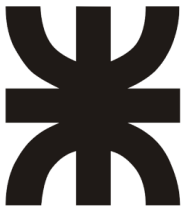


UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL CONCEPCIÓN DEL URUGUAY

TESIS PARA MAESTRIA EN
EN CIENCIAS DE LA COMPUTACIÓN
CON ORIENTACIÓN EN BASES DE DATOS

Autor:

Juan Pablo Nuñez

Año 2020

Aplicación de técnicas de minería de datos para la detección de fraudes en empresas del sector seguros

Director: Dr. Mario Guillermo Leguizamón

Codirectora: M.Cs. Anabella C. De Battista

Miembros del Tribunal de Tesis

Dr/a. Mgter.:

Dr/a. Mgter.:

Dr/a. Mgter.:

A Debo,

por su apoyo incondicional

Agradecimientos

En primer lugar quiero agradecer a la Facultad Regional Concepción del Uruguay de la Universidad Tecnológica Nacional por darme la posibilidad y el apoyo necesarios para concretar esta etapa de mi formación académica.

A Río Uruguay Seguros, que no solo me ha permitido desarrollarme profesionalmente, sino que también me permitió llevar adelante este trabajo logrando una aplicación real. Dentro de esta gran empresa quiero agradecer al equipo de siniestros quienes estuvieron brindándome información y lo necesario para concretar este trabajo, a una gran compañera, Belén Gomez que siempre estuvo allí con sus palabras de aliento para que avance con este trabajo.

Quiero agradecer profundamente a Anabella De Battista, en primer lugar por ser la responsable de que me decidiera a llevar adelante este gran desafío, y además por acompañarme durante todo este proceso, por su comprensión, conocimientos, dedicación y sobre todo por su paciencia.

Agradezco también al DR. Guillermo Leguizamón, quien no solo fue un gran docente durante el cursado de esta formación sino que me guío y acompaño en durante este trabajo.

A Lautaro, que no solo es un gran amigo, sino una gran persona, al que conocí dentro del Grupo de Investigación de UTN, y me acompañó en cada uno de los pasos de este trabajo, sin su apoyo, no hubiese sido posible.

A mis Padres, mi hermano y mis sobrinos que de una u otra forma siempre están presentes en lo que me propongo

A mi familia de amigos, todo el equipo de JARANA, en especial a Pablo, Nati, Ivan y Eve que estuvieron siempre alentando a concluir esta etapa.

Índice general

1. Introducción **1**

1.1. Planteamiento del problema y justificación 3

1.2. Fundamentación del tema elegido 6

1.3. Objetivos 8

1.4. Aportes de la tesis 8

1.5. Organización del Informe 9

2. Conceptos básicos **10**

2.1. Descubrimiento de Conocimientos en Bases de Datos 10

2.1.1. Etapas del proceso de KDD 11

2.2. Etapa de Minería de datos 13

2.2.1. Métodos de Minería de Datos 14

2.2.2. Tareas de Minería de Datos 16

2.2.3. Aplicaciones 23

2.3. Software empleado para Minería de Datos 24

2.3.1. R 24

2.3.2. R Studio 26

2.3.3. Librerías de R utilizadas 27

3. Comprensión del negocio **36**

3.1. El fraude en seguros 36

3.2. Tipos de fraudes 40

3.3. Factores a analizar para la detección de fraudes 41

3.4. Recolección de los datos 44

3.5. Descripción del conjunto de datos 45

4. Pre-procesamiento y preparación de los datos **49**

4.1. Análisis exploratorio de los datos 49

4.2. Selección de datos 54

4.3. Limpieza y Preparación de los datos 56

5. Tareas de minería de datos aplicadas **58**

5.1. Introducción 58

5.2. Selección de algoritmos de minería de datos 59

5.2.1. CART 59

5.2.2. GBM 60

5.3. Generación de los conjuntos de entrenamiento y prueba 62

5.4. Balanceo de datos 62

5.5. Construcción del modelo 65

5.5.1. Cross Validation 65

5.5.2. Grid Search 67

5.5.3. Ejecución Algortimo CART 68

5.5.4. Ejecución GBM 72

5.6. Generación del plan de pruebas 76

5.7. Ajuste del modelo 77

5.7.1. Evaluación del Modelo 78

5.7.2. Comparativa 80

6. Conclusiones y líneas de trabajo futuras **82**

6.1. Conclusiones principales 82

6.2. Consideraciones para la implementación 84

6.3. Trabajo futuro 84

Índice de Figuras

1.1. Triágulo del fraude 2

2.1. Proceso de Descubrimiento de Conocimientos en Bases de Datos 12

4.1. Proporción de siniestros con fraude comprobado y sin indicios de fraude. . 51
4.2. Cantidad de siniestros fraudulentos y sin indicio de fraude por tipo de accidente 51
4.3. Diferencia en días entre Fecha de emisión de la póliza y Ocurrencia del Siniestro 52
4.4. Diferencia en días entre Fecha de inicio de vigencia de la póliza y Ocurrencia del siniestro 53
4.5. Diferencia Fecha de ingreso de denuncia y Fecha de Ocurrencia del Siniestro 54

5.1. Resultado entrenamiento CART sin balancear 69
5.2. Resultados ejecución CART downsampling 70
5.3. Resultados ejecución CART upsampling 71
5.4. Resultados ejecución CART weighted 71
5.5. Resultado entrenamiento GBM sin balancear 73
5.6. Resultados ejecución GBM downsampling 74
5.7. Resultados ejecución GBM upsampling 75
5.8. Resultados ejecución GBM weighted 76
5.9. Curvas ROC CART y GBM sin balanceo 79
5.10. Curvas ROC CART y GBM balanceado con down sampling 79
5.11. Curvas ROC CART y GBM balanceado con up sampling 80
5.12. Curvas ROC CART y GBM balanceado con weighted 80

Índice de Tablas

4.1. Resumen de atributos numéricos 50
4.2. Resumen atributos no numéricos 50

5.1. Comparación de resultados Área bajo la Curva 78
5.2. Tabla comparativa 81

Capítulo 1

Introducción

El fraude es uno de los mayores problemas del sector de los seguros y el causante de importantes pérdidas financieras. En términos generales se refiere a un hecho deliberado en el que se realiza un reclamo a la compañía aseguradora que no se ajusta totalmente a la realidad para obtener un beneficio económico. En el año 1953 el criminólogo Donald Cressey formuló un modelo denominado Triángulo del Fraude [Cre54] que indica que el fraude ocurre cuando el estafador siente presión financiera, se le presenta una oportunidad y puede racionalizar el robo (ver Figura 1.1). El fraude tiene varios efectos adversos para las aseguradoras como perjuicios financieros y a la reputación o imagen de la organización. Por los motivos antes mencionados es de interés para quienes toman decisiones en las empresas de seguros poder anticiparse y detectar operaciones fraudulentas.

En las últimas décadas se ha incrementado notablemente la cantidad y la variedad de tipos de datos que las empresas almacenan, ya sea propios o provenientes de fuentes externas. En el caso de las empresas de seguros se resguardan grandes cantidades de datos relacionados con sus clientes, los productos que éstos contratan, los siniestros o denuncias asociados a dichos productos, entre otros. Los tipos más comunes de seguros que se han ofrecido en los últimos años son seguros de vida o de salud, seguros para vehículos, embarcaciones, agrícola, bienes inmuebles, entre otros.

Como respuesta a la necesidad de las organizaciones de analizar la gran cantidad de datos almacenada para brindar soporte a la toma de decisiones surge la Minería de Datos, que se define como el proceso de extraer conocimiento comprensible y potencialmente útil,

Figura 1.1: Triágulo del fraude

previamente desconocido, a partir de grandes volúmenes de datos [HORQFR04].

La detección de fraude es un área en la que resulta de utilidad la aplicación de técnicas de minería de datos. Desde la perspectiva del análisis de datos, los fraudes se asocian generalmente con observaciones inusuales, es decir, con comportamientos que se desvían de la norma. Estas desviaciones se conocen comúnmente como *outliers* en varias disciplinas de análisis de datos [Tor10]. Mediante la detección de las variables más relevantes para el caso de estudio, la minería de datos permite obtener modelos que superan a los que se podrían lograr a través de métodos estadísticos tradicionales [YL15a, SB13, IFFN+12].

En esta tesis se aborda el problema de la detección de fraudes en seguros del automotor. Se presenta un modelo de minería de datos que permite detectar con una precisión mayor al 80 % denuncias fraudulentas de siniestros de automotores. Para la generación de dicho modelo se trabajó con datos reales provistos por la empresa de Seguros Río Uruguay Cooperativa Limitada [RUS]. La base de datos contiene registros de denuncias de siniestros correspondientes a un período de tres años, contando cada registro con una variable indicativa de si el registro se pudo categorizar como fraude o no. En la construcción del modelo propuesto se empleó la técnica de árboles de decisión debido a que es ampliamente conocida y ha obtenido buenos resultados en casos similares, como

los detallados en [Bho11, UP17, GHKB12].

Este trabajo presenta un aporte fundamental al aplicar técnicas desarrolladas y probadas en otros ámbitos a un caso particular, trabajando además con datos reales en la formulación de un modelo que será de soporte para la empresa en la toma de decisiones dentro proceso de detección de fraudes. Otro factor diferencial es la posibilidad de validar el modelo con los usuarios finales de la compañía, facilitando la realización de ajustes que permitan su evolución en el tiempo para permitir la detección de nuevas modalidades de fraude.

1.1. Planteamiento del problema y justificación

En las últimas décadas se ha incrementado notablemente la cantidad, la velocidad y los tipos de datos que las empresas almacenan, ya sea propios y/o provenientes de fuentes externas. La Minería de Datos surge a partir de la necesidad de las empresas de analizar esas grandes cantidades de datos almacenados, y se define como el proceso de extraer conocimiento comprensible y potencialmente útil, previamente desconocido, a partir de grandes volúmenes de datos.

Actualmente las empresas de seguros resguardan grandes cantidades de datos relacionados con sus clientes, con los productos que éstos contratan, con siniestros asociados a dichos productos, entre otros. Los productos que ofrecen las empresas de seguros pueden ser seguros de vida o de salud, seguros para vehículos, embarcaciones, agrícola, bienes inmuebles, entre otros. Una problemática que preocupa especialmente a quienes toman decisiones en las empresas de seguros es la posibilidad de detectar operaciones fraudulentas. El fraude se puede definir como una acción contraria a la ley, regla o política con la intención de obtener beneficios propios.

La detección de fraude es un área importante en la que se pueden aplicar técnicas de minería de datos. Desde la perspectiva del análisis de datos, los fraudes se asocian generalmente con observaciones inusuales, es decir, con comportamientos que se desvían de la norma. Estas desviaciones se conocen comúnmente como *outliers* en varias disciplinas de la analítica de datos [Tor10].

Existen varios antecedentes de aplicaciones de técnicas de minería de datos en el sector seguros. En [Ali18] se presentan resultados de la aplicación del algoritmo de clustering *K-Means* para la identificación de patrones en seguros del automotor, en [YL15b] se presenta una aplicación de métodos basados en el vecino más cercano para la detección de fraudes en seguro de automóviles. En [RA15] se presentan dos técnicas de minería de datos supervisadas y no supervisadas, sus ventajas y desventajas y se propone un nuevo enfoque híbrido que combina las ventajas de ambas. En [Bho11] se presenta la aplicación de algoritmos basados en árboles de decisión y clasificación basadas en reglas del enfoque bayesiano. En [JAK+16] se explica cómo mediante el uso de técnicas de minería de datos como asociación, agrupamiento, previsión y clasificación, se analizan datos de clientes. En [SWB00] se presenta un análisis de patrones en retención de clientes y reclamos de seguros empleando técnicas de minería de datos. En [SSM16] se presenta una revisión y aplicación de varias técnicas de minería de datos aplicadas a la detección de fraudes en entidades financieras. En [KN18] se presenta la aplicación de algoritmos de predicción y clasificación en la detección de fraudes en reclamos de siniestros del automotor.

Existe evidencia de la gran variedad de áreas en las que las técnicas de minería de datos pueden agregar valor al negocio. La predicción del fraude en empresas de seguros es sin dudas un área de aplicación de estas técnicas ya que, a través de la detección de las variables más relevantes para el caso de estudio, permiten obtener modelos que superan a los que se podrían lograr a través de métodos estadísticos tradicionales en capacidad de detección de anomalías [YL15a, SB13, IFFN+12].

En esta tesis se pretende generar un modelo que permita detectar con una precisión mayor al 80 % denuncias fraudulentas de siniestros de automotores. Para la generación de dicho modelo se trabajará con datos reales provistos por la empresa de Seguros Río Uruguay Cooperativa Limitada. La base de datos puesta a disposición por parte de la empresa comprende registros de denuncias de siniestros de automotor correspondientes a un período de tres años, contando cada registro con una variable indicativa de si el registro se pudo categorizar como fraude o no.

Para la construcción del modelo propuesto se utilizará la técnica de árboles de decisión debido a que es ampliamente conocida y ha sido probada obteniendo buenos resultados en casos similares, como los detallados en [Bho11, GHKB12]. Se trabajará con los datos

provistos, empleando el 70 % de los mismos como conjunto de datos de entrenamiento y el 30 % restante será empleado como conjunto de datos de prueba. Está previsto comparar la técnica de árboles de decisión con otros algoritmos de minería de datos, para evaluar su eficacia en esta aplicación real y particular.

El aporte fundamental de esta tesis es la aplicación de técnicas de minería de datos desarrolladas y probadas en otros ámbitos, a un modelo e negocios real de una empresa de seguros. Es de destacar el valor agregado que genera este trabajo, ya que se trabaja con datos reales en la formulación de un modelo que la empresa adoptará como soporte a la toma de decisiones en el proceso de detección de fraudes. Otro factor diferencial es la posibilidad de validar el modelo con los usuarios finales de la compañía, facilitando la realización de ajustes que permitan su evolución en el tiempo, brindando la posibilidad de detectar nuevas modalidades de fraude.

Una tarea clave en el desarrollo de este trabajo es la integración, limpieza y depuración de datos, justamente por tratar con datos reales, provenientes de distintos sistemas informáticos transaccionales con los que opera la compañía. Se espera que esta tarea insuma tiempo ya que de la calidad de los datos dependerá obtener buenos resultados en la aplicación del modelo de minería de datos.

Este trabajo de tesis se desarrolla en el marco del proyecto "Descubrimiento de Conocimiento de Base de Datos"del grupo de investigación en Base de Datos de UTN-FRCU. Mediante un convenio firmado entre la Facultad Regional Concepción del Uruguay (UTN) y Rio Uruguay Seguros, la empresa se compromete a brindar la base de datos anonimizados para que el tesista, que se desempeña como docente de dicha casa de estudios y trabaja en relación de dependencia con dicha empresa, pueda realizar las pruebas necesarias para la generación de un modelo que permita detectar denuncias fraudulentas en siniestros de automotor. La empresa declara que adoptará el modelo generado en el marco de esta tesis como herramienta aplicable al análisis de siniestros de seguros de automotor.

1.2. Fundamentación del tema elegido

EL objetivo de esta tesis es la generación de un modelo predictivo para la detección de fraudes en denuncias de siniestros de automotores mediante la aplicación de árboles de decisión, una técnica ampliamente conocida dentro del área de minería de datos.

En [WFH11, TSK05] se define la minería de datos como el proceso de extraer conocimiento útil y comprensible, desconocido previamente, desde grandes cantidades de datos almacenados en distintos formatos. Para que este proceso sea efectivo debería ser automático o semi-automático y el uso de los patrones hallados debería ayudar en la toma de decisiones a la organización.

Por lo tanto la minería de datos plantea dos retos, por un lado procesar grandes volúmenes de datos (heterogéneos y en ocasiones no estructurados) y por el otro lado utilizar las técnicas adecuadas para obtener valor de los mismos. La utilidad del conocimiento obtenido está muy ligada a la comprensión del modelo inferido, ya que en la mayoría de los casos, los usuarios finales de estos modelos no son expertos en técnicas de minería de datos, sino que están estrechamente ligados al negocio.

La minería de datos puede aplicarse a información de cualquier tipo y de acuerdo al tipo de datos con el que se trabaja se selecciona la técnica que resulta más adecuada. Si bien existen grandes cantidades de tipos de datos (enteros, reales, texto, fecha, hora, entre otros), desde el punto de vista de las técnicas de minería de datos sólo es de interés distinguir entre numéricos (que pueden ser enteros o reales) y categóricos o discretos (datos que toman valores en un conjunto finito de categorías).

En muchos casos se utiliza el término Minería de Datos como sinónimo de Descubrimiento de Conocimiento (KDD por sus siglas en inglés), en cambio algunos autores indican que la Minería de Datos constituye un paso fundamental en el proceso de descubrimiento de conocimiento. El Descubrimiento de Conocimiento en Bases de Datos (*KDD*, por sus siglas en inglés) se define como: el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos [FPSS96b].

El *KDD* consiste en una secuencia iterativa de los siguientes pasos:

1. *Limpieza de Datos*: limpieza de ruido y datos inconsistentes.
2. *Integración de Datos*: combinación de datos provenientes de distintas fuentes.
3. *Selección de Datos*: recuperación de datos relevantes para el análisis desde la base de datos.
4. *Transformación de Datos*: construcción de una "vista minable" mediante, por ejemplo, ejecutar operaciones de resumen o agregación de los datos.
5. *Minería de Datos*: aplicación de métodos para extraer patrones de los datos.
6. *Evaluación de Patrones*: identificación de patrones de interés que representan conocimiento basados en métricas de interés.
7. *Presentación de Conocimiento*: utilización de técnicas de visualización o representación para presentar al usuario el conocimiento minado.

En este proceso de descubrimiento de conocimiento es posible aplicar distintos modelos o algoritmos dependiendo de la naturaleza del problema. Los tipos de problemas pueden ser de clasificación, segmentación, asociación o regresión. Dentro de las técnicas de clasificación se encuentran árboles de decisión, redes bayesianas o redes neuronales, k-vecinos más cercanos y máquinas de vectores de soporte. Los árboles de decisión constituyen una herramienta que permite: jerarquizar las variables independientes según su capacidad de predecir la variable objetivo, modelar relaciones no lineales de alta complejidad manejando un gran número de variables, describir el camino que sigue la variable explicada mostrando su dinámica hasta llegar al resultado final.

En el marco de esta tesis, previo a la aplicación de la técnica de minería de datos, es necesario realizar un análisis descriptivo de las variables involucradas para determinar cuáles son las más relevantes para la generación de modelos predictivos.

Una vez construidos los diferentes modelos de árboles de decisión se realizará una medición de su capacidad predictiva y una comparación con otros modelos, para comprobar si permiten la detección de casos fraudulentos de denuncias de siniestros de automotores.

1.3. Objetivos

Objetivo General

Aplicar técnicas predictivas de minería de datos en una empresa de seguros a fin de detectar casos de fraudulentos en denuncias de siniestros de automotores.

Objetivos Específicos

- Evaluar técnicas predictivas de Minería de Datos y su aplicabilidad al caso de estudio.
- Generar un modelo que permita detectar fraudes en las denuncias de siniestros de automotores.
- Comparar el rendimiento de la técnica predictiva seleccionada con otra técnica conocida para evaluar la precisión del modelo generado.

1.4. Aportes de la tesis

Los aportes fundamentales de esta tesis son:

1. La generación de un modelo que permite detectar fraudes en siniestros del seguro automotor mediante la aplicación de técnicas de minería de datos. Este desarrollo ha sido validado y ajustado con usuarios finales, lo que ha permitido lograr un modelo que se ajusta a la problemática actual de la empresa.
2. La determinación de las variables más influyentes a considerar para la detección de siniestros en seguros del automotor.
3. El desarrollo y aplicación de este modelo permitirá dimensionar los beneficios de implementar técnicas de minería de datos para la detección de fraude. En consecuencia se espera que genere interés en adaptar el modelo para aplicarlo a otro tipo de seguros.

4. Permitir mediante este modelo que la empresa analice la totalidad de sus siniestros ingresados, quedando para el análisis del sector correspondiente aquellos casos para los cuales el modelo indique una alta probabilidad de que resulte fraudulento.

1.5. Organización del Informe

Este informe de tesis está organizado en cinco capítulos, sobre los que podemos clasificarlos en dos grupos, por un lado los capítulos 1, 2 y 3 son los que conforman el marco teórico de este trabajo y los Capítulos 4 y 5 es donde se detalla el trabajo de esta tesis.

En el Capítulo 1 se da una introducción respecto del informe, y se detallan el trabajo y los aportes que se han dado a través del desarrollo de este trabajo, en el Capítulo 2 detallan las tareas de minería de datos, por último en el Capítulo 3 se detalla la comprensión del domino de la aplicación con un detalle de la identificación del problema, y los datos para aplicar las técnicas detalladas en los puntos anteriores. El Capítulo 4 se refiere al trabajo de procesamiento y preparación de datos, análisis, limpieza, transformación y selección. Por último el Capítulo 5 explica las tareas de minería de datos aplicadas, junto con las conclusiones y las lineas de trabajo a futuro.

Capítulo 2

Conceptos básicos

En este Capítulo se presentan los conceptos fundamentales del proceso de descubrimiento de conocimiento en Bases de Datos. Se detallan las etapas del proceso y se aborda de manera particular la etapa de Minería de Datos que constituye la base de este trabajo de tesis. Se especifican los objetivos de la Minería de Datos, así como las tareas y los métodos existentes, que permiten procesar diferentes conjuntos de datos e identificar los patrones subyacentes en los mismos. También se realiza una revisión de aplicaciones de minería de datos en diversos modelos de negocios.

En la última parte del capítulo se presenta una breve reseña del software estadístico R y de las librerías que se han empleado para desarrollar este trabajo.

2.1. Descubrimiento de Conocimientos en Bases de Datos

El proceso de Descubrimiento de Conocimientos en Bases de Datos (o *Knowledge Discovery in Databases* KDD) se define como el proceso no trivial de identificación de patrones válidos, novedosos, potencialmente útiles y comprensibles a partir del análisis de datos [FPSS96a].

En esta definición se resumen las propiedades deseables del conocimiento extraído:

- *válido*: los patrones deben seguir siendo precisos para datos nuevos y no sólo para

aquellos que se han utilizado en su obtención.

- *novedoso*: que aporte algo desconocido para el usuario.
- *potencialmente útil*: la información debe conducir a acciones que reporten algún tipo de beneficio para el usuario.
- *comprensible*: los patrones obtenidos deben resultar comprensibles, es decir, que se deben poder revisar, interpretar, validar y emplear en la toma de decisiones.

2.1.1. Etapas del proceso de KDD

El proceso de *KDD* abarca desde la comprensión de los datos y su preparación hasta la interpretación de resultados obtenidos a partir de ellos. Consiste en una secuencia iterativa de los siguientes pasos:

- *Comprensión del negocio*: el primer paso consiste en la comprensión del dominio de aplicación y en la identificación del objetivo del proceso de KDD desde el punto de vista del usuario.
- *Generación de la vista minable*: en esta etapa se selecciona el conjunto de datos sobre el que se aplicarán los métodos para descubrir conocimiento.
- *Limpieza y preprocesamiento de datos*: etapa que consiste en la implementación de estrategias para la limpieza de datos, la eliminación de ruido, tomar decisiones respecto a datos faltantes o inconsistentes.
- *Reducción y proyección de datos*: identificación de características de utilidad para representar los datos en base al objetivo de la tarea. Se trabaja en la reducción de la cantidad de variables a considerar mediante la aplicación de métodos de reducción de la dimensionalidad y de ser necesario, se aplican estrategias de transformación para contar con la información en un formato adecuado.
- *Selección del método de Minería de Datos en base a los objetivos del KDD*: selección del método adecuado de minería de datos, que puede ser clasificación, agrupamiento, regresión, entre otros.
- *Análisis exploratorio*: selección de los algoritmos de minería de datos a aplicar.

Esta etapa consiste en la definición de modelos y parámetros apropiados para el descubrimiento de conocimiento. No se debe perder de vista que es necesario lograr un modelo que el usuario final pueda comprender para que realmente resulte útil para soportar la toma de decisiones.

- *Minería de Datos*: búsqueda de patrones de interés que se representan bajo alguna forma particular como reglas o árboles de clasificación, regresión o agrupamiento.
- *Interpretación o evaluación de patrones obtenidos*: esta etapa puede derivar en la necesidad de iterar sobre alguno de los pasos anteriores como así también consistir en la generación de visualizaciones de los modelos extraídos y de los datos en base a los modelos hallados.
- *Tomar acciones en base a los patrones descubiertos*: en este último paso es posible emplear directamente el conocimiento, incorporar el conocimiento en otro sistema como insumo para otras acciones, o simplemente documentar e informar los hallazgos a los interesados. Debe además considerarse la posibilidad de que se requiera subsanar conflictos con bases de conocimiento existentes previamente.

La Figura 2.1 presenta los pasos básicos del proceso de KDD. Es necesario mencionar que el proceso puede requerir varias iteraciones e incluso puede contener ciclos entre algunos pasos.

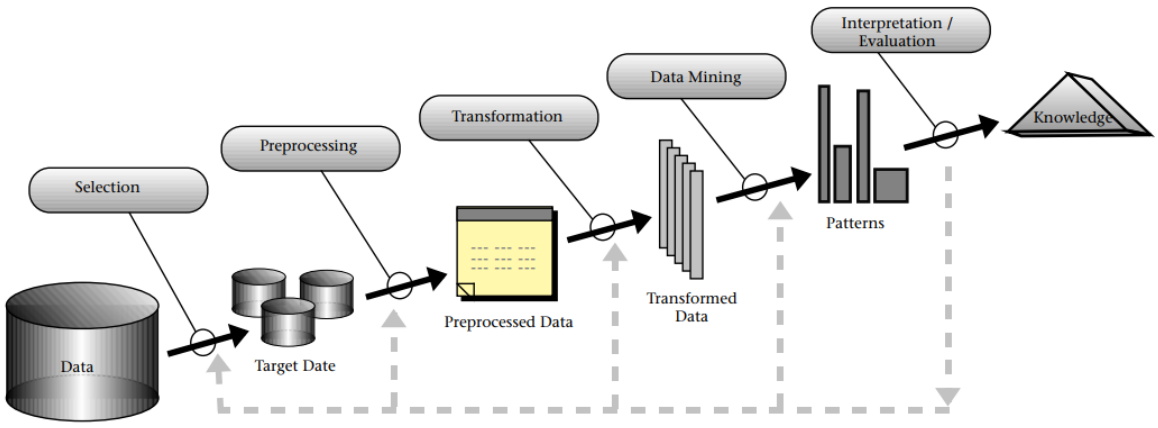


Figura 2.1: Proceso de Descubrimiento de Conocimientos en Bases de Datos

En este proceso de descubrimiento de conocimiento es posible aplicar distintos

modelos o algoritmos dependiendo de la naturaleza del problema. Los tipos de problemas pueden ser de clasificación, segmentación, asociación o regresión. Dentro de las técnicas de clasificación se encuentran árboles de decisión, redes bayesianas o redes neuronales, k-vecinos más cercanos y máquinas de vectores de soporte. Los árboles de decisión constituyen una herramienta que permite: jerarquizar las variables independientes según su capacidad de predecir la variable objetivo, modelar relaciones no lineales de alta complejidad manejando un gran número de variables, describir el camino que sigue la variable explicada mostrando su dinámica hasta llegar al resultado final.

2.2. Etapa de Minería de datos

La Minería de Datos permite a las organizaciones analizar grandes cantidades de datos almacenados en bases de datos propias, o incluso datos externos, para obtener conocimiento que sirva como insumo en la toma de decisiones.

Existe gran cantidad de áreas en las que resulta de utilidad la aplicación de técnicas de minería de datos: análisis de perfiles de clientes en el otorgamiento de créditos, análisis de comportamientos de compras en supermercados, obtención de patrones para detección de fraudes en tarjetas de crédito, reducción de fuga de clientes, predicción de fallas en modelos de producción, detección de spam en aplicaciones de correo electrónico, entre otras.

La minería de datos plantea dos retos, por un lado procesar grandes volúmenes de datos y, por el otro lado, utilizar las técnicas adecuadas para obtener valor de los mismos. La utilidad del conocimiento obtenido está muy ligada a la comprensión del modelo inferido, ya que en la mayoría de los casos, los usuarios finales de estos modelos no son expertos en técnicas de minería de datos sino que están estrechamente ligados al negocio. El modelo obtenido es una descripción de los patrones y relaciones que existen entre los datos que pueden emplearse para realizar predicciones, entender mejor los datos o explicar situaciones del pasado.

En base al tipo de datos que se requiere analizar y al problema a resolver se seleccionan: la tarea de Minería de Datos a aplicar, el tipo de modelo y finalmente el algoritmo de

minería de datos que permita obtener el tipo de modelo predefinido. Si bien existen diferentes tipos de datos (enteros, reales, texto, fecha, entre otros), desde el punto de vista de las técnicas de minería de datos sólo es de interés distinguir entre numéricos (que pueden ser enteros o reales) y categóricos o discretos (datos que toman valores en un conjunto finito de categorías).

Los patrones obtenidos luego de la aplicación de técnicas de Minería de Datos se pueden ver como un resumen de los datos de entrada y pueden ser utilizados en el análisis adicional o, por ejemplo, en el aprendizaje automático y análisis predictivo. El la etapa de minería de datos podría identificar varios grupos en los datos, que luego pueden ser utilizados para obtener resultados más precisos de predicción por un sistema de soporte de decisiones.

2.2.1. Métodos de Minería de Datos

Existen diversas técnicas de minería de datos, siendo las más representativas las que se enumeran a continuación:

Redes Neuronales

Una Red Neuronal Artificial es una técnica de aprendizaje automático que simula el mecanismo de aprendizaje de los organismos biológicos (Argawall 2018). Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Algunos ejemplos de Redes Neuronales son:

- El Perceptrón.
- El perceptrón multicapa.
- Los mapas autoorganizados, también conocidos como redes de Kohonen.

Regresión lineal

Esta técnica es la que se utiliza con más frecuencia para formar relaciones entre datos. Rápida y eficaz pero se torna insuficiente en espacios multidimensionales donde puedan

relacionarse más de 2 variables.

Árboles de decisión

Un árbol de decisión es un modelo de predicción utilizado en el ámbito de la inteligencia artificial y el análisis predictivo. Dada una base de datos se generan diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema.

Modelos estadísticos

Es una expresión simbólica en forma de igualdad o ecuación que se emplea en todos los diseños experimentales y en la regresión para indicar los diferentes factores que modifican la variable de respuesta.

Agrupamiento o Clustering

Es un procedimiento de agrupación de una serie de vectores según criterios de distancia; se dispondrán los vectores de forma que estén más cercanos a aquellos que tengan características comunes. Ejemplos:

- Algoritmo K-means
- Algoritmo K-medoids

Reglas de asociación

Se utilizan para descubrir hechos que ocurren en común dentro de un determinado conjunto de datos. Según el objetivo del análisis de los datos, los algoritmos utilizados se clasifican en supervisados y no supervisados (Weiss y Indurkhya, 1998):

- Algoritmos supervisados (o predictivos): predicen un dato (o un conjunto de ellos) desconocido a priori, a partir de otros conocidos.

- Algoritmos no supervisados (o del descubrimiento del conocimiento): se descubren patrones y tendencias en los datos.

2.2.2. Tareas de Minería de Datos

Como se mencionó anteriormente existen diferentes tipos de fuentes de datos (bases de datos relacionales, data warehouses, *streams* de datos, bases de datos transaccionales, objeto-relacionales, temporales, espaciales, textuales, multimedia, entre otras) sobre las que se aplican distintas técnicas de minería de datos, que en general pueden clasificarse en dos grandes categorías: descriptivas y predictivas.

Tareas descriptivas

Tienen como objetivo derivar patrones (correlaciones, tendencias, clusters, anomalías, etc.) que resumen la relación entre los datos. Éstas se denominan tareas exploratorias y requieren generalmente post-procesamiento para poder explicar y validar los resultados [HORQFR04, HKP11].

- *Agrupamiento (clustering)*: el funcionamiento de los algoritmos de agrupamiento está basado en la optimización de una función objetivo, que normalmente es la suma ponderada de las distancias a los centros, aunque estas funciones pueden variar, y muchas veces los distintos algoritmos de reconocimiento de patrones se distinguen principalmente en la definición de sus funciones objetivo a optimizar. Uno de los pasos en los algoritmos de agrupamiento es el de asignar a cada objeto una medida de semejanza al patrón o centroide de cada cluster, con el fin de determinar a cuál de los grupos detectados pertenece el objeto en cuestión. Esta medida de semejanza entre objetos de un conjunto de datos se basa normalmente en el cálculo de una función de distancia.
- *Correlaciones y factorizaciones*: tareas que se emplean para examinar el grado de similitud de los valores de dos variables numéricas. Una fórmula estándar para medir la correlación lineal es el coeficiente de correlación r, que es un valor real comprendido entre -1 y 1. Si r es 1 las variables están perfectamente correlacionadas, si r es -1 las variables están perfectamente correlacionadas negativamente. Esto

quiere decir, que cuando r es positivo, las variables tienen un comportamiento similar (ambas crecen o decrecen al mismo tiempo) y cuando r es negativo si una variable crece la otra decrece. En el caso en que r es 0 no hay correlación.

- *Reglas de asociación*: forman parte del grupo de tareas descriptivas, muy similares a las correlaciones, que tienen como objetivo identificar relaciones no explícitas entre atributos categóricos. Pueden ser de muchas formas, aunque la formulación más común es del estilo *si el atributo X toma el valor d entonces el atributo Y toma el valor b*. Un caso especial de reglas de asociación son las *reglas de asociación secuenciales* que se utilizan para determinar patrones secuenciales en los datos. Estos patrones se basan en secuencias temporales de acciones y difieren de las reglas de asociación en que las relaciones entre los datos se basan en el tiempo.

- *Dependencias funcionales*: una dependencia funcional (aproximada o absoluta) es un patrón en el que se establece que uno o más atributos determinan el valor de otro.

$$A \wedge B \wedge C \rightarrow D$$

  Es decir, que para los mismos valores de A, B y C tenemos un solo valor de D. Siendo D función de A, B y C. Si representamos la parte izquierda como un conjunto de condiciones, podemos establecer una relación de orden entre las dependencias funcionales, esto genera un semi-retículo y la búsqueda se realiza sobre él.

- *Detección de valores e instancias anómalas*: los valores anómalos son un problema omnipresente en la recolección de datos, son observaciones que se desvían en alguna dirección respecto al comportamiento general del resto del conjunto de datos y pueden afectar los resultados de aplicar métodos estadísticos univariantes o multivariantes. Es fundamental la detección de estos valores, ya sea para eliminarlos o para atenuar sus efectos en el análisis. Se han desarrollado varios métodos para la detección de valores anómalos, entre ellos están la Distancia Robusta de Mahalanobis (DRM), la Curtosis-1 y el método FGR de Filzmoser, Garrett y Reimann [KPR$^+$12, MF13].

Tareas predictivas

El objetivo de estas tareas es predecir el valor de un atributo basado en los valores de otros atributos. El atributo a predecir es conocido como el atributo destino o variable dependiente y los atributos usados para la predicción como variables independientes. Dependiendo de cómo sea la correspondencia entre los ejemplos y los valores de salida y la presentación de los ejemplos podemos definir varias tareas predictivas:

- *Clasificación (o discriminación)*: a cada ejemplo del tipo de objeto a clasificar (registro de la base de datos) se le asigna una clase, representada por el valor de un atributo (atributo de clase). El dominio del atributo de clase es discreto, cada valor representa una clase de objeto. Los restantes atributos que sean significativos para determinar la clase, son utilizadas por las técnicas de clasificación para generar funciones (reglas) que permiten determinar la clase de un ejemplo a partir de los valores de sus atributos significativos. El objetivo de la tarea es poder predecir la clase de nuevos ejemplos a partir del valor de sus atributos significativos, utilizando las reglas generadas.
- *Clasificación suave*: no es más que una extensión de la anterior (Clasificación), en la que se introduce un grado de certeza en la predicción hecha. En símbolos:

  Entrada:

  - Dominio de ejemplos:

    $$D = \{e\langle v_1, v_2, ..., v_n\rangle / v_i \in D_i\}$$

  - Conjunto de ejemplos (muestra):

    $$E \subseteq D$$

  - m clases:

    $$S = \{c_1, c_2, ..., c_m\}$$

- Conjunto de ejemplos etiquetado:

$$\{\langle e, s\rangle : e \subseteq E, s \subseteq S\}$$

Salida:

- Función Clasificador:

$$\lambda : E \to S$$

- Función de certeza (Grado de certeza de la predicción hecho por la función λ):

$$\theta : E \to R$$

- *Estimación de probabilidad de clasificación*: puede definirse como una generalización de la clasificación suave. En símbolos:

Entrada:

- Dominio de ejemplos:

$$D = \{e\langle v_1, v_2, ..., v_n\rangle / v_i \in D_i\}$$

- Conjunto de ejemplos (Muestra)

$$E \subseteq D$$

- m clases:

$$S = \{c_1, c_2, ..., c_m\}\}$$

- Conjunto de ejemplos etiquetado

$$\{\langle e, s\rangle : e \subseteq E, s \subseteq S\}$$

Salida:

- Función de certeza (grado de certeza de que un ejemplo sea de la clase i)

$$\theta : E \rightarrow R(i : 1..m) :$$

- *Categorización:* En este caso lo que se aprende es una correspondencia, esto es, para cada entrada, no sólo hay una correspondencia, sino que puede haber varias, cada elemento puede estar etiquetado con mas de una clase (Ej: los artículos de un blog tienen mas de una etiqueta, por lo que categorizar un artículo significa predecir que etiquetas tendrá en función de los ejemplos que ya existen en el archivo). En este caso también se puede presentar una categorización suave o un estimador de probabilidad.

  En símbolos:

  Entrada:

  - Dominio de ejemplos:

$$D = \{e\langle v_1, v_2, ..., v_n\rangle / v_i \in D_i\}$$

  - Conjunto de ejemplos (Muestra)

$$E \subseteq D$$

  - m clases:

$$S = \{c_1, c_2, ..., c_m\}$$

  - Conjunto de ejemplos etiquetado

$$\{\langle e, s\rangle : e \subseteq E, s \subseteq S\}$$

Salida:

- Correspondencia de clasificación:

$$\lambda \subseteq E \times S$$

- *Regresión*: es un caso particular de la tarea de clasificación, cuando el dominio de salida de la función es numérico. Se busca una función real entre un atributo (atributo objetivo) y un conjunto de atributos significativos del tipo de objeto. Los dominios de los atributos deben ser numéricos. El objetivo de la tarea es poder predecir el valor del atributo objetivo de nuevos ejemplos a partir del valor de sus atributos significativos, utilizando la función generada.

  En símbolos:

  Entrada:

  - Dominio de ejemplos:

$$D = \{e\langle v_1, v_2, ..., v_n\rangle / v_i \in D_i\}$$

  - Conjunto de ejemplos (Muestra)

$$E \subseteq D$$

  - m clases:

$$S = \{c_1, c_2, ..., c_m\}$$

  - Conjunto de ejemplos etiquetado

$$\{\langle e, s\rangle : e \subseteq E, s \subseteq S\}$$

Salida:

- Función de regresión:

$$\lambda : E \rightarrow S$$

Árboles de Decision

Los Arboles de decisión corresponden a uno de los métodos inductivos de aprendizaje supervisado, el cual realiza divisiones sucesivas del conjunto de datos, utilizando algún criterio de selección, manteniendo organizada su estructura de forma jerárquica, con el fin de maximizar la distancia entre los grupos de datos generados en cada iteración.

Son una manera de representar una serie de reglas que llevan hacia una clase o valor de los datos, y se utilizan para examinarlos y realizar predicciones.

Los árboles de decisión poseen una estructura formada por los siguientes elementos:

- Nodo: Se corresponden a los nombres o identificadores de los atributos que caracterizan al conjunto de datos. El nodo inicial o nodo raíz contiene la muestra total de atributos que definen a los datos.
- Rama: Representan a las variables de decisión o las condiciones que cumplen los objetos para separarse unos de otros.
- Hoja: Son finalmente los conjuntos o grupos de datos resultantes de la división que realiza el algoritmo.

El algoritmo realiza una clasificación discreta de los objetos, determinando a qué clase pertenece, mediante la decisión de qué rama escoger. Para esto, se asume que los grupos o clases que se formarán serán disjuntas, es decir, una instancia u objeto no puede pertenecer a dos clases a la vez. Esta misma condición se cumplirá para cada partición o sub-árbol que se forme, característica particular que tienen los árboles de decisión conocida como propiedad exhaustiva.

Existen diversos algoritmos de aprendizaje que se pueden utilizar para obtener un árbol de decisión. El algoritmo utilizado puede determinar aspectos como la compatibilidad con el tipo de variables de entrada y salida, el procedimiento de evaluación de la distancia

entre los grupos generados en cada división, y también la cantidad de ramas que se obtengan cada vez que un nodo se divide. En secciones posteriores se explicará el algoritmo CART con el cual se pueden obtener árboles con sólo dos ramas por cada división de los nodos, y es por esta razón que se les llaman árboles binarios.

Una de las ventajas de utilizar los árboles de decisión es que funcionan muy bien con variables categóricas, evitando realizar transformaciones de los datos. Otra ventaja de su uso es que permite una interpretación sencilla aún para usuarios sin conocimiento técnico de minerías de datos de las decisiones tomadas por el modelo para sus predicciones, situación que en algoritmos como las redes neuronales no es posible deducir.

Desventajas de los árboles de decisión es que son sensibles ante pequeños cambios en los datos, y además, dado que las decisiones para clasificar se realizan considerando una variable predictora a la vez, es difícil detectar las posibles relaciones entre los atributos y se pueden llegar a omitir algunos.

2.2.3. Aplicaciones

La capacidad predictiva de los modelos de minería de datos han cambiado el diseño de las estrategias empresariales, ya que permiten entender el presente para anticiparse al futuro. En la actualidad existen diversos ámbitos de aplicación para estas técnicas que cada vez son más importantes:

- Marketing: la minería de datos se utiliza para explorar bases de datos cada vez mayores y mejorar la segmentación del mercado. Analizando las relaciones entre parámetros como edad de los clientes, género, gustos, etc., es posible predecir su comportamiento para dirigir campañas personalizadas de fidelización o captación. La Minería de Datos en marketing predice también qué usuarios pueden darse de baja de un servicio, qué les interesa según sus búsquedas o qué debe incluir una lista de correo para lograr una tasa de respuesta mayor.
- Comercio minorista: los supermercados, por ejemplo, emplean los patrones de compra conjunta para identificar asociaciones de productos y decidir cómo situarlos en los diferentes pasillos y estanterías. La Minería de Datos detecta además qué ofertas son las más valoradas por los clientes.

- Bancos: éstos recurren a la minería de datos para entender mejor los riesgos del mercado. Es habitual que se aplique a la calificación crediticia (rating) y a sistemas inteligentes antifraude para analizar transacciones, movimientos de tarjetas, patrones de compra y datos financieros de los clientes. La Minería de Datos también permite a la banca conocer más sobre las preferencias de sus clientes o hábitos en internet para optimizar el retorno de sus campañas de marketing, estudiar el rendimiento de los canales de venta o gestionar las obligaciones de cumplimiento de las regulaciones.
- Medicina: el uso de estas técnicas favorece diagnósticos más precisos. Al contar con toda la información del paciente —su historial, examen físico y patrones de terapias anteriores— se pueden prescribir tratamientos más efectivos. También posibilita una gestión más eficaz, eficiente y económica de los recursos sanitarios al identificar riesgos, predecir enfermedades en ciertos segmentos de la población o pronosticar la duración del ingreso hospitalario.
- Fraude: como se expresa en este trabajo, un ámbito de aplicación es la detección de fraude, sobre todo en la actualidad a través de uso de la tecnología e internet los fraudes han aumentado, pero también su control y la detección de patrones de conducta a través de estrategias como Minería de Datos. Por ejemplo, en la banca existen fraudes en uso de tarjetas de créditos, estas técnicas nos ayudan a establecer patrones de perfiles, analizando grandes volúmenes de datos para poder predecir los mismos.

2.3. Software empleado para Minería de Datos

2.3.1. R

R [R] es un lenguaje y un entorno para gráficos y computación estadística. Es un proyecto GNU que es similar al lenguaje y entornos S que fue desarrollado en Bell Laboratories (antes ATT, ahora Lucent Technologies) por John Chambers y sus colegas. R se puede considerar como una implementación diferente de S. Hay algunas diferencias importantes, pero gran parte del código escrito para S se ejecuta inalterado en R. A su vez se compila y se ejecuta en una amplia variedad de plataformas UNIX y sistemas similares

(incluidos FreeBSD y Linux), Windows y MacOS.

R proporciona una amplia variedad de técnicas estadísticas (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, agrupamiento, etc) y técnicas gráficas y es altamente extensible. El lenguaje S es a menudo el vehículo de elección para la investigación en metodología estadística y R proporciona una ruta de código abierto para participar en esa actividad.

Uno de los puntos fuertes de R es la facilidad con la que se pueden producir gráficos con calidad de publicación bien diseñados, incluidos símbolos matemáticos y fórmulas cuando sea necesario. Se ha tenido mucho cuidado con los valores predeterminados para las opciones de diseño menores en los gráficos, pero el usuario conserva el control total.

Este software incluye:

- Una instalación eficaz de manejo y almacenamiento de datos.
- Un conjunto de operadores para cálculos en matrices.
- Una colección amplia, coherente e integrada de herramientas intermedias para el análisis de datos.
- Facilidades gráficas para el análisis de datos y visualización en pantalla o en papel.
- Un lenguaje de programación bien desarrollado, simple y efectivo que incluye condicionales, bucles, funciones recursivas definidas por el usuario e instalaciones de entrada y salida.

Un entorno de programación es un programa o conjunto de programas que engloban todas las tareas necesarias para el desarrollo de un programa o aplicación. Esta clasificación de entorno pretende caracterizarlo como un sistema coherente y totalmente planificado, en lugar de una acumulación incremental de herramientas muy específicas e inflexibles, como suele ser el caso de otros programas de análisis de datos.

R, como S, está diseñado en torno a un verdadero lenguaje informático y permite a los usuarios agregar funciones adicionales mediante la definición de nuevas funciones. Gran parte del sistema está escrito en el dialecto R de S, lo que facilita a los usuarios seguir las elecciones algorítmicas realizadas. Para tareas de computación intensiva, el código C, C++ y Fortran se puede vincular y llamar en tiempo de ejecución. Los usuarios

avanzados pueden escribir código C para manipular objetos R directamente.

Muchos usuarios piensan en R como un sistema de estadísticas. Se prefiere pensar en él como un entorno en el que se implementan técnicas estadísticas. R se puede ampliar (fácilmente) mediante paquetes. En la actualidad hay unos ocho paquetes que se suministran con la distribución R y muchos mas están disponibles a través de la familia de sitios de Internet CRAN que cubren una amplia gama de estadísticas modernas.

R tiene su propio formato de documentación similar a LaTeX, que se utiliza para proporcionar documentación completa, tanto en línea en varios formatos como en papel.

2.3.2. R Studio

Es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

RStudio está disponible para Windows, Mac y Linux o para navegadores conectados a RStudio Server o RStudio Server Pro (Debian / Ubuntu, RedHat / CentOS, y SUSE Linux). RStudio permite un análisis y desarrollo para que cualquiera pueda analizar los datos con R.

Características

Es un IDE construido exclusivo para R; al ser exclusivo obtiene al momento de programar diversas ventajas: como ser, el resaltado de sintaxis, auto completado de código y sangría inteligente. Permite a su vez ejecutar el código directamente desde el editor de código fuente y colaboración. Cuenta con Documentación y soporte integrado. Administración sencilla de múltiples directorios de trabajo mediante proyectos, navegación en espacios de trabajo y visor de datos, tiene un depurador interactivo para diagnosticar y corregir los errores rápidamente, siendo a su vez muy potente para auditar utilizando, sweave y r Markdown.

2.3.3. Librerías de R utilizadas

caret

La librería caret (Classification And Regression Training) provee funciones que facilitan el uso de decenas de métodos complejos de clasificación y regresión. Utilizar este paquete en lugar de las funciones originales presenta dos ventajas:

- Permite utilizar un código unificado para aplicar reglas de clasificación muy distintas, implementadas en diferentes paquetes.
- Es más fácil poner en práctica algunos procedimientos usuales en problemas de clasificación. Por ejemplo, hay funciones específicas para dividir la muestra en datos de entrenamiento y datos de test o para ajustar parámetros mediante validación cruzada.

Contiene herramientas para:

- División de conjunto de datos.
- Pre-procesamiento.
- Selección de características.
- Ajuste de parámetros de los modelos.
- Estimación de la importancia de las variables.
- entre otras funcionalidades.

Para realizar el split de conjuntos de datos la librería cuenta con la función *createDataPartition* que, de forma predeterminada, realiza una división aleatoria estratificada de los datos. Otra de las herramientas principales de la librería es la función *train* que se puede utilizar para:

- evaluar, mediante remuestreo, el efecto de los parámetros de ajuste del modelo en el rendimiento,
- elegir el modelo *óptimo* a través de estos parámetros,
- estimar el rendimiento del modelo a partir de un conjunto de entrenamiento.

Provee además funcionalidades para visualización de datos, permitiendo generar Matrices de scatterplots, gráficos de densidades acumuladas, box plots, scatter slots, entre otros.

rpart

Para crear modelos basados en decisión y predecir en base a ellos se emplea la librería rpart, siendo mas implementada para este tipo de modelos.

La libreria rpart() fue creada en 2012 por Therry Therneau, Beth Akinson y Brian Ripley, con licencias GPL 2 y 3 que brinda la posibilidad de trabajar con CARTs.

Para trabajar con esta librería se emplea la siguiente estrategia:

1. *Generar árbol de desición:* Para ello usaremos la función rpart(fórmula,data,method,control). cuyos parámetros mas importantes son:
   - data: Incluir tabla de datos.
   - method: Indicar método *class* para clasificación o *anova* para regresión.
   - control: una lista de características que controlan el algoritmo.

rpart plot

Esta función es una interfaz simplificada de las gráficas *prp*, con sólo los argumentos más útiles de esa función y con diferentes valores predeterminados para algunos de los argumentos. Los diferentes valores predeterminados significan que esta función crea automáticamente un gráfico de color adecuado para el tipo de modelo (mientras que *prp* por defecto crea un gráfico mínimo).

función

$rpart.plot(x = stop(\text{"}no'x'arg\text{"}), type = 2, extra = \text{"}auto\text{"}, under = FALSE,$
$fallen.leaves = TRUE, digits = 2, varlen = 0, faclen = 0, roundint = TRUE,$
$cex = NULL, tweak = 1, clip.facs = FALSE, clip.right.labs = TRUE,$
$snip = FALSE, box.palette = \text{"}auto\text{"}, shadow.col = 0, ...)$

Atributos

- x: un rpart objeto. El único argumento requerido.
- *type*: tipo de gráfico, puede tomar valores de 0 a 9 indicando la representación seleccionada.
- *extra*: muestra información adicional en los nodos, puede tomar diversos valores, si esta variable ingresamos el valor "auto", tomará automáticamente un valor para mostrar según el modelo representado. Por otro lado se pueden ingresar valores del 0 a 100 con los cuales asume diversos valores para su uso.
- *under*: se aplica sólo si extra >0. Por defecto es FALSE, lo que significa poner el texto adicional en el cuadro. Se debe indicar TRUE para poner el texto debajo del cuadro.
- $fallen.leaves$: predeterminado TRUE para colocar los nodos hoja en la parte inferior del gráfico. Puede ser útil usarlo FALSE si el gráfico está demasiado lleno y el tamaño del texto es demasiado pequeño.
- *digits*: el número de dígitos significativos en los números mostrados. Por defecto 2. Si es 0, usa getOption("digits"). Si es negativo, use la formatfunción estándar (con el valor absoluto de digits). Cuando digits es positivo, se aplican los siguientes detalles: los números del 0.001 a 9999 se representan sin un exponente. Los números fuera de ese rango se imprimen con un exponente (múltiplo de 3).
- *varlen*; longitud de los nombres de las variables en el texto en las divisiones (y, para las respuestas de clase, la clase en la etiqueta del nodo). Predeterminado 0, lo que significa mostrar los nombres completos de las variables. El valor predeterminado es 0 donde usa el valor completo, luego puede tomar valores, menores o mayores a 0.
- $faclen$; longitud de los nombres de nivel de factor en divisiones. Predeterminado 0, lo que significa mostrar los nombres completos de los factores. Los valores posibles son los valores anteriores, excepto que para la retro compatibilidad con text.rpartel.
- *roundint*; si roundint=TRUE(predeterminado) y todos los valores de un predictor en los datos de entrenamiento son números enteros, las divisiones para ese predictor se redondean a un número entero. Por ejemplo, mostrar en nsiblings <3 lugar de nsiblings <2.5. Si roundint=TRUE los datos utilizados para construir el modelo

ya no están disponibles, se emitirá una advertencia. El uso roundint=FALSE se aconseja si los valores no enteros son de hecho posible que un predictor, a pesar de que todos los valores en los datos de entrenamiento para que el predictor son integrales.

- *cex*: predeterminado NULL, lo que significa calcular el tamaño del texto automáticamente. Dado que los tamaños de fuente son discretos, *cex* es posible que el que solicite no sea exactamente el *cex* que obtenga.
- *tweak*: éste ajusta la variable anterior cex, Usar tweakes a menudo más fácil que especificar cex. El valor predeterminado tweakes 1, lo que significa que no hay ajuste. Por ejemplo, usar tweak=1.2 para agrandar el texto un 20 %. Dado que los tamaños de fuente son discretos, es posible que un pequeño cambio para ajustar en realidad no cambie el tamaño de la letra o lo cambie más de lo que desea.
- *clip.facs*: por defecto FALSE. Si TRUE, imprime divisiones en factores como en female lugar de sex = female; se elimina el nombre de la variable y es igual.
- *clip.right.labs*: se aplica solo si type=3 o 4. El valor predeterminado TRUE significa recortar las etiquetas divididas a la derecha, es decir, no imprimir variable.
- *snip*: por defecto FALSE. Configure TRUE para recortar el árbol de forma interactiva con el mouse.
- *box.palett*: paleta para colorear los cuadros de nodo según el valor ajustado. Éste es un vector de colors, por ejemplo box.palette=c("green", "green2", "green4"). Los valores ajustados pequeños se muestran con colores al comienzo del vector; valores grandes con colores al final. Los cuantiles se utilizan para dividir los valores ajustados.
- *shadow.col*: color de la sombra debajo de las cajas. Por defecto 0, sin sombra.
- ...: se pasan argumentos adicionales prp y las rutinas de trazado.

ROSE

Crea una muestra de datos sintéticos ampliando el espacio de características de ejemplos de clases minoritarias y mayoritarias. Operacionalmente, los nuevos ejemplos se

extraen de una estimación de densidad de kernel condicional de las dos clases, como se describe en Menardi y Torelli (2013).

Uso

$ROSE(formula, data, N, p = 0{,}5, hmult.majo = 1, hmult.mino = 1, subset = options("subset")subset, na.action = options("na.action")na.action, seed)$

Parámetros

- *Formula*, un objeto de clase formula (o uno que pueda ser forzado a esa clase). El lado izquierdo (respuesta) debe ser un vector que especifique las etiquetas de clase. El lado derecho debe ser una serie de vectores con los predictores.
- *Datos*, un marco de datos, una lista o un entorno opcional (u objeto coercible a un marco de datos as.data.frame) en el que interpretar preferentemente la fórmula. Si no se especifica, las variables se toman de 'entorno (fórmula)'.
- *N*, tamaño de muestra deseado del conjunto de datos resultante generado por ROSE. Si falta, se establece igual a la longitud de la variable de respuesta en formula.
- *P*, la probabilidad de los ejemplos de clases minoritarias en el conjunto de datos resultante generado por ROSE.
- *hmult.majo*, factor de contracción opcional que se multiplicará por los parámetros de suavizado para estimar la densidad del núcleo condicional de la clase mayoritaria.
- *hmult.mino*, factor de contracción opcional que se multiplicará por los parámetros de suavizado para estimar la densidad de núcleo condicional de la clase minoritaria.
- *subset*, un vector opcional que especifica un subconjunto de observaciones que se utilizarán en el proceso de muestreo.
- *na.action*, una función que indica lo que debería suceder cuando los datos contienen 'NA'
- *seed*, un valor único, interpretado como un número entero, recomendado para especificar semillas y realizar un seguimiento de la muestra generada

Detalles ROSE ayuda en la tarea de clasificación binaria en presencia de clases raras. Produce una muestra sintética, posiblemente equilibrada, de datos simulados de acuerdo

con un enfoque de bootstrap suavizado.

Denotado por y la respuesta binaria y por x un vector de predictores numéricos observados en n clases $i = \{i_1, i_2, ..., i_n\}$, ejemplos sintéticos con etiqueta de clase $k = \{0, 1\}$ se generan a partir de una estimación del núcleo de la densidad condicional $f = (y/x) = k$. El kernel es una función de producto normal centrada en cada uno de los x_i con matriz de covarianza diagonal H_K . Aquí H_k, es la matriz de suavizado asintóticamente óptima bajo el supuesto de normalidad multivariante.

Básicamente, ROSE selecciona una observación que pertenece a la clase k y genera nuevos ejemplos en su vecindario, donde el ancho del vecindario está determinado por H_K . El usuario puede encogerse H_K variando argumentos h.mult.majo y h.mult.mino. El equilibrio está regulado por el argumento p, es decir, la probabilidad de generar ejemplos de la clase.

En su forma actual, los métodos basados en kernel se pueden aplicar sólo a datos continuos. Sin embargo, como ROSE incluye la combinación de muestreo excesivo y insuficiente como un caso especial cuando H_K tienden a cero, el supuesto de continuidad puede ser eludido utilizando una distribución de núcleo degenerada para dibujar ejemplos categóricos sintéticos. Básicamente, si el $j^{ésimo}$ componente de X_i es categórico, un clon sintético de X_i tendrá como $j^{ésimo}$ componente el mismo valor de la $j^{ésimo}$ componente de Xi .

Valor

El valor es un objeto de clase ROSE que tiene componentes.

- *Lamada*, la llamada coincidente.
- *Método*, el método utilizado para equilibrar la muestra. La única opción posible es ROSE.
- *Datos*, un objeto de clase que data.frame contiene nuevos ejemplos generados por ROSE.

Consideraciones El propósito de ROSE es generar nuevos ejemplos sintéticos en el espacio de características. El uso de fórmula está destinado únicamente a distinguir la variable de respuesta de los predictores. Por lo tanto, fórmula no debe confundirse con el

suministrado para ajustar un clasificador en el que la especificación de transformaciones o interacciones entre variables puede ser sensible / necesaria. En la versión actual se ROSE descartan posibles interacciones y transformaciones de predictores especificados en fórmula automáticamente. El análisis automático de fórmula es capaz de gestionar prácticamente todos los casos en los que se ha probado, pero se advierte al usuario que tenga cuidado en la especificación de funciones entrelazadas de predictores. Cualquier informe sobre un posible mal funcionamiento del mecanismo de análisis es bienvenido.

ggplot2

La librería ggplot2 es un paquete de visualización de datos para el lenguaje R que implementa lo que se conoce como la "Gramática de los Gráficos", que no es más que una representación esquemática y en capas de lo que se dibuja en dichos gráficos, como lo pueden ser los marcos y los ejes, el texto de los mismos, los títulos, así como, por supuesto, los datos o la información que se grafica, el tipo de gráfico que se utiliza, los colores, los símbolos y tamaños, entre otros.

Las funciones de ggplot2 permiten obtener gráficas de gran calidad y con muchas opciones para representar los datos y extraer así información relevante de los conjuntos de datos que se estudien, como relaciones, distribuciones, patrones y demás comportamientos aplicables tanto al análisis de datos exploratorio como a los modelos predictivos.

Esta librería presenta diferentes tipos de gráficas para la visualización y análisis de datos, a continuación se mostrarán dos ejemplos de ellas que son las usadas en el presente trabajo:

Gráficos de Barras (Barplot)

Uno de los tipos de gráficos más comunes en el análisis de datos es el gráfico de barras, en el que simplemente se representa con barras de distintas "alturas" las dimensiones de una cantidad numérica comparada con otra.

La fórmula a aplicar es la siguiente:

$ggplot(data =, aes(x =)) + geom\_bar()$

todo gráfico hecho con ggplot debe tener un primer argumento ggplot() en donde

debe especificarse cuál es el dataset que contiene la información que se desea graficar. Esta representa la primera capa que guarda la información del conjunto de datos de partida. Con el argumento $data =$ se establece que el dataset es el indicado, y en términos de R este conjunto de datos puede ser cualquier dataframe construido o cargado previamente. Luego, se observa en la fórmula el comando aes(), que se refiere a la "estética" del gráfico, es decir, en este caso se especifica o "mapea" cuál variable del conjunto de datos es el que se va a representar en el eje "x". Una vez completada esta función, se agrega una segunda capa usando el operador + , y luego se establece que el gráfico a construir es de tipo barra con la función $geom\_bar()$.

Histograma (Histogram) y Gráfico de Densidad (Density Plot)

Un histograma es un tipo de gráfico de barras en donde la altura de éstas hacen referencia a la frecuencia con la que aparecen los valores que se representan. Los histogramas son utilizados en la mayoría de las ocasiones para observar la distribución de alguna variable continua, lo que nos permite de una manera sencilla y rápida obtener información como el comportamiento de los datos, tendencias, variabilidad u homogeneidades, entre otros.

La formular a aplicar es la siguiente:

$ggplot(diamonds) + geom\_histogram(binwidth =, aes(x =), fill =) + xlab("t") + ylab("") + ggtitle("") + theme\_minimal()$

Un histograma permite ver la distribución de los datos, que podria presentarse sesgada hacia uno de los lados. Sin embargo,con la función geom_histogram(), uno de los argumentos es el 'bidwidth', es decir, el ancho de las barras que recogen los rangos de representación de la variable. Este argumento es importante ya que de no establecer un valor adecuado, se puede perder la forma o los detalles de la distribución.

doParallel

El doparallel es un paquete de R 2.14.0 y posteriores proporciona funciones para la ejecución paralela de código R en máquinas con múltiples núcleos o procesadores o múltiples computadoras. Es esencialmente una mezcla de los paquetes snowy multicore. De forma predeterminada, el paquete doParallel utiliza una funcionalidad snow similar. Esta

debería funcionar bien en sistemas similares a Unix, pero la funcionalidad multicore está limitada a un sólo trabajador secuencial en sistemas Windows. En estaciones de trabajo con múltiples núcleos que ejecutan sistemas operativos similares a Unix, la llamada fork al sistema se usa para generar copias del proceso actual.

El backend admite opciones multicore indicadas a través de la función foreach. Las opciones multicore son compatibles con preschedule, set.seed, silent, y cores, que son análogos a los argumentos con nombres similares a mclapply, y se transmiten mediante el .options.multicore argumento para foreach. Las opciones admitidas son preschedule, que, al igual que su análogo, se pueden usar para dividir las tareas de modo que cada usuario obtenga una parte preprogramada de tareas, y attachExportEnvque se pueden usar para adjuntar el entorno de exportación en ciertos casos donde el alcance léxico de R no puede encuentre una exportación necesaria.

La función stopImplicitClusterse puede utilizar en viñetas y otros lugares donde es importante cerrar explícitamente el clúster creado implícitamente.

Uso

$registerDoParallel(cl, cores = NULL, \dots)stopImplicitCluster()$

Parámetros

- *cl*, un objeto de clúster devuelto por makeCluster o el número de nodos que se crearán en el clúster. Si no se especifica, en Windows se crea y utiliza un clúster de tres trabajadores.
- *cores*, el número de núcleos que se utilizarán para la ejecución en paralelo. Si no se especifica, el número de núcleos se establece en el valor de options("cores"), si se especifica, o en la mitad del número de núcleos detectados por el paquete parallel.
- .., opciones de paquete. Actualmente, sólo nocompilese admite la opción. Si nocompilese establece en TRUE, el soporte del compilador está deshabilitado.

Capítulo 3

Comprensión del negocio

En este Capítulo se presentan la importancia de la prevención y detección del fraude dentro de una empresa de seguros. También se detallan los diferentes tipos de fraudes que existen, junto con algunos ejemplos para su comprensión y los factores relevantes para la detección de los mismos.

En la última parte del capítulo se presenta brevemente un detalle con la descripción de los datos y el proceso de recolección de los mismos.

3.1. El fraude en seguros

La prevención y detección temprana de fraude dentro de una aseguradora es una tarea central, tanto para quienes están en la suscripción del riesgo, como la gestión de siniestros.

La disminución del fraude es uno de los principales retos de la industria aseguradora tanto a nivel nacional como mundial. Las pérdidas económicas derivadas del delito y los costos derivados de la adopción de los marcos de prevención y detección, convierten esta realidad en un asunto de absoluta trascendencia para dicho sector.

Una de las características que presenta un obstáculo en la prevención, es que el fraude en los seguros puede tomar diferentes formas y ser llevado adelante por cualquiera de las partes involucradas en la operatoria, incluso de las misas aseguradoras, a través de

funcionarios, intermediarios, contadores, auditores, consultores, liquidadores de siniestros, terceros denunciantes y asegurados.

¿Qué tienen en común el caso de una camioneta que se incendia en un campo antes de contratar el seguro, de un futbolista amateur que denuncia una lesión ocurrida durante el juego como accidente de trabajo, la denuncia de un siniestro automotor en el que las personas que intervienen ocultan un vínculo familiar excluido de la cobertura?

En todos ellos subyace un intento de fraude a las aseguradoras. Claro está que en algunos casos se trata de tentativas individuales y en otros de bandas organizadas que persiguen un fin de lucro ilegítimo.

Estas organizaciones violan derechos humanos básicos aprovechándose de la necesidad de personas que resultan más vulnerables por el contexto social que las rodea, lo que facilita la tarea de utilizarlas como parte de sus maniobras delictivas.

El fraude en el seguro tiene consecuencias que van más allá del perjuicio económico para la propia aseguradora, puesto que afecta al conjunto del sistema y por ello es importante generar conciencia dentro de todos los eslabones de la cadena.

Defensas empleadas en casos de intento de fraude

Existen infinidad de casos donde se detectan intentos de fraude. Estos casos dejan como enseñanza la importancia de contar con actitudes defensivas fundadas en el uso de herramientas tecnológicas, el cruce de datos y la colaboración entre aseguradoras.

- La importancia de las herramientas tecnológicas: Una camioneta aparece incendiada. La denuncia del siniestro indica que fue a consecuencia de un incendio en un campo donde el dueño del vehículo se encontraba trabajando. Surgió una alerta que indicaba que podría tratarse de un caso de fraude: la póliza fue modificada con una ampliación de cobertura y el siniestro fue denunciado inmediatamente. Gracias a herramientas de geolocalización y fotografías satélitales, se podría detectar que el incendio en el campo fue anterior a la ampliación de cobertura, por lo cual el siniestro es rechazado por no estar cubierto.
- El cruce de datos y la colaboración entre aseguradoras: Una mujer denuncia lesiones por haberse caído en un supermercado. En el cruce de datos de los sistemas

informáticos, vinculando el DNI de la denunciante, aparecían antecedentes similares. En la denuncia, señalaba haber sufrido una fractura de cadera, pero también había denunciado una lesión similar en los años anteriores con otras tres aseguradoras. En la mayoría de los casos, la denuncia era en el ámbito del transporte público: una por una caída dentro de la unidad, la otra por un desmayo y la tercera por un golpe en el descenso del vehículo. Si se cruzara esta información entre las aseguradoras, podría llegar a detectarse o alertarse la posibilidad del fraude..

- La globalización de la información en el mundo digital: Las herramientas de comunicación, como las redes sociales son también aliadas para la investigación de posibles fraudes en seguros. Un ejemplo una denuncia de un siniestro en el que están 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 podría llevar adelante una investigación para evaluar si se trataba de un fraude, en el cual detectar, por ejemplo, a través de las redes sociales, que entre ambos había una relación familiar directa, hecho que podría motivar el rechazo del siniestro.

Las TICS como herramientas de prevención del fraude

La perspicacia de la persona que inicia una investigación para confirmar una maniobra de fraude, en la actualidad, tiene como aliadas a las tecnologías de la información y la comunicación, que cada vez brindan mayores posibilidades de investigación.

Existen en el mercado diferentes tipos de herramientas informáticas que permiten el cruce de datos y son de uso generalizado por todos los operadores de seguros. El sistema Orion de Cesvi Argentina [ORI] y el soporte que brinda la Superintendencia de Seguros de la Nación [SSN] a través de sus sistemas son ejemplos claros de ello.

Estas herramientas son funcionales a uno de los principios básicos de la lucha contra el fraude en el mercado asegurador: la colaboración permanente entre aseguradoras. Hoy, el sector asegurador tiene en claro que el fraude no es un problema de una sola empresa, sino del mercado en su conjunto y en función de ese precepto, se trabaja a diario.

Además de las herramientas disponibles para todo el mercado, cada aseguradora también enriquece ese proceso con sistemas propios.

Algunos de los ejemplos mencionados permiten observar que los avances tecnológicos aplicados de manera sistemática son aliados fundamentales de una política de prevención y detección de fraudes en el seguro, siempre que exista un compromiso de todo el personal de la empresa para detectar y reportar las tentativas que existen.

El Hecho de que las aseguradoras resguarden su información en sistemas de base de datos, o también la posibilidad de compartir esta información a través de entes regulatorios como la Superintendencia de Seguros, presenta una oportunidad para aplicar técnicas de minería de datos generando de esta forma herramientas que apoyen la toma de decisiones en éste u otros ámbitos.

La política antifraude y su influencia en la competitividad

Más allá de los casos puntuales donde se percibe una tentativa fraudulenta para sacar beneficio por parte de una persona en particular u organizaciones con fines delictivos, hay un aspecto que deja expuesto claramente que el fraude es un tema que afecta a todos los actores del mercado (Productores Asesores de Seguros, Proveedores, Re Aseguradores, etc) y no sólo a las aseguradoras.

Las acciones fraudulentas que no se detectan, se pagan, por ejemplo en los siniestros, y tienen un impacto directo en los costos de las aseguradoras. Una empresa de seguros, como la de cualquier otra actividad, debe ser sustentable y competitiva; es decir, no puede soportar pérdidas sin hacer medidas correctivas y una de ellas, es la tarifa que cobra por el servicio que brinda.

Por ello, si no se le otorga a la prevención y lucha contra el fraude la debida importancia, la concreción de estas conductas tiene un impacto directo en el precio que los asegurados pagan por las coberturas que contratan. Es decir, si existe una ineficiente gestión para evitar el fraude, el resultado será un prorrateo de las pérdidas a través de las tarifas que pagan los asegurados como consecuencia de no tomar acciones contra la detección del fraude.

Una proactiva y eficaz política de prevención de fraudes termina convirtiéndose así en una poderosa herramienta para mejorar la competitividad de la aseguradora y un acto de responsabilidad en beneficio de los asegurados.

3.2. Tipos de fraudes

La Superintendencia de Seguros de la Nación [SSN] define al fraude como "una acción u omisión, perpetrada en el marco de una relación de seguros, incluyendo la conducta de comercializadores no autorizados, para recabar una ventaja o beneficio indebido, para provecho propio o de un tercero".

Entre las modalidades, la Superintendencia destaca de manera enunciativa:

- Engaño.
- Aserción de lo que es falso o disimulación de lo verdadero.
- Artificio.
- Astucia.
- Abuso de confianza.

Como se menciono en párrafos anteriores el fraude puede configurarse con la complicidad de personal de la propia entidad, de servicios tercerizados, de profesionales que actúan como auxiliares de la actividad aseguradora u otros canales de comercialización. El combate al fraude en los seguros, se justifica en los serios perjuicios financieros que genera, como así también los impactos a la reputación y otros costos sociales y económicos.

La SSN bajo la Resolución 38.477 del 17 Julio de 2014, en su expediente 62.550 de NORMAS SOBRE POLÍTICAS, PROCEDIMIENTOS Y CONTROLES INTERNOS PARA COMBATIR EL FRAUDE establece en el artículo tercero que: las entidades aseguradoras deberán adoptar una política para combatir el fraude que como mínimo, observe los siguientes aspectos:

- La elaboración de un manual que contemple los mecanismos y procedimientos para luchar contra el fraude de seguros.
- La designación de un responsable de contacto que deberá adecuarse a los recaudos previstos en el Artículo 2, párrafo segundo, de la citada Resolución.
- La elaboración de una memoria de casos investigados por sospecha de fraude de seguros, en la que se registre un resumen o síntesis que describa brevemente los principales contenidos del caso, acorde a las pautas establecidas en dicha resolución.

El control del fraude es un objetivo compartido por los operadores del mercado y por el organismo de control dado que por sus elevados niveles, los efectos repercuten en los usuarios que ven incrementados sus costos o disminuidas sus garantías como consecuencia de esta actividad ilícita.

IRIS es un sistema de Control de Fraudes para la rama automotor desarrollado por la Superintendencia de Seguros de la Nación. Su objetivo básico es detectar denuncias de siniestros que, por sus características, podrían ser parte de maniobras fraudulentas.

El principal motivo que impulsa el proyecto IRIS es alcanzar la finalidad descripta haciendo uso de la capacidad que tiene la SSN para recolectar información.

Al inicio del sistema la base de datos IRIS [IRI] se alimenta diariamente con los siniestros que afectan las coberturas del casco. IRIS también permite:

- Conocer la Historia Siniestral de un vehículo.
- Obtener estadísticas.
- Efectuar el seguimiento de la siniestralidad.
- Conocer periódicamente sus variaciones.

3.3. Factores a analizar para la detección de fraudes

En casos de fraude en siniestros existen diversos datos sobre el hecho ocurrido a tener en cuenta, esto es aun mas complejo por que las coberturas de cada seguro son distintas y los hechos que ocasionan lo siniestros aun mas, por ello existen análisis para cada uno de ellos. A continuación se mostrará por tipo de siniestros dentro de seguros de automotores cuáles son aquellos datos o declaraciones en cada uno de los hechos que identifican alertas de fraude para su futuro análisis.

Daño Parcial

- Los comprobantes de reparación son de lugares distantes al domicilio del socio.
- El siniestro ocurre en una ubicación remota
- El daño es de magnitud y no hay indicación de servicio de remolque o lesionados

- Daños de magnitud y el socio al momento de la denuncia presenta factura de reparación

Daños Totales

- El daño es de magnitud y no hay indicación de servicio de remolque o lesionados

Incendio Parcial

- El asegurado/tercero dificulta la verificación de su unidad.
- El asegurado se presenta con una copia del título o el título no está a su nombre
- El siniestro ocurre en una ubicación remota
- El daño es de magnitud y no hay indicación de servicio de remolque o lesionados.
- Daños de magnitud y el socio al momento de la denuncia presenta factura de reparación.
- Los comprobantes de reparación son de lugares distantes al domicilio del socio.

Incendio Total

- El daño es de magnitud y no hay indicación de servicio de remolque o lesionados.

Robo Parcial

- Colisión con árboles, muros, pilares, etc.
- Detalles vagos e inexactos, falta de precisión de datos.
- Colisión en reversa con daños de magnitud en el vehículo del tercero, o donde el vehículo asegurado es de mayor porte.
- Las partes involucradas son familiares, conocidos o amigos, vecinos, etc.
- Daño por colisión debido a frenado imprevisto (paso de animales, peatones, etc.).
- El asegurado lleva en su propio vehículo al lesionado, sin intervención de policía y servicios de emergencias.
- El lugar de atención primaria es distante del lugar del accidente o del domicilio del tercero.

- Documentación con evidencias de adulteración.
- El accidente ocurre en lugares pocos transitados, en horarios pocos frecuentes o en zona de boliches, pubs, etc.
- El/los transportados no tienen relación de parentesco o amistad (no corresponde para hecho NO transportado).

Robo Total

- El asegurado se muestra muy tranquilo o indiferente al momento de la denuncia.
- Notifica a la policía luego de denunciar el siniestro en la aseguradora.
- El asegurado/tercero dificulta la verificación de su unidad.
- El Asegurado se presenta con una copia del título o el título no está a su nombre.
- El siniestro ocurre en una ubicación remota.
- El daño es de magnitud y no hay indicación de servicio de remolque o lesionados.
- Daños de magnitud y el socio al momento de la denuncia presenta factura de reparación-
- Los comprobantes de reparación son de lugares distantes al domicilio del socio.

Terceros

- Colisión con árboles, muros, pilares, etc.
- Detalles vagos e inexactos, falta de precisión de datos
- Colisión en reversa con daños de magnitud en el vehículo del tercero, o donde el vehículo asegurado es de mayor porte
- Las partes involucradas son familiares, conocidos o amigos, vecinos, etc.
- Daño por colisión debido a frenado imprevisto (paso de animales, peatones, etc.)

Lesionados

- Colisión con árboles, muros, pilares, etc.
- Detalles vagos e inexactos, falta de precisión de datos

- Colisión en reversa con daños de magnitud en el vehículo del tercero, o donde el vehículo asegurado es de mayor porte
- Las partes involucradas son familiares, conocidos o amigos, vecinos, etc.
- Daño por colisión debido a frenado imprevisto (paso de animales, peatones, etc.)
- El asegurado lleva en su propio vehículo al lesionado, sin intervención de policía y servicios de emergencias
- El lugar de atención primaria es distante del lugar del accidente o del domicilio del tercero
- Documentación con evidencias de adulteración
- El accidente ocurre en lugares pocos transitados, en horarios pocos frecuentes o en zona de boliches, pub, etc.
- El/los transportados no tienen relación de parentesco o amistad (no corresponde para hecho NO transportado).

—-

3.4. Recolección de los datos

Para realizar este trabajo, se contó con los datos provistos por la empresa Rio Uruguay Seguros, la cual dio acceso a la base de datos transaccional mediante datos anonimizados para abordar esta tesis.

Además de estos datos, que incluyen datos de los asociados, siniestros y pólizas, también se obtuvieron datos correspondiente al proceso de detección de fraude que la empresa realiza actualmente.

En la actualidad, si bien la empresa cuenta con aproximadamente 8000 siniestros mensuales, el análisis de fraude no se hace sobre la totalidad de siniestros, sino sobre una muestra acotada. Que se define mediante algunos parámetros que se seleccionan al momento de ingresar el siniestro. Se conoce también que la conformación de dicho universo en la actualidad no es confiable. Sobre el universo acotado, se hacen las peritaciones y

estudios correspondiente mediante los cuales se define si el siniestro es fraudulento o no. Esta información queda indicada en el siniestro y la misma será utilizada también como insumo dentro de este trabajo.

3.5. Descripción del conjunto de datos

Como se mencionó en la sección anterior los datos utilizados para el presente trabajo no representan el volumen total de siniestros de la empresa, sino que se ha trabajado con un subconjunto de registros sobre el que la empresa analiza la presencia de indicios de fraude.

La determinación sobre si un caso debe ser analizado o no, se realiza de la siguiente manera:

Al registrarse un nuevo siniestro en la base de datos:

- se evalúan ciertos parámetros.
- se determina si ese siniestro pasa al proceso de análisis de fraudes.
- se realizan las peritaciones, análisis y estudios correspondientes para determinar si el hecho se debe categorizar como fraudulento o no.

Luego de la correspondiente evaluación se registra el resultado en la base de datos, indicando específicamente en un atributo si el registro presentó fraude o no. Dicho resultado de estas evaluaciones se ha utilizado en el contexto del presente trabajo.

El conjunto de datos provisto del sector de Análisis de Fraudes de la compañía está conformada por los siguientes atributos:

1. *nro_siniestro*: es un valor numérico que identifica unívocamente al siniestro en cuestión. Es un valor incremental que la empresa asigna para hacer referencia a un siniestro que sólo tiene valor interno, aunque el mismo también es informado a los socios y entes de control como la Superintendencia de Seguros de la Nación (SSN).
2. *franquicia*: consiste en un valor monetario fijo o un porcentaje que, en caso de siniestro, el asegurado soportará con su patrimonio. Sirve para reducir el importe de

la prima de un seguro. En algunos países se denomina “deducible”. Debido a que no todas las coberturas poseen franquicia en algunos casos este valor es igual a 0.

3. *sexo*: esta variable representa al género del socio (cliente que contrata la póliza) de la compañía. Puede tomar los valores: Masculino o Femenino.

4. *tipo_accidente*: valor que contempla los posibles tipos de accidente en los que se clasifican los siniestros de seguros de automotores esta compañía (si corresponde). Pude tomar los siguientes valores: Desplazamiento, En cadena, Robo, Incendio, Frontal, Lateral, Posterior, Vuelco, Otros. Siendo Otros, un valor posible para la carga.

5. *interv_municipal*: atributo que puede tomar dos valores: SI o NO. Indica si en el siniestro hubo intervención del ente municipal de la localidad en la que se produjo el siniestro.

6. *interv_policial*: atributo que puede tomar dos valores: SI o NO. Indica si en el siniestro hubo intervención policial en el lugar donde ocurrió el siniestro.

7. *horario_siniestro*: atributo que indica el momento del día en el que se produjo el siniestro. Puede tomar dos valores: Diurno o Nocturno.

8. *hora_dia*: atributo de tipo numérico que indica el horario en el que se produjo el siniestro de acuerdo a la declaración de quien realiza la denuncia del mismo.

9. *anio_vehiculo*: atributo en el que se registra el año de fabricación del vehículo asegurado.

10. *limite_casco*: atributo en el que se registra el valor numérico que expresa el monto máximo de cobertura por daños parciales y/o totales ocasionados al vehículo asegurado como consecuencia de robo/hurto, incendio y/o destrucción ante un posible siniestro.

11. *limite_RC*: atributo en el que se registra el valor numérico que expresa el monto máximo de cobertura ante un posible siniestro. Esto se conoce generalmente como Seguro de Responsabilidad Civil (RC) o Seguro contra Terceros y su objetivo es cubrir los reclamos que el asegurado pueda recibir por daños o perjuicios que él mismo o su vehículo puedan ocasionar a terceros. Es la cobertura mínima con la

que todo vehículo debe contar.

12. *es_fraude*: atributo que puede tomar dos valores: 0 y 1. Los registros que presentan valor 1 son aquellos que han sido reconocidos como fraudulentos. En el contexto de este trabajo es la variable objetivo.

13. *tipo_conductor*: atributo que indica si quien conducía el vehículo en el momento que ocurrió el siniestro era asegurado u otra persona. Puede tomar dos valores: Asegurados y Otros.

14. *antiguedad_socio*: atributo que toma un valor numérico que indica la antigüedad del socio en la empresa expresada en años. Dicho valor expresa la cantidad de años en los que el socio tuvo póliza con la aseguradora de manera ininterrumpida, por lo que se calcula como la diferencia entre la fecha de vencimiento de la póliza vigente y la fecha de inicio de vigencia de la primer póliza contratada.

15. *cant_rec_lesiones*: atributo que puede tomar un valor numérico que indica la cantidad de reclamos existentes para un siniestro con lesiones, ya que cada siniestro puede tener mas de un reclamo siempre que existan terceros involucrados en el siniestro.

16. *cant_rec_danios*: atributo que puede tomar un valor numérico que indica la cantidad de reclamos que existen en el siniestro de daños materiales, ya que cada siniestro puede tener mas de un reclamo siempre que existan terceros involucrados en el siniestro.

17. *cant_rec_casco*: atributo que puede tomar un valor numérico que indica la cantidad de reclamos que existen en el siniestro de casco, ya que cada siniestro puede tener mas de un reclamo siempre que existan terceros involucrados en el siniestro.

18. *edad_socio*: atributo que puede tomar un valor numérico que indica la edad del socio titular de la póliza en el siniestro en cuestión.

19. *monto_reclamado*: atributo que puede tomar un valor numérico que indica el valor monetario reclamado por ese siniestro para la compañía. Este valor es estimado y se carga al momento del ingresar el siniestro en la base de datos.

20. *dif_emision*: atributo que puede tomar un valor numérico que indica la diferencia

en días entre la fecha de emisión de la póliza y la fecha en la cual ocurrió el siniestro.

21. *dif_inicio_vigencia*: atributo que puede tomar un valor numérico que indica la diferencia en días entre la fecha de inicio de vigencia de la póliza y la fecha en la cual ocurrió el siniestro. La diferencia con el atributo *dif_emision* es que la fecha de inicio de la póliza indica el inicio de la cobertura y la fecha de emisión es la fecha en la cual se da de alta la póliza en la base de datos.

22. *en_localidad_guarda*: atributo que puede tomar un valor de tipo texto que indica el nombre de la localidad en la que está asignada la póliza del automóvil. La misma corresponde a la localidad en la que radica el vehículo y es declarada por el asegurado en el momento en el que se realiza el alta de la póliza.

23. *dos_ruedas_vigencia*: atributo que puede tomar dos valores: SI o NO. Indica si el asegurado tiene mas de dos denuncias por robo de ruedas dentro del período de vigencia de una misma póliza.

24. *tenencia_poliza*: atributo que puede tomar un valor numérico que indica desde cuando el asegurado tiene la misma póliza activa. Se indica en cantidad de años

25. *dif_denuncia*: atributo que puede tomar un valor numérico que indica la diferencia en días entre la fecha en la que ocurrió el siniestro y la fecha en que se realizó la denuncia del mismo.

En el próximo capítulo se detallan las actividades realizadas para obtener la vista minable empleada en este trabajo. Se presenta un análisis detallado del estado inicial del conjunto de datos y los pasos requeridos para completar y preparar los datos para aplicar los algoritmos de minería.

Capítulo 4

Pre-procesamiento y preparación de los datos

El primer paso en el proceso de extracción de conocimiento es recolectar los datos con los que se va a trabajar. En ocasiones los datos pueden pertenecer a distintas unidades de negocios de la propia organización, o incluso provenir de fuentes externas, lo que plantea la necesidad de integrar diversas fuentes de datos. Esta tarea no es sencilla, ya que se debe considerar que cada origen de datos emplea distintos formatos, granularidades, claves, entre otros aspectos, para el registro de la información. En este Capítulo se presentan las tareas de pre-procesamiento y preparación de los datos obtenidos para lograr la vista minable. En la primera Sección se presenta el Análisis Exploratorio de los Datos, a continuación el criterio empleado para la selección de atributos definitivos y finalmente las tareas propias de preparación de los datos.

4.1. Análisis exploratorio de los datos

Los resultados del proceso de descubrimiento de conocimiento dependen de seleccionar el algoritmo de minería de datos adecuados, como así también, de la calidad del conjunto de datos empleado. Es necesario identificar la relevancia de cada uno de los atributos presentes en el conjunto de datos, como así también la presencia de valores anómalos. En aplicaciones de detección de fraudes, los casos extraños o que no se ajustan al comportamiento regular

son de gran interés porque pueden estar indicando la necesidad de analizar con mayor detalle algún registro.

El conjunto de datos que ha facilitado la empresa para la ejecución de este trabajo cuenta con 982 registros y 21 atributos.

Para conocer en detalle los diferentes atributos que lo componen y la distribución de valores de cada uno se realizó un Análisis Exploratorio de los Datos. A continuación se presenta mediante las tablas 4.2 y 4.1 y las gráficas correspondientes que muestran las características principales.

Atributo	Cant.	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
franquicia	982	1,639.002	3,284.780	0	0	0	15,000
hora_dia	982	10.793	7.217	0	5	18	23
anio_vehiculo	982	2,005.916	7.339	1,969	2,000	2,011	2,018
limite_casco	965	173,355.900	223,221.200	0.000	87,000.000	194,000.000	3,850,000.000
limite_RC	982	1,079,430.000	2,693,634.000	0	0	0	18,000,000
antiguedad_socio	981	3.529	9.845	0.000	0.000	2.000	48.000
cant_rec_lesiones	982	0.029	0.205	0	0	0	2
cant_rec_danios	982	0.187	0.440	0	0	0	3
cant_rec_casco	982	1.048	0.601	0	1	1	7
edad_socio	643	27.680	270.647	−6,817.000	29.000	45.000	89.000
monto_reclamado	982	33,625.460	74,942.260	0	0	30,755.8	1,162,500
dif_emision	981	89.274	55.742	−3.000	43.000	133.000	386.000
dif_inicio_vigencia	981	82.540	54.008	0.000	36.000	125.000	352.000
tenencia_poliza	981	0.668	1.328	0.000	0.000	1.000	13.000
dif_denuncia	982	4.730	18.941	0	0	3	409

Tabla 4.1: Resumen de atributos numéricos

Atributo	Cant. no nulos	Cant. distintos
sexo	776	2
tipo_accidente	982	9
es_fraude	982	2
tipo_conductor	256	2
en_localidad_guarda	982	2
dos_ruedas_vigencia	982	2

Tabla 4.2: Resumen atributos no numéricos

La variable objetivo de este trabajo es el atributo es_fraude cuyo valor positivo es *SI* y representa el 26 % del total de los registros, como se puede observar en la Figura 4.1. El hecho de que el porcentaje de siniestros indicados como fraudulentos es significativamente bajo respecto del total de siniestros analizados denota un desbalance entre las clases, situación que puede presentar problemas al momento de identificar los fraudes debido a que están sub-representados.

También se desprende de este análisis exploratorio que tenemos datos erróneos, como la edad del socio, que en algunos casos da negativa, razón por la cual fue necesario, como explicaremos mas adelante llevar adelante una limpieza y reconstrucción de estos datos.

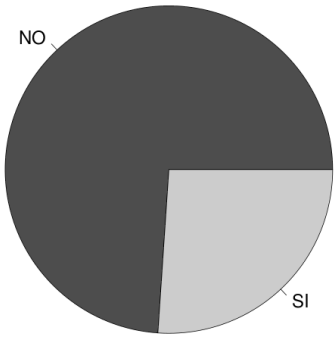


Figura 4.1: Proporción de siniestros con fraude comprobado y sin indicios de fraude.

Analizando los datos por tipos de accidente (atributo *tipo_accidente*) se observa que la mayor proporción de siniestros corresponde a hechos de robo, sin embargo en el porcentaje de fraude para cada tipo de accidente no existe una diferencia significativa, por lo que no se puede deducir a priori que un tipo de accidente presenta mayor indicio de fraude que otro (Figura 4.2).

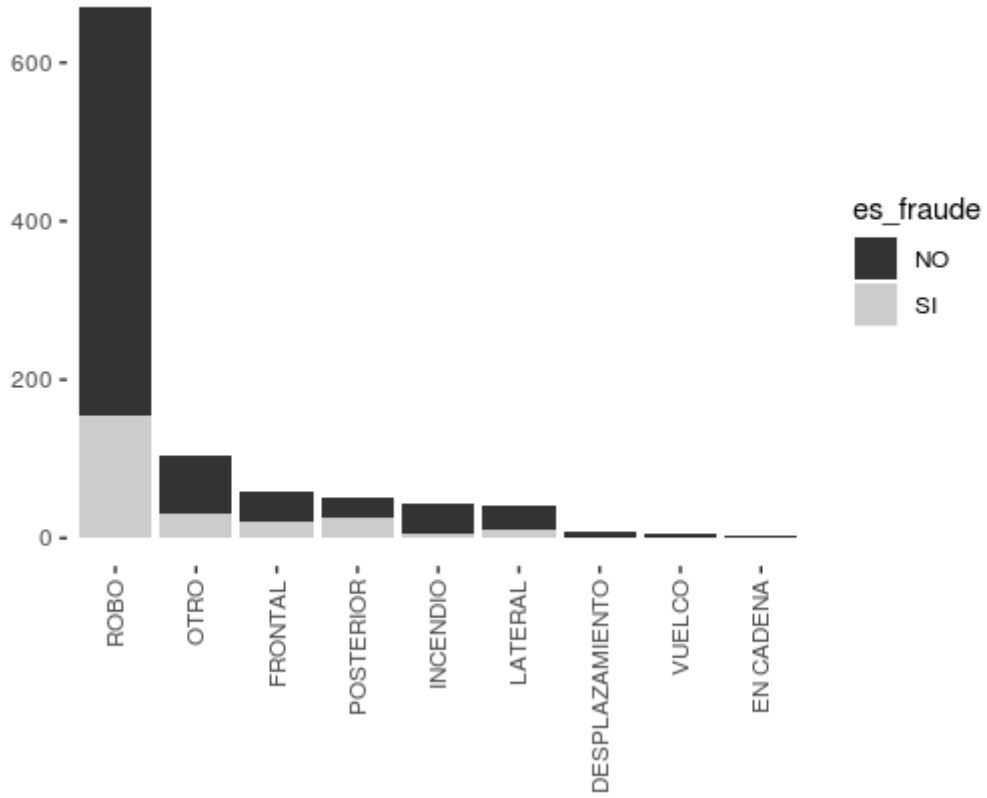


Figura 4.2: Cantidad de siniestros fraudulentos y sin indicio de fraude por tipo de accidente

Si se analizan los datos a partir de la diferencia en días entre ocurrencia del siniestro y la fecha de emisión de la póliza (o fecha del último endoso de la póliza) se observa que la distribución es similar para los casos fraudulentos y no fraudulentos (ver Figura 4.3)

Sin embargo se observan algunos detalles que requieren análisis:

- Si se analiza la caja correspondiente a los casos no fraudulentos se observa que existe un registro que toma un valor negativo, lo que indica que la fecha del siniestro fue anterior a la fecha de emisión de la póliza.
- Se observan algunos valores atípicos que indican que existen siniestros que ocurrieron luego del año de la emisión de la póliza. Posiblemente estos valores se deban a un error en el registro de la fecha de ocurrencia del siniestro, ya que la cobertura de la póliza no excede los 365 días.

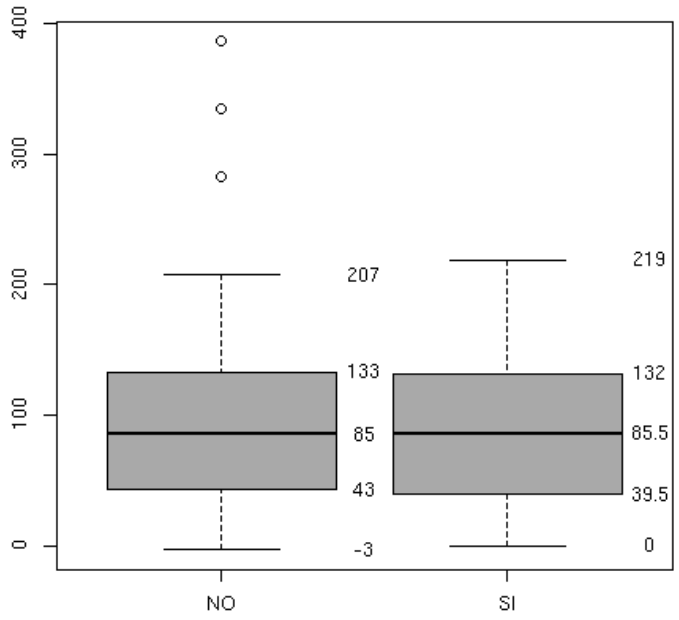


Figura 4.3: Diferencia en días entre Fecha de emisión de la póliza y Ocurrencia del Siniestro

Al analizar los registros tomando en consideración la diferencia en días entre la fecha de inicio de vigencia y la fecha de ocurrencia del siniestro (ver Figura 4.4) se observa que la distribución es similar tanto para los casos fraudulentos como para los que no lo son. Sin embargo se observa que existen registros en los que la diferencia entre estas fechas es

igual a 0, lo que indica que el siniestro ocurrió el mismo día que el inicio de vigencia de la póliza.

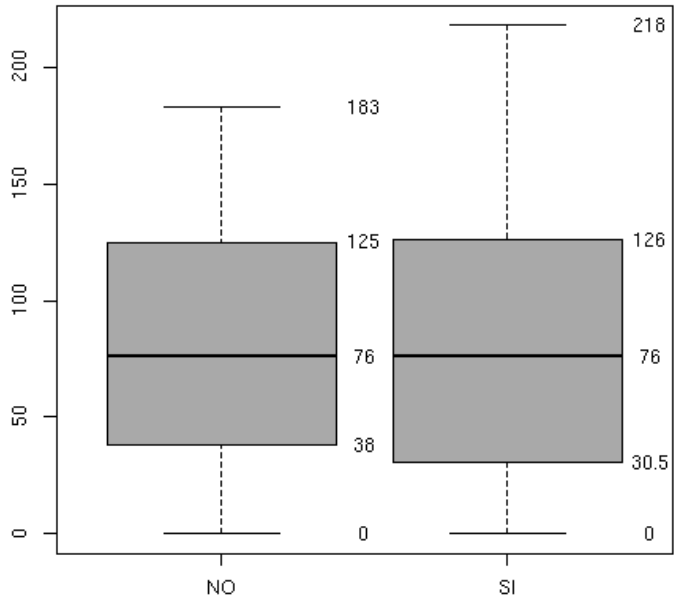


Figura 4.4: Diferencia en días entre Fecha de inicio de vigencia de la póliza y Ocurrencia del siniestro

Otro enfoque de interés para analizar los datos es verificar la diferencia en días entre la fecha de ingreso de la denuncia y la fecha de ocurrencia del siniestro (ver Figura 4.5). Se observa que la distribución de los registros para este enfoque presenta ciertas diferencias para los casos fraudulentos y los que no lo son. En el caso de siniestros fraudulentos se ve que no han sido denunciados el mismo día en que ocurrieron. Para el caso de siniestros no fraudulentos se observa que el 25 % de los registros se han denunciado el mismo día de la ocurrencia del evento.

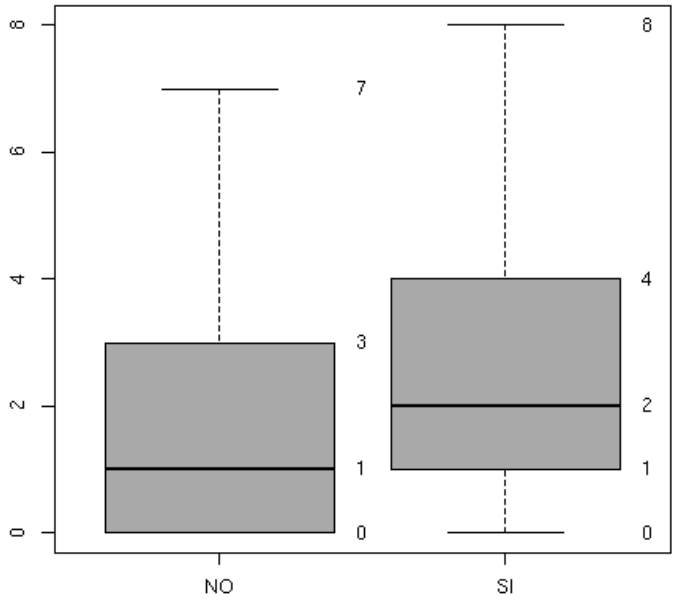


Figura 4.5: Diferencia Fecha de ingreso de denuncia y Fecha de Ocurrencia del Siniestro

4.2. Selección de datos

Luego de explorar el conjunto de datos provistos por la empresa y haber realizado un análisis detallado de los registros, se procedió a descartar los atributos que se detallan a continuación:

1. *nro_siniestro*: este atributo almacena un identificador numérico secuencial de uso interno de la empresa que no representa ninguna característica que permita identificar patrones en relación a si un registro de este tipo es fraudulento o no, por lo que se decidió descartarlo.

2. *interv_municipal*: este atributo registra la intervención municipal en el lugar de ocurrencia del siniestro. Se decidió descartarlo ya que desde el Sector de Análisis de Fraude de la empresa indicaron que existen inconsistencias en el registro de los datos.

3. *interv_policial*: este atributo registra la intervención policial en el lugar de ocurrencia del siniestro. Se decidió descartarlo ya que desde el Sector de Análisis de Fraude

de la empresa indicaron que existen inconsistencias en el registro de los datos.

4. *horario_siniestro*: este atributo registra el momento del día en el que ocurrió el siniestro mediante una descripción (DIURNO o NOCTURNO). Se decidió descartar el atributo ya que se cuenta con el atributo *hora_dia* que permite obtener información precisa sobre el horario de ocurrencia del siniestro.

Una vez realizado el proceso de limpieza se generó la vista minable que quedó conformada por los siguientes atributos:

- $franquicia$
- $sexo$
- $tipo\_accidente$
- $hora\_dia$
- $anio\_vehiculo$
- $limite\_casco$
- $limite\_RC$
- es_fraude
- $tipo\_conductor$
- $antiguedad\_socio$
- $cant\_rec\_lesiones$
- $cant\_rec\_danios$
- $cant\_rec\_casco$
- $edad\_socio$
- $monto\_reclamado$
- $dif\_emision$
- $dif\_inicio\_vigencia$
- $en\_localidad\_guarda$

- *dos_ruedas_vigencia*
- *tenencia_poliza*
- *dif_denuncia*

4.3. Limpieza y Preparación de los datos

La actividad de limpieza y reparación de los datos es fundamental para el uso de los algoritmos ya que el faltante de algunos de ellos puede alterar el funcionamiento del mismo.

En este trabajo, al ser una aplicación sobre un caso real y contar con datos reales provistos por la empresa, estas tareas demandaron gran parte del esfuerzo total de este trabajo, debido a que la base de datos transaccional y de clientes de la empresa data muchos años atrás y existen datos que por ejemplo no eran obligatorios y luego si, o que por ciertas definiciones del negocio no eran solicitados, o que simplemente están mal informados, debido a las restricciones de los sistemas de ingreso que también fueron mejorando con el tiempo.

Por lo expuesto anteriormente, además de la selección de atributos realizada, donde se descartaron algunos de ellos, también fue necesario realizar una preparación de los mismos, ya que no todos se encontraban completos o con el mismo formato de datos. Razón por la cual, se procedió a completar y corregir los mismos, para ello fue necesario consultar otras bases de datos de la misma empresa, lo que incluyó las siguientes tareas:

- Obtener base de datos de otras fuentes de la empresa (Sistema transaccional).
- Cruzar la información de la base provista con los análisis de fraude, con otras fuentes.
- Completar los datos faltantes.
- Corregir/ Actualizar datos erróneos.
- Unificar parámetros en una sola base de datos.

En esta etapa de limpieza, actualización y corrección de datos, en gran medida los datos que se corrigieron fueron los datos propios del socio, como: DNI, sexo y fecha de

nacimiento. También en muchos casos, la corrección estuvo relacionada al formato de los mismos.

Luego de estas tareas de limpieza, de adecuación de formato de los datos y de haber completado los datos faltantes, se logró generar la vista minable, que es el insumo fundamental para la aplicación de algoritmos de minería de datos que permitan descubrir conocimiento, en este caso, la identificación de denuncias de siniestros que cumplen con ciertos criterios que permiten clasificarlos como casos sospechosos de fraude.

Capítulo 5

Tareas de minería de datos aplicadas

5.1. Introducción

El empleo de técnicas de minería de datos para la extracción de conocimiento útil de grandes bases de datos ha sido ampliamente aceptado en el ámbito de seguros y se ha consolidado como una alternativa que permite obtener información de calidad para respaldar la toma de decisiones. Así lo demuestra la literatura científica citada en capítulos anteriores, como las revisiones del estado del arte propuestas en [SKK18] que identifica las técnicas de aprendizaje automático más utilizadas, sus fortalezas y debilidades; o en [GT16] que presenta un estudio realizado para un amplio período de tiempo sobre el uso de minería de datos en detección de fraudes en la industria del seguro y muestra que técnicas como modelos logísticos, bayesianos y árboles de decisión han sido ampliamente empleados y han demostrado grandes aportes. También en [GF18] los autores presentan los beneficios del uso del algoritmo K-means en la detección de fraudes.

Entre los métodos de aprendizaje supervisado los árboles de decisión se destacan por su facilidad de uso y comprensión. Un árbol de decisión organiza condiciones en una forma jerárquica que permite identificar la decisión a tomar siguiendo las condiciones que se cumplen partiendo de la raíz del árbol y llegando hasta alguna de sus hojas. Una de sus principales ventajas es que, partiendo de una condición, las opciones posibles son excluyentes, lo que permite realizar un análisis de la situación siguiendo adecuadamente su desarrollo en el árbol y arribar a una única decisión.

Los problemas de clasificación presentan la característica de que las clases objetivo son disjuntas. Para el caso objeto de este estudio, en el que se debe clasificar un registro de un siniestro automotor de acuerdo a si pertenece a la clase de casos fraudulentos o no, la aplicación de árboles de decisión se considera adecuada, ya que a partir del recorrido del árbol será posible determinar a cuál de las clases pertenece cada observación.

Esta característica de los árboles de decisión resulta de interés al momento de tener que analizar los resultados con usuarios del negocio, que no necesariamente poseen conocimiento técnico. Ante la clasificación de un registro como fraudulento, por parte del modelo, se genera una alerta que indica al usuario que ese caso debe ser analizado con mayor detalle para verificar fehacientemente si se trata de un verdadero hecho de fraude.

En este capítulo se presenta la metodología empleada en la preparación de los conjuntos de datos de entrenamiento y de prueba y los algoritmos de minería de datos empleados.

5.2. Selección de algoritmos de minería de datos

En este trabajo se realizaron pruebas con dos algoritmos Classification And Regression Trees (CART) y Generalized Boosted Regression Modeling (GBM). La elección de estos algoritmos se debe por un lado a que ambos presentan buena performance para tareas de clasificación, y también a que los árboles de decisión presentan gran poder explicativo y sencillez en su interpretación para usuarios finales, sin conocimientos técnicos.

En las siguientes secciones se presentan las características principales de cada algoritmo empleado en esta tesis, la configuración de los conjuntos de entrenamiento y prueba, y las estrategias de balanceo de clases. Al final del capítulo se detallan los pasos requeridos para la construcción del modelo, la generación del plan de pruebas y la evaluación de la calidad de los modelos.

5.2.1. CART

Este algoritmo se origina en el ámbito de la estadística, fue desarrollado por matemáticos de la Universidad de Berkeley y Stanford (Breiman, Friedman, Olshen y Stone) a

mediados de los años 80 [BFSO84].

El uso de este tipo de algoritmos representa una alternativa al análisis tradicional de clasificación/discriminación o a la predicción tradicional (regresión).

Los árboles de clasificación y regresión (CART) son una técnica de aprendizaje de árboles de decisión no paramétrica que produce árboles de clasificación o de regresión, dependiendo de si la variable dependiente es categórica o numérica, respectivamente.

Los árboles de decisión están formados por una colección de reglas basadas en variables del conjunto de datos de modelado:

- Las reglas basadas en los valores de las variables se seleccionan para obtener la mejor división para diferenciar las observaciones basadas en la variable dependiente.
- Una vez que se selecciona una regla y se divide un nodo en dos, se aplica el mismo proceso a cada nodo "secundario"(es decir, es un procedimiento recursivo).
- Cada rama del árbol finaliza en un nodo terminal.
- Cada observación cae en uno y exactamente un nodo terminal, y cada nodo terminal está definido de forma única por un conjunto de reglas.

Entre las ventajas de los árboles CART podemos destacar su robustez a outliers, la invarianza en la estructura de sus árboles de clasificación o de regresión a transformaciones monótonas de las variables independientes, y sobre todo, su interpretabilidad.

5.2.2. GBM

También se realizaron pruebas con el algoritmo Generalized Boosted Regression Modeling (GBM) cuya implementación en R [Rgb] se basa en los enfoques propuestos en [Fri01], [Fri02].

Este modelo solo se puede ejecutar empleando el paquete GBM provisto por R.

GBM es un tipo de algoritmo que produce diferentes modelos individuales (árboles de decisión) cuyos resultados se van agregando de modo que el resultado final (clasificador de ensamble) está formado por un modelo que es una combinación de los anteriores (clasificadores débiles), pero con una capacidad de predicción muy superior a la de los

modelos individuales en los que se basa. GBM, en sus sucesivas iteraciones, aprende y minimiza los errores de los modelos anteriores, y ajusta los árboles de decisión a los residuos o errores con el fin de ir actualizando y minimizando los residuos. Precisamente, una de las características más significativas de este algoritmo basado en 'Boosting' es que aprende de los errores de los múltiples modelos a medida que los va generando. Entre los diferentes métodos que hacen uso del 'Boosting', los más comunes son 'AdaBoost', 'Gradient Boosting' y 'Stochastic Gradient Boosting'.

En este trabajo se uso el algoritmo 'stochastic Gradient Boosting'. Una de las particularidades de éste es la construcción secuencial de cada nuevo árbol de decisión en función de los árboles que se hayan construido previamente. Además, la estocasticidad mejora los resultados predictivos y reduce la varianza final del modelo, pues implica utilizar solo un subconjunto aleatorio de las observaciones para ajustar cada uno de los árboles individuales de los modelos que integran el modelo final.

Esto significa, que en cada iteración se selecciona una muestra aleatoria sin reemplazo de la totalidad de los datos disponibles.

La fortaleza del algoritmo GBM reside en el hecho de que mediante la construcción de numerosos árboles de decisión se hace frente al principal problema de los modelos basados en un único árbol, que es su escasa capacidad predictiva. El modelo GBM resultante es en realidad una combinación lineal de muchos árboles de decisión (normalmente cientos o miles) que puede entenderse como un modelo de regresión donde cada término es un árbol de decisión individual. Asimismo, GBM permite la existencia de valores extremos, correlaciones altas entre las variables, relaciones no lineales, la presencia de valores perdidos y admite el uso de variables categóricas como independientes.

La utilización de GBM supone necesariamente la especificación de tres parámetros importantes: el ratio de aprendizaje o parámetro de contracción, la profundidad de los árboles de decisión (número de cortes o divisiones de los árboles desde un nodo terminal hasta el nodo raíz) y el número de árboles de decisión. Es habitual obtener el valor óptimo de estos parámetros mediante el uso de técnicas de validación cruzada, muy utilizadas entre los usuarios del aprendizaje automático.

5.3. Generación de los conjuntos de entrenamiento y prueba

Para la generación de los conjuntos de entrenamiento y de prueba se particionó la vista minable utilizando un criterio 70-30, quedando el 70 % de los registros para entrenamiento y el 30 % para pruebas respecto de la variable objetivo, esto implica respetar el mismo porcentaje de casos fraudulentos y no fraudulentos en las dos particiones.

Para esta actividad se utilizó la función createDataPartition del paquete Caret. Caret proporciona un conjunto de funciones que intentan agilizar el proceso de creación de modelos predictivos. Este paquete tiene herramientas que nos permiten realizar: división de datos, pre-procesamiento, selección de características, ajuste del modelo mediante remuestreo, estimación de importancia variable entre otras funcionalidades.

En particular la función createDataPartition se puede utilizar para crear divisiones equilibradas de los datos. Si el argumento de esta función es un factor, el muestreo aleatorio se produce dentro de cada clase y debe preservar la distribución de clase general de los datos.

A continuación se presenta el código en R que se ha ejecutado para generar dicha partición:

```
set.seed(seed)
partition <- createDataPartition(y = fraud_data$es_fraude, p = 0.7,
                                 list = F)
train_data <- fraud_data[partition,]
test_data <- fraud_data[-partition,]
```

5.4. Balanceo de datos

Generalmente las bases de datos empleadas para detección de fraude poseen una distribución de la clase objetivo desbalanceada, generalmente solo alrededor del 1 o 2 % (o incluso menos) de las transacciones son fraudulentas. Esto genera problemas para ciertas técnicas analíticas ya que se les presenta un alto porcentaje de registros no fraudulentos y,

por lo tanto, presentarán una tendencia a clasificar la mayor parte de las observaciones como no fraudulentas. Esto podría generar modelos con una alta precisión (por ejemplo 98 o 99 %) pero que no detecten ninguna observación de la clase fraudulenta.

El dataset empleado en este trabajo presenta esta característica de datos desbalanceados, como se puede apreciar en la Figura 4.1.

Por ejemplo, se puede presentar un problema de clasificación de 2 clases (binario) con 100 instancias (filas). Un total de 80 instancias están etiquetadas con Clase-1 y las 20 instancias restantes están etiquetadas con Clase-2.

Este es un conjunto de datos desbalanceado y la proporción de instancias de Clase-1 a Clase-2 es 80:20.

también se puede presentar un problema de desbalanceo en problemas de clasificación de varias clases. La mayoría de las técnicas se pueden usar en cualquiera de las dos situaciones.

Los conjuntos de datos de clasificación en un gran porcentaje no tienen exactamente el mismo número de instancias en cada clase, pero una pequeña diferencia en ocasiones es irrelevante.

Hay problemas en los que un desbalanceo de clase no sólo es común, sino que es esperable. El caso de estudio en este trabajo de tesis es un ejemplo muy común que presenta clases desbalanceadas. La gran mayoría de las transacciones estarán en la clase *No Fraude* y una minoría estará en la clase *Fraude*.

Otro ejemplo son los conjuntos de datos de abandono de clientes, donde la gran mayoría de los clientes se quedan con el servicio y una minoría cancela su suscripción.

Existen algunas formas de abordaje sobre la problemática planteada anteriormente., los cuales se detallan a continuación:

1. Cambiar las métricas de rendimiento: al trabajar con conjunto de datos desbalanceados, la precisión no es la mejor opción, por eso es necesario conocer y evaluar diferentes métricas que nos permitan evaluar el rendimiento del modelo. Existen diversas medidas de rendimiento que nos pueden dar mas precisión sobre el modelo de clasificación que estamos desarrollando:

- Confusión Mátrix: un desglose de las predicciones en una Tabla que muestra las predicciones correctas (la diagonal) y los tipos de predicciones incorrectas realizadas (qué clases se asignaron las predicciones incorrectas).
- Precisión: Una medida de la exactitud de un clasificador.
- Recall: Una medida de la integridad de un clasificador
- F-score: Un promedio ponderado de precisión y recuperación.

Además de las nombradas anteriormente existen dos formas mas de medir el rendimiendo que son las mas recomendadas:

- Kappa (o kappa de Cohen): precisión de clasificación normalizada por el desequilibrio de las clases en los datos.
- Curvas ROC: al igual que la precisión y la recuperación, la precisión se divide en sensibilidad y especificidad y los modelos se pueden elegir en función de los umbrales de equilibrio de estos valores.

2. Modificar el conjunto de datos: Según [BEP] existe la posibilidad y es conocido como una buena técnica poder modificar el conjunto de datos que se usa para construir un modelo predictivo y así poder tener datos más equilibrados.

   Existen dos métodos principales que puede utilizar para disminuir la situación de desbalanceo:

   - Agregar copias de instancias de la clase sub-representada llamada over-sampling [CBHK02].
   - Eliminar instancias de la clase sobre-representada, llamada under-sampling [LLL98].

   Estos enfoques suelen ser muy fáciles de implementar y rápidos de ejecutar.

3. Probar diferentes algoritmos: una recomendación para solucionar este tipo de inconvenientes suele ser realizar el modelado con diferentes algoritmos, por mas que se conozca la mayor eficiencia de un modelo siempre es recomendable no usar el favorito sino probar con diversos ya que para cada modelo puede que alguno se ajuste mas que otro. Al menos, se debe verificar con diferentes tipos de algoritmos

ante un problema determinado.

Dicho esto, los árboles de decisión a menudo funcionan bien en conjuntos de datos desequilibrados. Las reglas de división que observan la variable de clase utilizada en la creación de los árboles pueden obligar a que se aborden ambas clases.

En relación a este punto y a este trabajo se probó con los algortimos de CART y GBM detallados en apartados anteriores.

5.5. Construcción del modelo

En este apartado se presentará como se realizo la construcción del modelo, el detalle de parámetros utilizados, el código de las ejecuciones del software R, etc.

Muchos modelos contienen parámetros que no pueden aprenderse a partir de los datos de entrenamiento de minería de datos y que, por lo tanto, deben ser establecidos por el analista, estos se conocen como hiperparámetros. Los resultados de un modelo pueden depender en gran medida del valor que tomen sus hiperparámetros.

Sin embargo es recomendable utilizar para cada algoritmo un conjunto de datos para su entrenamiento y atributos que mejor se ajusten para obtener el mejor rendimiento en la clasificación. Para el presente trabajo, se utilizaron herramientas que automáticamente nos den estos dos parámetros, éstas son cross_validation y grid_search.

5.5.1. Cross Validation

Uno de los pasos importantes a tener en cuenta cuando se quiere entrenar un algoritmo de aprendizaje es estimar cuál es su comportamiento, es decir, con qué precisión clasificará el modelo creado aquellos datos nuevos que no han sido vistos previamente.

Un conocido error metodológico para estos casos es el de sobreentrenar el algoritmo con datos para los que ya conocemos el resultado deseado, esto es comúnmente conocido como overfitting o, en castellano, “sobreajuste”. Por otro lado es posible que el modelo sea demasiado simple y se obtengan valores adecuados directamente en su entrenamiento. De

esta forma se estará perdiendo la tendencia de los datos, en este caso estaríamos sufriendo "underfitting" o también "subajuste".

Es necesario buscar la mejor forma de evaluar el modelo cuidadosamente para obtener el mejor rendimiento del mismo. Para ello existen técnicas de validación cruzada (cross-validation) que indican cómo de preciso se comportará el algoritmo para datos nuevos no observados [RTL09]. Esta técnica se emplea para poder evaluar los resultados de un análisis para garantizar que son independientes de la partición entre los datos que usamos del "conjunto de entrenamiento" y los datos del "conjunto de prueba".

Esta técnica consiste en dividir el conjunto de datos disponible para entrenar en dos partes, una llamada "conjunto de entrenamiento" y otra "conjunto de prueba".

El conjunto de entrenamiento es usado para entrenar el algoritmo, y el conjunto de pruebas es usado para testear el rendimiento del mismo. Como se conoce la salida que debe proporcionar, podemos obtener del algoritmo entrenado qué salidas deben tener los datos del conjunto de prueba y de esta forma saber si tiene una idea acertada de lo que debe predecir.
Para este trabajo se utilizo la estrategia de dividir el conjunto de datos en 70 % para los datos de entrenamiento y un 30 % para los datos de pruebas

Otro punto a tener en cuenta es saber cómo están distribuídos las clases en ambos conjuntos, el de entrenamiento y el de prueba. Lo más adecuado es que exista una distribución equitativa de todas las clases en ambos conjuntos. Imaginemos que los datos están ordenados y que el 70 % que hemos entrenado no tiene todas las clases, no estaríamos entrenando todas las posibilidades. Por lo tanto, el método de cross validation entra en juego, siempre y cuando podamos encontrar los datos balanceados. En caso que no sea así la técnica de validación cruzada no servirá.

El método k-fold es similar al mencionado anteriormente, consta de dividir el conjunto en 70/30 pero aplicada a más subconjuntos. Lo que se hace con estos métodos es dividir el conjunto en k subconjuntos y entrenar k-1 de esos subconjuntos para comprobar luego con el último conjunto que no se ha entrenado.

5.5.2. Grid Search

En la búsqueda de parámetros, se dice que el aprendizaje automático de un algoritmo tiene dos tipos de parámetros: uno de los tipos es el conjunto de parámetros de modelo, estos son datos que ingresan para su entrenamiento y sobre los que el algoritmo se ajusta. Otro tipo de parámetro son los valores que podemos ajustar al mismo algoritmo para realizar los cálculos necesarios en su aprendizaje, estos son conocidos como hiper-parámetros, aquellos que no se aprenden dentro del mismo algoritmo en su entrenamiento.

Los hiper-parámetros pueden aceptar valores en distintos rangos y dependiendo de cada valor se obtiene un rendimiento mejor o peor. Como es evidente, se quiere obtener el mejor rendimiento de cada algoritmo, pero hacer una evaluación del comportamiento de cada uno con cada uno de los hiper-parámetros puede ser tedioso y muy costoso en tiempo si lo hacemos manualmente.

Por ello se utiliza una técnica llamada grid search, que comprueba exhaustivamente todas las combinaciones en el espacio de parámetros y sus valores posibles.

La búsqueda de los parámetros consiste en:

- La elección del algoritmo al que se quiere estimar los parámetros.
- Un espacio de parámetros.
- Un método para la búsqueda de los parámetros.
- Un esquema de cross-validation.
- Una función de puntuación.

Cómo es esperable para este tipo de método, los requerimientos de capacidad de computos crecen exponencialmente a medida que se incrementa el número de los candidatos. Por lo que resulta necesario tener en cuenta el tiempo necesario para la búsqueda del mejor candidato y que dependerá del espacio a probar y del conjunto de datos que se trate.

Esta técnica se encarga de la búsqueda exhaustiva del mejor candidato a través de una matriz de valores posibles especificados en su parámetro *paramgrid*, que es un diccionario o lista de diccionarios con los distintos parámetros posibles y los valores que

queremos que testee.

Por último, para evaluar el comportamiento del algoritmo en el conjunto de posibles parámetros elegidos, *GridSearch* usa como puntuación la misma que el algoritmo, ésta se ingresa en el parámetro *estimator*, por defecto es Accuracy Score: métrica que indica el porcentaje de ejemplos del conjunto de entrenamiento que el algoritmo ha clasificado correctamente

5.5.3. Ejecución Algortimo CART

En este apartado se presenta la ejecución del algoritmo CART, indicando cada uno de los pasos, valores y atributos aplicados a este algoritmo.

Se comienza creando diferentes conFiguraciones de parámetros para el algoritmo CART.

```
grid.cart <- expand.grid(maxdepth = seq(4, 30, 1))
```

Se entrena el modelo ajustando la máxima profundidad del árbol

```
set.seed(seed)
unwantedoutput <- capture.output(fit.cart <- train(es_fraude~.,
                     data = train_data,
                     method = "rpart2",
                     trControl = control,
                     tuneGrid = grid.cart,
                     metric=metric,
                      na.action = na.pass))
```

En las siguientes subsecciones se muestra mediante gráficas los resultados de las ejecuciones del Algoritmo CART, con los modelos balanceados y sin balancear. Estas gráficas se presentan mediante curva ROC (Receiver Operating Characteristic) que es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación.

Conjuntos de datos sin balancear y balanceados mediante Downsampling

En las siguientes Figuras se muestran las métricas obtenidas en el entrenamiento empleando el algoritmo CART utilizando el conjunto original y un conjunto balanceado mediante downsampling, en ellos podemos observar que la máxima profundidad que tenemos sin balancear es 12 como se puede observa en la figura 5.1, sin embargo en 5.2 aplicando downsampling la máxima profundidad del árbol es 4.

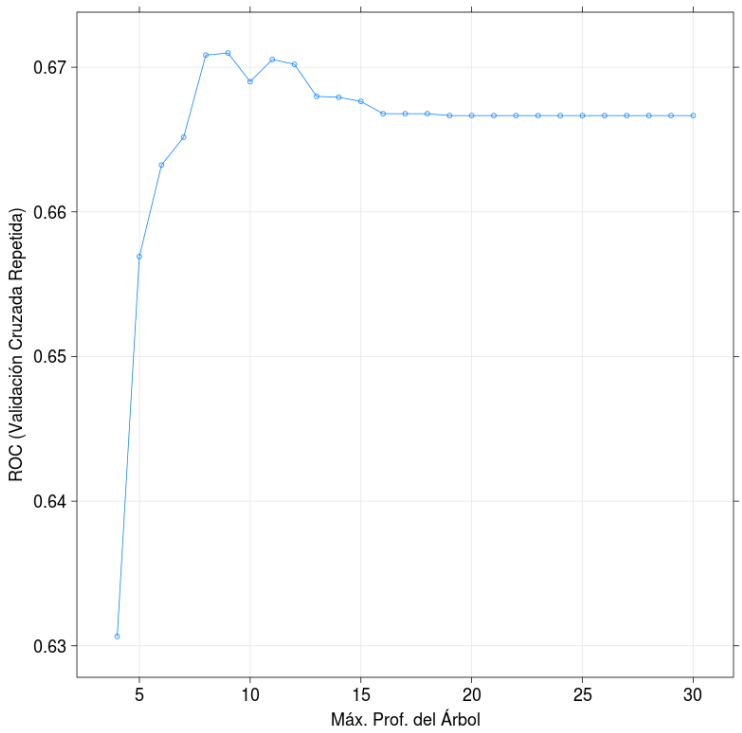


Figura 5.1: Resultado entrenamiento CART sin balancear

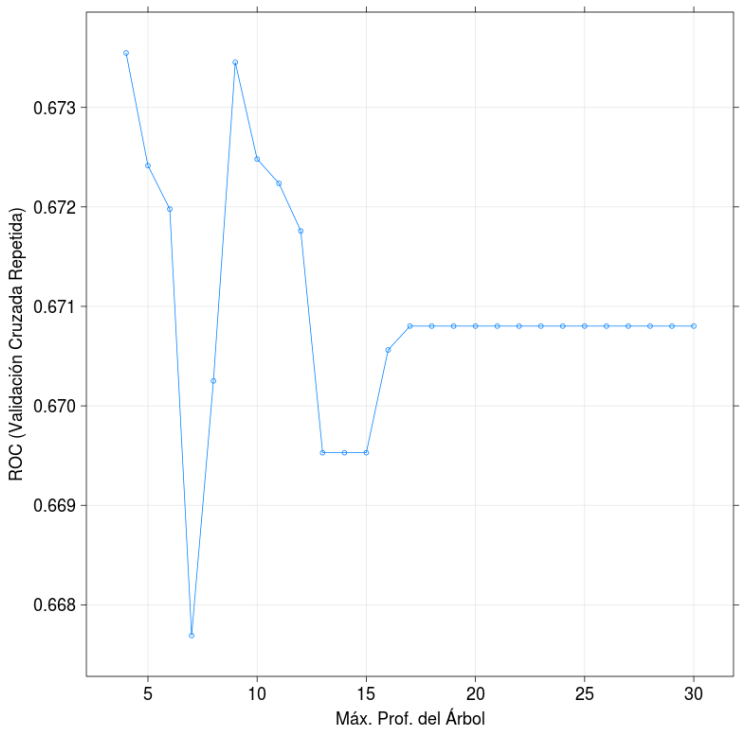


Figura 5.2: Resultados ejecución CART downsampling

Conjunto de datos balanceados mediante Upsampling y Weighted

En las siguientes Figuran se muestran las métricas obtenidas en el entrenamiento empleando el algoritmo CART utilizando un conjunto de datos balanceado por upsampling 5.3 y weighted **??** en ellos podemos observar que la máxima profundidad que tenemos del árbol en upsampling es de 7 , sin embargo aplicando weighted es de la máxima profundidad del árbol es 5.

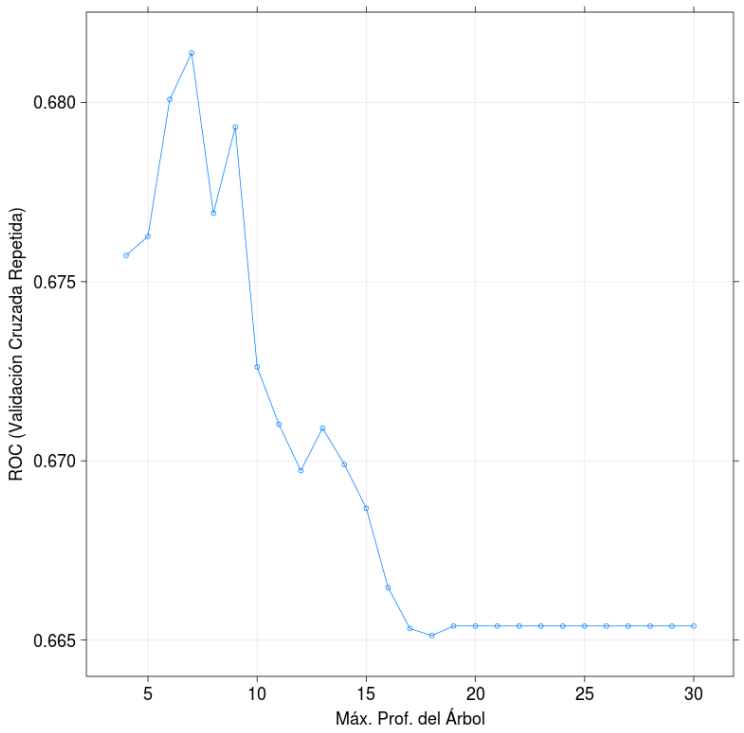


Figura 5.3: Resultados ejecución CART upsampling

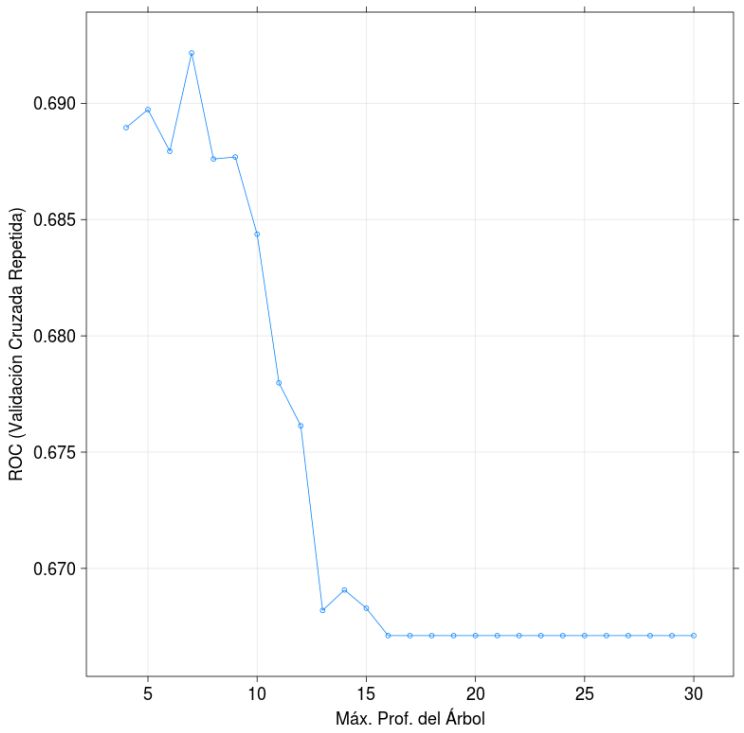


Figura 5.4: Resultados ejecución CART weighted

Adelante en el Capítulo se presenta un análisis de resultados y comparativas de las ejecuciones de ambos algoritmos, de todas formas, se observa en los valores de las gráficas

expuestas en esta sesión, que para este algoritmo si bien las técnicas de balancero mejoran el modelo, entre ellas no ofrecen diferencias significativas.

5.5.4. Ejecución GBM

En este apartado se presenta la ejecución del algoritmo GBM, indicando cada uno de los pasos, valores y atributos aplicados a este algoritmo y sus correspondientes gráficas, que luego nos permitirán su evaluación y comparación. De la misma forma que en el algoritmo anterior, las gráficas se mostraran mediante la curva ROC (Receiver Operating Characteristic) que como mencionamos, es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación.

A continuación se presenta el código en lenguaje R empleado para la ejecución del algoritmo y sus configuraciones.

En el siguiente código se indica la configuración para la ejecución de GBM indicando que la cantidad de árboles variara entre 1000 y 1500.

```
grid.gbm <- expand.grid(n.trees = c(1000,1500), interaction.depth=c(1:4),
    shrinkage=c(0.001,0.01),
    n.minobsinnode=c
    (20))



library(doParallel)
```

```
registerDoParallel(detectCores()-1)
```

Se entrena el modelo aplicando las diferentes configuraciones

```
set.seed(seed)
unwantedoutput <- capture.output(fit.gbm <- train(es_fraude~.,
             data = train_data,
            method = "gbm",
            trControl = control,
```

```
            tuneGrid = grid.gbm,
        metric=metric,
        na.action = na.pass))
```

En las siguientes subsecciones se muestra mediante gráficas los resultados de las ejecuciones del Algoritmo GBM, con los modelos balanceados y sin balancear.

Conjunto de datos sin balancear

En este caso si analizamos el algoritmo trabajando con datos sin balancear, como podemos verlo en la Gráfica 5.5, presenta 1500 albores, con una profundidad de 2, una tasa de aprendizaje 0.01 y cuenta con un mínimo de observaciones de 25 por cada nodo.

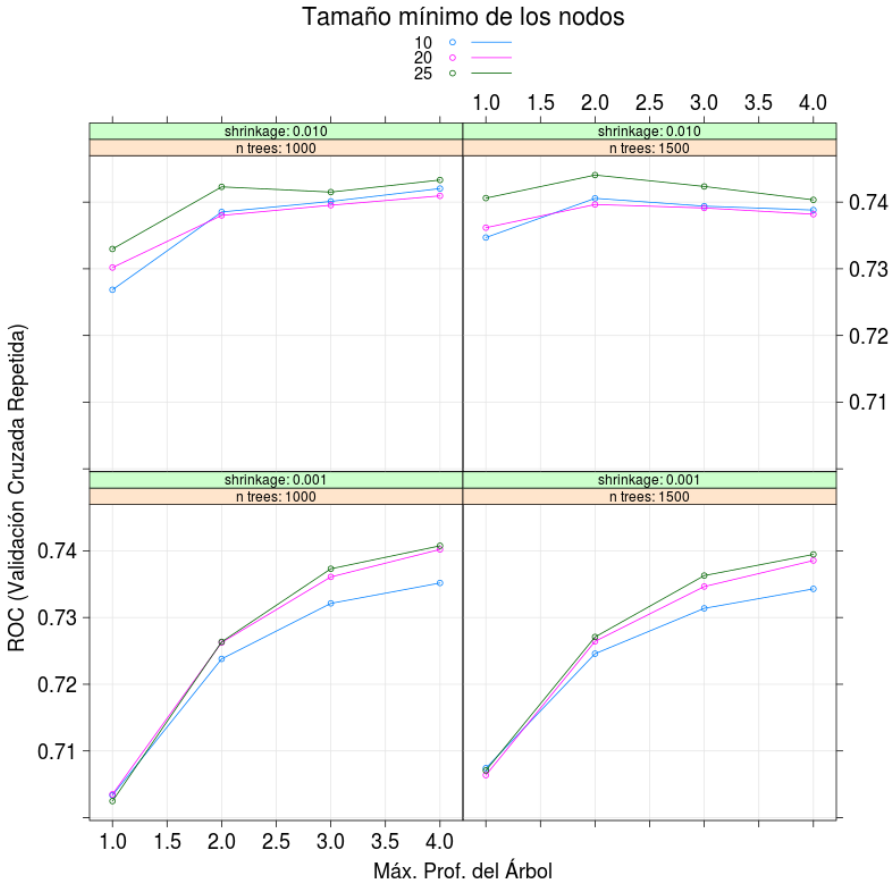


Figura 5.5: Resultado entrenamiento GBM sin balancear

Downsampling

Aplicando balanceo Downsampling, como se presenta en la Gráfica 5.6, el algoritmo presenta 1000 arboles, con una profundidad de 4, una tasa de aprendizaje 0.01 y cuenta con un mínimo de observaciones de 20 por cada nodo.

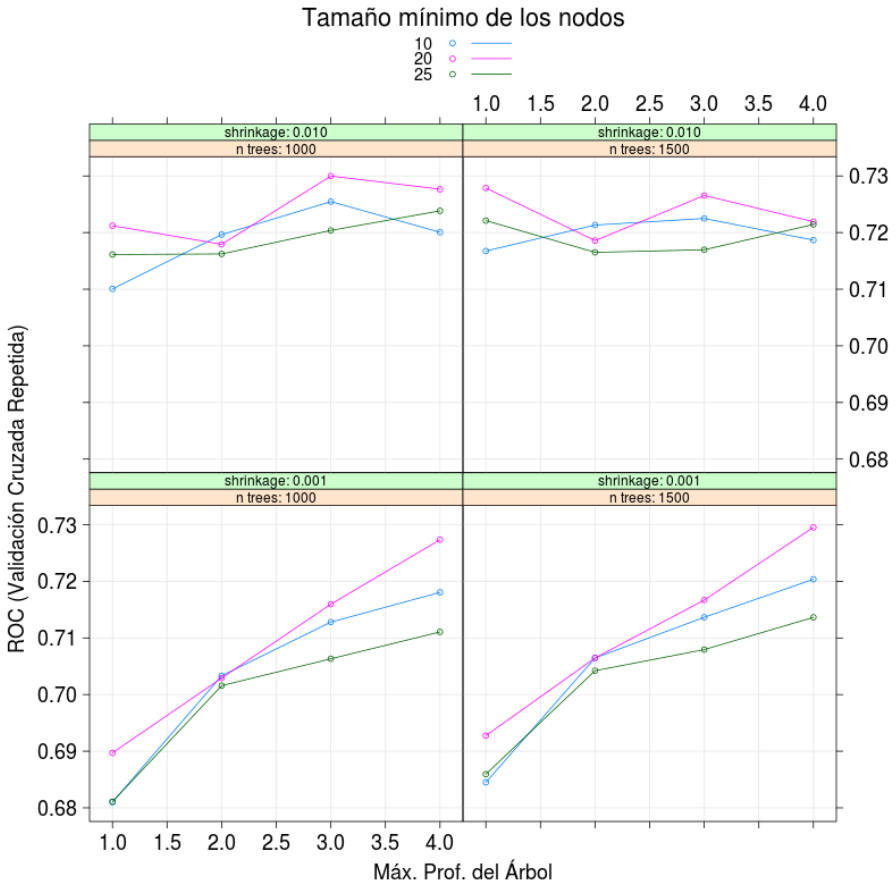


Figura 5.6: Resultados ejecución GBM downsampling

Upsampling

Aplicando balanceo Upsampling, como se presenta en la Gráfica 5.7, el algoritmo presenta 1500 arboles, con una profundidad de 3, una tasa de aprendizaje 0.01 y cuenta con un mínimo de observaciones de 20 por cada nodo al igual que el balanceo anterior.

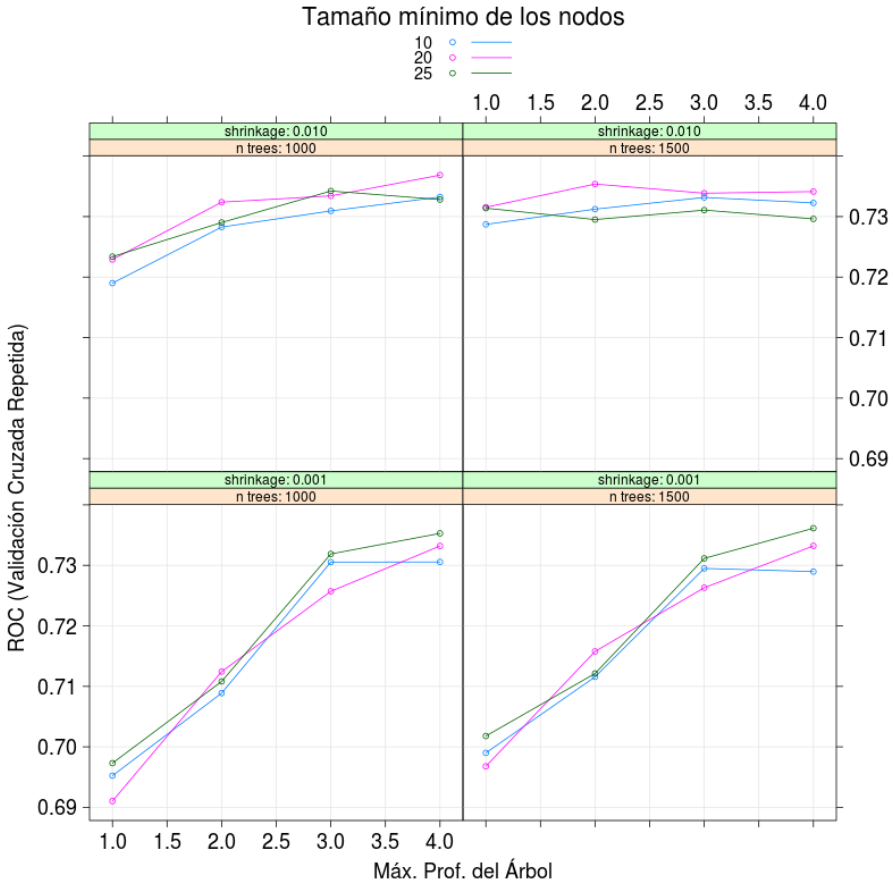


Figura 5.7: Resultados ejecución GBM upsampling

Weighted

Aplicando balanceo Weighted,como se presenta en la Gráfica 5.8, el algoritmo presenta 1000 arboles,en este caso coincidente con Downsampling, cuenta con una profundidad de 4, una tasa de aprendizaje 0.01 y cuenta con un mínimo de observaciones de 25 por cada nodo.

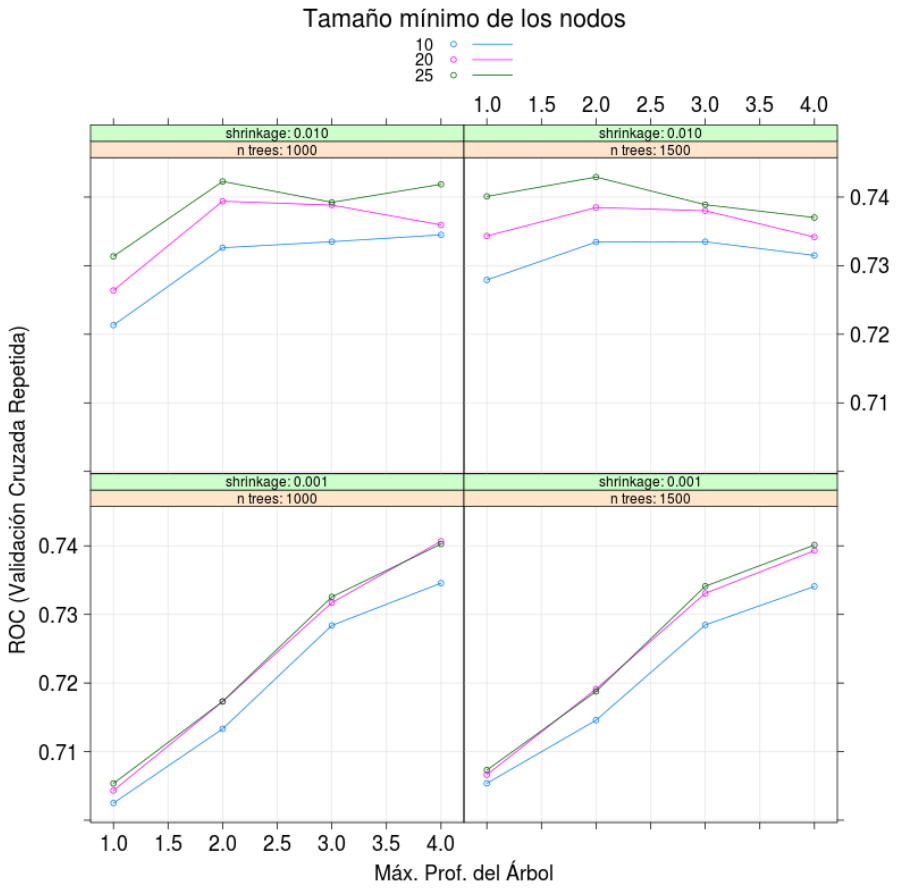


Figura 5.8: Resultados ejecución GBM weighted

Para el algoritmo GBM, como se observa en las gráficas, las técnicas de balancero mejoran el rendimiento del modelo, pero este resultado es independiente de la técnica empleada como lo demostraremos sobre el final de este Capítulo.

5.6. Generación del plan de pruebas

El plan de pruebas se configuró de la siguiente manera:

1. Se realizaron pruebas con los algoritmos CART y GBM.
2. Cada algoritmo se ejecutó sobre el conjunto de datos sin balancear y luego con los datos balanceados mediante 3 técnicas: Class Weights, Undersampling y Oversampling.

3. Para cada ejecución se obtuvieron los valores de las métricas: precisión, sensibilidad y especificidad.

A continuación se indica la interpretación de las métricas para este caso de estudio:

- La sensibilidad indica la capacidad del estimador para identificar como casos positivos a los casos fraudulentos. Esta métrica caracteriza la capacidad de la prueba para detectar la capacidad del modelo en siniestros con indicios de fraude.

$$Sensibilidad = \frac{VP}{VP + FN}$$

  Donde VP es verdaderos positivos y FN falsos negativos.

- La especificidad indica la capacidad del estimador para identificar como casos negativos a los casos sin indicios de fraude; caracteriza la capacidad del modelo para detectar la ausencia ausencia de fraude en siniestros no fraudulentos.

$$Especificidad = \frac{VN}{VN + FP}$$

  Donde VN, serían los verdaderos negativos; y FP, los falsos positivos.

La sensibilidad es la fracción de verdaderos positivos y la especificidad la fracción de verdaderos negativos (FVN).

Lo que interesa para este caso de estudio, son los verdaderos positivos, razón por la cual se dará mayor peso a la métrica sensibilidad.

5.7. Ajuste del modelo

En este apartado se evaluará el modelo y su capacidad de predicción para cada uno de los algoritmos presentados y la capacidad de predicción de los mismos, expresando las variables sensibilidad y especificad de cada uno. Esta evaluación se lleva adelante a través del análisis de la curva ROC (Receiver Operating Characteristic) que es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación.

Otra interpretación de este gráfico es la representación de la razón o proporción de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o proporción de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual se decide que un caso es un positivo). ROC también puede significar Relative Operating Characteristic (Característica Operativa Relativa) porque es una comparación de dos características operativas (VPR y FPR) según cambia el umbral para la decisión.

En este apartado se presenta un gráfico por cada ejecución de cada algoritmo para poder observar y analizar mediante el área bajo la curva ROC cuál es el modelo más adecuado. Además de los gráficos individuales de cada curva, se presenta mediante la gráfica, una comparativa de todas las ejecuciones para facilitar el análisis de los resultados.

5.7.1. Evaluación del Modelo

Técnica de balanceo	CART			GBM		
	Acc	Sens	Spec	Acc	Sens	Spec
Ninguna	0.7304	0.2632	0.8940	0.7816	0.3290	0.9401
Class weights	0.5768	0.8421	0.4839	0.6689	0.6447	0.6774
Undersampling	0.5939	0.7632	0.5346	0.6416	0.7237	0.6129
Oversamplig	0.6177	0.8158	0.5484	0.6792	0.6974	0.6728

Tabla 5.1: Comparación de resultados Área bajo la Curva

A partir de la ejecución del plan de pruebas se genero la Tabla 5.1 que presenta los valores de las métricas estimadas.

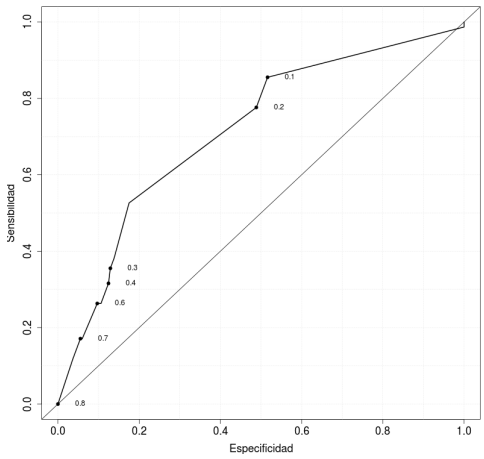
En la Tabla se observa que en el dataset sin balancear la precisión para ambos algoritmos es un valor considerablemente aceptable, ronda entre el 70 y el 80 por ciento, pero si se observa la especificidad presenta una gran diferencia respecto de la sensibilidad, por lo tanto se puede concluir que los modelos clasifican mejor los casos que no presentan indicios de fraudes y presentan un bajo rendimiento al clasificar los casos fraudulentos.

Para la ejecución de los algoritmos sobre el conjunto de datos aplicando las tres técnicas de balanceo se puede concluir que:

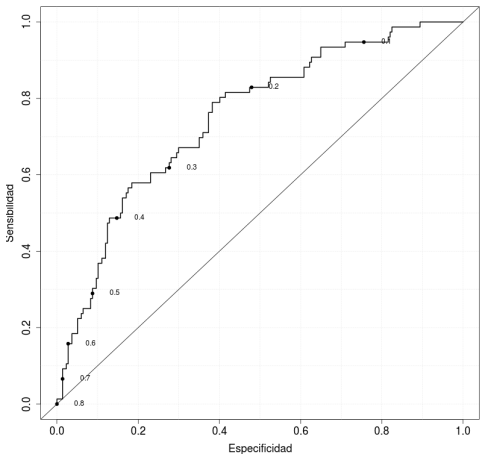
Para el algoritmo GBM, las técnicas de balancero mejoran el rendimiento del modelo,

pero este resultado es independiente de la técnica empleada. Por otro lado no existen diferencias considerables entre las métricas (precisión, sensibilidad y especificidad).

Para el algoritmo CART, también se observa que si bien las técnicas de balancero mejoran el modelo, entre ellas no ofrecen diferencias significativas. En cuando a las métricas, la ponderación de la sensibilidad es mayor que en el primer algoritmo, sin embargo la especificidad es considerablemente menor.

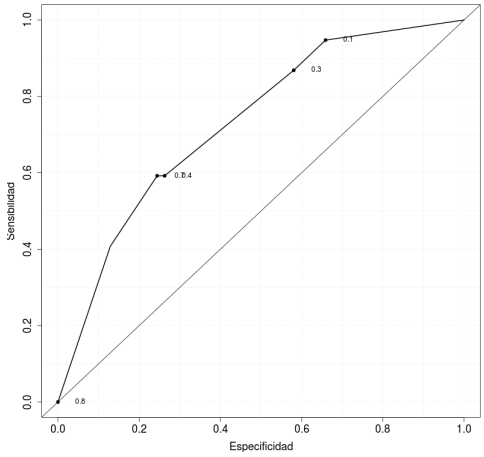


(a) Curva ROC CART sin balanceo

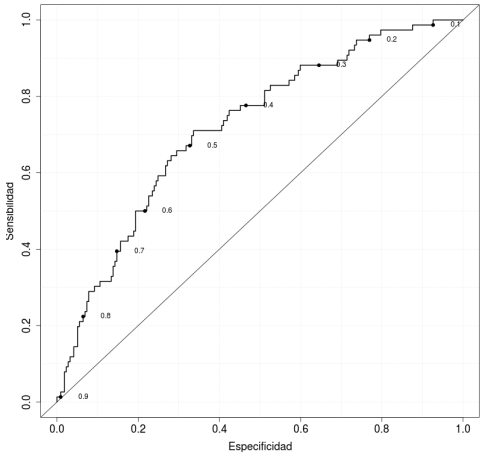


(b) Curva ROC GBM sin balanceo

Figura 5.9: Curvas ROC CART y GBM sin balanceo

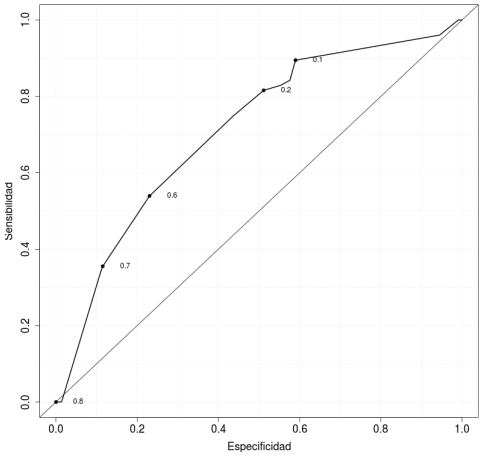


(a) Curva ROC CART down sampling

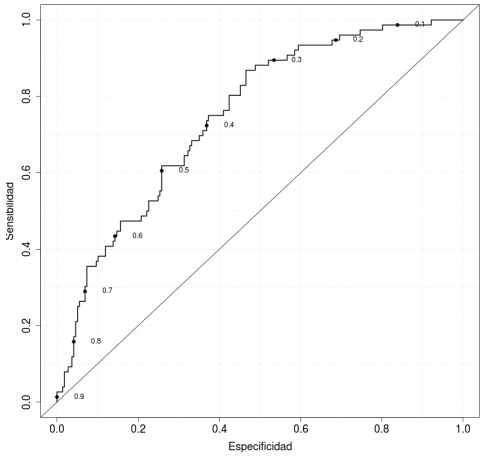


(b) Curva ROC GBM down sampling

Figura 5.10: Curvas ROC CART y GBM balanceado con down sampling

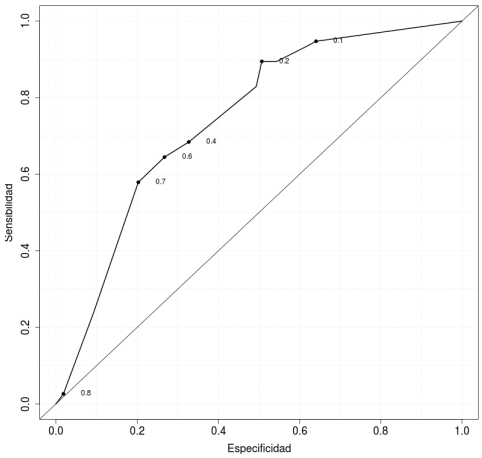


(a) Curva ROC CART up sampling

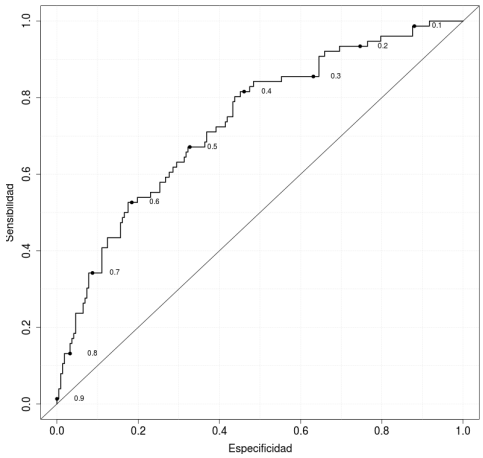


(b) Curva ROC GBM up sampling

Figura 5.11: Curvas ROC CART y GBM balanceado con up sampling



(a) Curva ROC CART weighted

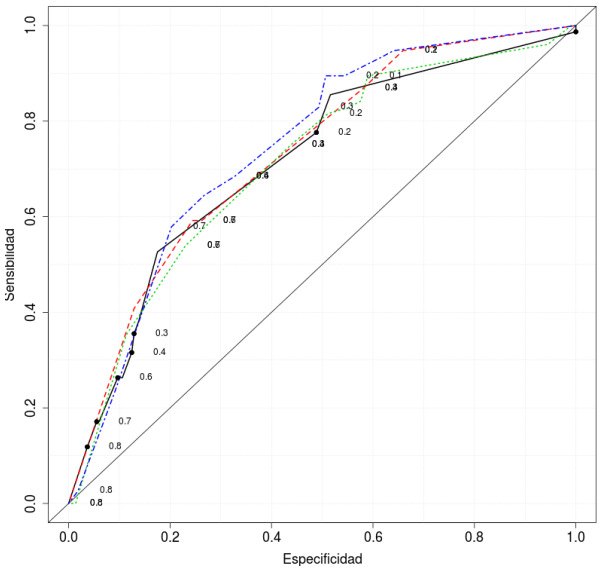


(b) Curva ROC GBM weighted

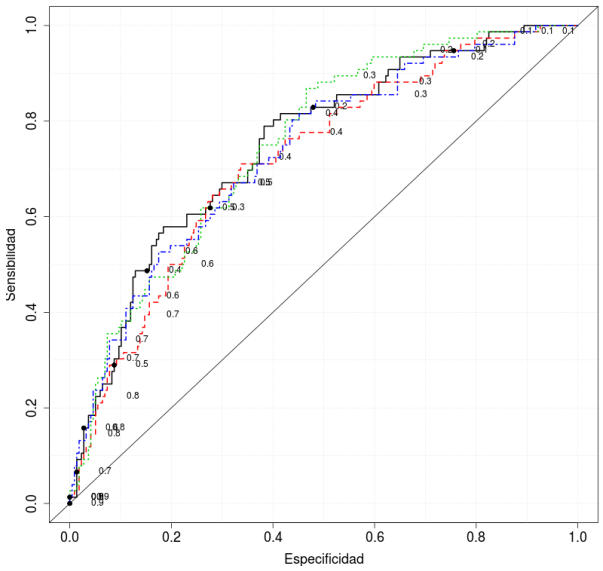
Figura 5.12: Curvas ROC CART y GBM balanceado con weighted

5.7.2. Comparativa

En este apartado se presentan mediante la Figura 5.13b dos gráficas las comparativas para ambos algoritmos con los distintos conjuntos de datos. En la Tabla 5.2 donde se puede observar la comparativa del área bajo la curva.



(a) Comparativa curvas ROC CART



(b) Comparativa curvas ROC GBM

Algoritmo/Balanceo	Sin balancear	Down samplig	Up sampling	Weighted
CART	0.7154075	0.7302025	0.7100412	0.7438455
GBM	0.7495149	0.7201674	0.7493936	0.7346592

Tabla 5.2: Tabla comparativa

Se observa mediante el análisis de la curva ROC, que para este modelo es la mejor alterativa es la que resulta de aplicar el algoritmo GBM mediante el balanceo con Upsampling.

Capítulo 6

Conclusiones y líneas de trabajo futuras

6.1. Conclusiones principales

La prevención y detección temprana de fraude dentro de una aseguradora es una tarea central, tanto para quienes están en la suscripción del riesgo, como la gestión de siniestros.

La disminución del fraude es uno de los principales retos de la industria aseguradora tanto a nivel nacional como mundial. Las pérdidas económicas derivadas del delito y los costos derivados de la adopción de los marcos de prevención y detección, convierten esta realidad en un asunto de absoluta trascendencia para dicho sector.

En este trabajo se abordó la problemática de detección de fraudes en siniestros de seguros del automotor mediante la aplicación de técnicas de minería de datos. El conjunto de datos para realizar este estudio fue provisto por la Empresa Río Uruguay Seguros Ltda.

Sobre los datos provistos por la empresa hubo que hacer tareas de limpieza que demandaron gran parte del esfuerzo total de este trabajo, debido a que la base de datos existen datos que no eran obligatorios y luego si, o que por ciertas definiciones del negocio no eran solicitados, o que simplemente están mal informados, debido a las restricciones de los sistemas de ingreso que también fueron mejorando con el tiempo.

A partir del análisis exploratorio de los datos y de la generación de la vista minable se pudo realizar un análisis exhaustivo de las variables mas relevantes en el contexto de detección de fraudes. La aplicación de técnicas de minería de datos basadas en árboles de decisión permitió determinar cuáles son aquellos atributos que ofrecen mayor ganancia de información en la detección de patrones que determinan conductas fraudulentas en denuncias de siniestros del automotor.

La elección los algoritmos CART y GBM se debe por un lado a que ambos presentan buena performance para tareas de clasificación, y también a que los árboles de decisión presentan gran poder explicativo y sencillez en su interpretación para usuarios finales, sin conocimientos técnicos.

Por otro lado, los problemas de clasificación presentan la característica de que las clases objetivo son disjuntas. Para el caso objeto de este estudio, en el que se debe clasificar un registro de un siniestro automotor de acuerdo a si pertenece a la clase de casos fraudulentos o no, la aplicación de árboles de decisión se considera adecuada, ya que a partir del recorrido del árbol será posible determinar a cuál de las clases pertenece cada observación. Debido a que el conjunto de datos empleado en este trabajo, presentaba clases desbalanceadas, es decir, la gran mayoría de las transacciones estarán en la clase *No Fraude* y una minoría estará en la clase *Fraude*. Por lo cual fue necesario aplicar técnicas de balanceos para lograr un mejor rendimiento de los algoritmos.

Luego de realizar las pruebas de los algoritmos CART y GBM según las diferentes configuraciones de parámetros, se pudo verificar que el algoritmo GBM presentó un mejor desempeño al ser ejecutado sobre el conjunto de datos balanceado. La técnica de balanceo de clases que mejora los resultados de los algoritmos es Upsampling

Ante la clasificación de un registro como fraudulento, por parte del modelo, se genera una alerta que indica al usuario que ese caso debe ser analizado con mayor detalle para verificar fehacientemente si se trata de un verdadero hecho de fraude.

6.2. Consideraciones para la implementación

El modelo generado como resultado de este trabajo será implementado en la Cooperativa Río Uruguay Seguros.

La implementación de este modelo permitirá a la organización realizar un análisis automático de la totalidad de los siniestros del automotor, ofreciendo a quienes toman decisiones una herramienta que respalda sus acciones.

Cabe destacar que hasta el momento la detección de posibles casos fraudulentos se realizaba a partir de una recomendación u observación de ciertos valores de atributos informados por algún agente de la organización. Contar con un modelo que genere alertas de manera automática cuando se superan ciertos valores de las variables que permiten detectar un fraude, no solo redundará en un beneficio económico para la empresa sino en la optimización del tiempo asignado a esta tarea por parte de su personal.

Es importante mencionar que este modelo no es definitivo, es decir, así como las modalidades de fraude van cambiando, se deben realizar ajustes al modelo una vez que se realice su implementación. De esta manera, un modelo actualizado puede seguir brindando respuestas adecuadas y servir como soporte a quienes toman decisiones.

6.3. Trabajo futuro

Este trabajo ha demostrado que la aplicación de técnicas de minería de datos en el ámbito de detección de fraudes resulta adecuada y permite contar con nuevas herramientas que soporten la detección automática de situaciones que se corresponden con un patrón asociado a conductas fraudulentas.

En base a la experiencia adquirida en este trabajo se considera que es factible y resultaría de suma utilidad contar con modelos que permitan dar soporte a la toma de decisiones en el ámbito del análisis de fraudes para otro tipo de seguros (de vivienda, agropecuarios, de vida, entre otros).

En tal sentido está previsto analizar si serían de utilidad los mismos algoritmos de minería de datos que se emplearon para analizar fraudes del seguro de automotores, o si

es conveniente considerar otros algoritmos.

Bibliografía

[Ali18] Ali Ghorbani and Sara Farzai. Fraud Detection in Automobile Insurance using a Data Mining Based Approach. *International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)*, 8(27):3764–3771, 2018.

[BEP] E. Burnaev, P. Erofeev, and A. Papanov. Influence of resampling on accuracy of imbalanced classification. *Eighth International Conference on Machine Vision (ICMV 2015)*.

[BFSO84] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[Bho11] Rekha Bhowmik. Detecting Auto Insurance Fraud by Data Mining Techniques. 2(4), 2011.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[Cre54] Donald Cressey. *Other People's Money. A Study in the Social Psychology of Embezzlement*, volume 19. 06 1954.

[FPSS96a] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37, Mar. 1996.

[FPSS96b] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Advances in knowledge discovery and data mining. chapter From Data Mining

to Knowledge Discovery: An Overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[Fri01] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001.

[Fri02] Jerome H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, February 2002.

[GF18] Ali Ghorbani and Sara Farzai. Fraud detection in automobile insurance using a data mining based approach. *International Journal of Mechatronics, Elektrical and Computer Technology (IJMEC)*, 8(27):3764–3771, 2018.

[GHKB12] Adrian Gepp, J Holton Wilson, Kuldeep Kumar, and Sukanto Bhattacharya. A Comparative Analysis of Decision Trees Vis-‘ a-vis Other Computational Data Mining Techniques in Automotive Insurance Fraud Detection. Technical report, 2012.

[GT16] Leila Goleiji and Mohammad Jafar Tarokh. Survey of detecting fraud in automobile insurance using data mining techniques, 2016.

[HKP11] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[HORQFR04] José Hernández Orallo, María José Ramírez Quintana, and César Ferri Ramírez. *Introducción a la Minería de Datos*. Pearson Educación, 2004.

[IFFN+12] Seyyed Mahmood Izadparast, Ahmad Farahi, Faramarz Fath Nejad, Babak Teimourpour, and AWT_TAG. Using data mining techniques to predict the detriment level of car insurance customers. *Journal of Information Processing and Management*, 27, 2012.

[IRI] Iris-ssn. Sitio WEB:http://seguro.ssn.gob.ar/iris. Accedido 01-10-2020.

[JAK+16] S.N. John, C. Anele, O. Okokpujie Kennedy, F. Olajide, and Chinyere Grace Kennedy. Realtime Fraud Detection in the Banking Sector Using Data Mining Techniques/Algorithm. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1186–1191. IEEE, dec 2016.

[KN18] G. Kowshalya and M. Nandhini. Predicting Fraudulent Claims in Automobile Insurance. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1338–1343. IEEE, apr 2018.

[KPR+12] Horacio Daniel Kuna, J. Germán A. Pautsch, Martín Rey, C. Cuba, Alice Rambo, Sergio Caballero, Ramón García Martínez, and Francisco Villatoro. Comparación de la efectividad de procedimientos de la explotación de información para la identificación de outliers en bases de datos. 2012.

[LLL98] Charles Ling, , Charles X. Ling, and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98*, pages 73–79. AAAI Press, 1998.

[MF13] Luis Marcano and Wilmer Fermín. *Comparación de métodos de detección de datos anómalos multivariantes mediante un estudio de simulación*, volume 25. UDO, Consejo de Investigacion, 2013.

[ORI] Orion de sesvi. Sitio WEB:https://www.sistema-orion.com/. Accedido 01-10-2020.

[R] R: The r project for statistical computing. Sitio WEB:https://www.r-project.org/. Accedido 03-01-2019.

[RA15] Vipula Rawte and G Anuradha. Fraud detection in health insurance using data mining techniques. In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–5. IEEE, jan 2015.

[Rgb] gbm: Generalized Boosted Regression Models. Sitio WEB:https://CRAN.R-project.org/package=gbm. Accedido 03-01-2019.

[RTL09] Payam Refaeilzadeh, Lei Tang, and Huan Liu. *Cross-Validation*, pages 532–538. Springer US, Boston, MA, 2009.

[RUS] Río uruguay seguros. Sitio WEB: https://www.riouruguay.com.ar/. Accedido 01-10-2020.

[SB13] H. Lookman Sithic and T. Balasubramanian. Survey of insurance fraud detection using data mining techniques. *CoRR*, abs/1309.0806, 2013.

[SKK18] J. O. Sinayobye, F. Kiwanuka, and S. Kaawaase Kyanda. A state-of-the-art review of machine learning techniques for fraud detection research. In *2018 IEEE/ACM Symposium on Software Engineering in Africa (SEiA)*, pages 11–19, 2018.

[SSM16] B. B. Sagar, P. Singh, and S. Mallika. Online transaction fraud detection techniques: A review of data mining approaches. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 3756–3761, March 2016.

[SSN] Superintendecia de seguros de la nación. Sitio WEB:https://www.argentina.gob.ar/superintendencia-de-seguros. Accedido 01-10-2020.

[SWB00] K A Smith, R J Willis, and M Brooks. An analysis of customer retention and insurance claim patterns using data mining: a case study. *Journal of the Operational Research Society*, 51(5):532–541, may 2000.

[Tor10] Luis Torgo. *Data Mining with R: Learning with Case Studies*. Chapman & Hall/CRC, 1st edition, 2010.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[UP17] K Ulaga Priya and S Pushpa. A Survey on Fraud Analytics Using Predictive Model in Insurance Claims. *International Journal of Pure and Applied Mathematics*, 116(21):629–640, 2017.

[WFH11] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[YL15a] Chun Yan and Yaqi Li. The identification algorithm and model construction of automobile insurance fraud based on data mining. In *Instrumentation*

and Measurement, Computer, Communication and Control (IMCCC), 2015 Fifth International Conference on, pages 1922–1928. IEEE, 2015.

[YL15b] Chun Yan and Yaqi Li. The Identification Algorithm and Model Construction of Automobile Insurance Fraud Based on Data Mining. In *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pages 1922–1928. IEEE, sep 2015.