INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE INGENIERÍA Y GESTIÓN

DETECCIÓN DE FRAUDES EN SEGUROS DE AUTOMOVILES UTILIZANDO ALGORITMOS DE MACHINE LEARNING

AUTOR: Fabbiano, Pablo Miguel (Leg. Nº 104873)

TUTOR: Denicolay, Gustavo

TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE ESPECIALISTA EN CIENCIA DE DATOS

BUENOS AIRES

PRIMER CUATRIMESTRE, 2020

ITBA

Contenidos

1. Introducción

El presente trabajo desarrolla una posible solución para detectar el fraude en siniestros en el Seguro de Automóviles, utilizando herramientas de bajo costo y un modelado de solución basado en datos.

La detección de fraudes en el Seguro Automotor se ha vuelto de vital importancia para reducir los costos de las Compañías de Seguros. La mayoría de las Aseguradoras adoptan conocimientos especializados para detectar el fraude. El conocimiento basado en la experiencia es interpretable y reutilizable, pero dicho conocimiento utilizado en la práctica conduce a algunos grados de error de juicio.

Este documento tiene como objetivo desarrollar una solución sobre la que basar la decisión de disparar un posterior proceso de investigación, ante una probabilidad razonable de intento de fraude de parte de un Asegurado.

La solución combina múltiples piezas de evidencia, indicadores basados en la experiencia y probabilidades de fraude obtenidos de datos históricos.

Se propone resolver el problema mediante la utilización de algoritmos de machine learning que permitan predecir el fraude en una etapa temprana del ciclo de vida de una Póliza de Seguros.

El impacto buscado alcanza a todo el Mercado Asegurador, dado que al bajar las tasas de siniestralidad evitando el fraude se logrará bajar la prima de los seguros, creando un Mercado más justo y transparente para beneficio de todos los involucrados.

2. Fraude en Seguros de Automóviles

Se define como **Fraude en Seguros** a toda acción u omisión, contraria a la verdad y a la rectitud, que perjudica a una de las partes contratantes de una póliza de seguro, beneficiando a una de ellas o a un tercero, es decir a todo engaño realizado con el objeto de obtener ilegítimamente un beneficio para sí mismo o para un tercero.

La Superintendencia de Seguros de la Nación (SSN) mediante la Resolución Nº 38477 del 17/07/2014, exige a las entidades supervisadas la adopción de normas sobre políticas, procedimientos y controles internos tendientes a combatir el fraude en los seguros.

El siguiente cuadro resume cuáles son los intentos de fraude habituales en Seguros de Automóviles:

Intentos de Fraude habituales (entrevista a expertos, ver texto en "Referencias Bibliografía [6]").	
Reclamos por lesiones de una misma persona ante diferentes Aseguradoras	Simular una lesión en el cuello tras un supuesto accidente de tráfico es una de las grandes estafas al seguro automotor nuestro país. Los latigazos cervicales son un tipo de lesión frecuente cuando sufrimos un golpe trasero y no resulta sencillo comprobar si el daño es real o no. El punto central radica en que se trata de una afección que se manifiesta en forma de dolor intenso, algo muy subjetivo que desde un punto de vista médico resulta difícil de probar si es verdad o no, por lo que el accidentado puede fingir.
Vehículos indemnizados por destrucción total en una Aseguradora…	*…y luego asegurados sin inspección previa en otra Compañía,* ante la cual se efectúa denuncia por robo.
Auto-robo	
Reclamos de terceros por lesiones…	…donde se *"arman"* los accidentes con la intervención de Asegurados, talleres, terceros, testigos, abogados, etc.
El préstamo de la póliza…	…es decir fraguar un hecho para lograr la reparación de un vehículo determinado sin cobertura.
indemnizaciones por lesiones originadas en sucesos que nada tienen que ver	Pretender lograr indemnizaciones por lesiones originadas en sucesos que nada tienen que ver con accidentes de autos.
Cuando se inventan o exageran los daños	Se declaran averías como si fueran siniestros. En estas situaciones la entidad deberá estudiar el nexo de causalidad entre la colisión y el accidente que figura en el parte con las lesiones y daños con especial detalle.
Presupuestos de arreglos recargados	Los Asegurados se confabulan con los talleristas para que el presupuesto del arreglo del coche por un siniestro sea por un monto que no se corresponde con los daños reales.
Encubrir consumos de alcohol o drogas	Lo cierto es que, en los últimos años, debido a que muchos accidentes peligrosos son a consecuencia del alcohol y las drogas consumidos por los conductores, se han endurecido los límites permitidos de consumo y las sanciones. En este

	contexto, hemos de resaltar que las compañías cuando reciben un parte de siniestro causado por la ingesta excesiva de alcohol no se suelen responsabilizar de los daños propios, aunque el Asegurado tenga contratado un seguro Todo Riesgo. Y el caso se agrava en las coberturas de responsabilidad civil porque tras abonar la indemnización correspondiente la Aseguradora puede reclamar a su Asegurado por este importe. En consecuencia, hay Asegurados que intentan tapar el consumo de alcohol y drogas al volante si están implicados en un percance.
Pluralidad de Seguros	
Mentir u ocultar datos a la compañía en el contrato	Otro de los engaños que se produce llega en el momento de la contratación de un seguro (proceso de suscripción). Las Compañías indican en sus condiciones de contratación que los datos que aporte el Asegurado deben ser reales ya que, en caso contrario, puede no brindar cobertura ante un siniestro.

Tabla 1 – Intentos de fraude habituales

Existen tres grandes grupos de defraudadores:

Grupo	Descripción
Bandas organizadas	Cuyos procedimientos desbordan generalmente la capacidad de actuación aislada de las Entidades. Muchas veces se cuenta con la complicidad activa o pasiva de las autoridades, lo cual dificulta desbaratar a estas organizaciones.
Profesionales	Defraudan en el ejercicio de su profesión. Entre ellos encontramos peritos, liquidadores, talleristas, abogados, productores de seguros, médicos y mediadores. Su actuación de forma ocasional puede convertirse en habitual cuando encuentran facilidades basadas en la confianza profesional para obtener beneficio extra.
Particulares	Aprovechan el contrato de seguros (póliza) como una solución a sus problemas económico financieros En ocasión de un siniestro real obtienen un beneficio extra. En este grupo se encuentran Tomadores de Seguro, Asegurados, terceros y beneficiarios.

Tabla 2 – Grupos de defraudadores

Si se realiza una distinción entre los autores materiales de los hechos, los fraudes cometidos por bandas organizadas se refieren, fundamentalmente, a siniestros con lesiones y auto-robo de automotores; entre los estafadores ocasionales es más común encontrar los casos de préstamos de póliza para cubrir a terceros sin cobertura.

Debido a las dificultades para la obtención de pruebas documentales, al lógico rigor probatorio exigido y a las dificultades para no incurrir en errores de procedimiento, el número de acciones fraudulentas detectadas que son llevadas

a la justicia es mínimo. Debe tenerse presente que el fraude al seguro no tiene una figura penal típica. Esta realidad implica que toda actividad investigativa tendiente a una posterior denuncia penal deba orientarse sobre la figura penal de la estafa; en consecuencia, todo relevamiento de pruebas resulta de muy difícil realización.

No existen datos ciertos acerca de cuánto afecta esta situación al pago de siniestros de las Aseguradoras. Para distintos analistas, el fraude representaría entre el 7% y el 10% del monto total pagado por el mercado en concepto de Siniestros Automotores.

Existen actualmente en el mercado distintas herramientas informáticas que permiten el cruce de datos y son de uso generalizado por todos los operadores de Seguros. El desarrollo de CESVI (Orion y Sofía) y el soporte que brinda la SSN a través de sus sistemas son ejemplos claros de ello.

Las redes sociales son herramientas que también pueden ser aliadas para la investigación de posibles fraudes en seguros. Un ejemplo de ello es una denuncia de un siniestro en el que estaban involucrados un automóvil y una moto. En la denuncia, se daba cuenta de un siniestro en el que participaban dos personas, pero al detectar un argumento llamativamente similar entre el Asegurado y el tercero, se inició una investigación para evaluar si se trataba de un fraude y se detectó, a través de las redes sociales, que entre ambos había una relación familiar directa, hecho que motivó el rechazo del siniestro.

También se trabaja con software que permite la detección de la edición digital de las fotografías. Ocurren situaciones, sobre todo en intentos de fraude con accidentes automotores, en las que se modifican digitalmente los datos del vehículo. Estos sistemas permiten detectar retoques a las fotografías originales, que son imperceptibles al ojo humano. Esta tarea se complementa con procedimientos internos que requieren el análisis del origen de las fotografías que se presentan en la denuncia de un siniestro o la que se obtiene en la inspección previa.

3. Definición del Problema

Imposibilidad con los mecanismos actuales de detectar el fraude en siniestros en el Seguro Automotor, en una etapa temprana del ciclo de vida de una Póliza.

4. Justificación del estudio

Es una necesidad del Mercado Asegurador (figura 1)

Figura 1 – Mercado Asegurador

Servirá para brindar transparencia, dado que al hacer más reales las tasas de siniestralidad, las primas de seguro que se calculan en función del riesgo serán reales, permitiendo de esta forma bajar los precios del seguro en función del riesgo real, en beneficio de las Compañías, Productores y Asegurados.

5. Alcances del trabajo y limitaciones

Pese a que el Fraude en Seguros se presenta en todos los ramos de la industria, se acota el presente trabajo al ramo Automotores.

La información utilizada para modelar la solución y verificar los resultados del presente trabajo, es provista por una Compañía actual del Mercado Asegurador local que exige confidencialidad y anonimato. Los datos se mantendrán de forma no pública, siendo los mismos o cualquier inferencia que pudiera hacerse de ellos, estrictamente confidenciales.

El trabajo está orientado a brindar una solución de bajo costo, pudiendo de esta forma ser la solución implementada por Aseguradoras pequeñas y medianas con bajos presupuestos.

El prototipo se diseñará para ser ejecutado en equipos configurados al menos con 32 gigabytes de RAM, procesador I7, disco de 1 terabyte, similar a los existentes en estas Aseguradoras, es decir equipos estándar de bajo costo.

6. Hipótesis

Es posible detectar un potencial fraude en siniestros de Automotores en la etapa de suscripción del riesgo.

Las variables involucradas en la hipótesis surgen de la estructura de datos de la Póliza y los Siniestros, y de indicadores estadísticos del Mercado.

Variable	Descripción
Aseguradora	variable de Contexto. Hace referencia a la Compañía Aseguradora que se analiza.
Datos de la Póliza/Endoso	
Asegurado	Tipo ID. En el caso de automóviles también es el Tomador del seguro, en general el dueño del vehículo o quien lo usa.
Productor	Tipo ID. intermediario entre la Aseguradora y el Asegurado, es quien acerca al Asegurado a la Compañía de Seguros.
Fecha de Emisión	Tipo Fecha. fecha en que se emite el documento.
Vigencia	Tipo Fecha. fechas desde y hasta en que el documento tiene validez.
Tipo de Vehículo	Categórica. Tabla de Tipos de vehículo → por ejemplo "sedan", "4x4", "moto", etc.
Marca	Categórica. Tabla de Marcas → marca del vehículo asegurado, por ejemplo "Chevrolet", "VW", "Renault", etc.
Modelo	Categórica. Tabla de modelos del vehículo → por ejemplo "SURAN", "TRACKER", "9 TSI", etc.

Año	año de fabricación del vehículo.
Suma Asegurada	valor por el que se asegura el bien.
Cobertura	Categórica. Tabla de Coberturas → qué riesgo cubre el contrato de Seguros, por ejemplo "RC", "TODO RIESGO SIN FRANQUICIA", "TERCEROS COMPLETO", etc..
Datos del Siniestro	
Fecha Ocurrencia	Tipo Fecha.
Fecha de Denuncia	Tipo Fecha. Fecha de ingreso a la compañía de la Denuncia.
Tipo de Siniestro	Categórica. Tabla de Tipos de Siniestros → ejemplos "Robo Cubierta", "Robo Total", "Robo Parcial", "Daños", "Daños a Terceros", "Lesiones Transportados", etc.
Clima	Categórica. Según Tabla. Condiciones climáticas denunciadas en el momento del siniestro.
Datos del Asegurado	
Zona Geográfica	Categórica. Según localidad del Domicilio del Asegurado. Zona de riesgo "ALTA", "MEDIO", "BAJA".
Profesión	Tabla de Profesiones del Asegurado → ejemplos "Policía", "Abogado", "Mecánico", "Médico", "Jubilado", etc.
Condición Fiscal	Categórica. Tabla de Situación Fiscal del Asegurado → "Monotributista", "Jubilado", "Consumidor Final"
Historial de Siniestros	Referencia a la historia del Asegurado en la Compañía.
Historial de Pagos	Referencia a la historia del Asegurado en la Compañía.

Tabla 3 – Variables involucradas en la hipótesis

7. Objetivos

Desarrollar un modelo que permita predecir un potencial fraude en una operación de Seguros de Automóvil, de forma temprana en el ciclo de vida de una Póliza, concretamente en el proceso de suscripción del riesgo.

El objetivo general se desagrega en los siguientes objetivos específicos:

- Armar indicadores basados en experiencia para alimentar los modelos.
- Realizar un análisis técnico-funcional de la Compañía donde se implementará la solución, particularmente de los circuitos de los departamentos de Suscripción y Siniestros Automotores.
- Contar con un set de datos consolidado, independiente de la base de datos operativa de la Aseguradora, para ser ingerido por el modelo predictivo a desarrollar.
- Encontrar herramientas sólidas y de bajo costo para el desarrollo del modelo.
- Desarrollar un prototipo y evaluar dicho modelo.
- Desarrollar conclusiones para la recomendación del modelo final a implementar.

8. Metodología

Para llevar adelante el proyecto se utilizará la metodología CRISP-DM (Cross Industry Standard Process for Data Mining).

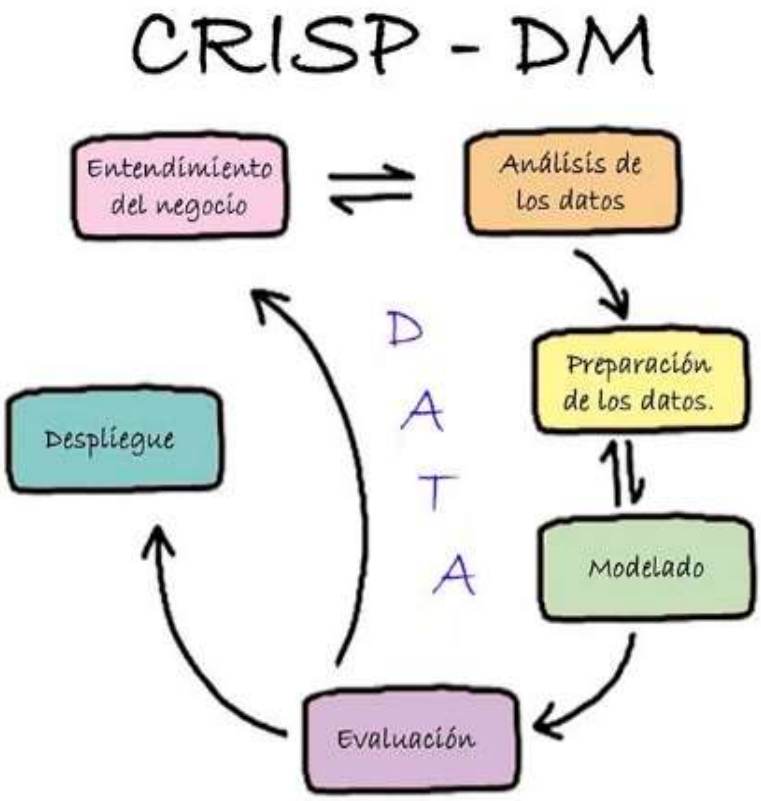


Figura 2 – Metodología CRISP-DM

Esta metodología surge de la necesidad de normalización del proceso de extraer conocimiento de los datos (ver Chapman 2000 "Referencias-Bibliografía [9]"). Se descompone el proceso de Data Mining en seis fases:

1) Entendimiento del negocio (*Business understanding*): Esta fase se centra en la comprensión de los objetivos y requisitos del proyecto desde el punto de vista del negocio. Esta perspectiva se utiliza para descubrir qué problemas resolver mediante el uso de la minería de datos.
2) Análisis de los Datos (*Data understanding*): El entendimiento de los datos nos permite familiarizarnos con ellos. Implica realizar un análisis de datos exploratorio con el objetivo de evaluar la calidad de los mismos, detectar cierta información a través de estadísticas descriptivas básicas o visualizaciones, y sacar las primeras conclusiones.

3) Preparación de los Datos (*Data preparation*): Esta fase puede considerarse como la más lenta del proceso de minería de datos. Implica una limpieza y un procesamiento previo de los datos crudos a través de diversas técnicas (reducción de datos, agrupamiento, selección de variables, transformación de variables, manejo de datos faltantes, valores atípicos, variables categóricas, etc.). Esta fase prepara el dataset final para el modelado.
4) Modelado (*Modelling*): En esta fase los datos pre-procesados se utilizan para construir y evaluar modelos en los que se utilizan algoritmos de Machine Learning.
5) Evaluación (*Evaluation*): Al realizar las 4 fases antes mencionadas, es importante evaluar los resultados acumulados, la performance predictiva de los diferentes modelos y las herramientas utilizadas, y revisar el proceso realizado hasta el momento para determinar si se cumplen o no los objetivos establecidos originalmente. En esta fase de evaluación, algunos hallazgos pueden generar nuevas ideas de proyectos para explorar.
6) Despliegue (*Deployment*): una vez que el modelo es de calidad satisfactoria, se implementa. La implementación puede variar desde un simple informe, una API a la que se puede acceder mediante llamadas programáticas, una aplicación web, etc. Asimismo, se deberá considerar los procesos para actualizar los datos de forma regular y cualquier otra actividad de mantenimiento.

Cada fase consta de una serie de tareas y salidas esperadas. El siguiente cuadro (figura 3) muestra las seis fases de CRISP-DM, y las principales características de cada una de ellas.

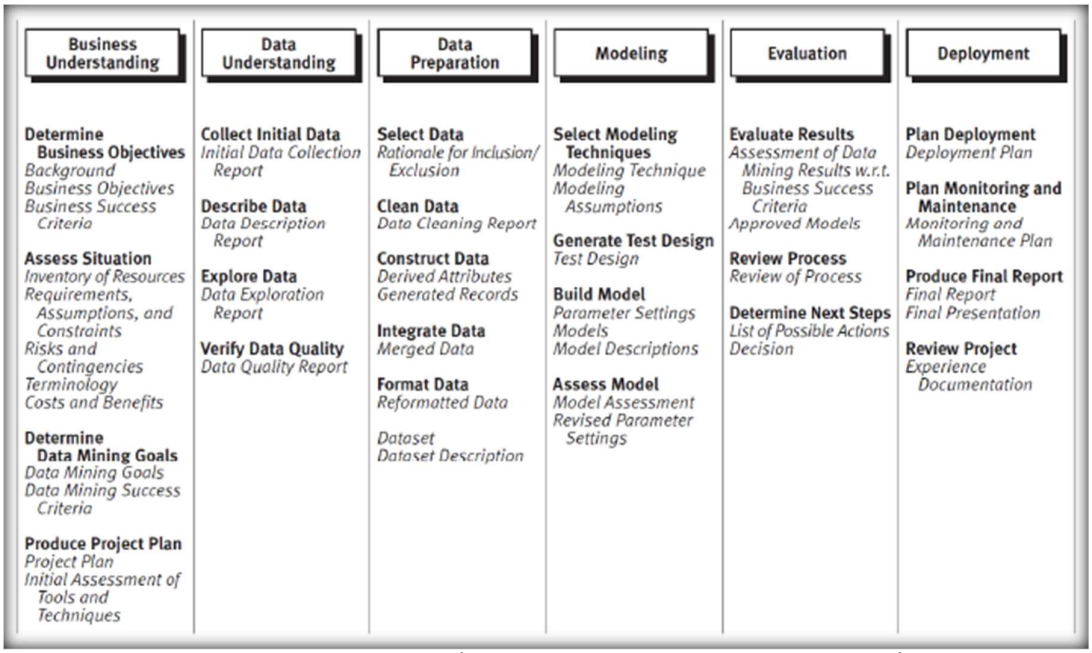


Figura 3 – Tareas y salidas de la metodología CRISP-DM (ver Chapman 2000 "Referencias-Bibliografía [9]")

Al tener las diferentes Aseguradoras sus sistemas de gestión propietarios, se desarrollará un proceso de ETL para, en base a los indicadores de la industria basados en la experiencia, generar un set de datos con el cual el modelo predictivo trabajará.

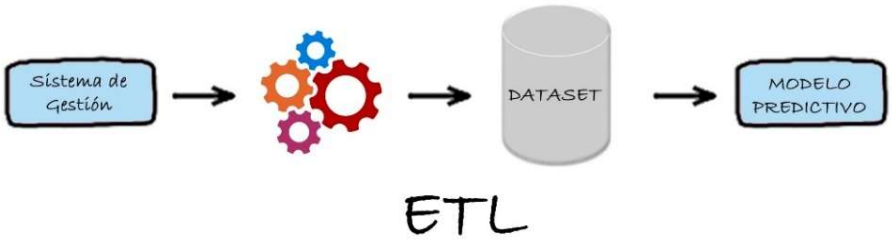


Figura 4 – Esquema del ETL

Desarrollado este proceso, se entrenará un modelo de machine learning que permita detectar patrones de fraude, retroalimentándose de un aprendizaje continuo. Con este modelo se busca obtener un score de la operación, con un enfoque en la detección del fraude y de mitigación del riesgo altamente automatizado para las Aseguradoras.

El score permitirá tomar mejores decisiones, constituye el fundamento de la solución e indica el riesgo de cada solicitud o póliza. El Ciclo de aprendizaje facilita una trayectoria de mejora continua para seguir por delante en una época dinámica y mantenerse de cara al futuro.

Las herramientas utilizadas son herramientas mayormente open source. Como DBMS se utilizó PostgreSQL 13 con interface pgAdmin 4 versión 5.2. Como plataforma de desarrollo de Data Science se utilizó Anaconda 3 2021.05 64 bit. Como entorno se utilizó Jupyter notebook con lenguaje de desarrollo se utilizó Python 3.8.0 v.1916. Las librerías Python utilizadas para los procesos de Machine Learning son numpy, pandas, matplotlib, seaborn, scikit-learn, bayes_opt, skopt, xgboost y lightgbm.

9. Desarrollo

9.1 Indicadores basados en experiencia

De la información relevada en las Aseguradoras surgen los siguientes indicadores que se deberán tener en cuenta a la hora de evaluar el riesgo de la suscripción:

Nombre del indicador	Tipo de Indicador	Operación	Descripción
Poliza_nueva_01	Prevención	Emisión	Pólizas nuevas en zonas de alto riesgo ciudad de Córdoba, Rosario, CABA y Gran Buenos Aires con la cobertura de Robo-Total.
Cambio_cobertura	Prevención	Endoso	Cambio de cobertura de vehículos en zona de alto riesgo ciudad de Córdoba, Rosario, CABA y Gran Buenos Aires (ampliación a cobertura por Robo Total o Parcial).
Poliza_nueva_02	Prevención	Emisión	Pólizas nuevas contratadas por policías, abogados, mecánicos y médicos.
Aumento_suma	Prevención	Endoso	Aumento en la suma asegurada por un importe mayor a $250.000,-
Poliza_nueva_03	Prevención	Emisión	Monotributistas, Consumidor Final y Jubilados que aseguran vehículos de Alta Gama.
Siniestro_Cobertura	Detección	Siniestros	Siniestros con vehículos de 7 años o mayor antigüedad con variación de cobertura.
Historial_Siniestros_01	Detección	Emisión Siniestros	Dominios de terceros con 3 o más siniestros en los últimos 2 años.
Siniestro_01	Detección	Siniestros	Siniestros ocurridos dentro de los 45 días de inicio de vigencia de la póliza.
Siniestro_02	Detección	Siniestros	Siniestros ocurridos quince días antes de finalizar la vigencia de la póliza.
Historial_Siniestros_02	Detección	Emisión Siniestros	Asegurados con cinco o más siniestros en los últimos dos años.
Historial_Siniestros_03	Detección	Emisión Siniestros	Asegurados con dos o más siniestros con suceso de Robo Total.
Historial_Siniestros_04	Detección	Emisión Siniestros	Asegurados con antecedentes de Robo de Cubiertas.
Siniestro_03	Detección	Siniestros	Siniestros con Fecha de Ocurrencia igual al día de pago de la cuota de la póliza.
Siniestro_04	Detección	Siniestros	Siniestros con ampliación de cobertura dentro de los 15 días de ocurrido el siniestro.
Historial_Siniestros_05	Detección	Emisión Siniestros	Robo de dos o más ruedas.
Historial_Siniestros_06	Detección	Emisión Siniestros	Pólizas con siniestros en los cuales intervienen lesionados o motos.

Tabla 4 – Indicadores basados en experiencia

9.2 Análisis técnico funcional de los circuitos de Suscripción y Siniestros

El análisis técnico funcional de los circuitos de los departamentos de Suscripción y Siniestros Automotores, fue realizado en función a los procedimientos de cada Aseguradora, y de sus respectivos flujos de información. Gracias a este análisis fue posible determinar en qué parte de los circuitos se puede intervenir para desarrollar e implementar la interfaz que alimenta a los modelos predictivos.

Consecuentemente, una vez ejecutados dichos modelos, se hace posible retroalimentar los circuitos con las alertas de potencial fraude generadas.

9.3 Preparación de los datos – ETL (Extracción, transformación y Carga)

Con el fin de obtener la información estandarizada, se desarrolló un ETL (proceso de extracción, transformación y carga) genérico para las distintas Aseguradoras. Se generó un dataset unificado con la información de pólizas, endosos y solitudes, tanto en operaciones nuevas como en modificaciones y renovaciones. El momento elegido para su ejecución es el momento de la Suscripción del riesgo, cumpliendo con lo definido en el punto *"7. Objetivos"* referido a *"Desarrollar un modelo que permita predecir un potencial fraude…, de forma temprana en el ciclo de vida de una Póliza…"*.

En una primera etapa, y dentro del alcance de este TFI, se trabajará de forma off-line, generando un dataset que diariamente permita procesar todas las operaciones, con el fin de obtener un informe con el indicador de fraude de cada operación. En un futuro, y dependiendo de la necesidad y decisión de cada Aseguradora, este proceso se podrá adaptar para trabajar de forma on-line y, ante la carga de la operación, disparar la corrida del proceso, ejecutando el modelo predictivo y retroalimentando el circuito operativo normal de la suscripción del riesgo con la respuesta del indicador de fraude, sin interrupciones. De esta forma permitirá tomar decisiones con información de la posibilidad de fraude.

El script del ETL se encuentra disponible en el siguiente link:
https://github.com/pfabbiano/fraude/blob/a36696c005d834282ea6d1918d42cd67ea206468/_ETL-script.pdf

Como se explicó anteriormente, cada Aseguradora tiene como soporte a su operatoria un sistema de Gestión particular. En algunos casos son desarrollos propios; en otros son sistemas alquilados a consultoras de software del Mercado Local con adaptaciones particulares para la Aseguradora. En todos los casos son sistemas basados en tecnología Cliente-Servidor con los datos almacenados en

RDBMS. La Aseguradora que se elige para el armado del prototipo tiene su base de datos montada en un DBMS SQLServer.

A continuación, se detallan un modelo reducido de las tablas OLTP involucradas en el proceso de ETL:

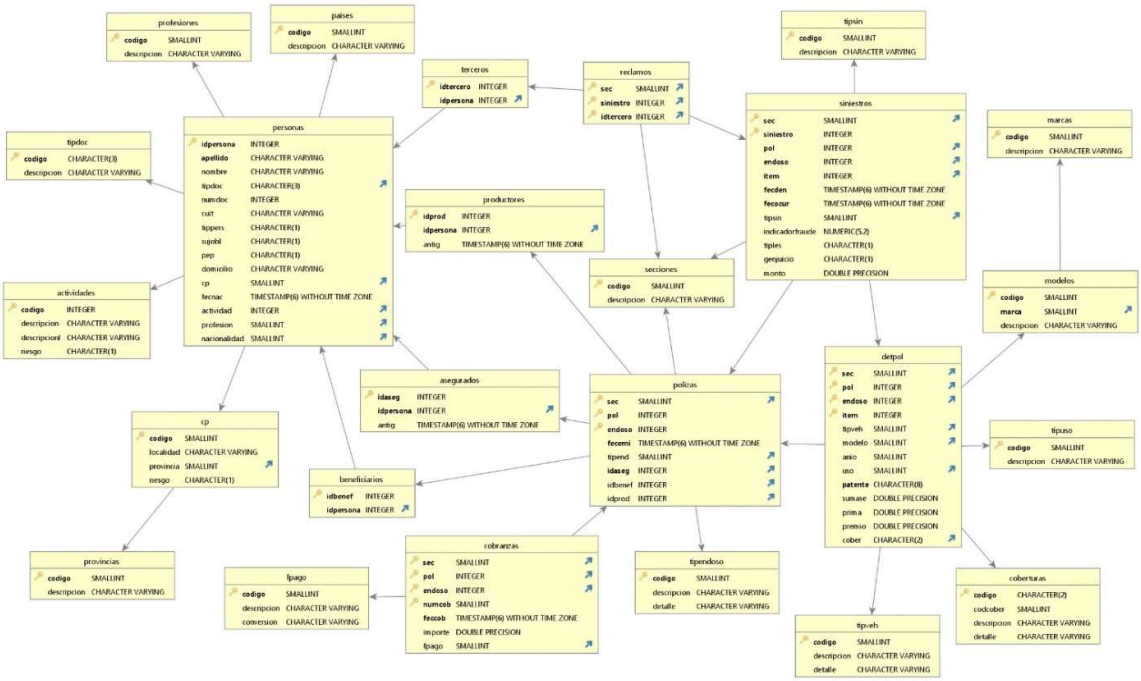


Figura 5 – ERD del modelo reducido de las tablas OLTP involucradas en el proceso de ETL

Los datos y la estructura de las tablas presentadas en el modelo descripto en el dibujo anterior (figura 5), pertenecen al sistema de gestión de una Aseguradora del Mercado Local. Fueron analizados y tienen la coherencia, consistencia e integridad necesaria. No obstante, en el proceso de ETL se llevan adelante algunos procesos de limpieza, depuración y mejora de la calidad del dato.

El proceso accede al repositorio de datos provisto por la Aseguradora a través de queries SQL desarrollados a tal fin. El procedimiento consta de una primera etapa de preparación de los datos, luego un proceso central que genera el dataset en formato de tabla temporal, para luego proceder a la generación del dataset en formato 'csv'.

Dicho proceso corrido con una cantidad de operaciones emitidas del orden de los 500,000 registros, demora alrededor de 35 segundos.

9.4 Análisis exploratorio del dataset generado en el ETL

Con el fin de familiarizarnos y comprender el dataset, refinar la ingeniería de atributos y re-verificar la calidad de los datos que servirán de ingesta del modelo predictivo, se realizó un análisis exploratorio del dataset (EDA). El mismo consiste en una serie de adaptaciones, operaciones y visualizaciones de los datos, que permiten tomar decisiones con respecto a qué datos tener en cuenta para procesar y preparar un dataframe consistente con el consiguiente proceso de machine learning.

En el caso de esta primera corrida, luego de un análisis superficial del dataset, surge que el mismo consta de 469,370 filas/registros y 25 columnas/variables.

Del análisis anterior surge que la variable 'ase_prof' tiene NA en todos los registros, la profesión del Asegurado es un dato que está definido como no obligatorio en el Sistema de Gestión y por tanto no lo están cargando. Se realiza una recomendación de setearlo como obligatorio y así se hará a partir de ahora, pero lamentablemente es un dato con el que hoy no cuenta esta Aseguradora en el momento de la Suscripción del riesgo. Pese a que el rendimiento del algoritmo de machine learning elegido no se ve el afectado por la columna llena de valores NA, se elige eliminarla del dataframe.

También vemos que la variable 'ase_nac', nacionalidad del Asegurado tiene NA en un 25% de los registros, consultada el área Comercial operativa de la Aseguradora, nos indican que cuando no tienen la nacionalidad se asume nacionalidad 'Argentina', código '200'.

La variable 'cob_fecuma' tiene muchos valores NA y es razonable que así sea, son operaciones que aún no tuvieron cobranza.

La variable 'cob_ef' cobranza en EFECTIVO es una variable categórica ('S'/'N'), por lo analizado de sus valores, cuando es 'S' tiene ese valor, pero cuando es 'N' tiene NA. Se arregla el contenido de la variable.

Para entender más aún los datos, se utiliza el método ‘describe’ del dataframe entero para mostrar las estadísticas descriptivas incluyendo valor medio, máx, mín, desvío std, etc. El método ‘describe’ solo retorna los valores de estas estadísticas para las columnas numéricas (salvo que se instruya al método con el argumento "include='all'").

También se analizan frecuencias y distribución de valores de cada variable, y luego en alguna variables, se realiza un análisis de los outliers. Pese a que el algoritmo de machine learning elegido no se ve afectado por los outliers, el análisis nos permite evaluar la calidad del dato, y ante el hecho de detectar un caso inconsistente, revisar los procesos para ver dónde está el error o la inconsistencia, y resolverlo.

En el caso de la variable 'item', se comprueba con personal de Suscripciones de la Aseguradora que la mayoría de las operaciones tienen un solo vehículo asegurado, pero existen operaciones flota con hasta 45 vehículos. Esto es correcto.

Con respecto a la variable 'ase_antig_an' su media está en el orden de los 3 años. Se encuentran 3 clientes con una antigüedad mayor a 20 años. Se verifica con personal de Comercial de la Aseguradora y esto es correcto, corresponde a 3 clientes, códigos 35957109, 35880056 y 15357066. Se dejan los registros como están.

Por último, como parte del EDA, se realiza un análisis multivariable a fin de determinar que variables pueden estar más relacionadas con la variable objetivo 'fraude'. Se determina que la variable objetivo no tiene dependencia fuerte de ninguna otra de las variables procesadas.

El notebook con el EDA realizado se encuentra disponible en el siguiente link:
https://github.com/pfabbiano/fraude/blob/967f36eafe811141bfc8a578b833a666626d5f99/v3.6%20TFI-EDA.ipynb

9.5 Modelado y Análisis de Resultados

Con el dataframe generado, se procede al modelado de la solución.

Lo primero que se hizo fue dividir la muestra con una proporción de 70/30 entre training y testing.

El dataset de 469,370 registros contiene 17,236 registros con la variable objetivo 'fraude' en '1'. Es decir, un 0.0367 de los registros (surge del EDA).

Al dividir la muestra con la función 'train_test_split', genera un número menor de registros positivos. Para resolverlo se utilizó el parámetro 'stratify' que permite obtener proporcionalmente la misma cantidad de registros positivos y negativos que el dataset entero.

Finalmente, en el dataframe de training quedaron 328,559 registros, de los cuales 12,065 tienen la variable 'fraude' en '1', es decir el 0.0367.

A continuación, se probó modelar la solución con cinco algoritmos diferentes, para evaluar los resultados con el dataset actual y determinar cómo seguir adelante con el pipeline.

Los algoritmos elegidos son Decision Tree, LightGBM, Random Forest, Logistic Regression y XGBoost.

A efectos de establecer un base-line y familiarizarse con el funcionamiento de los algoritmos, se realiza una primera corrida con parámetros por default y sin pre-procesamiento de ningún tipo en los datos.

A continuación, y con el objetivo de determinar que algoritmo tiene mejor rendimiento como solución del problema planteado, se detallan los resultados individuales de las pruebas realizadas sobre estos algoritmos.

9.5.1 Algoritmo XGBoost

En una segunda corrida de este algoritmo, se realiza una optimización bayesiana de hiperparámetros, sin realizar aún feature engineering.

Algunos de los parámetros que fueron optimizados para la corrida fueron 'learning_rate'=0.2; 'n_estimators'=45. La métrica de evaluación elegida para el corte es 'auc'.

Las features que más tiene en cuenta el clasificador en esta corrida se muestran en el siguiente cuadro (figura 6):

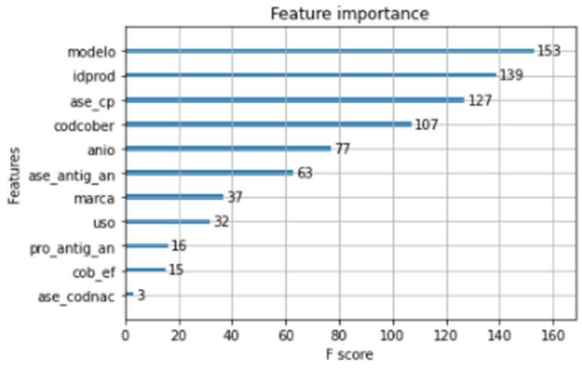


Figura 6 – Feature Importance

Teniendo en cuenta la importancia de estas variables en el ámbito del negocio, tiene sentido que el "modelo del vehículo asegurado" o el "código postal del domicilio del Asegurado" estén entre las características que más tenga en cuenta el algoritmo, ya que de la experiencia humana surgen tablas que indican el riesgo de fraude y siniestro en función a ellas. El id del productor podría ser un descubrimiento del algoritmo, y seguramente ayudaría a actualizar las condiciones comerciales particulares en cada caso.

En la figura siguiente (figura 7) se muestran las métricas de rendimiento tradicionales y el AUC como referencia dentro del gráfico de la curva ROC:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	135640
1	0.56	0.59	0.58	5171
accuracy			0.97	140811
macro avg	0.77	0.79	0.78	140811
weighted avg	0.97	0.97	0.97	140811

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

```
array([[133215,   2425],
       [  2098,   3073]], dtype=int64)
```

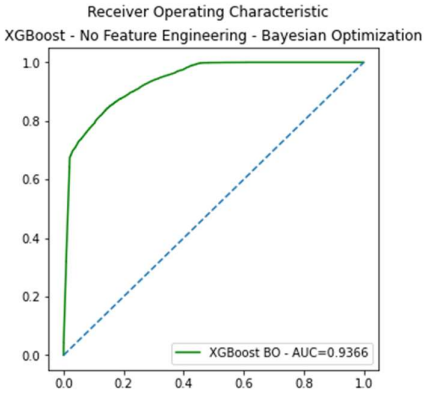


Figura 7 – Métricas de Rendimiento - Algoritmo XGBoost

Se repiten las mismas métricas y gráficos para los restantes modelos.

9.5.2 Decision Tree

Para armar el modelo de 'decisión tree' se utiliza la librería 'sklearn.tree DecisionTreeClassifier'.

Con el objetivo de evitar la mala perfomance del algoritmo a causa del desbalanceo de la clase, se ajusta el parámetro 'class_weight'='1:27', siendo 1 en 27 la relación que hay entre los '0' y los '1' en la variable objetivo.

Con la modificación de este parámetro se logró pasar de un AUC de 0.69 en el modelo corrido como línea base, a un AUC de 0.89.

```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.99	0.88	0.93	135640
1	0.20	0.79	0.33	5171
accuracy			0.88	140811
macro avg	0.60	0.84	0.63	140811
weighted avg	0.96	0.88	0.91	140811

```
confusion_matrix(y_test, y_pred)
```

```
array([[119828,  15812],
       [  1099,   4072]], dtype=int64)
```

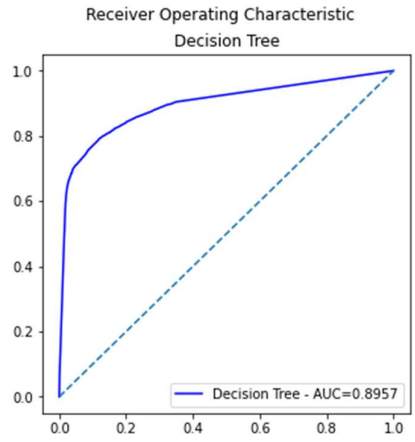
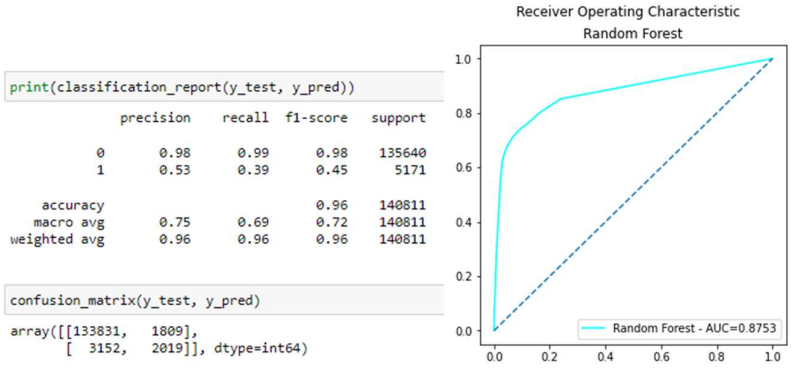


Figura 8 – Métricas de Rendimiento - Algoritmo Decision Tree

También se modificaron los parámetros por defecto 'criterion'=entropy; 'min_samples_split'=20 que se refiere a la cantidad mínima de muestras que debe tener un nodo para poder subdividir; y 'min_samples_leaf'=5 que refiere a la cantidad mínima que puede tener una hoja final.

9.5.3 Random Forest



```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	135640
1	0.53	0.39	0.45	5171
accuracy			0.96	140811
macro avg	0.75	0.69	0.72	140811
weighted avg	0.96	0.96	0.96	140811

```
confusion_matrix(y_test, y_pred)
```

```
array([[133831,   1809],
       [  3152,   2019]], dtype=int64)
```


Figura 9 – Métricas de Rendimiento - Algoritmo Random Forest

En este caso, modificando algunos parámetros, también se logra obtener una mejora en el rendimiento. Se modifica 'class_weight'='1:27' (siguiendo el mismo criterio y argumento que en 'decision tree') y 'n_estimators'=100. Existe una mejora en el rendimiento, aunque no tan notoria como en el algoritmo 'decision tree'.

9.5.4 Logistic Regression

La regresión logística, a diferencia de los algoritmos de Machine Learning, demanda un cuidadoso tratamiento previo del dataset.

A fin de evitar el desbalanceo, se intentó pre-procesar los datos, haciendo un 'subsampling' de la clase mayoritaria, pero al intentar correr los procesos usando 'imblearn.under_sampling' como librería, el algoritmo luego de procesar un rato tira un error de memoria insuficiente que no fue posible salvar manteniendo la configuración actual del equipo (que es uno de los objetivos planteados en este trabajo), lo que obliga a cambiar de estrategia.

Se procede a realizar un 'oversampling' de la clase minoritaria, usando 'RandomOverSampler'. Con esto se logra una mejora importante en el rendimiento del algoritmo.

También, y sin necesidad de pre-procesar los datos, se probó utilizando el parámetro 'class_weight'='balanced' obteniéndose similares resultados.

A continuación, se muestran las métricas de rendimiento del mejor proceso realizado:

```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.99	0.88	0.93	135640
1	0.19	0.74	0.30	5171
accuracy			0.87	140811
macro avg	0.59	0.81	0.61	140811
weighted avg	0.96	0.87	0.91	140811

```
confusion_matrix(y_test, y_pred)
```

```
array([[118745,  16895],
       [  1324,   3847]], dtype=int64)
```

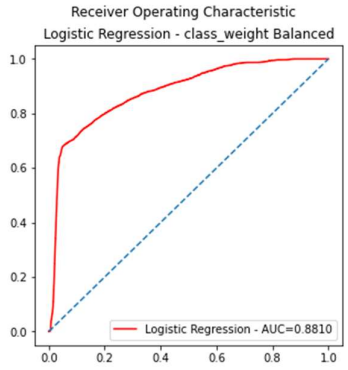


Figura 10 – Métricas de Rendimiento - Algoritmo Logistic Regression

9.5.5 Algoritmo LightGBM

Con el algoritmo LightGBM, se intentó infructuosamente optimizar usando las librerías 'bayes_opt' que habían sido utilizadas para optimizar XGBoost, pero todos los procesos probados terminaban con errores.

Luego de investigar las causas del no funcionamiento, se optó por cambiar de librerías, usando 'skopt'. Con estas librerías se pudo ejecutar satisfactoriamente la optimización Bayesiana. Se limitó el proceso a 60' de ejecución máxima, pero finalmente demoró 22', siendo los hiperparámetros optimizados *colsample_bytree, learning_rate, max_bin, max_depth, min_child_samples, min_child_weight, n_estimators, num_leaves, reg_alpha, reg_lambda, scale_pos_weight, subsample y subsample_freq*

```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	135640
1	0.56	0.59	0.58	5171
accuracy			0.97	140811
macro avg	0.77	0.79	0.78	140811
weighted avg	0.97	0.97	0.97	140811

```
confusion_matrix(y_test, y_pred)
```

```
array([[133274,   2366],
       [  2118,   3053]], dtype=int64)
```

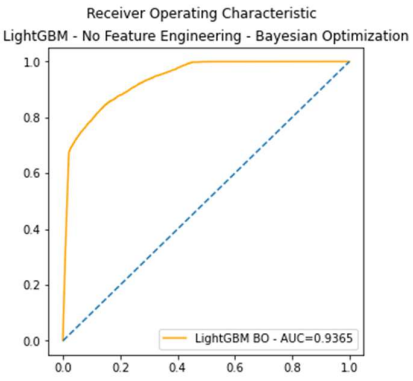


Figura 12 – Métricas de Rendimiento - Algoritmo LightGBM

El siguiente cuadro (figura 14) muestra el resultado comparativo del rendimiento de los 5 modelos, evaluando la curva ROC y el AUC:

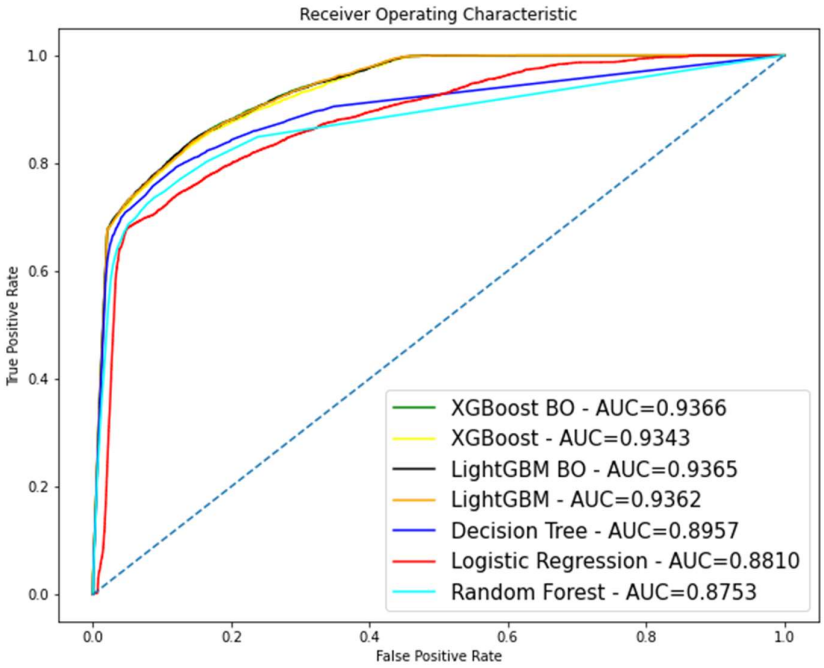


Figura 14 – Curva ROC y AUC de todos los Modelos probados

La curva ROC muestra la relación entre sensibilidad (o TPR) y especificidad (1 – FPR). Los clasificadores que dan curvas más cercanas a la esquina superior izquierda indican un mejor desempeño. Como línea de base, se espera que un clasificador aleatorio proporcione puntos a lo largo de la diagonal (FPR = TPR). Cuanto más se acerca la curva a la diagonal de 45 grados del espacio ROC, menos precisa es la prueba.

Las curvas ROC, las AUC y las métricas mostradas en los párrafos anteriores dan el sustento técnico para comparar los modelos obtenidos. De los 5 modelos, el de mejor rendimiento fue el algoritmo XGBoost. Esto coincide con la literatura publicada a la fecha, donde se señala que los resultados del algoritmo XGBoost generalmente superan a otros algoritmos. En nuestro caso particular, los resultados obtenidos por LightGBM son muy similares.

Se considera que la simple división utilizada entre training y testing, más allá de que sea estratificada, es insuficiente para validar los modelos. Por tanto, en este punto se decide validar los modelos utilizando la técnica de 5-fold cross validation.

Los resultados obtenidos son los siguientes:

```
XGBoost BO - 5-fold cross validation - AUC mean         : 0.9370
Decision Tree - 5-fold cross validation - AUC mean      : 0.8850
Random Forest - 5-fold cross validation - AUC mean      : 0.8743
Logistic Regression - 5-fold cross validation - AUC mean: 0.8782
LightGBM BO - 5-fold cross validation - AUC mean        : 0.9370
```

El siguiente gráfico (figura 15) muestra los diferentes AUC obtenidos en las distintas iteraciones del proceso:

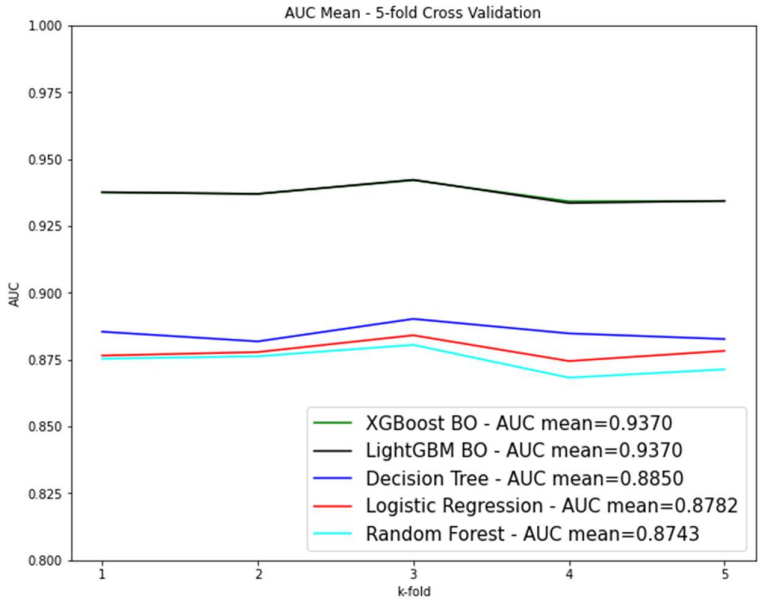


Figura 15 – AUC - 5-fold Cross Validation

Se debe tener en cuenta que LightGBM y XGBoost tienen el mismo AUC de 0.9370, por eso las líneas verde y negra se superponen.

Se decide desde este punto, y en virtud de lo expuesto, continuar el proceso del modelado final solamente con los algoritmos XGBoost y LightGBM.

9.6 Modelo Final

Como parte del armado del modelo final, se llevaron adelante operaciones de feature engineering sobre el dataset. Se probaron diferentes combinaciones de features, y se crearon 5 nuevas relevantes, en base a dos de las ya existentes: 'fecemi' que es la fecha de emisión de la operación y 'cob_fecuma' que es la fecha de la última cobranza.

Las features que se crearon son:

- Año de la fecha de última cobranza
- Mes de la fecha de última cobranza
- Año de la fecha de emisión
- Mes de la fecha de emisión
- Cantidad de días entre fecha de última cobranza y fecha de proceso

Se realizó sobre ambos algoritmos (XGBoost y LightGBM) una optimización Bayesiana final con las nuevas features en el dataset. En el caso de XGBoost usando 'bayes_opt' y en el caso de LightGBM usando 'skopt' como se explicó previamente.

La optimización Bayesiana del XGBoost ('bayes_opt') demoró cerca de 4 hs. y los hiperparámetros optimizados y sus valores fueron los siguientes:

colsample_bytree = 0.5
learning_rate = 0.2
gamma = 8.141210926403971
max_depth = 10
min_child_weight = 6.779723064345376
alpha = 10
n_estimators = 50
reg_alpha = 0.08672170218151724
reg_lambda = 0.048019737256687234

La optimización Bayesiana de LightGBM ('skopt') demoró 23' y los hiperparámetros optimizados y sus valores fueron los siguientes:

colsample_bytree → 0.9224409289683465
learning_rate → 0.23319839843327927
max_bin →.958
max_depth → 256
min_child_samples → 246
min_child_weight → 8.564041284164047
n_estimators → 312
num_leaves → 13
reg_alpha → 1.482952729978173e-06
reg_lambda → 2.463330811829892
scale_pos_weight → 1.0
subsample → 1.0
subsample_freq → 2

El gráfico siguiente (figura 16), muestra la evolución de los dos algoritmos y de los distintos modelos desarrollados, configurándolos con parámetros por default, con la pertinente optimización de hiperparámetros, realizando operaciones de feature engineering sobre el dataset y los modelos finales que involucran todas las operaciones anteriores. Las diferencias de rendimiento son sútiles y casi imperceptibles como muestran las curvas graficadas en la (figura 16), ejecutándose de forma más rápida los modelos desarrollados con LightGBM, pero teniendo un mejor rendimiento (mínimo) con XGBoost.

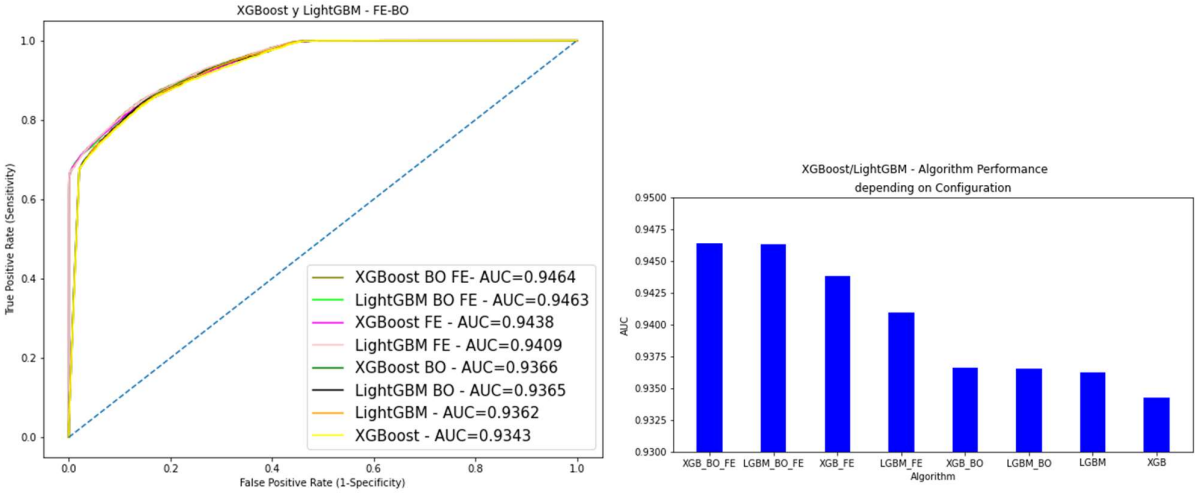


Figura 16 – Curva ROC y AUC de los distintos modelos XGBoost y LightGBM

La conclusión de estas últimas visualizaciones, y en base al análisis del rendimiento de los algoritmos, es que se podrá desarrollar e implementar la solución utilizando cualquiera de ellos.

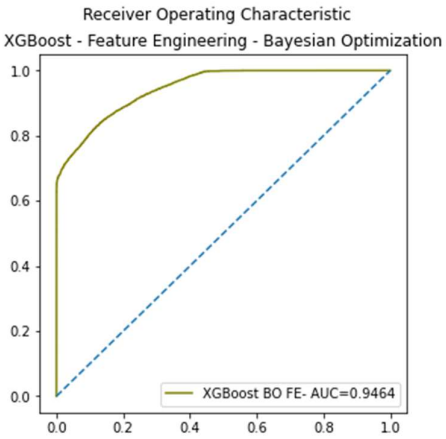
Las métricas generadas por el modelo de mejor performance son las siguientes:

```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.99	1.00	0.99	135640
1	0.96	0.65	0.78	5171
accuracy			0.99	140811
macro avg	0.97	0.83	0.89	140811
weighted avg	0.99	0.99	0.98	140811

```
confusion_matrix(y_test, y_pred)
```

```
array([[135504,    136],
       [  1798,   3373]], dtype=int64)
```



En los casos testeados, de las 5,171 operaciones que habían sido indicadas como fraudulentas por expertos humanos, el algoritmo pudo tempranamente identificar correctamente a 3,373 operaciones; mientras que las restantes 1,798 operaciones indicadas como fraudulentas por expertos humanos no fueron detectadas.

Por otro lado, el algoritmo aceptó la suscripción sin riesgo de fraude de manera correcta en 135,504 operaciones; y aceptó la suscripción sin riesgo de fraude de manera incorrecta en 136 casos, casos estos últimos 136, que basados en la experiencia humana y en una etapa posterior del ciclo de vida de la Póliza de Seguros, resultaron ser fraudulentos.

El notebook con los procesos realizados se encuentra disponible en el link:
https://github.com/pfabbiano/fraude/blob/a94a979a7237c71315bfb01d20f40bffba52d5f0/v3.7%20TFI-ML%20Modelos.ipynb

10. Conclusiones

Es posible desarrollar un algoritmo de bajo costo, que ayude a detectar un intento de fraude en el momento de la suscripción del riesgo. La hipótesis del trabajo se demuestra.

Los objetivos propuestos para este TFI (ver "7-Objetivos") se cumplieron.

No todos los indicadores que fueron relevados y que son usados por los expertos humanos para detectar fraudes pudieron implementarse en el modelo. En algunos casos por no estar el dato presente en el momento de la suscripción del riesgo, dado que muchas veces, en favor de generar prima, la Compañía no completa el legajo en la Suscripción del riesgo, sino que es el personal de Siniestros quien lo completa solo en el caso de que haya siniestro. Con algunas de las Aseguradoras que se está intentando implementar esta solución, se está trabajando coordinadamente con el personal de Datos de la Empresa para poder tener esta información disponible en tiempo y forma.

Una alternativa que se presenta para continuar desarrollando este trabajo implica determinar cuál es el costo de suscribir un riesgo que luego pudiera resultar en fraude. Así como también determinar cuál es el costo de no suscribir un riesgo que luego pudiera no resultar en fraude. Para lograr esto se deberá trabajar con personal de las áreas Emisión, Técnica y Siniestros de las distintas Aseguradoras, con el fin de lograr armar una función de maximización o minimización dependiendo si se desarrolla en función de la suscripción (prima) o en función del siniestro (pérdida directa).

También es una alternativa de mejora que el modelo desarrollado se pueda implementar integrándolo con la recepción de solicitudes, devolviendo en tiempo real una predicción de fraude. Para lograr esto se deberá ajustar el pipeline de machine learning al actual sistema de Gestión de la Aseguradora, tarea que deberá desarrollarse en forma conjunta con las áreas de IT de las Compañías.

Independientemente de esto, es de destacar que ya sea para su proceso en tiempo real, o en modo diferido, será necesario realizar un constante monitoreo del modelo y establecer mecanismos de ajustes del mismo en caso de requerirse.

Referencias-Bibliografía

[1] Hakim Ghazzai (07/04/92020) "A Secure AI-Driven Architecture for Automated Insurance Systems: Fraud Detection and Risk Measurement" - IEEEXplore - Digital Library.
https://ieeexplore.ieee.org/document/9046765

[2] Raymond R (28/06/2020) "Machine Learning at Insurers - Insights into the slow adoption of advanced models at insurance companies".
https://towardsdatascience.com/machine-learning-at-insurance-companies-2ac7ad109c65

[3] Esdras Christo Moura dos Santos (02/2018) "Using Data Mining to Predict Automobile Insurance Fraud" - Tesis de Maestría UNIVERSIDAD CATÓLICA PORTUGUESA.
https://www.semanticscholar.org/paper/Using-Data-Mining-to-Predict-Automobile-Insurance-Rafael/6a4ec63863553bc49baf404d8538d3a4b3602991?p2df

[4] Igor Bobriakov (09/07/2018) "Top 10 Data Science Use Cases in Insurance".
https://medium.com/activewizards-machine-learning-company/top-10-data-science-use-cases-in-insurance-8cade8a13ee1

[5] Kat Campise (2021) "Data Science in the Insurance Industry".
https://www.discoverdatascience.org/industries/insurance/

[6] Aníbal E. Cejas (2020)"MercadoAsegurador.com.ar – Fraude en Automotores"*.*
http://mercadoasegurador.com.ar/backup/adetail.asp?id=1682

[7] Aníbal Cejas (29/04/2019) "Informe Operadores de Mercado - Cómo detectar y prevenir el fraude contra las Aseguradoras".
http://www.informeoperadores.com.ar/2019/04/29/como-detectar-y-prevenir-el-fraude-contra-las-aseguradoras/

[8] Superintendencia de Seguros de la Nación (17/07/2014) "Normas sobre políticas, procedimientos y controles internos para combatir el fraude."
http://servicios.infoleg.gob.ar/infolegInternet/anexos/230000-234999/232554/norma.htm

[9] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth (2000) "CRISP-DM 1.0 Step-by-step data mining guide".

[10] Vishal Morde (2019) "XGBoost Algorithm: Long May She Reign!".
https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d