propia

Los resultados obtenidos sobre el conjunto de datos de 191 variables indican que aproximadamente el 90% de la varianza explicada acumulada corresponden a más de 125 componentes, lo cual continúa siendo un espacio dimensional bastante elevado. Probablemente esta técnica en particular no sea la más idónea para ese conjunto de datos. Por el contrario, la misma técnica aplicada sobre el conjunto de 13 variables, si presenta un resultado mejor.

Así, con las primeras 6 componentes es posible representar un poco más del 90% de la varianza explicada acumulada. Es decir, un poco menos de la mitad del número de variables originales pueden ser empleadas en este nuevo espacio vectorial para generalizar la realidad expresada en el conjunto de datos evaluado.

En el siguiente apartado utilizaremos el conjunto de datos de 6 dimensiones para construir el modelo predictivo y comparemos los resultados obtenidos empleando el conjunto de datos con 13 variables para concluir cuál de ellos ofrece el mejor rendimiento.

3.4 Elección del modelo de clasificación

Existen diversos algoritmos y técnicas para el modelamiento predictivo de problemas de clasificación. A priori, no una teoría exacta sobre cómo mapear algoritmos para cada tipo de problema.

Cada conjunto de datos representa una realidad diferente: cantidad de variables, calidad y tipo de datos, volumen de observaciones, contexto del problema…etc. Son factores que generan complejidad e impiden generalizar completamente el algoritmo adecuado para la tarea adecuada.

Los algoritmos de modelado predictivo de clasificación se evalúan en función de sus resultados. La exactitud (accuracy) de la clasificación es una métrica popular que se utiliza para evaluar el rendimiento de un modelo en función de las etiquetas de clase previstas. La exactitud de la clasificación no es perfecta, pero es un buen punto de partida para muchas tareas de clasificación.

En lugar de etiquetas de clase, algunas tareas pueden requerir la predicción de una probabilidad de pertenencia a una clase para cada observación. Esto proporciona incertidumbre adicional en la predicción que se interpreta a partir del modelo. Un diagnóstico popular para evaluar las probabilidades predichas es la curva ROC que se detalla más adelante.

Para nuestro caso particular, la tarea clasificación consiste en determinar la ocurrencia o no de una reclamación sobre una determinada cartera de asegurados que mantienen contratado un seguro de asistencia en viaje. Este tipo de tareas de clasificación bianaria involucran una clase que representa el estado normal (claim = No) y otra clase que es el estado anormal (claim = Yes).

En este sentido, la clase para el estado normal suele ser asignada a la etiqueta de clase ‘0’ y la clase con el estado anormal con la etiqueta de clase ‘1’, una notación dicotómica ampliamente utilizada para estudiar este tipo de fenómenos.

Los algoritmos más populares y que se encuentran detallados largamente en la literatura científica para los problemas de clasificación binaria son:

- Regresión Logística
- K-vecinos más cercanos
- Arboles de decisión
- Máquinas de vectores de soporte
- Naive Bayes

En este trabajo, se utilizará como punto de partida el modelo basado en una regresión logística, dada su simplicidad y efectividad demostrada para abordar tareas de clasificación binaria. Posteriormente se destacan otros métodos para comparar el rendimiento obtenido por cada uno de los clasificadores en virtud de las métricas de evaluación establecidas en la sección 3.4.1.

Regresión Logística

Los modelos de regresión permiten evaluar la relación entre una variable (dependiente) respecto a otras variables en conjunto (independientes). Los modelos regresión se expresan de la siguiente forma: $Y = f(x1, x2, \ldots) + \varepsilon$.

El objetivo principal de la construcción de un modelo de regresión es la evaluacion del cambio en unas características determinadas (variables independientes) sobre otra característica en concreto (variable dependiente), cuando la variable dependiente es una variable continua, el modelo de regresión más frecuentemente utilizado es la regresión lineal, mientras que cuando la variable de interés es dicotómica (es decir, toma dos valores como sí/no, hombre/mujer, votó/no votó) se utiliza la regresión logística.

En este último caso, se construye una función basada en el cálculo de la probabilidad de que la variable de interés adopte el valor del evento previamente definido (Peláez, 2016), de la siguiente manera:

$$Y = \ln (p / (1-p))$$

En el modelo de regresión logística se estima un modelo de regresión que en lugar de realizar estimaciones para la variable dependiente real, las realizará sobre la función de probabilidad asociada a ella, pudiendo entonces aplicar los métodos de estimación aplicables al modelo de regresión lineal, diferenciándose entonces ambos modelos únicamente en la interpretación de resultados.

Para entender en que consiste un modelo de regresión de acuerdo con Peláez (2016) debemos relacionar dos conceptos: el coeficiente de correlación y el análisis de la varianza. Se puede demostrar que existe una relación entre el coeficiente de correlación (r) y el análisis de la varianza de la regresión, de tal forma que el cuadrado

de r, llamado coeficiente de determinación, multiplicado por 100 se interpreta como el porcentaje de la varianza de la variable dependiente que queda explicada por el modelo de regresión.

Tambien es fundamental hablar de los criterios para seleccionar las variables del modelo, las cuales presentamos a continuacion:

- Introducir en el modelo aquellas variables que resultaron estadísticamente significativas en las comparaciones bivariantes realizadas previamente.

- Debería considerarse la conveniencia de incluir en el modelo adicionalmente aquellas variables que consideremos especialmente importantes o influyentes, como por ejemplo la edad o el género, si sospechamos que a pesar de no haber resultado estadísticamente significativas, podrían modificar o intervenir en nuestros resultados, o otra serie de variables de las que hayamos tenido conocimiento de su influencia a través de estudios previos.

3.4.1 Ajuste del modelo sobre los datos de entrenamiento

Definido el modelo inicial a evaluar, en esta etapa de la metodología propuesta es importante introducir algunos conceptos clave a la construcción del modelo predictivo, permitiendo que se lleve a cabo un entrenamiento adecuado del mismo y se reduzca el margen de error al máximo:

Separación del conjunto de datos en entrenamiento y prueba

El procedimiento habitual en el aprendizaje automático o machine learning es la división de los datos en dos subconjuntos: uno de entrenamiento y otro de comprobación o test.

Datos de entrenamiento

Las observaciones en el conjunto de entrenamiento forman la experiencia que el algoritmo usa para aprender. Por lo general corresponde a una proporción aleatoria entre el 70% y 80% del conjunto de datos original.

Datos de comprobación o test

El conjunto de test es un conjunto de observaciones utilizadas para evaluar el rendimiento del modelo utilizando alguna medida de rendimiento. Por lo general, corresponde a una proporción aleatoria entre el 20% y 30% del conjunto de datos original.

Es importante que no se incluyan observaciones del conjunto de entrenamiento en el conjunto de prueba. Si el conjunto de prueba contiene ejemplos del conjunto de entrenamiento, será difícil evaluar si el algoritmo ha aprendido a generalizarse bien a partir del conjunto de entrenamiento o simplemente lo ha memorizado.

Cuando se hace referencia a que un modelo generaliza bien, tendrá la capacidad de realizar efectivamente una tarea de predicción sobre nuevos datos. En contraste, un modelo que memoriza los datos de entrenamiento aprendiendo un modelo demasiado complejo podría predecir los valores de la variable de respuesta para el conjunto de entrenamiento con precisión, pero no podrá predecir el valor de la variable objetivo para nuevas observaciones. Memorizar el conjunto de entrenamiento se conoce como overfitting o ajuste excesivo.

Un modelo que memoriza sus observaciones puede no realizar bien su tarea, ya que podría memorizar relaciones y estructuras que son ruido o coincidencia. La memorización y generalización del equilibrio, o el ajuste excesivo y el ajuste insuficiente, es un problema común a muchos modelos de aprendizaje automático. (Mayorga, 2018)

En función de los algoritmos utilizados nos podemos encontrar modelos que explican muy bien los datos de entrenamiento, pero ienen una escasa capacidad predictiva.

Tratamiento de datos desbalanceados en la variable objetivo

Previo a realizar el ajuste del modelo de regresión logística, es indispensable retomar el hallazgo asociado a la distribución de frecuencias de la variable objetivo, en el cual se identifica un desbalance considerable entre las categorías Yes/No, con clase mayoritaria el nivel No.

La literatura propone tres enfoques de tratamiento del desbalance en los datos: a nivel de datos, la modificación de algoritmos y las matrices de costos. En este apartado, se evalúa la metodología a nivel de datos, que considera el re-muestreo de los datos para balancear las clases. En esencia, consiste en alcanzar un balance entre las clases mediante la eliminación de objetos de la clase mayoritaria (sub - muestreo) o la inclusión de objetos en la clase minoritaria (sobre - muestreo) (Cardenas, 2019).

En la sección 3.5 se explorarán algunos algoritmos modificados para abordar el procesamiento de clases desbalanceadas y se comparan los resultados obtenidos con las técnicas de re-muestreo.

Cuando se hace sub-muestreo se puede tener el problema de excluir objetos representativos o valiosos para entrenar el clasificador. Si lo que se realiza es un sobre-muestreo es posible que se incluyan objetos artificiales y el clasificador se pueda sobre - entrenar.

En el presente análisis se emplean solo algunas técnicas que pretenden modificar la distribución de los datos cuando estos son desbalanceados:

Métodos de sub-muestreo

- RU (*Random Under-sampling*), en este caso se selecciona de manera aleatoria instancias de la clase mayoritaria y se eliminan sin reemplazo hasta que ambas clases queden balanceadas (Cardenas, 2019).

- Tomek Links, en este se eliminan las instancias de la clase mayoritaria que sean redundantes o que se encuentren muy cerca de instancias de la clase minoritaria (Mohamed M. Shoukri T. A.-H., 2016)

Métodos de sobre-muestreo

- SMOTE (Synthetic Minority Over-sampling TEchnique), el algoritmo genera nuevas observaciones de la clase minoritaria interpolando los valores de las instancias de esta clase más cercanas a una dada. (Cardenas, 2019)

Para efectos ilustrativos de la investigación, se utiliza el método hibrido SMOTE + RU, en la cual aplica sobre muestreo SMOTE en la clase minoritaria y luego aplica Random Undersampling a ambas clases. El contraste se realiza con los resultados obtenidos de la combinación de las técnicas SMOTE+ Tomek Links, en esta técnica se realiza el sobre muestreo en la clase minoritaria y luego se aplica el Tomek Links a ambas clases.

En el proceso de ajuste del modelo, se contrastarán los resultados obtenidos al aplicar las mismas técnicas de re-muestreo sobre el conjunto de datos PCA de la sección anterior.

Validación cruzada

Durante el desarrollo, y particularmente cuando los datos de entrenamiento son escasos, se puede usar una práctica llamada validación cruzada para entrenar y validar un algoritmo con los mismos datos. En la validación cruzada, los datos de entrenamiento están divididos. El algoritmo se entrena usando todas menos una de las particiones, y se prueba en la partición restante. Las particiones se rotan varias veces para que el algoritmo se entrene y evalúe en todos los datos. Esta técnica nos ayudará a medir el comportamiento del modelo que creamos y nos ayudará a encontrar un mejor modelo rápidamente. (aprendemachinelearning, 2020)

Técnicas de validación cruzada

K-fold, el procedimiento de validación tiene un único parámetro llamado k que se refiere al número de grupos en los que se dividirá el conjunto de entrenamiento. Como tal, el procedimiento a menudo se denomina validación cruzada de k veces. Cuando se elige un valor específico para k, puede usarse en lugar de k en la referencia al modelo, por ejemplo, k = 10 se convierte en una validación cruzada de 10 veces. (aprendemachinelearning, 2020)

Stratified K-fold, es una variante mejorada de K-fold, que cuando hace las divisiones del conjunto de entrenamiento tiene en cuenta mantener equilibradas las clases. Esto es muy útil, porque si tenemos que clasificar en “SI/NO” y si una de las iteraciones del K-fold normal tuviera muestras con etiquetas sólo “SI” el modelo no podría aprender a generalizar y aprenderá para cualquier observación a responder “SI”. Esto lo soluciona el Stratified K-fold. (aprendemachinelearning, 2020)

Métricas de evaluación del rendimiento del modelo

La métrica de evaluación es una representación numérica que nos indica que también el modelo realiza su tarea, en este caso, de predicción, generalmente entre mayor es su valor, mejor es el rendimiento del modelo.

En el caso de tareas de clasificación con datos desbalanceados es común encontrar en artículos científicos (Jeni, Cohn, & De la Torre) y sitios web algunas de las siguientes :

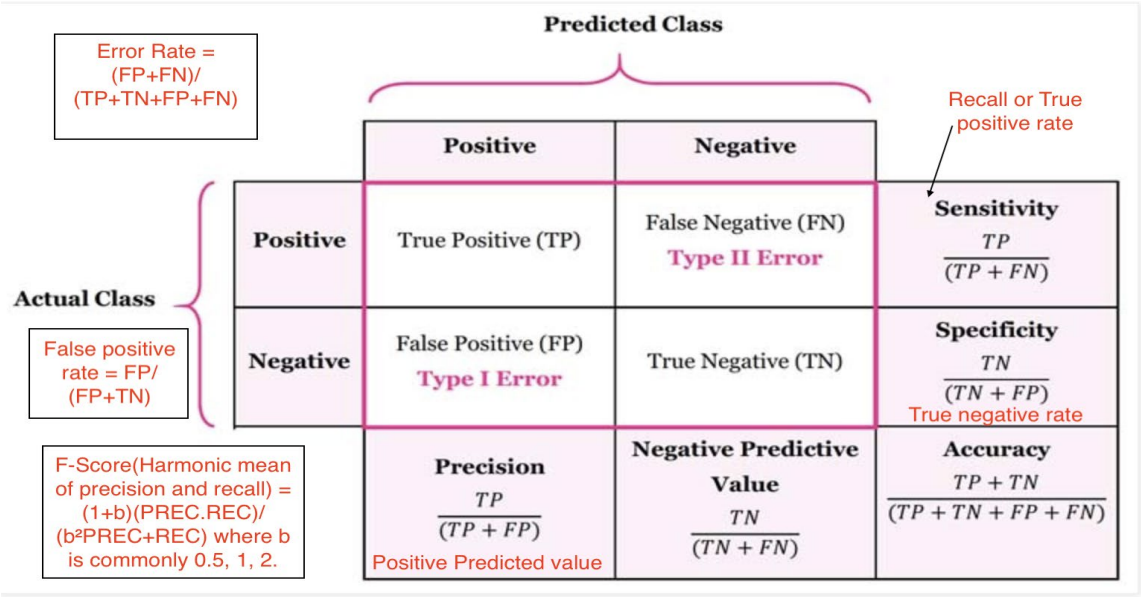
Métricas basadas en umbrales

Las métricas de umbral son aquellas que cuantifican los errores de predicción de clasificación. Es decir, están diseñadas para resumir la fracción, la proporción o la tasa de cuando una clase predicha no coincide con la clase esperada sobre un conjunto de datos dado.

La mayoría de las métricas de umbral se pueden comprender mejor mediante los términos utilizados en una matriz de confusión para un problema de clasificación binario (de dos clases). Esto no significa que las métricas estén limitadas para su uso en la clasificación binaria; es solo una manera fácil de comprender rápidamente lo que se está midiendo. (Brownlee, 2020c)

La matriz de confusión proporciona más información no solo sobre el rendimiento de un modelo predictivo, sino también sobre qué clases se predicen correctamente, cuáles incorrectamente y qué tipo de errores se están cometiendo. En este tipo de matriz de confusión, cada celda de la tabla tiene un nombre específico y bien entendido, que se resume a continuación:

Figura 14: Matriz de confusión



Fuente: https://medium.com/@cmukesh8688/evaluation-machine-learning-by-confusion-matrix-a4196051cf8d

Accuracy: fracción entre las predicciones correctas y el total de las predicciones

Correct Predictions / Total Predictions

Precision: La precisión resume la fracción de ejemplos asignados a la clase positiva que pertenecen a la clase positiva.

TruePositive / (TruePositive + FalsePositive)

Recall: El recall resume qué tan bien se predijo la clase positiva y es el mismo cálculo que la sensibilidad.

TruePositive / (TruePositive + FalseNegative)

F-Measure: La precisión y el recall se pueden combinar en una única puntuación que busca equilibrar ambos conceptos, denominada puntuación F o medida F.

(2 * Precision * Recall) / (Precision + Recall)

Kappa: En palabras simples, el coeficiente Cohen Kappa indica cuánto mejor es el modelo sobre el clasificador aleatorio que predice en función de las frecuencias de clase.

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

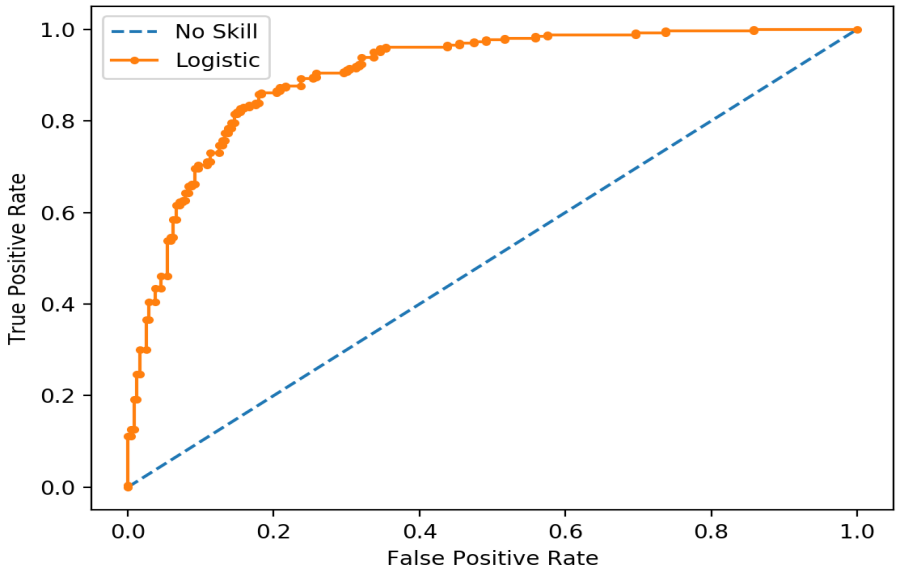
Para calcularlo, es necesario calcular dos cosas: "acuerdo observado" (po) y "acuerdo esperado" (pe). El acuerdo observado (po) es simplemente cómo nuestras predicciones del clasificador concuerdan con la verdad básica, lo que significa que es solo precisión. La concordancia esperada (pe) es cómo las predicciones del clasificador aleatorio que muestra según las frecuencias de clase concuerdan con la verdad básica o la precisión del clasificador aleatorio. (Czakon, 2019)

Métricas basadas en clases o rangos

Las métricas de rango están más relacionadas con evaluar clasificadores en función de su eficacia para separar clases. Estas métricas requieren que un clasificador prediga una puntuación o una probabilidad de pertenencia a una clase. (Brownlee, 2020c)

ROC Curve (AUC_ROC): Una curva ROC (o curva característica de funcionamiento del receptor) es un gráfico que resume el rendimiento de un modelo de clasificación binaria en la clase positiva. El eje x indica la tasa de falsos positivos y el eje y indica la tasa de verdaderos positivos.

Figura 15: Curva ROC



Fuente: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/

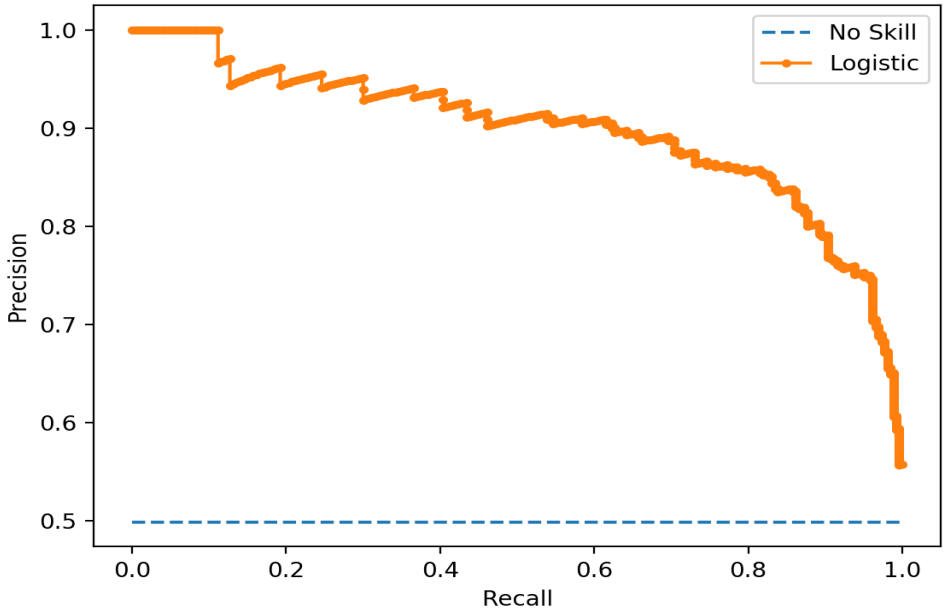
Idealmente, se quiere que la fracción de predicciones de clases positivas correctas sea 1 (parte superior de la gráfica) y la fracción de predicciones de clases negativas incorrectas sea 0 (izquierda de la gráfica). Esto resalta que el mejor clasificador posible que logra una habilidad perfecta es la parte superior izquierda de la gráfica (coordenada 0,1). (Brownlee, 2020d).

Aunque la curva ROC es una herramienta de diagnóstico útil, puede resultar difícil comparar dos o más clasificadores en función de sus curvas.

En cambio, el área debajo de la curva se puede calcular para dar una sola puntuación para un modelo de clasificación en todos los valores de umbral. Esto se denomina área bajo la curva ROC o AUC ROC o, a veces, ROCAUC. La puntuación es un valor entre 0.0 y 1.0 para un clasificador perfecto. (Brownlee, 2020d)

Precision – Recall Curve (AUC_PR): Es un gráfico de la precisión (eje y) y el recall (eje x) para diferentes umbrales de probabilidad.

Figura 16: Curva PR-RC



Fuente: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/

Un modelo con habilidad perfecta se representa como un punto en una coordenada de (1,1). Un modelo hábil está representado por una curva que se inclina hacia una coordenada de (1,1). Un clasificador sin habilidad será una línea horizontal en la gráfica con una precisión que es proporcional al número de ejemplos positivos en el conjunto de datos. Para un conjunto de datos equilibrado, será 0,5.

El enfoque de la curva PR en la clase minoritaria la convierte en un diagnóstico eficaz para modelos de clasificación binaria desbalanceados. (Brownlee, 2020d)

Precision-Recall AUC es como el ROC AUC, ya que resume la curva con un rango de valores de umbral como una sola puntuación. La puntuación se puede utilizar como punto de comparación entre diferentes modelos en un problema de clasificación binaria donde una puntuación de 1.0 representa un modelo con habilidad perfecta. (Brownlee, 2020d)

Observación: Si bien el accuracy es utilizado ampliamente en la evaluación del rendimiento de modelos predictivos, en este caso, dado el desbalance de la variable objetivo, se convierte en una medida engañosa, pues el modelo explicado desde este punto de vista no representa el comportamiento esperado de la realidad modelada porque prevalecen las predicciones sobre la clase mayoritaria.

En nuestro caso, se utiliza la métrica AUC_ROC para evaluar el rendimiento del modelo sobre los datos de entrenamiento, y las demás métricas se analizan sobre el conjunto de datos de prueba para contrastar los resultados sobre las predicciones. (Brownlee, 2020c)

3.4.2 Evaluar la calidad de predicción del modelo

Una se vez se obtiene el modelo ajustado sobre los datos de entrenamiento, el siguiente paso consiste en evaluar el rendimiento del mismo sobre los datos de prueba con el objetivo de medir la precisión del clasificador seleccionado para realizar pronósticos.

Para la evaluación del modelo de regresión logística se utilizan las métricas descritas en la sección 3.4.1 con los datos obtenidos del algoritmo Boruta (13 variables) y una versión reducida de los mismos empleando PCA (6 variables). De igual forma se incluyen las dos técnicas de re-muestreo expuestas con anterioridad para el tratamiento de los datos desbalanceados.

Tabla 1: Rendimiento del modelo de regresión logística sobre datos de prueba

AUC_ROC	**Boruta**	**PCA**
SMOTE + RU	0.825	0.822
SMOTE + Tomek Links	0.827	0.825

AUC_PR	**Boruta**	**PCA**
SMOTE + RU	0.081	0.078
SMOTE + Tomek Links	0.082	0.078

Kappa	**Boruta**	**PCA**
SMOTE + RU	0.099	0.097
SMOTE + Tomek Links	0.072	0.074

Accuracy	**Boruta**	**PCA**
SMOTE + RU	0.864	0.887
SMOTE + Tomek Links	0.804	0.808

Precision (binary)	**Boruta**	**PCA**
SMOTE + RU	0.068	0.068
SMOTE + Tomek Links	0.053	0.054

Recall (binary)	**Boruta**	**PCA**
SMOTE + RU	0.632	0.513
SMOTE + Tomek Links	0.704	0.707

F-measure (binary)	Boruta	PCA
SMOTE + RU	0.124	0.121
SMOTE + Tomek Links	0.098	0.101

Fuente: Elaboración Propia

Aunque los resultados obtenidos sobre ambos conjuntos de datos para las dos técnicas de re-muestreo son similares, el modelo evaluado sobre el conjunto de datos original utilizando SMOTE + RU, presenta en la mayoría de las métricas evaluadas los mejores indicadores.

El recall obtenido con SMOTE + Tomek Links presenta el mejor indicador para la clase positiva (claim='yes'), se castiga un poco el accuracy global del modelo, es decir, la cantidad de falsos negativos tiende a incrementarse con esta técnica de re-muestreo. Finalmente lo que se busca es no incrementar el número de instancias predichas que no pertenecen a su clase, lo ideal es mantener un balance adecuado, predecir una gran cantidad observaciones que no presentan reclamaciones como si las tuviesen, tampoco es un buen indicador para el modelo.

3.4.3 Refinamiento del modelo: optimización por hiper parámetros

El modelo ajustado en la sección anterior utiliza la configuración por defecto provista en sklearn para el método de regresión logística. En este sentido, suele ser bastante aceptable para la construcción del modelo en la mayoría de los casos.

Así, la optimización por hiper parámetros tiene como objetivo encontrar el conjunto de restricciones bajo los cuales el modelo obtiene su mejor rendimiento en función de las métricas de evaluación establecidas.

Un procedimiento de optimización implica definir un espacio de búsqueda. Esto se puede considerar geométricamente como un volumen de n dimensiones, donde cada hiperparámetro representa una dimensión diferente y la escala de la dimensión son los valores que puede tomar el hiperparámetro, como valores reales, valores enteros o categóricos.

Espacio de búsqueda: Volumen que se buscará donde cada dimensión representa un hiperparámetro y cada punto representa una configuración de modelo.

Un punto en el espacio de búsqueda es un vector con un valor específico para cada valor de hiperparámetro. El objetivo del procedimiento de optimización es encontrar un vector que dé como resultado el mejor rendimiento del modelo después del aprendizaje, como máxima precisión o mínimo error.

Se puede utilizar una variedad de algoritmos de optimización diferentes, aunque dos de los métodos más simples y comunes son la búsqueda aleatoria y la búsqueda en cuadrícula.

Búsqueda aleatoria (random search). Define un espacio de búsqueda como un dominio limitado de valores de hiperparámetros y puntos de muestreo aleatorios en ese dominio.

Búsqueda de cuadrícula (grid search). Define un espacio de búsqueda como una cuadrícula de valores de hiperparámetros y evalua cada posición en la cuadrícula.

Para efectos de este trabajo, se utiliza la búsqueda de cuadricula. A continuación se presentan los resultados sobre el modelo original:

Parámetros evaluados

'C' = [100, 50, 10, 1 , 0.1, 0.001, 0.0001]
'solver' = ['lbfgs','saga','liblinear','newton-cg']
'penalty' = ['l1', 'l2', 'elasticnet']

Mejor conjunto de parámetros obtenido sobre datos de entrenamiento

Best: 0.820269 using {'C': 1, 'penalty': 'l1', 'solver': 'saga'}

Tabla 2: Optimización hiper parámetros: métricas evaluadas sobre conjunto de datos de prueba

	AUC ROC	AUC PR	Accuracy	Kappa	Precision	Recall	F1
Regresión Logística sin ajuste de parámetros	0.825	0.081	0.864	0.099	0.068	0.632	0.124
Regresión Logística con ajuste de parámetros	0.825	0.081	0.864	0.099	0.068	0.632	0.124

Fuente: Elaboración Propia

Las métricas obtenidas para el conjunto de parámetros optimizado no representan una mejora significativa respecto al modelo de regresión logística inicial. En este sentido, el modelo original sin ajuste de parámetros ofrece un rendimiento adecuado para esta tarea de clasificación.

3.5 Combinación de clasificadores

En el ámbito del aprendizaje automático, para las tareas de clasificación es común ajustar un modelo con sus respectivos parámetros sobre un conjunto de datos en particular o la combinación de un conjunto de clasificadores, de manera que se obtenga una buena aproximación cercana a la realidad que se busca representar. Otra de las aproximaciones, consiste en combinar un conjunto de clasificadores mas o menos sencillos, para crear uno más complejo, de forma que la decisión tomada sea una combinación de cientos o miles de decisiones parciales. Obviamente, los clasificadores usados como base para construir el clasificador combinado deben ser lo más diversos posible. De este modo, los errores cometidos por un clasificador base concreto sean minoritarios con respecto al resto, de forma que los errores puntuales no alteren una decisión basada en la opinión correcta de la mayoría. (Gironés Roig et al., 2017)

En este apartado, se realiza una exploración de esta aproximación empleando algunos de los métodos de combinación de clasificadores

Combinación paralela de clasificadores base igual

Existen dos métodos característicos para este tipo de clasificadores.

Bagging: La idea esencial del bagging es promediar muchos modelos ruidosos pero aproximadamente imparciales, y por tanto reducir la variación. Los árboles de decisión son los candidatos ideales para el bagging, dado que ellos pueden registrar estructuras de interacción compleja en los datos, y si crecen suficientemente profundo, tienen relativamente baja parcialidad. Producto de que los árboles son notoriamente ruidosos, ellos se benefician enormemente al promediar. El principal objetivo intrínseco de los algoritmos de bagging es el de la reducción de la varianza.

El algoritmo Random Forest es uno de los más conocidos en esta categoría de clasificadores. Posteriormente será analizado sobre el conjunto de datos objeto de nuestro análisis.

Boosting: consiste en combinar los resultados de varios clasificadores débiles para obtener un clasificador robusto. Cuando se añaden estos clasificadores débiles, se lo hace de modo que estos tengan diferente peso en función de la exactitud de sus predicciones. Luego de que se añade un clasificador débil, los datos cambian su estructura de pesos: los casos que son mal clasificados ganan peso y los que son clasificados correctamente pierden peso. Así, los clasificadores débiles se centran de mayor manera en los casos que fueron mal clasificados por los clasificadores débiles. El principal objetivo de los algoritmos de boosting es el de la reducción del sesgo.

Los algortimos Gradient Boosting y eXtreme Gradient Boosting son algunos de los más conocidos en este categoría, posteriormente también evaluaremos sobre nuestro conjunto de datos.

Combinación secuencial de clasificadores base diferente

En función de la información que recibe el clasificador combinado de los clasificadores parciales, se distinguen dos métodos llamados stacking y cascading.

Stacking: la idea básica es construir diferentes clasificadores base de forma que cada uno de ellos genere una decisión parcial. Entonces se construye un nuevo clasificador usando como datos de entrada todas las predicciones parciales, en lugar de los datos originales de entrada. Este segundo clasificador suele ser un árbol de decisión, una red neuronal sencilla o una regresión logística.

Cascading: es similar al stacking pero no solo utiliza las predicciones parciales de los clasificadores base, sino también los datos originales e incluso otros datos que se hayan podido generar durante la toma de decisiones. La idea básica es alimentar al clasificador combinado con decisiones parciales, así como los motivos que han llevado a tomar dichas decisiones. (Gironés Roig et al., 2017)

Para efectos de este trabajo, se realiza una aproximación al método de stackking utilizando la clase StackingClassifier de python.

Combinación de clasificadores para datos desbalanceados

Un segundo enfoque en el contexto de las tareas de clasificación sobre datos desbalanceados corresponde a los algoritmos modificados. Este tipo de algoritmos están diseñados para operar nativamente sobre datos desbalanceados sin necesidad de aplicar técnicas de re-muestro sobre los mismos.

El proyecto imbalanced learn para python (Lemaitre, et al., 2017), provee versiones modificadas de clasificadores combinados para ajustar modelos con datos desbalanceados. En este apartado se exploran los clasificadores:

- BalancedBaggingClassifier
- BalancedRandomForestClassifier
- EasyEnsembleClassifier

Sobre los cuales validamos su pertinencia y efectividad para abordar este tipo de problemáticas.

Como variante a esta metodología, también se exploran un par de algoritmos que utilizan parámetros de penalización para datos desbalanceados al inicializar el clasificador como: class_weight y scale_pos_weight en los algoritmos Random Forest y XGBoost respectivamente.

A continuación se presentan los resultados de aplicar algunas de las técnicas mencionadas sobre el conjunto de datos derivado del algoritmo Boruta, con y sin re-muestreo sobre los datos de entrenamiento para el ajuste de los respectivos clasificadores.

Tabla 3: Resultados obtenidos para cada una de las métricas evaluadas sobre los clasificadores combinados con base igual

Clasificador/Métrica	AUC ROC	AUC PR	Accuracy	Kappa	Precision	Recall	F1
BaggingClassifier SMOTE+ RU	0.775	0.061	0.882	0.075	0.057	0.434	0.100
BalancedBaggingClassifier	0.797	0.067	0.807	0.066	0.050	0.648	0.093
RandomForestClassifier SMOTE+ RU	0.777	0.058	0.882	0.079	0.059	0.451	0.104
RandomForestClassifier (class_weight='balanced')	0.693	0.046	0.976	0.025	0.047	0.030	0.036
BalancedRandomForestClassifier	0.806	0.069	0.731	0.050	0.041	0.740	0.077
GradientBoostingClassifier SMOTE + RU	0.819	0.073	0.892	0.105	0.073	0.530	0.129
XGBClassifier (scale_pos_weight)	0.723	0.051	0.906	0.072	0.056	0.329	0.096
XGBClassifier SMOTE + RU (scale_pos_weight)	0.776	0.066	0.875	0.082	0.060	0.497	0.107
EasyEnsembleClassifier	0.823	0.067	0.764	0.059	0.045	0.727	0.045

Fuente: Elaboración Propia

Partiendo de un razonamiento similar al de la sección 3.4.2, el modelo que presenta un rendimiento adecuado y balanceado para esta tarea de clasificación es el BalancedBaggingClassifier sin aplicar re-muestreo sobre los datos de entrenamiento, sus resultados son muy similares a los obtenidos en el modelo de regresión logística con re-muestreo sobre los datos de entrenamiento.

Tabla 4: Resultados obtenidos al aplicar la técnica de stacking sobre el conjunto de datos boruta más re-muestro

Clasificador/Métrica	AUC ROC	AUC PR	Accuracy	Kappa	Precision	Recall	F1
DecisionTreeClassifier	0.634	0.232	0.846	0.049	0.041	0.414	0.075
SVC	0.780	0.064	0.865	0.099	0.068	0.628	0.123
KNeighborsClassifier	0.741	0.137	0.830	0.062	0.048	0.543	0.088
StackingClassifier final_estimator=LogisticRegression	0.768	0.057	0.883	0.083	0.061	0.467	0.108

Fuente: Elaboración Propia

El conjunto de clasificadores base comprende los algoritmos DecisionTreeClassifier, SVC y KNeighborsClassifier, como estimador final se utiliza LogisticRegression. En este caso el modelo stacking permite que la cantidad de falsos negativos disminuya respecto a los clasificadores base, por lo cual su métrica accuracy es la mejor luego

del proceso de combinación, sin embargo el recall obtenido no es el mejor, de las reclamaciones efectivas en el conjunto de prueba (claim='yes') clasifica como correctas, un poco menos de la mitad de las observaciones. La técnica de stacking seleccionada para este conjunto de datos es probable que no sea la más adecuada, el modelo obtenido al aplicar SVC sobre el conjunto de datos parece arrojar mejores resultados en este sentido.

3.6 Modelos predictivos con AutoML

El aprendizaje automático automatizado, o AutoML para abreviar, implica la selección automática de la preparación de datos, el modelo de aprendizaje automático y los hiperparámetros del modelo para una tarea de modelado predictivo.

Se refiere a técnicas que permiten a los profesionales del aprendizaje automático semi-sofisticados y a los no expertos descubrir rápidamente una buena canalización de modelos predictivos para su tarea de aprendizaje automático, con muy poca intervención más que proporcionar un conjunto de datos.

En esta sección vamos a explorar en términos generales como operan este tipo de técnicas sobre datos desbalanceados en la variable objetivo, para ello, se utilizará el conjunto de datos de entrenamiento extraído a partir de la aplicación del algorimo boruta en adición con la técnica de re-muestreo SMOTE – RandomUnderSampling, y se contrastan los resultados utilizando el conjunto de datos de entrenamiento original sin aplicar estas 2 técnicas, es decir, el conjunto de datos con 191 variables sin re-muestreo, con el fin de analizar la veracidad o ingerencia del concepto AutoML sobre este tipo de datos.

Para el lenguaje python, se han implementado tres de los más populares proyectos open source que incorporan esta metodología de trabajo, sobre los cuales en este apartado se realizará una exploración básica de los resultados obtenidos como parte de las tareas de optimización inherentes a este paradigma, estos tres proyectos son: Hyperopt-Sklearn, Auto-Sklearn, y TPOT. (Brownlee, 2020e)

3.6.1 Auto-Sklearn

Es un proyecto de python desarrollado por Matthias Feurer que utiliza modelos de aprendizaje automático de la librería sklearn. (Feurer, 2015).

Auto-Sklearn busca automáticamente el algoritmo de aprendizaje adecuado para un nuevo conjunto de datos y optimiza sus hiper parámetros. Por lo tanto, libera al usuario de estas tediosas tareas y le permite concentrarse en el problema real.

Documentación del proyecto: https://automl.github.io/auto-sklearn/master/

3.6.2 Hyperopt-Sklearn

HyperOpt es una libreria de python de código abierto basado en optimización bayesiana desarrollada por James Bergstra. (Bergstra et al., 2013).

Está diseñado para la optimización a gran escala de modelos con cientos de parámetros y permite escalar el procedimiento de optimización en múltiples núcleos y múltiples máquinas. Además es utilizada explícitamente para optimizar pipelines de machine learning, incluyendo la preparación de los datos, la selección de modelos y la optimización de los hiper parámetros.

Documentación del proyecto: http://hyperopt.github.io/hyperopt-sklearn/

3.6.3 Tree-based Pipeline Optimization Tool (TPOT)

Es una librería de código abierto para realizar AutoML en python. Hace uso de la popular biblioteca de aprendizaje automático Scikit-Learn para transformaciones de datos y algoritmos de aprendizaje automático y utiliza un procedimiento de búsqueda global estocástica de programación genética para descubrir de manera eficiente un modelo de alto rendimiento para un conjunto de datos determinado. (Olson et al., 2016)

TPOT utiliza una estructura basada en árboles para representar un modelo base con sus principales transformaciones y parámetros (model pipeline) para un problema de modelado predictivo, incluye la preparación de datos y los algoritmos de modelado, al igual que los respectivos hiper parámetros.

Documentación del proyecto: https://epistasislab.github.io/tpot/

A continuación se presentan los resultados de la aplicación de las tres librerías mencionadas sobre el conjunto de datos derivado del algoritmo boruta con re-muestreo (13 variables) y el conjunto de datos original (191 variables).

Tabla 5: AutoML, resultados para cada uno de los modelos presentados con su respectiva métrica de evaluación sobre los datos de prueba

Clasificador/Métrica	AUC ROC	AUC PR	Accuracy	Kappa	Precision	Recall	F1
TPOTClassifier I	0.827	0.080	0.985	0.000	0.000	0.000	0.000
TPOTClassifier II	0.765	0.063	0.886	0.079	0.059	0.431	0.103
HyperoptEstimator I	0.837	0.104	0.985	0.000	0.000	0.000	0.000
HyperoptEstimator II	0.752	0.071	0.895	0.071	0.055	0.365	0.096
AutoSklearnClassifier I	0.831	0.079	0.985	0.000	0.000	0.000	0.000
AutoSklearnClassifier II	0.778	0.060	0.848	0.065	0.050	0.500	0.091

Fuente: Elaboración Propia

TPOTClassifier I:
MLPClassifier(SelectPercentile(input_matrix,percentile=9), alpha=0.0001, learning_rate_init=0.001) , sobre data original

TPOTClassifier II:
ExtraTreesClassifier(input_matrix,bootstrap=False, criterion=entropy, max_features=0.55, min_samples_leaf=1, min_samples_split=4, n_estimators=100), sobre data boruta + re-sample

HyperoptEstimator I:
AdaBoostClassifier(algorithm='SAMME', learning_rate=0.12351405659014378, n_estimators=326, random_state=2), 'preprocs': (StandardScaler(with_mean=False),) , sobre data original

HyperoptEstimator II:
GradientBoostingClassifier(learning_rate=0.44711445571797426, max_depth=None, max_features=0.34461637289486735, min_samples_leaf=3, n_estimators=322, random_state=3, subsample=0.9440637356264416), sobre data boruta + re-sample

AutoSklearnClassifier I: sobre data original

AutoSklearnClassifier II: sobre data boruta + re-sample

Es posible apreciar que sobre los datos originales, ninguno de los clasificadores logró predecir reclamaciones efectivas como verdaderas (claim='yes'), por lo que concluimos que su efectividad sobre datos altamente desbalanceados no es buena, los modelos generados solo tienen la capacidad de predecir la clase negativa.

En contraste, los resultados obtenidos sobre los datos con boruta + resample si arrojan modelos con mejor rendimiento y, a que diferencia de los primeros, si tienen la capacidad de predecir reclamaciones efectivas como verdaderas, aunque no superan los indicadores que presentan, por ejemplo, el modelo de regresión logística o el clasificador bagging balanceado de la librería imblearn.

4. Conclusiones

El análisis realizado sobre el conjunto de datos *travel insurance* empleando la metodología y técnicas del aprendizaje automático permitió identificar hallazgos relevantes:

- Previo al análisis exploratorio de los datos, se parte del supuesto que el problema a estudiar consistía en una tarea clásica de clasificación binaria con datos balanceados. Posterior a este análisis, se comprueba que el supuesto no era válido, por lo que fue necesario ajustar la planificación del trabajo para su fase de implementación e incorporar a la metodología nuevos conceptos y técnicas para el tratamiento de datos desbalanceados sobre la variable objetivo, que no fueron contemplados al inicio del proyecto.

- El modelamiento de tareas de clasificación sobre datos altamente desbalanceados, en la cual la clase minoritaria (claim = Yes), representaba menos de 1 % de los datos analizados. Para este fin, se exploraron dos técnicas propuestas en la literatura que permiten abordar adecuadamente este tipo de modelamiento: el re-muestreo sobre los datos de entrenamiento y los algoritmos de clasificación modificados para el procesamiento de datos desbalanceados.

- El modelo analítico con el mejor rendimiento y balance para predecir la clase minoritaria, acorde a los criterios de evaluación establecidos, fue el de Regresión Logística utilizando la técnica de re-muestreo SMOTE – RandomUndersampling. Se hizo especial énfasis en los resultados de evaluación obtenidos a partir de las matrices de confusión y las métricas de recall, precision , f1 y kappa sobre los datos de prueba y la métrica AUC ROC para los datos de entrenamiento como métrica global de evaluación, la cual considera relevante tanto la clase minoritaria como la mayoritaria.

- Los resultados obtenidos sobre el conjunto de pruebas con las diferentes técnicas analizadas ofrecen unos nivel de precisión alrededor del 64% sobre la clase minoritaria y superior al 80% sobre la clase mayoritaria, aunque no son métricas muy altas en comparación a las tareas de clasificación con datos balanceados, son indicadores aceptables dada la complejidad que trae consigo este tipo de tareas de clasificación desbalanceadas, mas aún sobre conjuntos de datos que presentan una variedad en su estructura, al considerar atributos numéricos y categóricos con diferentes características en las distribución de sus datos.

- La metodología utilizada para la selección del mejor modelo predictivo fue adecuada porque permitió analizar una serie técnicas para el tratamiento de datos desbalanceados, explorar un amplio conjunto de algoritmos de clasificación y contrastar sus respectivas métricas de rendimiento, de manera que la decisión para elegir el mejor modelo no quedase sesgada a una técnica de preferencia o fuese fruto del azar.

- Como se referenció en el capítulo del estado del arte, las metodologías emergentes como AutoML parecen ser prometedoras en cuanto a la simplificación de algunas de las tareas inherentes al proceso del aprendizaje automático, sin embargo, para este caso de estudio los resultados no son lo suficientemente buenos en comparación con los métodos tradicionales utilizados. AutoML es un concepto en constate evolución que posiblemente en algunos años tendrá un mayor nivel de madurez y se ajustará adecuadamente a este tipo de problemas con datos desbalanceados.

- Si bien los resultados obtenidos con las técnicas de re-muestreo para el tratamiento de la clase desbalanceada fueron buenos, como línea de trabajo futuro se sugiere explorar en mayor detalle el uso de algoritmos modificados para este tipo de distribución de datos en la clase objetivo, haciendo especial énfasis en la optimización de sus hiper parámetros. En el presente trabajo se utilizaron los valores por defecto solo para algunos de los algoritmos provistos por la librería de python imblearn, sin embargo existen en la literatura científica una cantidad considerable de estudios relacionados a este tipo de tareas clasificación con resultados bastante buenos, por mencionar algunos ejemplos, versiones modificadas del método XGBoost y algoritmos probabilísticos avanzados no abordados en este trabajo.

5. Bibliografía

Álvarez Díaz, S. (2017-2018). *TFM: Análisis del Big Data en los Seguros: Modelos Predictivos. Máster Universitario en Ciencias Actuariales y Financieras, Facultas de Ciencias Económicas y Empresariales, Universidad de León.*

Bergstra, J., Yamins D. & Cox, D.D. (2013). *Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures*. In Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28. JMLR.org, I-115-I-123.

Brownlee, J. (5 de Junio de 2020a). *How to Perform Feature Selection With Numerical Input Data*. Obtenido de Machine Learning Mastery: https://machinelearningmastery.com/feature-selection-with-numerical-input-data/

Brownlee, J. (17 de Junio de 2020b). *What Is Data Preparation in a Machine Learning Project*. Obtenido de https://machinelearningmastery.com/what-is-data-preparation-in-machine-learning/

Brownlee, J. (8 de enero de 2020c). *Tour of Evaluation Metrics for Imbalanced Classification*. Obtenido de Machine Learning Mastery: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

Brownlee, J. (16 de septiembre de 2020d). *ROC Curves and Precision-Recall Curves for Imbalanced Classification*. Obtenido de https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/

Brownlee, J. (18 de septiembre de 2020e). *Automated Machine Learning (AutoML) Libraries for Python*. Obtenido de https://machinelearningmastery.com/automl-libraries-for-python/

Cardenas, M. d. (2019). Mejoras en la clasificación de interacciones de proteínas de secuencias de la Arabidopsis Thaliana utilizando técnicas de bases de datos desbalanceadas. *Revista Cubana de ciencias informaticas* .

Cetinsoy, A. (2016). The Past, Present, and Future of Machine Learning APIs. *JMLR: Workshop and Conference Proceedings*, 50:43–49.

Czakon, J. (28 de agosto de 2019). *The ultimate guide to binary classification metrics*. Obtenido de https://towardsdatascience.com/the-ultimate-guide-to-binary-classification-metrics-c25c3627dd0a

Dutta, D. (22 de Marzo de 2016). *How to perform feature selection (i.e. pick important variables) using Boruta Package in R ?* Obtenido de https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/

Fauzan, M. A., & Murfi, H. (2018). *The Accuracy of XGBoost for Insurance Claim Prediction. Int. J. Advance Soft Compu. Appl, Vol. 10, No. 2. ISSN 2074-8523.*

Feurer, Matthias. (2015). *Efficient and Robust Automated Machine Learning.*

Frempong, N. K., Nicholas, N., & Boateng, M. A. (2017). *Decision Tree as a Predictive Modeling Tool for Auto Insurance Claims. International Journal of Statistics and Applications 2017, 7(2):117-120.* doi:10.5923/j.statistics.20170702.07

Gharamani, Z. (2018). *Automatic Machine Learning: Methods, Systems, Challenges.* San Francisco: University of Cambridge.

Gironés Roig, J., Casas Roma, J., Minguillón Alfonso, J., & Caihuelas Quiles, R. (2017). *Mineria de datos: modelos y algoritmos.* Barcelona: Editorial UOC.

Henke N, B. J. (diciembre de 2016). *McKinsey Global Institute.* Obtenido de https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/the-age-of-analytics-competing-in-a-data-driven-world.

IT, R. (3 de Agosto de 2018). *Reclu IT*. Obtenido de Reclu IT: https://recluit.com/historia-y-evolucion-del-machine-learning/#.X683bmhKjIU

Jeni, L. A., Cohn, J. F., & De la Torre, F. (s.f.). *Facing Imbalanced Data: Recommendations for the Use of Performance Metrics.*

Kent, G. (Marzo de 2019). *Adext AI*. Obtenido de https://blog.adext.com/machine-learning-guia-completa/#El_nacimiento_del_machine_learning

Kuriyak, S. (7 de noviembre de 2017). *Producia*. Obtenido de https://blog.produvia.com/artificial-intelligence-ai-in-insurance-d9035ea2d0b9

Kursa, M. B., & Rudnicki, W. R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software, 36(11)*, 1-13.

Kuznetsov, V. (s.f.). *Indico*. Obtenido de https://indico.cern.ch/event/822631/contributions/3476195/attachments/1871522/3079864/MLaaS4HEPTalk.pdf

Lemaitre, G., Nogueira, F., Oliveira, D., & C., A. (2017). *imbalanced-learn*. Obtenido de https://imbalanced-learn.readthedocs.io/en/stable/install.html

Mayorga, L. (25 de septiembre de 2018). *zonaia*. Obtenido de https://zonaia.com/tutoriales-machine-learning/datos-de-entrenamiento-y-prueba/

Mendes Antunes, J., de Valeriola, S., Mahy, S., & Maréchal, X. (2017). *Machine Learning application to non-life pricing. Frecuency modelling: An educational case study. Reacfin.*

Mohamed M. Shoukri, T. A.-H. (2016). Bias and Mean Square Error of Reliability Estimators under the One and Two Random Effects Models: The Effect of Non-Normality. *Open Journal of Statistics*.

Ogunnaike, R. M., & Si, D. (2017). *Prediction of Insurance Claim Severity Loss Using Regression Models.* doi:10.1007/978-3-319-62416-7 17

Olson, Randal S., Bartley, Nathan., Urbanowicz, Ryan J. & Moore, Jason H. (2016). *Evaluation of a Tree-based Pipeline Optimization Tool for Data Science*. In Proceedings of the Genetic and Evolutionary Computation Conference 2016. Association for Computing Machinery, New York, NY, USA, 485–492. DOI:https://doi.org/10.1145/2908812.2908918.

Peláez, I. M. (2016). Modelos de regresión: lineal simple y lógistica. *Revista Seden*.

Quan, Z., & Valdez, E. A. (2018). *Predictive analytics of insurance claims using multivariate decisión trees.* doi: 10.1515/demo-2018-0022

Rahul Kumar Sevakula, W.-T. M.-Y. (2020). State-of-the-Art Machine Learning Techniques Aiming to Improve Patient Outcomes Pertaining to the Cardiovascular System. *Journal of the American Heart Association*.

Real, E. (2020). AutoML-Zero: Evolving Machine Learning Algorithms From Scratch. *Arxiv.org*.

Russell SJ, N. P. (2016). Artificial Intelligence; A modern Approach. *Pearson Education Limited*.

SHARMA, P. (27 de Agosto de 2018). *Analytics Vidhya*. Obtenido de https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/

Siddhant, S. (4 de Junio de 2020). *Feature selection using Boruta algorithm*. Obtenido de linkedin.com: https://www.linkedin.com/pulse/feature-selection-using-boruta-algorithm-shashwat-siddhant/?articleId=6674081448502398976

Todo BI. (5 de abril de 2019). Obtenido de https://todobi.com/una-breve-historia-del-machine-learning/

Universidad de Alcalá. (2019). *CONCEPTOS CLAVE DEL DEEP LEARNING*. Obtenido de https://master-deeplearning.com/conceptos-clave-deep-learning/

6. Anexos

Esta memoria, junto con el set de datos analizado y el notebook de Jupyter con la implementación del trabajo desarrollado en python se encuentra publicado en el repositorio público de github:

https://github.com/juanpabosu/TFM