UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DISEÑO Y DESARROLLO DE UN MODELO PREDICTIVO DE FUGA PARA UN SEGMENTO DE CLIENTES EN EL MERCADO DE *FOODSERVICE* EN AGROSUPER S.A. UTILIZANDO HERRAMIENTAS DE *MACHINE LEARNING*

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

RAIMUNDO JOSÉ VICENTE LUCHSINGER

PROFESOR GUÍA:
HÉCTOR ÁLVAREZ GÓMEZ

MIEMBROS DE LA COMISIÓN:
PAULA BRAVO BERRÍOS
ANDRÉS CABA RUTTE

SANTIAGO DE CHILE
2023

Resumen

La fuga de un cliente es una situación que se quiere evitar en todo tipo de industria, puesto que en la mayoría de los casos, los clientes representan un factor económico sumamente importante para las empresas, y además, es más rentable retener un cliente antiguo que captar uno nuevo [4]. De esto nace la necesidad de un modelo estadístico capaz de predecir esta situación de forma robusta y confiable con el fin de retener clientes y evitar su eventual fuga. El presente proyecto propone el diseño y desarrollo de un modelo predictivo de fuga de un segmento específico de clientes punto a punto en el mercado de *foodservice* en la empresa Agrosuper, para el cual se desarrollan, validan y comparan 3 tipos de modelos predictivos diferentes: regresión logística, *random forest* y *support vector machine*. Para este fin, siguiendo con la metodología de *multiple time slicing*, se construye un *dataset* de características explicativas utilizando 6 ventanas de tiempo de 12 meses cada una, que se utiliza para el entrenamiento y validación de los modelos, y un *dataset* de validación *out time* para evaluar el desempeño y la estabilidad temporal de los modelos. Los datos son analizados y transformados mediante transformaciones estadísticas univariadas, técnicas de tratamiento de multicolinealidad y reducción de dimensionalidad. Se realiza una segmentación de clientes en base a un análisis estadístico de comportamiento transaccional, donde se obtienen los clientes más estables y recurrentes. Finalmente se entrenan, validan y comparan los modelos mediante matrices de confusión y métricas de desempeño tanto en conjunto de validación como validación *out time*, en donde el modelo con el mejor desempeño en validación *out time*, y que seguirá a la implementación futura, resultó ser la regresión logística, siguiendo con *random forest* y terminando con *support vector machine*, el cual tuvo un desempeño relativamente bueno en validación pero disminuyó notablemente en validación *out time*. Con los resultados se concluye acerca de la importancia de la validación *out time*, puesto que permite evidenciar el real desempeño de un modelo y su estabilidad temporal, permitiendo escoger el mejor modelo para la implementación en producción.

Agradecimientos

A mi familia y amigos que siempre estuvieron ahí para apoyarme y sacarme una sonrisa cuando lo necesité. A los del voley que me ayudaron a distraerme adentro y afuera de la cancha. A todos los que creyeron y me ayudaron a creer en mi. Al guly por su calculadora.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Sobre la empresa

Agrosuper es un holding de empresas alimentarias chilenas, dedicadas particularmente a la producción, industrialización, distribución y comercialización de alimentos frescos y congelados de cerdo, aves (pollos y pavos), salmones y productos procesados (cecinas y elaborados).

La empresa comienza el año 1955 con la producción de huevos en la localidad de Lo Miranda, Región de O'Higgins. Cinco años más tarde, el negocio se expande hacia la producción y comercialización de pollos vivos y en 1974 se amplía hacia el procesamiento y comercialización de carne de pollo. En 1983 la compañía ingresa al negocio de la carne de cerdo, aprovechando la experiencia en la crianza de animales vivos y la infraestructura existente, y seis años más tarde, la compañía ingresa al negocio de la elaboración de cecinas, mismo año en el que se inicia la producción y comercialización de truchas y salmones.

En la actualidad, Agrosuper cuenta con 5 diferentes tipos de clientes. El primero de estos segmentos llamado **tradicional**, los cuales contemplan negocios de barrio, como lo son las carnicerías, *minimarkets*, por mencionar algunos. El segundo segmento de clientes se llama **foodservice**, el cual contempla negocios como restaurantes, casinos, hoteles y distribuidores, por mencionar algunos. El siguiente tipo de cliente, llamado **supermercado**, como su nombre da a entender, contempla clientes que son supermercados, como Tottus y Walmart, por mencionar un par. Luego, se tienen los clientes pertenecientes al segmento **industrial**, los cuales son clientes que compran su materia prima a Agrosuper para hacer sus propios productos como cecinas y elaborados. Finalmente, se tienen los **grandes clientes**, los cuales corresponden a clientes que son muy grandes para ser considerados en el segmento Tradicional, pero no alcanzan a ser supermercados de cadenas regionales. Un ejemplo de este último segmento vendría siendo la carnicería Doña Carne.

Como se evidencia en el título de este trabajo, el modelo estará enfocado a un segmento de clientes de la unidad de negocio de Foodservice, más específicamente, a los clientes "punto a punto", que son los clientes de Foodservice que compran y reciben la mercadería en cada punto, y se destacan por ser de bajo volumen y con un máximo aproximado de 4 locales.

Luego, es importante contextualizar acerca de la manera en que Agrosuper se relaciona con un cliente de este segmento. En el momento en que un local de Foodservice comienza a hacer negocios con Agrosuper, se genera un contrato que relaciona al cliente con la empresa, no obstante, no existe la necesidad u obligatoriedad de terminar dicho contrato para que el cliente deje de comprar o adquirir productos. De aquí nace la definición básica de la fuga de un cliente, que es el momento en que un cliente, que adquiere productos o servicios de cierta empresa de manera regular, deja de hacerlo. El hecho de que no exista la necesidad de terminar el contrato para dejar de comprar, genera incertidumbre sobre la situación real del cliente, ya que al no existir una instancia que asegure el desligamiento del cliente con Agrosuper, no se tiene seguridad absoluta sobre si el cliente volverá a comprar o simplemente se fugó. Esto último permite introducir la motivación del presente trabajo.

1.2. Motivación

La fuga de un cliente, en todo tipo de industria, es una situación alarmante y se intenta evitar a toda costa, ya que normalmente los clientes representan un factor económico sumamente importante en las empresas, y resulta menos costoso para las empresas retener clientes que captar nuevos [4]. Normalmente, para lidiar con este fenómeno, existen diversas estrategias para retener clientes, como proponer descuentos, ofertas especiales, despacho gratuito, por mencionar algunas.

Si bien existen acciones específicas que se siguen al detectar la fuga de un cliente, la principal dificultad radica precisamente en la capacidad de poder detectar a los clientes que potencialmente se fugarán, ya que, si bien suena algo simple, la verdad es que no siempre lo es, y depende fuertemente de la naturaleza de la industria en la que se está estudiando este fenómeno.

Si se considera el caso de un banco, la fuga de un cliente es relativamente sencilla de detectar, ya que al momento de fugarse, el cliente cierra su cuenta bancaria, y al tener esa información, es posible asegurar su fuga inmediatamente. Por otro lado, si consideramos el caso de Agrosuper, una empresa proveedora de alimentos de origen animal, de naturaleza no contractual, esta detección deja de ser una tarea trivial, puesto que al no ser necesaria una terminación de contrato por parte del cliente para dejar de comprar, existen muchas incertidumbres que no permiten asegurar la desvinculación de dicho cliente con la empresa, ya que el cliente pudo haber dejado de comprar temporalmente sin haberse fugado del todo, pudo haber cerrado el negocio por vacaciones o bien pudo haberse cambiado a la competencia directamente.

De esta forma nace la importancia de un modelo de fuga que sea capaz de detectar, de manera confiable y eficaz, clientes que muestran indicios de fuga, con el fin de poder retenerlos con tiempo mediante diversos protocolos y estrategias de marketing, y evitar su eventual fuga.

El presente informe muestra el diseño y desarrollo de un modelo predictivo de fuga para un segmento de clientes en el mercado de *Foodservice* en la empresa Agrosuper.

1.3. Objetivos

En lo que sigue, se presentan los principales objetivos del presente proyecto, tanto a nivel general como en ámbitos específicos.

1.3.1. Objetivo general

1. Diseñar y desarrollar un modelo predictivo de fuga para un segmento de clientes en el mercado de *Foodservice* en la empresa Agrosuper S.A. utilizando herramientas de *machine learning*.

1.3.2. Objetivos específicos

- Caracterizar a los clientes de *Foodservice* en cuanto a su comportamiento transaccional y escoger el segmento más relevante para la empresa.
- Realizar análisis univariados y multivariados de los datos.
- Diseñar, desarrollar y validar modelos predictivos para dicho segmento.
- Comparar y escoger el mejor modelo para la implementación futura.

Capítulo 2

Estado del arte

A continuación se muestran las soluciones que se plantean y utilizan en la actualidad para resolver la problemática principal del proyecto, correspondiente a la predicción de fuga de clientes en negocios de carácter no contractuales.

2.1. Fuga en negocios no contractuales

Como se mencionó previamente, la fuga de clientes es una situación que afecta a todo tipo de industrias, y constantemente se están desarrollando metodologías y maneras de prevenir esto, ya que la fuga de clientes, especialmente los más valiosos, es sumamente costoso para las industrias, dado que en la mayoría de los casos, son los que representan la mayor parte de la rentabilidad de un negocio, además de que, para estos, resulta menos costoso retener clientes antiguos que captar nuevos [4].

A modo de definición formal, la fuga de un cliente es el acto de abandono por parte de los clientes de la relación existente con la empresa a la que compran actualmente [12]. A diferencia de otros negocios, la principal dificultad que existe en predecir la fuga de un cliente en un **negocio no contractual** es aterrizar una correcta definición de fuga, es decir, cuando y bajo que criterios es posible etiquetar a un cliente como fugado, y cuando no. Esto es particularmente complejo, ya que al no existir una instancia o hecho que confirme la desvinculación de un cliente con la empresa (como lo son los bancos, empresas de telefonía móvil, entre otros), no se tiene conocimiento o registro real sobre si el cliente se desvinculó por un tiempo, para siempre, se fue con la competencia o simplemente no está comprando por el momento, y es por esto que la definición de fuga, en esta clase de industrias, es relativa, y depende del enfoque que se le de.

En la actualidad existen diversos casos de estudio que se han realizado para analizar y predecir la fuga de un cliente en negocios no contractuales. Estos estudios permiten evaluar diferentes maneras de aterrizar una solución a la misma problemática, utilizando diferentes maneras de definir la fuga, procesamiento de datos, modelos predictivos, por mencionar algunos, pero que finalmente todos apuntan a el mismo objetivo, correspondiente a lograr

predecir la fuga de un cliente en un negocio de naturaleza no contractual. Todos estos estudios toman definiciones diferentes de fuga de clientes, ya que como se mencionó, no hay una definición única en este tipo de negocios, por lo que dependerá estrictamente de quien esté realizando el trabajo, el tipo de negocio, las características de los clientes con los que se trabaja, entre otros, por lo que se deben tomar en consideración todos estos factores para lograr definir de la mejor manera posible si un cliente es fugado o no.

2.1.1. Construcción del *dataset*

En cuanto a la construcción del *dataset*, los proyectos actuales siguen la dinámica clásica de la gran mayoría de los proyectos de *machine learning* y *data mining*, lo que implica una selección de variables, un preprocesamiento de los datos, un etiquetado de los datos y la generación de los conjuntos de entrenamiento y validación que finalmente alimentarán los modelos predictivos y permitirán evaluar el rendimiento de estos.

Selección de variables

La selección de variables se realiza comenzando por la data sin procesar, la cual tiene información sobre el comportamiento de los clientes con respecto a la empresa, como puede ser información transaccional o de compra [10][13][5], actividad telefónica [3], o bien cualquier información que permita caracterizar el comportamiento del cliente dentro de una empresa. Desde esta información normalmente se extraen diversas características que entregaran valor al modelo sobre el cliente y sus patrones de compra, ya sean históricos o desde cierto periodo de tiempo por determinar.

Dentro de las características más utilizadas en los proyectos actuales, son los extraídos a través del análisis RFM, correspondiente a recencia, frecuencia y valor monetario. Estas permiten saber cuando fue la última compra de un cliente, con que frecuencia han comprado en el pasado y cuanto han gastado en total, respectivamente. Estas características han demostrado ser eficaces a la hora de entregar información sobre la disposición de un cliente a escuchar ofertas y propuestas de retención. Si bien en casos se utilizan como variables dentro del modelo, también son utilizadas para segmentar clientes, y de esta forma enfocar el modelo a un segmento específico de clientes.

Otra metodología de segmentación que se realiza, con el fin de enfocar el modelo a los clientes más importantes (económicamente hablando) para una empresa, es calcular la lealtad y valor de cada cliente [12], y de esta forma, poder generar un modelo para los clientes que más aportan a la rentabilidad de la empresa.

Luego, otra característica utilizada comúnmente para la segmentación de clientes, es la frecuencia (ya sea de compra, de utilización de servicio, entre otras)[3]. Es sumamente importante puesto que permite diferenciar a los clientes recurrentes de los clientes irregulares. Si bien en ciertos mercados, la regularidad de los clientes es más o menos estándar, existen otros en que se tiene clientes de todo tipo, y por lo tanto, la definición de fuga no puede ser la misma para todos. Para ejemplificar, se considera el caso de un cliente que compra

para 2 meses enteros, ya que le queda muy lejos para ir a comprar más seguido. Si se define un criterio de fuga general como el cliente que no ha comprado en 1 mes, bajo este criterio, el cliente recién mencionado sería etiquetado como fugado, cuando en realidad no lo está, simplemente tiene otro comportamiento transaccional. Aquí nace la importancia de escoger un correcto criterio de etiqueta de fuga, y la segmentación por frecuencia es una opción muy viable, en donde se le aplica el mismo criterio a clientes con frecuencias de compra similares.

Además de las características ya mencionadas y variables transaccionales de los clientes, normalmente se escogen variables personales como edad, genero, educación, localidad, por mencionar algunas, ya que a veces dichas características aportan información hacia el comportamiento del cliente dentro de la empresa.

Etiqueta de clientes

Con respecto al etiquetado de clientes, esto corresponde principalmente a escoger el criterio de fuga que más se adecúe a los clientes para poder etiquetarlos con la mayor precisión posible.

Como ya se mencionó anteriormente, este criterio dependerá fuertemente de la naturaleza del negocio y los clientes, por lo que no existe una manera estándar de etiquetar como fugado o no, y dado esto, se observan diversos criterios que se utilizan en los recientes proyectos.

Uno de los métodos o criterios más simples para etiquetar a los clientes es estandarizar un tiempo de inactividad máximo, y si un cliente no ha comprado o ha sido inactivo por dicho tiempo de manera consecutiva, se considerará como fugado, mientras que el caso contrario, es decir, si el cliente mostró actividad en dicho tiempo, se considerará como no fugado. Dicho tiempo de inactividad máximo, en algunos casos, se varía, con el fin de ver como afecta a la precisión de decisión de los modelos, como por ejemplo, evaluar tiempos de inactividad máximos de 1, 2 y 3 meses [10].

Cuando los clientes muestran actividades o frecuencias de actividad muy diferentes entre si, es importante generar criterios de fuga diferentes entre ellos, ya que, si se considera el caso de un proovedor de internet prepago, y se genera un criterio de fuga para los clientes que utilizan el celular diariamente, un cliente que utiliza su celular una vez por semana sería etiquetado de manera errónea como fugado, cuando en realidad no lo está [3]. En este caso, lo que se realiza es segmentar a los clientes en función de sus frecuencias de uso, y de esta forma agrupar a los clientes que tengan comportamientos de uso similar, y luego se utiliza un criterio similar de fuga para cada segmento.

Otro sistema de etiquetado es uno basado en lealtad y rentabilidad de clientes [12]. Una empresa define dos periodos de evaluación $P1$ y $P2$, y en primer lugar, escoge el 66 % de los clientes más rentables de ambos periodos (66 % más valiosos del P1 y 66 % más valiosos del P2), y luego se obtienen los clientes que se repiten en ambas listas, es decir, que están en el 66 % más valiosos de ambos periodos por separado, donde se obtiene la base de clientes valiosos de la empresa. Luego, se define la lealtad por cliente, que se calcula como la cantidad de productos de cierto tipo comprado en la empresa dividido en la cantidad de productos del mismo tipo comprado total. De esta forma, de los clientes más rentables, si estos poseían

una lealtad sobre el 33 % en el $P1$ son considerados como leales. Dicho todo esto, el criterio de fuga que se toma son los siguientes:

- Clientes cuyo porcentaje de lealtad está entre 33 % y 50 % en el período $P1$ se etiquetará como fugado si su porcentaje de lealtad es 0 (ninguna compra en la empresa) en el período $P2$
- Clientes cuyo porcentaje de lealtad sea sobre 50 % en el período $P1$ se etiquetará como fugado si su porcentaje de lealtad es al menos 50 % menor en el período $P2$ comparado con el $P1$

Finalmente, se tiene el criterio de fuga basado en la rentabilidad o ingresos generados por un cliente en un periodo de 3 meses [13]. Este criterio indica que un cliente será etiquetado como fugado si en la rentabilidad o ingresos generados por dicho cliente en un periodo de tres meses cumplen cualquiera de los siguientes criterios:

- En al menos un segmento o tipo de producto, la rentabilidad generada por dicho cliente disminuyó igual o más del 50 % en comparación con los 3 meses previos
- La rentabilidad total generada por dicho cliente disminuyó igual o más que un 30 % en comparación con los 3 meses previos

Luego, si alguno de dichos criterios se cumplen, el cliente será etiquetado como fugado, en caso contrario, no estará fugado.

De esta forma, es posible ver que en la actualidad existen diversos criterios para definir si un cliente está fugado o no, por lo que no existe una manera correcta de definirla, más bien existen formas más lógicas de hacerlo.

Separación en conjuntos de entrenamiento y prueba

Para la creación de conjuntos de entrenamiento y prueba, existen predominantemente 2 maneras de realizarlo. El primero, corresponde al método *two-time slicing*:

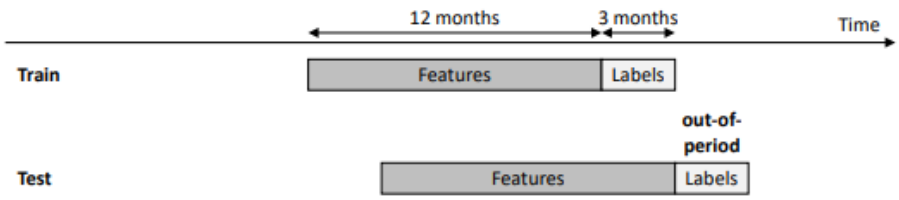

Figura 2.1: Metodología de *two-time slicing*.

Esto significa escoger un periodo T_{train} en el tiempo, usualmente en el pasado, y definir un periodo de M meses (que en el caso de la figura 2.1 son 12 meses), que representará cuantos meses hacia atrás se utilizarán para la construcción de variables y caracterización

del cliente. Luego, desde el periodo T_{train} se tomará la predicción hacia el futuro, o bien el conocido *forecasting* (que en el caso de la figura 2.1 son 3 meses). Para este método se utiliza este periodo pasado para generar la data de entrenamiento, y realizando un proceso análogo, pero para el tiempo presente, se genera un *dataset* de prueba más pequeño que el de entrenamiento. Esto se hace ya que es más realista, y por lo tanto válido, probar los modelos con data actual, y no con data del pasado. El método se llama *two-time slicing* ya que utiliza solo dos periodos de tiempo para definir el *dataset*.

El segundo método para separar en conjuntos de entrenamiento y prueba, que se basa en el anterior, es el *multiple-time slicing*:

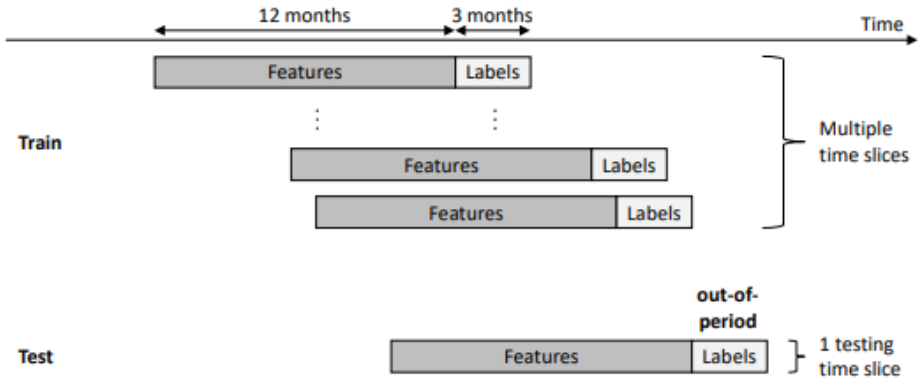

Figura 2.2: Metodología de *multiple-time slicing*.

Este, a diferencia del método anterior y como se evidencia en la figura 2.2, utiliza muchos periodos en el pasado para generar la data de entrenamiento, lo que beneficia notoriamente el entrenamiento de los modelos [14], ya que al utilizar diversos periodos del pasado, se toma información de los clientes bajo diferentes condiciones, lo que entrega robustez al modelo y a la predicción en general.

El otro método corresponde simplemente a tomar, tanto los conjuntos de entrenamiento como de prueba, de la misma ventana de tiempo, lo cual, según lo planteado anteriormente, no es lo óptimo para el entrenamiento y aprendizaje de los modelos.

2.1.2. Modelos predictivos

En cuanto a los modelos predictivos que se utilizan para la predicción de fuga como tal, se observa una gran variedad en los trabajos actuales, pero hay tres modelos que se repiten de manera consistente, y son algoritmos de clasificación muy reconocidos y altamente utilizados para todo tipo de problemáticas.

El primer modelo, y probablemente el más utilizado, es ***random forest*** [10][13][12][6]. Este algoritmo tiene muchas ventajas que lo hacen ser tan popular. En primer lugar, es un algoritmo que puede ser utilizado tanto para problemas de regresión como clasificación, por lo que abarca una inmensidad de aplicaciones. Otro punto positivo de este modelo es que posee baja varianza ya que se basa en el algoritmo *bagging*, que consiste en generar una predicción en base a muchos modelos iguales en estructura y entrenados con *datasets* que son diferentes

entre si, pero generados desde el mismo conjunto de entrenamiento original (*bootstrap)*, lo que permite tener modelos entrenados para diferentes tipos de datos y características, aportando robustez a la predicción y una varianza relativamente baja. Finalmente, este algoritmo, al predecir en función a mayoría de voto, es capaz no solo de entregar una predicción para una clase, si no que entrega también la probabilidad de predicción, lo cual para el contexto de fuga de clientes, resulta de suma utilidad.

El segundo modelo que más se utiliza para este tipo de soluciones es la conocida **regresión logística** [10][12][6]. Este algoritmo de clasificación, proveniente de la familia de regresiones, se utiliza para conocer la relación entre una serie de variables predictivas y la probabilidad de una clase, que usualmente suele ser binaria, por lo que al igual que el caso del modelo *random forest*, este algoritmo no solo es capaz de entregar una predicción de fuga o no fuga, si no que lo hace con cierta probabilidad, lo cual es una de las tantas razones por la que es tan utilizado. Otra razón que justifica su popularidad es su fácil implementación, interpretación y eficiencia a la hora de entrenarse. Por último, dada su naturaleza de regresión, es posible interpretar los coeficientes de la regresión como el nivel de importancia de las variables, y de esta forma analizar que variables aportan mayor información en la probabilidad de una clase y cuales son menos decisivas.

El tercer modelo predictivo utilizado para la detección de fugas y problemas de clasificación binaria es ***support vector machine*** [10][6]. Este es un modelo predictivo de aprendizaje supervisado, capaz de resolver problemas tanto de clasificación como regresión, aunque su uso más frecuente suele ser en problemas de clasificación. El algoritmo se basa en buscar el mejor hiperplano capaz de dividir y clasificar la data con el mayor margen de clase posible, por lo que es mucho más efectivo cuando la data es linealmente separable (existe un hiperplano en el cual los datos son divididos por un hiperplano de forma perfecta), por lo que una de sus ventajas es funcionar excepcionalmente bien cuando existe un margen de disociación entre las clases a clasificar. También, este modelo ha demostrado funcionar eficientemente específicamente en *datasets* con alta dimensionalidad. Uno de sus puntos bajos, es que, a diferencia de los anteriores, *support vector machine* no es capaz de entregar probabilidades de clase de forma nativa, no obstante, existen métodos de calibración probabilísticos, como la regresión isotónica o escalamiento de Platt para transformar la salida en probabilidades [2]. Otro punto bajo, y probablemente la razón por la que no es tan común ver su utilización como los otros dos modelos predictivos, es que su interpretación no es sencilla, por lo que analizar los resultados obtenidos con este modelo puede resultar más complejo en comparación con los modelos mencionados anteriormente.

Capítulo 3

Marco teórico

A continuación, se procede a describir el marco teórico del proyecto, en donde se presentarán y explicarán los trasfondos matemáticos y algebraicos detrás de los modelos y metodologías de procesamiento de datos que se utilizan para la realización del presente trabajo.

3.1. *Weight of evidence*

WOE (por sus siglas en inglés *weight of evidence*) es una técnica estadística de transformación de datos. Esta transformación permite modificar los valores de una variable predictiva, dividiendo esta en rangos y asignando valores nuevos a cada rango generado. La principal finalidad de esta transformación es lograr transmitir el poder predictivo de cada rango dentro de esta variable con respecto a la variable respuesta, puesto que el nuevo valor se define en base a la cantidad de positivos sobre los negativos que hay en dicho rango.

Esta técnica permite transformar una variable independiente de tal forma que se establezca una relación monótona con respecto a la variable respuesta, ordenando los nuevos valores en una escala logística (por la forma en que se construye el *WOE*), lo que lo hace altamente utilizado para el tratamiento de variables en modelos de regresión logística.

WOE entrega una herramienta para verificar la relación lineal entre una variable predictiva y la variable respuesta. Es por esto que es sumamente importante considerar la linealidad de probabilidades de correcta clasificación a la hora de definir los intervalos o puntos de corte, donde es esperable que las probabilidades de correcta clasificación en cada intervalo aumenten linealmente a medida que aumenta el intervalo, o disminuyan linealmente, en caso de existir una correlación negativa entre la característica evaluada y la variable respuesta.

Para un mejor entendimiento de la construcción de los nuevos valores, se supone el caso de una variable predictiva C, y una variable respuesta binaria Y, donde Y puede ser 1 (caso positivo) o 0 (caso negativo). Luego, la variable C tiene un rango de valores muy amplio, por lo que es posible definir 3 intervalos, los que llamaremos C_1,C_2 y C_3. Luego, los puntos de corte de dichos intervalos se definirán como x_1,x_2.

Por otro lado, se define B_k como la cantidad de eventos negativos en el intervalo k, o bien $Y = 0$. Análogamente, se define G_k como la cantidad de eventos positivos en el intervalo k, o bien $Y = 1$. Finalmente, se define b_k como el porcentaje de eventos negativos total dentro del rango k, y g_k como el porcentaje de eventos positivos total dentro del rango k, tal como se muestra a continuación:

$$b_k = \frac{B_k}{\sum_{k=1}^{3} B_k} \tag{3.1}$$

$$g_k = \frac{G_k}{\sum_{k=1}^{3} G_k} \tag{3.2}$$

donde en este caso, $k \in [1, 2, 3]$, ya que se está considerando el ejemplo de 3 intervalos.

Con estas definiciones, se desprende el valor *WOE*, definido de la siguiente manera:

$$WOE_k = \ln\left(\frac{g_k}{b_k}\right) \tag{3.3}$$

que es precisamente el valor transformado que se reemplazará por todos los datos que estén dentro del rango k.

Teniendo estas definiciones, se ejemplifica el funcionamiento de *WOE* mediante la siguiente tabla:

Tabla 3.1: Ejemplificación de WOE para intervalos de valores.

C	**Intervalo**	$\boldsymbol{B_k}$	$\boldsymbol{G_k}$	$\boldsymbol{b_k}$	$\boldsymbol{g_k}$	***WOE***
1	$[-\infty, x_1]$	90	2400	0.281	0.489	0.554
2	$[x_1, x_2]$	130	1300	0.406	0.265	-0.427
3	$[x_2, \infty]$	100	1210	0.313	0.246	-0.241
	SUM	**320**	**4910**			

Luego, se puede observar en la tabla 3.1 como se calculan los diferentes valores *WOE* dependiendo del intervalo al cual pertenece la variable predictiva, y como este valor es calculado a partir de la capacidad predictiva de dicho intervalo con respecto a la variable respuesta.

Si algún intervalo de cierta característica posee una proporción mayor de eventos positivos en comparación con eventos negativos, a dicho intervalo se le asignará un mayor valor *WOE*, lo que indica a su vez que dicho intervalo separa los eventos positivos de los negativos.

Al analizar la tabla 3.1, es posible ver, para el primer intervalo, que se tiene un valor de porcentaje de eventos negativos b_k igual a 0.281, mientras que se tiene un porcentaje de eventos positivos g_k igual a 0.406. Esto implica que si el valor de la variable predictiva C se encuentra en el primer intervalo, es más probable que la variable respuesta sea positiva, es decir, que $Y = 1$.

3.2. Reducción de dimensionalidad

El problema de reducción de dimensionalidad consiste con construir una representación de dimensión estrictamente menor que los datos originales con la finalidad de interpretar de mejor forma la información contenida en nuestros datos así como también disminuir el costo computacional en el entrenamiento.[15]

Para este fin, existen diversas técnicas, siendo de las más utilizadas el análisis de componentes principales (*PCA*), el cual se utilizará para el presente proyecto, permitiendo eliminar características que no entregan suficiente información minimizando la pérdida de información y mejorando la eficiencia computacional de los modelos.

3.2.1. *Principal Component Analysis*

Datasets de grandes tamaños son cada vez más comunes y usualmente son difíciles de interpretar a simple vista. *PCA* es una técnica para reducir la dimensionalidad de dichos conjuntos de datos minimizando la pérdida de información.[8]

Se considera un conjunto de observaciones $\{x_i\}_{i=1}^{N} \subset \mathbb{R}^M$, en donde $x_i = [x_{i1}, x_{i2}, ..., x_{iM}]^\top$. Luego, esto significa que se tienen N observaciones, y las observaciones tienen M atributos. Así, se referirá a x_{ij} como el j-ésimo atributo de la i-ésima observación. Es posible descomponer cada observación en la base canónica $\{e_i\}_{i=1}^{M}$ de $\mathbb{R}^M$ como sigue

$$x_i = \sum_{j=1}^{M} x_{ij} e_j \tag{3.4}$$

Notar que se puede representar cada vector x_i mediante una cantidad $M' < M$ de términos. Es decir:

$$x_i \approx \sum_{j=1}^{M'} x_{i\sigma(j)} e_{\sigma(j)} \tag{3.5}$$

donde $\sigma : \{1, 2, ..., M\} \mapsto \{1, 2, ..., M'\}$ es una permutación que prioriza las coordenadas más representativas de los datos.

No obstante, la base canónica no es el mejor candidato de base para descomponer las observaciones, puesto que por sí solos, conllevan poca información estructural que puede ser encontrada en los vectores observados.

En su defecto, se determinará una base cuyos componentes ordenados $\{c_1, c_2, ...\}$ capturan las M' direcciones ortogonales de máxima variabilidad de nuestros datos (lo que implica escoger las componentes que capturan las características con mayor variabilidad, y por lo tanto entregan más información al modelo). Luego, dado que $\langle c, x \rangle$ representa la proyección ortogonal de x sobre c, el primer elemento de la nueva base estará dado por:

$$c_1 = \arg\max_{||c||=1} \langle c, x \rangle \tag{3.6}$$

Este criterio es conocido como **análisis de componentes principales (*PCA*)**. La restricción sobre la norma $||c|| = 1$ es necesaria para evitar que $\langle c, x \rangle$ crezca indefinidamente, ya que $\langle \lambda c_1, x \rangle = \lambda \langle c_1, x \rangle$. Además de esto, es sumamente importante estandarizar los datos (atributos de media cero y varianzas marginales unitarias), con el fin de evitar sesgos de magnitud de datos en la determinación de máxima varianza.

Se considera una aproximación muestral de la varianza en la ec. 3.6 y se resuelve la siguiente definición:

$$c_1 = \arg\max_{||c||=1} \sum_{i=1}^{N} \langle c, x \rangle^2 \tag{3.7}$$

y utilizando la siguiente notación para la matriz de observaciones:

$$X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{bmatrix} = \begin{bmatrix} x_{11} & \dots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{NM} \end{bmatrix} \tag{3.8}$$

y así se reescribe la ecuación 3.7 de la siguiente forma:

$$c_1 = \arg\max_{||c||=1} ||Xc||^2 = \arg\max_{||c||=1} c^\top X^\top X c = \arg\max_{c} \frac{c^\top X^\top X c}{c^\top c} \tag{3.9}$$

Por otra parte, dada la propiedad de minimización del cociente de Rayleigh, la cual indica en pocas palabras, que si se tiene una matriz $M \in \mathcal{M}_{nn}(\mathbb{R})$ matriz simétrica y cuadrada, entonces para su cuociente de Rayleigh:

$$R(M, x) := \frac{x^\top M x}{x^\top x} \tag{3.10}$$

Su valor mínimo (máximo) corresponde al menor (mayor) valor propio de M, y es alcanzado en su vector propio asociado. De esta forma, dado que $X^\top X$ es simétrica, su cociente de Rayleigh es maximizado en el vector propio asociado al valor propio máximo de $X^\top X$. Consecuentemente, la proyección de una observación x_i en la dirección de máxima varianza, o bien la primera componente principal, está dada por

$$x_i^{(1)} = \langle x_i, c_1 \rangle \tag{3.11}$$

donde c_1 es el vector propio asociado al mayor valor propio de la matriz de covarianza muestral $X^\top X$.

El cálculo de las siguientes componentes se realiza de forma iterativa sobre los residuos del conjunto de observaciones con respecto a las componentes anteriores. De esta forma, *PCA* encuentra una nueva base ortonormal tal que las componentes maximicen la variabilidad, donde en algunos casos se puede perder interpretabilidad de las nuevas características generadas, pues son combinaciones lineales de las características originales de los datos. [15]

3.3. Modelos Estadísticos

3.3.1. Regresión logística

Idea general

El análisis de regresión es un subcampo del aprendizaje automático supervisado y es uno de los métodos estadísticos más útiles y utilizados en la actualidad, cuyo objetivo es establecer un método para la relación entre un cierto número de características y una variable objetivo o variable respuesta.

Dentro del mundo de las regresiones, existen múltiples metodologías y tipos de regresiones dependiendo del problema que se quiere resolver. Para el presente objetivo, el cual es predecir la probabilidad de fuga de un cliente en un mercado específico, la utilización de una regresión logística se adapta de manera perfecta.

El análisis de regresión logística es una técnica estadística multivariante que permite estudiar la relación entre una o más variables independientes (las cuales pueden ser cuantitativas o cualitativas) y una variable dependiente de tipo dicotómica. Recordar que una variable dicotómica es una variable que puede tomar sólo uno de dos valores mutuamente excluyentes, donde por lo general se codifican como $Y = 1$ para éxito e $Y = 0$ para fracaso [15]. Para este caso, la variable dicotómica a predecir será la fuga o no fuga de un cliente.

Formulación del problema

Considerando un enfoque generativo, se modelan dos objetos principalmente: en primer lugar, se tiene la "probabilidad condicional de clase", la cual representa como distribuyen los valores de las características (o *inputs*) x cuando la clase, o variable dicotómica en este caso, es C_k, denota, da por $\mathbb{P}(x|C_k)$. En segundo lugar, las "probabilidades de clase", denotadas $\mathbb{P}(C_k)$. De esta forma, la densidad posterior sobre las clases, o variables objetivo, dadas las características x, usando el Teorema de Bayes, se obtiene de la siguiente manera:

$$\mathbb{P}(C_k|x) = \frac{\mathbb{P}(x|C_k)\mathbb{P}(C_k)}{\mathbb{P}(x)} \tag{3.12}$$

La ecuación 3.12 representa la densidad de probabilidad de que la instancia pertenezca a la clase C_k dadas las características x. Para el caso binario, como lo es el caso en el presente trabajo, en donde se tienen 2 clases dicotómicas C_1 y C_2, es posible calcular la probabilidad de una de las clases de la siguiente forma:

$$\mathbb{P}(C_1|x) = \frac{\mathbb{P}(x|C_1)\mathbb{P}(C_1)}{\mathbb{P}(x)} \tag{3.13}$$

Si desarrollamos la ecuación 3.13 de la siguiente manera:

$$\begin{aligned}
\mathbb{P}(C_1|x) &= \frac{\mathbb{P}(x|C_1)\mathbb{P}(C_1)}{\mathbb{P}(x)} \\
&= \frac{\mathbb{P}(x|C_1)\mathbb{P}(C_1)}{\mathbb{P}(x|C_1)\mathbb{P}(C_1) + \mathbb{P}(x|C_2)\mathbb{P}(C_2)} \\
&= \frac{1}{1 + \dfrac{\mathbb{P}(x|C_2)\mathbb{P}(C_2)}{\mathbb{P}(x|C_1)\mathbb{P}(C_1)}} \\
&= \frac{1}{1 + exp(-r)} = \sigma(r)
\end{aligned} \tag{3.14}$$

En donde se introdujo la notación $r = r(x) = \ln\left(\dfrac{\mathbb{P}(x|C_1)\mathbb{P}(C_1)}{\mathbb{P}(x|C_2)\mathbb{P}(C_2)}\right)$ y la función logística $\sigma(r)$ definida como $\sigma(r) = \dfrac{1}{1+e^{-r}}$.

Para encontrar el r en la ecuación 3.14 conocido en la regresión logística, se considera el caso binario donde las densidades condicionales de clase son gaussianas multivariadas. Dichas densidades de probabilidad son de la siguiente forma:

$$p(x|C_k) \sim \mathcal{N}(\mu_k, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \cdot exp\left(-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)\right) \tag{3.15}$$

Donde $u_k \in \mathbb{R}^{\mathbb{M}}$ corresponde al centroide de la clase C_k y $\Sigma \in \mathbb{R}^{M\times M}$ simétrica y definida positiva, corresponde a la matriz de covarianza de las clases, la cual es la misma para ambas clases. Luego, tomando la densidad de probabilidad definida en la ecuación 3.15, el valor de $r(x)$ se expresa de la siguiente manera:

$$r(x) = \ln\left(\frac{\mathbb{P}(x|C_1)\mathbb{P}(C_1)}{\mathbb{P}(x|C_2)\mathbb{P}(C_2)}\right) = \ln\left(\frac{exp\left(-\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1)\right)}{exp\left(-\frac{1}{2}(x-\mu_2)^\top\Sigma^{-1}(x-\mu_2)\right)}\right) \tag{3.16}$$

Si desarrollamos la ecuación 3.16, se obtiene la expresión

$$r = a^\top x + b \tag{3.17}$$

donde se utiliza que

$$a = \sigma^{-1}(\mu_1 - \mu_2) \tag{3.18}$$

$$b = \frac{1}{2}(\mu_2^\top \sigma^{-1}\mu_2 - \mu_1^\top \sigma^{-1}\mu_1) + \ln\left(\frac{p(C_1)}{p(C_2)}\right) \tag{3.19}$$

y el hecho de que Σ^{-1} es simétrica.

Luego, si reemplazamos el valor de r encontrado en la ecuación 3.17 en el término obtenido en 3.14, obtenemos el siguiente término:

$$p(C_1|x) = \sigma(a^\top x + b) = \frac{1}{1 + exp(-a^\top x - b)} \tag{3.20}$$

Finalmente, la ec. 3.20 nos entrega el modelo de clasificación binario conocido como regresión logística.

Ahora que se definió el modelo, es importante definir como se ajustan los parámetros que lo componen. El modelo de clasificación está compuesto principalmente por dos parámetros principales:

- **Probabilidad de clase** $p(C_k)$, donde consideraremos las probabilidades como

$$p(C_1) = \pi \tag{3.21}$$

$$p(C_2) = 1 - \pi \tag{3.22}$$

  donde π es por determinar. También se puede observar la dicotomidad de las variables, al sumar 1 ambas probabilidades.

- **Probabilidad condicional de clase** $p(x|C_k)$, donde se consideran las probabilidades como

$$p(x|C_k) = \mathcal{N}(\mu_k, \Sigma) \tag{3.23}$$

  con $k \in \{1, 2\}$, donde los parámetros $\mu_1, \mu_2 \in \mathbb{R}^M$ y $\Sigma \in \mathbb{R}^M \times \mathbb{R}^M$ por determinar.

Luego, se denotan todos los parámetros por determinar mediante el parámetro $\theta = \{\pi, \mu_1, \mu_2, \Sigma\}$.

Para encontrar θ, se utiliza el método de máxima verosimilitud. Para efectos de notación, se utilizará la variable $t \in \{0, 1\}$, la cual representará a que clase pertenece una observación en particular. Una i-ésima observación será representada por la tupla (x_i, t_i), y si la observación es de la clase C_1, entonces $t_i = 1$, de lo contrario, $t_i = 0$. Luego, la verosimilitud queda definida de la siguiente manera:

$$L_i(\theta) = p(x_i, t_i|\theta) = p(x_i, C_1|\theta)^{t_i} p(x_i, C_0|\theta)^{1-t_i} \tag{3.24}$$

Luego, la ecuación 3.24 representa la probabilidad de que una observación x_i pertenezca a la clase C_1 cuando $t_i = 1$ y a la probabilidad de pertenecer a la clase C_0 cuando $t_i = 0$.

Luego, si se considera un conjunto de datos $\mathcal{D}$ como el siguiente:

$$X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{bmatrix} \in \mathbb{R}^{N \times M}, \quad T = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} \in \{0, 1\}^N \tag{3.25}$$

Donde se consideran N observaciones, la verosimilitud se define de la siguiente manera:

$$L(\theta) = \prod_{i=1}^{N} L_i(\theta) = \prod_{i=1}^{N} p(x_i, t_i|\theta) \tag{3.26}$$

$$= \prod_{i=1}^{N} p(x_i, C_1|\theta)^{t_i} p(x_i, C_0|\theta)^{1-t_i} \tag{3.27}$$

$$= \prod_{i=1}^{N} (p(x_i|C_1, \theta)p(C_1|\theta))^{t_i} (p(x_i|C_0, \theta)p(C_0|\theta))^{1-t_i} \tag{3.28}$$

Usando las definiciones 2.10, 3.22 y 3.23; se obtiene el siguiente término para la verosimilitud:

$$L(\theta) = \prod_{i=1}^{N} (\pi \mathcal{N}(x_i|\mu_1, \Sigma))^{t_i} ((1-\pi)\mathcal{N}(x_i|\mu_2, \Sigma))^{1-t_i} \tag{3.29}$$

Luego, para obtener los parámetros θ óptimos del modelo de regresión logística en cuestión, se utiliza la log-verosimilitud, correspondiente a aplicar logaritmo al término de la ec. 3.29, la cual queda de la siguiente manera:

$$l(\theta) := \log(L(\theta)) = \sum_{i=1}^{N} (t_i(log(\pi) + log(\mathcal{N}(x_i|\mu_1, \Sigma))) + (1-t_i)(log(1-\pi) + log(\mathcal{N}(x_i|\mu_2, \Sigma)))) \tag{3.30}$$

Luego, de esta ecuación es posible obtener los parámetros del modelo, utilizando la condición de primer orden con respecto a los diferentes parámetros. En función de la extensión de esta sección, se omitirá el desarrollo de estos cálculos:

1. Con respecto a π:

$$\frac{\partial l(\theta)}{\partial \pi} = \frac{\partial \log(L(\theta))}{\partial \pi} = 0 \tag{3.31}$$

$$\Rightarrow \quad \pi = \frac{N_1}{N_1 + N_2} \tag{3.32}$$

   En donde N_i corresponde a la cantidad de observaciones que pertenecen a la clase i, o bien, $N_i := Card(x : x \in C_i)$.

2. Con respecto a μ_1:

$$\frac{\partial l(\theta)}{\partial \mu_1} = \frac{\partial \log(L(\theta))}{\partial \mu_1} = 0 \tag{3.33}$$

$$\Rightarrow \mu_1 = \frac{1}{N_1} \sum_{x_i \in C_1} x_i \tag{3.34}$$

   y de manera análoga a lo anterior:

$$\mu_2 = \frac{1}{N_2} \sum_{x_i \in C_2} x_i \tag{3.35}$$

Así, la ec. 3.32 indica que el parámetro óptimo π es la razón entre la cantidad de elementos pertenecientes a la clase C_1 y la cantidad total de datos. Esto se debe a que π es la probabilidad de pertenecer a la clase C_1, por lo que es simplemente un promedio aritmético. También, las ecuaciones 3.34 y 3.35 muestran que las medias de las clases óptimas es la media muestral de los datos disponibles de cada clase.

3.3.2. *Support vector machine*

Idea general

Una gran desventaja de los clasificadores lineales binarios es la falta de atención o la poca importancia que se le da al margen de las clases, o bien la distancia entre las muestras de ambas clases. Este concepto es claro en métodos como el perceptrón, donde todas las soluciones que dividen las clases en dos son "igual de buenas", es decir, el perceptrón es insensible al margen descrito arriba.[15]

Por otro lado, en los métodos que funcionan mediante gradiente descendiente, como la regresión logística, este margen se maximiza indirectamente, lo cual conlleva a inestabilidades en el caso donde el parámetro de la regresión logística diverge, por ejemplo.

Luego, el clasificador *SVM* (*Support Vector Machine* por sus siglas en inglés) resuelve esta problemática, siendo un clasificador de máximo margen, en la que la parametrización y solución se desprende de un problema de optimización en la que se maximiza el margen de clasificación entre clases con ciertas restricciones.

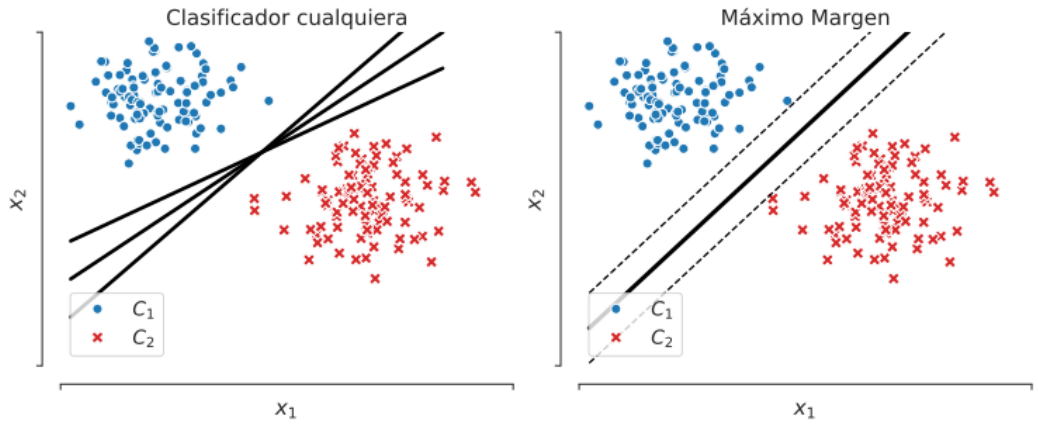

Figura 3.1: Hiperplanos de clasificación sobre conjunto de datos separables (izquierda). Clasificador de máximo margen sobre conjunto de datos separables (derecha).

Si se considera un problema de clasificación como el de la figura 3.1, es decir, donde las clases son linealmente separables, se puede ver que, en la figura de la izquierda, existen diversos hiperplanos que separan los datos de manera perfecta. Cada hiperplano define un modelo de clasificación capaz de asignar a una clase una nueva observación x_*. Cada uno de estos modelos definidos en el espacio de clases funcionará de manera similar, excepto en las zonas cercanas a los límites de clases. Esta problemática es justamente lo que *SVM* intenta resolver.

Por lo tanto, la idea general es escoger, de todos los hiperplanos o modelos capaces de separar de manera correcta las clases, aquel que nos entregue la máxima separación entre los datos y las regiones de clases, lo que corresponde a un clasificador de máximo margen (fig. 3.1 derecha).

El argumento de ocupar dicho criterio nace de asumir que si los datos generados en cada clase provienen de una distribución cualquiera, es de esperar que si se obtienen nuevos datos desde la misma distribución, estos estén cerca de los datos observados inicialmente. De este forma, con el máximo margen se pretende maximizar la probabilidad de que los nuevos datos de clase 1 (o bien clase -1) sean bien clasificados también.

Los datos (o vectores) que definen el margen los llamaremos vectores de soporte (*support vectors*), cuya función es restringir la rotación y expansión del margen. Una diferencia importante entre un clasificador de máximo margen como lo es *SVM*, y otros clasificadores binarios, es que estos últimos aprenden, o se parametrizan incorporando todos los datos de entrenamiento, mientras que *SVM* define el clasificador usando únicamente los vectores de soporte.

Formulación del problema

Para el entrenamiento del modelo, se denota un conjunto de entrenamiento de la forma $\{x_i\}_{i=1}^{N}$ con clases $\{1, -1\}$ y linealmente separable. Recordar que un hiperplano está definido por la siguiente ecuación:

$$\{x \in \mathbb{R}^n | w^\top x + b = 0\} \tag{3.36}$$

donde $w \in \mathbb{R}^n$ es el vector perpendicular al hiperplano y $b \in \mathbb{R}$ es el *offset*. De esta forma, si $w^\top x + b > 0$, se le asignará a x la clase 1, mientras que si $w^\top x + b < 0$, se le asignará a x la clase -1.

Es importante notar, que si $\{w, b\}$ es solución, también lo será $\{\lambda w, \lambda b\}$. Para evitar esta invarianza y asegurar que el problema tenga solución única, se puede imponer una restricción sobre los bordes del margen de clases. Denotando vectores soporte de cada clase mediante x_+ y x_-, se impone que para todo vector soporte x_+, x_-, estos pertenezcan a su respectiva clase, es decir:

$$w^\top x_+ + b = 1 \tag{3.37}$$
$$w^\top x_- + b = -1 \tag{3.38}$$

Luego, estos vectores de soporte pueden no ser únicos. Es importante notar que las ecs. 3.36, 2.26 y 3.38 definen 3 hiperplanos paralelos, puesto que los 3 poseen el mismo parámetro w. Esto se puede apreciar en la siguiente figura:

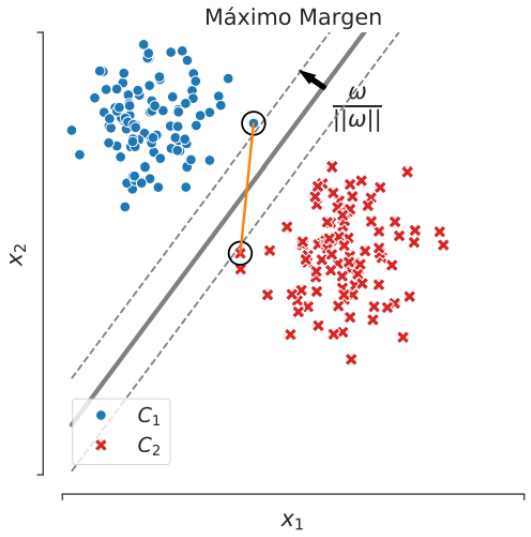

Figura 3.2: Clasificador de máximo margen con ancho de margen, vector unitario y vectores de soporte.

En donde además se muestra el vector unitario perpendicular a estos hiperplanos dado por $\frac{w}{||w||}$. Ahora se considera la variable m, que representará el ancho de margen, y es la distancia entre la región de decisión y cualquiera de las clases, o bien, corresponde a la mitad de la diferencia entre ambos vectores de soporte, proyectada en la dirección normal del hiperplano. El ancho de margen entonces estará definido de la siguiente manera:

$$m = ||\text{proy}_w(x_+ - x_-)|| = \frac{1}{2}||x_+ - x_-|| \cos(\theta) \tag{3.39}$$

$$= \frac{1}{2}||x_+ - x_-|| \left(\frac{w^\top (x_+ - x_-)}{||w|| \cdot ||x_+ - x_-||} \right) \tag{3.40}$$

$$= \frac{1}{2||w||} w^\top (x_+ - x_-) \tag{3.41}$$

donde se utilizó que $\cos(\angle(x, y)) = \frac{\langle x,y \rangle}{||x|| \cdot ||y||}$.

Si se incorporan las restricciones propuestas anteriormente, se obtiene la siguiente definición para m:

$$m = \frac{1}{2||w||} \left((w^\top x_+) - (w^\top x_-) \right) \tag{3.42}$$

$$= \frac{1}{2||w||} \left((1 - b) - (-1 - b) \right) \tag{3.43}$$

$$= \frac{1}{||w||} \tag{3.44}$$

Luego, considerando el ancho de margen m, y las siguientes restricciones para las clases:

$$y_i = +1 \Rightarrow w^\top x_i + b \geq +1 \tag{3.45}$$

$$y_i = -1 \Rightarrow w^\top x_i + b \leq -1 \tag{3.46}$$

Es posible formular el problema de clasificación de máximo margen mediante el siguiente problema de optimización:

$$\max_{w,b} \frac{1}{||w||} \tag{3.47}$$

$$\text{s.a} \quad y_i(w^\top x_i + b) \geq 1, \ i \in \{1, ..., N\} \tag{3.48}$$

Lo que quiere decir que se está maximizando el ancho de margen, sujeto a que todas las muestras están bien clasificadas (separabilidad).

Para evitar problemas de diferenciabilidad del recíproco de la raíz cuadrada en el objetivo del problema de optimización anterior, sobretodo cuando w es cercano a 0, se considerará la siguiente formulación equivalente del problema anterior:

$$(P) \quad \max_{w,b} \frac{1}{2}||w||^2 \tag{3.49}$$

$$\text{s.a} \quad y_i(w^\top x_i + b) \geq 1, \ i \in \{1, ..., N\} \tag{3.50}$$

Finalmente, este problema de optimización se resuelve mediante el método de Lagrange, donde se resuelve el problema dual, puesto que tiene una estructura más "amigable". Dicho esto, el lagrangiano del problema anterior está dado por:

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 + \sum_{i=1}^{N} \alpha_i (1 - y_i(w^\top x_i + b)) \tag{3.51}$$

donde los multiplicadores de Lagrange se definieron mediante $\alpha = (\alpha_1, \alpha_2, ..., \alpha_N)^\top$. Para obtener el lagrangiano dual del problema basta aplicar la condición de primer orden a L, puesto que es convexo. Esto nos entrega las siguientes definiciones:

$$\frac{\partial L}{\partial w} = w^\top - \sum_{i=1}^{N} \alpha_i y_i x_i^\top = 0 \Rightarrow \bar{w} = c\alpha_i y_i x_i \tag{3.52}$$

$$\frac{\partial L}{\partial w} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{3.53}$$

Así, utilizando las condiciones de primer orden del lagrangiano en la ecuación 3.51, se obtiene el lagrangiano dual que tiene la siguiente forma:

$$\theta(\alpha) = L(\bar{w}, \bar{b}, \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{3.54}$$

De esta forma, el problema dual consiste en maximizar $\theta(\alpha)$ sujeto a que $\alpha \geq 0$, o bien:

$$(D) \quad \max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{3.55}$$

$$\text{s.a} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.56}$$

$$\alpha_i \geq 0 \tag{3.57}$$

Luego, este problema es del tipo *QP* (*quadratic programming*), para el cual existen variados métodos para resolverlo de manera óptima y eficiente.

Una vez resuelto el problema dual (i.e., se han encontrado los valores óptimos para α), la predicción $\bar{y}$ de un nuevo punto x_* se expresa de la siguiente forma:

$$\bar{y}(x_*) = \text{sgn}(\bar{w}^\top x_* + b) = \text{sgn}\left(\left[\sum_{i=1}^{N} \alpha_i y_i \langle x_i, x_* \rangle \right] + b \right) \tag{3.58}$$

donde la función sgn(·) extrae el signo del argumento, es decir

$$\text{sgn}(x) := \begin{cases} -1 & \text{si } x < 0 \\ 0 & \text{si } x = 0 \\ +1 & \text{si } x > 0 \end{cases} \tag{3.59}$$

Finalmente, por el teorema de holgura complementaria, para un α óptimo, se tiene que

$$\alpha_i(1 - y_i(\bar{w}^\top x_i + b)) = 0, \ \forall i \in \{1, ..., N\} \quad \Rightarrow \quad \alpha_i = 0 \tag{3.60}$$

para toda observación x_i fuera del margen. Esto es sumamente importante e interesante, puesto que reafirma la propiedad que se mencionó al principio, y es que la predicción de clase depende únicamente de los vectores de soporte, o bien los que están en el margen, y las observaciones que estén fuera (que son generalmente la mayoría), no aportarán a la predicción de clase. Esto ayuda notoriamente a resolver el problema de optimización de manera más rápida y eficiente, ya que solo algunas variables duales α_i no serán nulas.

3.3.3. *Decision trees* y *random forest*

Decision trees

Las observaciones que definen el árbol de decisión se sitúan en un espacio, normalmente multidimensional, el cual se divide en regiones R_r, donde hay tantas regiones como hojas en el árbol. Para cierta región, cada observación que pertenezca a esta será etiquetada con la misma clase. Para la predicción, o etiqueta de una nueva observación $\vec{x}$, esta se definirá de la siguiente forma:

$$f(\vec{x}) = \sum_{r=1}^{R} \delta_{\vec{x} \in R_r} \underset{c=1,...,C}{\arg\max} \sum_{i:\vec{x} \in R_r} \delta(y^i, c) \tag{3.61}$$

donde δ es la función *Dirac* definida de la siguiente forma:

$$\delta : X \times X \to \{0, 1\} \tag{3.62}$$

$$\delta(u, v) \to \begin{cases} 1 & \text{si } u = v \\ 0 & \text{si } u \neq v \end{cases} \tag{3.63}$$

y c las etiquetas existentes.

La función definida en 3.63 define como se escogen las etiquetas para una nueva observación dentro del espacio. La primera sumatoria (izquierda) va sumando dentro de cada región

definida dentro del espacio de observaciones, producto del entrenamiento del árbol. El primer *Dirac* permite anular las regiones de la primera sumatoria, dejando solo la región a la cual pertenece la nueva observación, ya que en caso de estar en una región a la cual $\vec{x}$ no pertenece, el *Dirac* será 0, en el otro caso, será 1. La sumatoria interna (derecha) recorrerá todas las observaciones en la región a la que pertenece $\vec{x}$, revisando mediante la función *Dirac*, si la etiqueta de dichas observaciones son iguales a c o no. De esta forma, la función arg máx probará todos los valores que puede tomar c, es decir, todas las etiquetas posibles, y encontrará la que maximiza la sumatoria de *Diracs*, lo que significa que devolverá la predicción que más se repite en la región a la que pertenece la nueva observación $\vec{x}$. Así, la función 3.63 entrega la predicción de una nueva observación, en base a la moda de la etiqueta de la región en la que cayó.

Los nodos de los arboles representan un punto de separación de la data, en que se separan en base a un criterio de cierta variable, y se ramifica el árbol hacia abajo, pudiéndose encontrar otros nodos, o simplemente hojas, las cuales representan, en este caso, una clase o región. Para armar los nodos de los árboles de decisión, se deben generar criterios de separación, por lo que un nodo m representado por R_m se representará con los parámetros $\theta = (j, s_m)$, con j la variable de separación (característica sobre la cual se aplicará el criterio de separación) y s_m el umbral. De esta forma, bajo el nodo m se definen dos regiones en las cuales se divide el conjunto de clasificación:

$$R_m^{left}(\theta) = \{\vec{x} : x_j < s_m\} \tag{3.64}$$

$$R_m^{right}(\theta) = \{\vec{x} : x_j \geq s_m\} \tag{3.65}$$

Luego, es importante entender como escoger los parámetros θ para la división de las regiones de clasificación. Para esto, es necesario introducir el término de impureza, el cual denota que tan puro (valga la redundancia) es un conjunto de datos. Es posible definir la impureza de diversas formas, y dependerá del usuario que la compute. Por ahora, solo se llamará a la impureza como una función $I(\cdot)$ genérica.

Luego, se define lo siguiente:

$$G(R_m, \theta) = \frac{n_m^{left}}{n_m} I(R_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} I(R_m^{right}(\theta)) \tag{3.66}$$

donde n_m es la cantidad de observaciones en el nodo m, n_m^{right} la cantidad de observaciones en el nodo derecho de la división en el nodo m, y por ende, n_m^{left} la cantidad de observaciones en el nodo izquierdo de la división en el nodo m. Luego, 3.66 representa la impureza ponderada de la división realizada con los parámetros θ.

Dicho esto, para encontrar la división más eficiente de un nodo m en dos nodos hijos R_m^{left} y R_m^{right}, se busca seleccionar el parámetro θ^* que minimice la impureza de la división, o bien:

$$\theta^* = \arg\min_{\theta} G(R_m, \theta) \tag{3.67}$$

Como se mencionó anteriormente, la función de impureza puede definirse de diversas formas. En primer lugar, se aterriza la siguiente definición:

$$p_{mc} = \frac{1}{n_m} \sum_{i:\vec{x} \in R_m} \delta(y^i, c) \tag{3.68}$$

Así, la definición en 3.68 indica la proporción de muestras de entrenamiento en la región en la región m-ésima que son etiquetados con la clase c-ésima.

Dada esta última definición, es posible definir la primera impureza, llamada índice de Gini. Esta impureza se define de la siguiente forma:

$$G(R_m) = \sum_{c=1}^{C} p_{mc}(1 - p_{mc}) = \sum_{c=1}^{C} p_{mc} - \sum_{c=1}^{C} p_{mc}^2 = 1 - \sum_{c=1}^{C} p_{mc}^2 \tag{3.69}$$

El índice de Gini (3.69) se considera como una medida de la impureza de un nodo, donde un valor pequeño de este índica indica que un nodo contiene predominantemente muestras de una misma clase. Si todas las clases tienen la misma probabilidad de estar en dicho nodo (distribución uniforme), el índice es máximo y por lo tanto, la impureza del nodo. Por otro lado, si todas las probabilidades, o p_{mc} son cercanos a cero, y hay uno que es cercano a uno, el índice de Gini toma valores muy pequeños [16]. El índice de Gini se minimiza cuando todas las observaciones de una región son de la misma clase, por lo que todas las probabilidades serán cero, excepto una, que será 1, lo que entregará un índice de Gini de valor 0.

Un criterio alternativo para evaluar la impureza de un nodo es la entropía. Este criterio permite maximizar el *gain information*. Esta se define de la siguiente manera:

$$D = -\sum_{c=1}^{C} p_{mc} \log p_{mc} \geq 0 \tag{3.70}$$

Luego, el índice de Gini (3.69) y la entropía (3.70) son numéricamente similares, donde la entropía toma valores cercanos a cero si todas las probabilidades p_{mc} son cercanos a cero ó 1.

Random forest

El término *Random Forest*, nace de dos conceptos. El primero, y más evidente, es la utilización de diversos árboles de decisión, lo que genera un "bosque" de decisión. Luego, nace la natural duda de por qué se denota como un modelo aleatorio, y la respuesta es que en este modelo ocurren dos procesos aleatorios, que permiten reducir la varianza.

El primero de estos procesos aleatorios es el llamado *bootstrap*, la cual es una herramienta general para evaluar la precisión estadística. En palabras simples, *bootstrap* consiste en generar N conjuntos de entrenamientos diferentes entre si, los cuales son formados a partir de

la data de entrenamiento original, realizando un *sampling* aleatorio y con reposición, lo que significa que cada conjunto de entrenamiento nuevo será distinta al anterior, ya que tendrán elementos repetidos distintos entre si, lo que ayudará también a enfocar ciertos árboles a ciertos tipos de datos, aportando robustez al modelo en general. Esto se muestra de forma más clara en la siguiente figura:

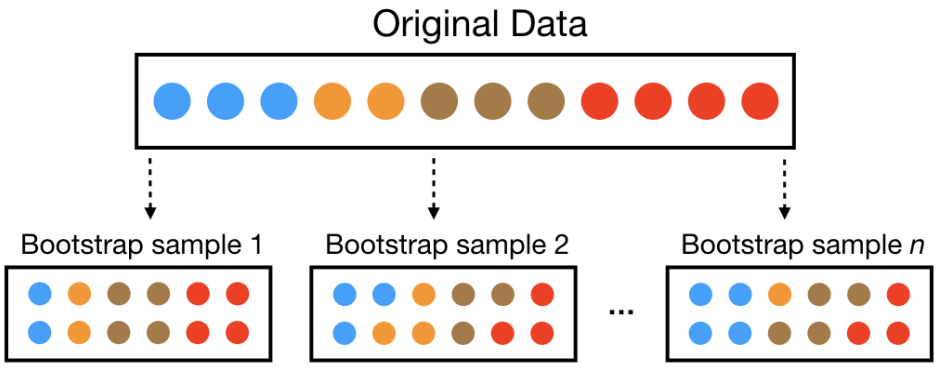

Figura 3.3: Diagrama de funcionamiento de la técnica *bootstrap*.

Luego, de cada conjunto de entrenamiento generado, se escogen cierta cantidad de características de manera aleatoria, con el objetivo de enfocar el entrenamiento de los árboles hacia distintas características, y así también reducir la varianza. Dado que estos dos procesos son aleatorios, es que recibe el nombre de *random forest*.

Luego de realizar un *bootstrap*, se entrenan N árboles de decisión con cada uno de los N conjuntos de entrenamiento mencionados previamente. Luego, para una observación de testeo de entrada, cada árbol genera una predicción sobre dicha observación, y la predicción final del modelo de *random forest* se basará en un sistema de votación, donde la clase más "votada" por los árboles será la predicción del modelo. Dicho esto, sea una observación x, y $C_n(x)$ la predicción del n-ésimo árbol construido. Luego, la predicción del modelo de *random forest* sobre la observación x será:

$$C_{RF}(x) = \text{MV}\{C_n\}_n^N \tag{3.71}$$

donde MV vendría siendo una función de mayoría de voto. El proceso de combinar resultados de diferentes modelos se llama agregación.

De esta forma, el término que se emplea al utilizar *bootstrap* con agregación se llama *bagging*. La técnica de *bagging* se utiliza principalmente para reducir la varianza de una función de predicción estimada. Esta metodología parece funcionar especialmente bien para procesos que presentan alta varianza y bajo sesgo, tal como los árboles de decisión. En el caso de clasificación, un "comité" de árboles decide la clase o etiqueta predicha para un nuevo ejemplo mediante un sistema de votación. [7]

La idea esencial detrás del concepto de *bagging* es promediar múltiples modelos ruidosos y no sesgados, y de esta manera, reducir la varianza. Es por esto, que los árboles de decisión son candidatos ideales para esta técnica, puesto que son capaces de capturar interacciones

complejas dentro de la estructura de la data, y de crecer lo suficientemente grandes, además de tener un sesgo relativamente bajo.

3.4. Metodologías de proyectos

3.4.1. CRISP-DM

CRIPS-DM (*Cross Industry Standard Process for Data Mining* por sus siglas en inglés) corresponde a una de las primeras metodologías que se estandarizaron para el desarrollo de proyectos de minería de datos, a principio de los años 90. Su funcionamiento se basa en el siguiente diagrama:

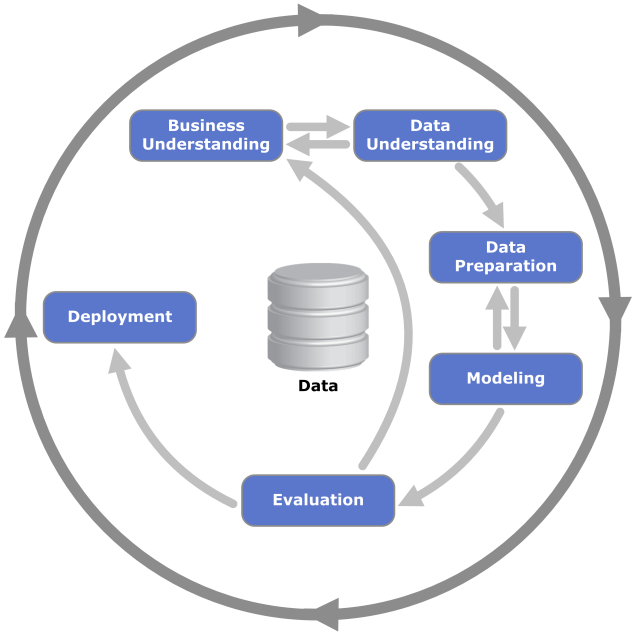

Figura 3.4: Diagrama de la metodología CRISP-DM.

Donde, como se puede observar de la figura 3.4, esta sigue una serie de pasos, desde el entendimiento del negocio hasta la implementación, pasando por una serie de iteraciones con el fin de asegurar el mejor resultado de cada proyecto o trabajo de minería de datos.

Dicho esto, estos 6 pasos se describen a continuación:

1. **Entendimiento del negocio:** Corresponde al proceso que se enfoca en el entendimiento de los objetivos y requerimientos del proyecto desde una perspectiva comercial, para posteriormente convertir este conocimiento en un problema de minería de datos, y con esto, generar un plan preliminar.
2. **Entendimiento de la data:** Este proceso comienza con la data inicial, y se procede a familiarizarse con esta, identificar problemas de calidad, limpieza, relaciones, *insights*, por mencionar algunos.

3. **Preparación de la data:** En este proceso se pretende modificar la data original con el fin de obtener la data final. Esto se refiere a todo lo que tenga que ver con análisis univariado y multivariado, limpieza, reducción de dimensionalidad, correlación, entre muchos otros.
4. **Modelar:** Utilizar modelos y técnicas de *data mining* y *machine learning* a los datos. Se itera sobre el preprocesamiento ya que usualmente los modelos requieren formatos específicos para ser alimentados, por lo que se debe volver al procesamiento de la data.
5. **Evaluar:** Implica analizar y asegurar el rendimiento de los datos bajo diferentes conjuntos de testeo no vistos anteriormente por los modelos.
6. **Implementar:** Implementar el modelo en el sistema para utilizarlo de forma automática a medida que se genera data nueva en tiempo real.

Es importante mencionar que esta metodología de trabajo es la base de la metodología que se utiliza para el desarrollo de modelos estadísticos en Agrosuper, y que por lo tanto se siguió para el desarrollo del presente proyecto, por lo que se revisará más en profundidad en la siguiente sección correspondiente a metodología (4.1), en donde se revisarán las diversas etapas de trabajo que la componen y como se llevó a cabo cada una de estas a lo largo del semestre.

Capítulo 4

Metodología

Continuando con el trabajo, se procede a explicitar la metodología que se sigue para la realización del proyecto.

4.1. Metodología del trabajo

Para la realización del presente trabajo, se sigue una metodología basada en el siguiente ciclo de trabajo por iniciativa, mostrado en la figura a continuación:

Figura 4.1: Diagrama de desarrollo de trabajo por iniciativa.

El ciclo mostrado en la figura 4.1 muestra como se lleva a cabo, de principio a fin, un proyecto de esta naturaleza, en la empresa Agrosuper. Es importantísimo recalcar que el diagrama mostrado en la figura 5.2 está basada en la metodología de desarrollo de proyectos de minería de datos CRISP-DM, la cual se explica en la sección 3.4.1.

El trabajo, como se desprende de la figura 4.1, es multidisciplinario en diversas etapas de su realización, en donde además de participar el área de *Data Science* en todo momento, participan también áreas de negocios y de tecnología.

El proyecto comienza con el **conocimiento y entendimiento del negocio** de *Foodservice* y las dinámicas de negociación entre proovedores y clientes. Entre los principales clientes de este mercado se tienen restaurantes locales y cadenas, *buffets*, casinos, por mencionar algunos. El entendimiento y conocimiento del negocio se lleva a cabo mediante reuniones virtuales con los jefes de ventas, ejecutivos y gerentes de la empresa del área de *Foodservice*. Con esto no solo se adquiriere conocimiento acerca del funcionamiento de la unidad de este negocio en específico, si no que también se identifican los principales dolores de los clientes de este mercado, lo cual es fundamental a la hora de caracterizarlos, puesto que permite entender los factores que más impactan en la permanencia o satisfacción de un cliente con la empresa.

Para el **levantamiento inicial de datos**, se realiza un acompañamiento de ruta con los encargados de preventa. Esto implica asistir, de manera oyente, a negociaciones presenciales entre los encargados de preventa y los clientes, consultando de local en local sobre el estado actual de los clientes. Además, se aprovecha la oportunidad para encuestar a los dueños de los locales sobre su nivel de satisfacción con Agrosuper, si están contentos o no, que problemas han vivido con la empresa, que factores harían o hacen que se deban fugar a la competencia, entre varios otros. De esta forma se adquiere conocimiento tanto acerca de las dinámicas de negociación que existen entre los clientes y la empresa, como los dolores de los clientes. Con esta información, se logra generar una lista preliminar de las variables más relevantes para los clientes, o bien, las que pueden generar un mayor impacto en su permanencia o fuga. Finalmente, se revisan que variables de la lista se pueden obtener y/o generar a partir de los datos que se tienen disponible, y finalmente se construye el *dataset* final. Luego, la **segmentación de clientes** se realiza mediante un análisis estadístico del comportamiento transaccional de los clientes. Para esto se estudiará el comportamiento de frecuencia de compra de cada cliente, con el fin de lograr segmentarlos en base a su mediana de frecuencia de compra y rango, permitiendo escoger el segmento de clientes más relevante para la empresa y enfocar el proyecto al modelamiento de estos.

Para el **análisis univariado y tratamiento de datos**, se realiza un estudio estadístico de la data a utilizar. Dentro de este estudio, se analizan representaciones gráficas y estadísticas como histogramas, percentiles, distribuciones, entre otros; con el fin de entender la naturaleza de la data y realizarle los tratamientos y transformaciones correctas. Para el tratamiento de los datos, se ordenará la data de forma que quede consistente entre todos los clientes, con el fin de no sesgar los modelos estadísticos. Para ejemplificar lo anterior, no es comparable utilizar los *tickets* medios generados de un emprendimiento que recibe pocos clientes, con una cadena nacional que genera notablemente más *tickets* que el emprendimiento. Una solución a esa diferencia de data, o bien un tratamiento que se puede aplicar, vendría siendo una escala logarítmica a los datos para redefinirlos en una escala similar, y de esta forma no sesgar el entrenamiento de los modelos estadísticos.

El **análisis multivariado y reducciones** se realiza mediante técnicas de reducción de dimensionalidad como *PCA* (*Principal Component Analisis* por sus siglas en inglés, sección 3.2.1), lo que permite reducir la dimensionalidad de la data, o en palabras más simples, eliminar variables o características que no entregan información suficiente para ser considerada en el entrenamiento de los modelos estadísticos. Esta técnica implica también construir y revisar tablas de correlación para identificar las variables más correlacionadas entre sí, agruparlas y aplicarles *PCA*, con el fin de obtener las principales componentes que expliquen la mayor parte de la varianza, reduciendo la dimensionalidad y la multicolinealidad de estas variables. Para ejemplificar la correlación de dos variables, se puede tomar el caso de cantidad de clientes mensuales y cantidad de *tickets* mensuales, donde evidentemente ambas variables estarán fuertemente correlacionadas, puesto que un negocio que tiene más clientes al mes intuitivamente tendrá más *tickets* al mes también, por lo que se podría eliminar una de esas variables, ya que la información, o varianza que entregan estas variables al modelo, están completamente representadas por una de las dos.

El **desarrollo de los modelos estadísticos** se realiza diseñando y construyendo 3 modelos predictivos de diferente naturaleza. El primero de ellos corresponde a un modelo *random forest* (sección 3.3.3), el cual es un método de predicción basado en la construcción de múltiples árboles de decisión y un sistema de votación. El segundo modelo es un *SVM* (*Support Vector Machine* por sus siglas en inglés, sección 3.3.2), el cual es un algoritmo de aprendizaje supervisado que se basa en encontrar un hiperplano que separe de la mejor forma posible un conjunto de datos. El tercer y último modelo predictivo que se construye es una regresión logística (sección 3.3.1), el cual es un tipo regresión utilizado para predecir el resultado de una variable categórica, en función de las variables independientes o predictoras. Es sumamente útil para modelar la probabilidad de un evento ocurriendo en función de otros factores, que es precisamente lo que se quiere lograr con este proyecto. Se desarrollan 3 modelos predictivos diferentes con el fin de analizar y comparar el desempeño de cada uno de estos, y eventualmente identificar cual es el más adecuado para predecir la fuga de un cliente.

Las **validaciones de los modelos estadísticos** se llevan a cabo mediante el análisis de diversas métricas de desempeño que permiten evaluar diferentes cualidades y capacidades de predicción de los modelos en cuestión, y de esta forma iterar sobre los parámetros de estos hasta maximizar (o minimizar) estas métricas. El proceso de validación de los modelos construidos comienza con el entrenamiento y validación de estos, en donde se definen hiperparámetros y se entrenan los modelos con un conjunto de entrenamiento, para posteriormente realizar la validación sobre un conjunto de datos nuevos. Luego, las métricas de desempeño obtenidas de la validación son analizadas y evaluadas, y se vuelve a iterar sobre los parámetros de los modelos. Una vez se tiene el modelo con el mejor desempeño del conjunto de validación, se procede a validar de manera *out time*, que implica utilizar un conjunto de datos absolutamente independiente de los conjuntos de entrenamiento y validación, con el fin de evaluar el rendimiento real de los modelos y su comportamiento predictivo en ejemplos más similares a los que serían en el caso de que pasasen a producción. En otras palabras, la importancia de la validación *out time* radica en que permite evaluar la estabilidad temporal de los modelos, permitiendo asegurar sus correctos funcionamientos en futuras predicciones.

En la presentación al negocio, se muestra tanto el mejor modelo como los resultados preliminares y las métricas de desempeño a los representantes del área de la unidad de

negocio *Foodservice*. Con esto, se pretende dar a entender a los encargados del área acerca del funcionamiento del modelo, que segmentos de clientes se modeló, que características se están considerando para su entrenamiento, cuales son las características más relevantes según el modelo obtenido y el rendimiento que se está alcanzando con el modelo. Para finalizar el proyecto, se implementa el modelo piloto y se deja en funcionamiento de forma automática quedando como un proceso almacenado y actualizándose periódicamente con los datos que se van adquiriendo con el tiempo. Finalmente, este proceso almacenado se incorpora con los sistemas de la empresa, gatillando procesos de retención de clientes, *mailing*, alertas a los preventa, por mencionar algunos.

4.2. Cronología del trabajo

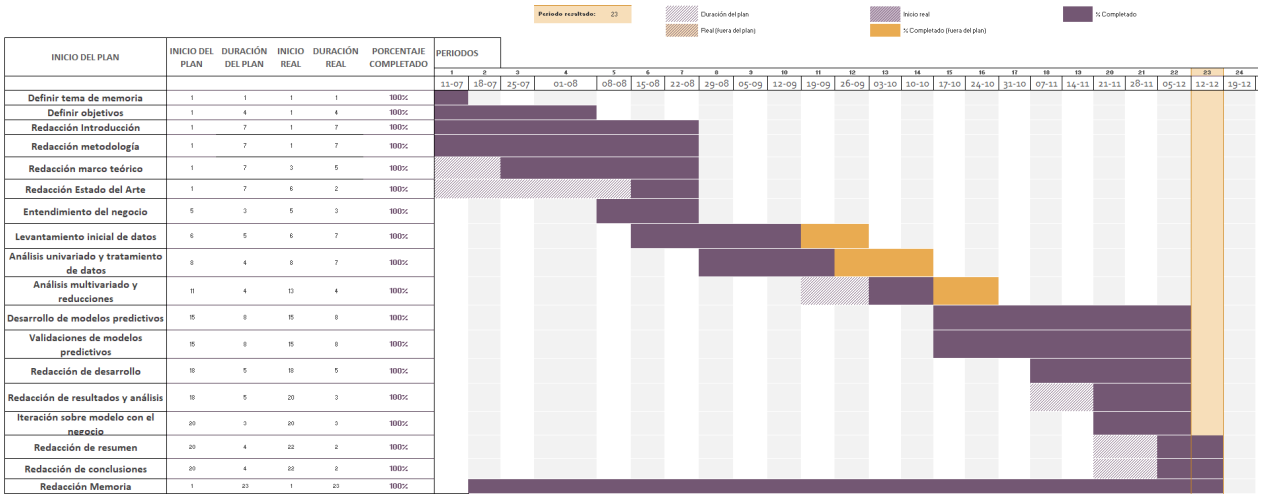

Figura 4.2: Carta gantt del desarrollo del trabajo.

Capítulo 5

Desarrollo

En lo que sigue del presente informe, se explica el desarrollo del proyecto, siguiendo la metodología propuesta anteriormente. En esta sección se pretende dar a conocer los pasos que se siguieron para completar tanto los objetivos específicos como generales del proyecto en cuestión.

5.1. Entendimiento del negocio

Como se menciona en la metodología del trabajo, el proyecto comienza con el entendimiento del negocio, lo que refiere a conocer las principales características y funcionamiento de la unidad de negocio de *Foodservice* en Agrosuper.

En primera instancia, se realizan cursos *online* acerca de las diversas líneas productivas que posee la empresa, tanto el proceso productivo de pollo como cerdo, pasando por la crianza, faena y distribución del producto desde las sucursales hasta los clientes. En estos cursos se adquirió conocimientos generales de los procesos productivos que ocurren para brindar un producto de calidad a los clientes, cumpliendo con las normativas de salud y seguridad impuestas por el gobierno de Chile.

Posterior a la inducción, se realizó la ruta con un preventa de *Foodservice* en la comuna de Colina. La ruta consiste en visitar, de manera presencial, diversos clientes de Agrosuper, con el fin de tener conocimiento acerca del estado de su negocio y su relación con respecto a nuestros servicios. De esta forma, es posible saber si el cliente está satisfecho, descontento, decepcionado, preocupado, por mencionar algunos; con nuestros servicios de entrega y/o calidad de producto. En esta instancia, los clientes son capaces de dar a conocer sus problemas de forma directa con el preventa, ya sea problemas de gestión como tiempos de envío, paquetes en mal estado, problemas de *stock*, etc.

En esta misma instancia, el preventa se encarga, además de escuchar y brindarle atención al cliente acerca de cualquier problema que pueda tener, de realizar pedidos para el cliente, a través de una aplicación desde su celular. Bajo este contexto, existen muchos clientes que esperan la visita del preventa (que regularmente es una vez por semana o semana de por

medio), para realizar pedidos para su negocio, lo que son categorizados como "pedidos por venta móvil".

Dicho esto, una gran parte de la realización de la ruta con el preventa se centraliza en la obtención de *feedback* acerca del sistema de atención y gestión de Agrosuper hacia sus clientes, y es por esto que es una instancia sumamente importante para conocer los dolores de estos, poder preguntarles y obtener información fundamental acerca de que características consideran importantes en términos de gestión o calidad de producto, que consideran bueno y que consideran malo, que situaciones harían o hacen que regularmente se fuguen, o bien dicho de otra forma, dejen de comprar a Agrosuper y se cambien a la competencia por un tiempo, ya que es importante enfatizar en que la fuga en negocios B2B no contractuales no es definida estrictamente como el cliente que no volvió a comprar nunca más, y la definición está sujeta a interpretaciones dependiendo de los clientes y sus frecuencias de compra.

La ruta permitió aterrizar un conjunto considerable de características que los clientes de *Foodservice* consideran críticos a la hora de realizar negocios con Agrosuper. Dentro de la información obtenida, se logró generar una panorámica de las razones principales de fuga de estos, siendo de las más recurrentes las fallas en la gestión, como envíos que no llegan a tiempo, cajas que llegan en mal estado (abiertas o maltratadas), el precio de los productos, cuando llega menos kilogramos de producto de los que se pidieron, pedidos realizados que nunca llegaron, falta de visitas presenciales, por mencionar algunos.

Luego, esta lista de características fue corroborada mediante reuniones virtuales con todo tipo de autoridades y trabajadores del área de *Foodservice* de la empresa, como la gerenta de ventas en *Foodservice*, el jefe de preventa, KAM líder, la jefa de procesos, entre otros.

Finalmente, se utilizan estas características recopiladas de la ruta y reuniones para generar una lista de variables preliminares, para posteriormente intentar construir la mayoría de ellas en base a la información que se dispone dentro de las bases de datos de la empresa, y utilizarlas para entrenar los modelos predictivos. Dentro de la lista de variables preliminares, se encuentran algunas como cantidad de pedidos mensuales por canal de venta (online, *call center* y venta móvil), *fillrate* de pedidos, elasticidad del cliente con respecto al precio, por mencionar algunas.

5.2. Construcción del *dataset*

A continuación, se realiza la construcción de la base de datos, correspondiente a la que se utilizará para el entrenamiento, validación y validación *out time* de los modelos.

Tanto para la construcción de las variables como el desarrollo y validación de los modelos de *machine learning*, se utilizan los lenguajes de programación R y SQL, codificando en la plataforma de *Databricks* en la nube de *Microsoft Azure.*

5.2.1. Tablones a utilizar

Antes de ahondar en la construcción de las variables y como se abordó cada una de estas, es sumamente importante dar a conocer las bases de datos con las que se trabajó y se trabaja día a día en la empresa, para contextualizar sobre que variables o campos se pueden obtener de los clientes.

Dicho lo anterior, existen **4 principales tablones o maestras** que se utilizan para obtener la información principal, las cuales se limpian para mantener únicamente la información que es de interés para la construcción de las variables:

- **Maestra de clientes:** En este tablón se encuentra la información completa de cada cliente que tiene o tuvo alguna vez Agrosuper (B.1). En esta base de datos, cada fila o registro es un local diferente, diferenciado por un código de local único. Un mismo cliente, diferenciado por un código de empresa único, puede tener diversos locales asociados, como lo son las cadenas de restaurantes o supermercados por ejemplo. Dentro de las características que se filtraron, se encuentra el código de local, región, fecha de creación (cuando empezó a ser cliente), tipo de local, entre otros. Al momento de limpiarla, también se filtró por clientes que fueran únicamente de *Foodservice* y nacionales.
- **Maestra de materiales:** En este tablón se encuentra la información de cada SKU que ofrece Agrosuper a sus clientes (B.4). En esta base de datos, cada fila o registro es un SKU o producto en específico, diferenciado entre ellos por un código SKU. Dentro de las características que se consideran de este tablón, se encuentra el código del SKU, fecha de creación, estado (congelado, refrigerado), entre otros.
- **Maestra de pedidos:** En este tablón se encuentra la información de cada SKU pedido por cada cliente de Agrosuper (B.3). En esta base de datos, cada fila o registro es un SKU pedido por algún cliente. De esta forma, cada pedido se diferencia por un número de documento, donde un número de documento puede aparecer en varias filas o registros, lo que implica que en un mismo pedido se incluyeron diversos SKU's. Dentro de las características importantes de este tablón se encuentra el código de local del cliente que realizó el pedido, el código del SKU pedido, el número de documento del pedido, la fecha en que se hizo el pedido, la cantidad de kilogramos pedidos, el precio por kilogramo del SKU y el valor total de dicho registro.
- **Maestra de facturas:** En este tablón se encuentra la información de cada SKU facturado por cada cliente de Agrosuper (B.2). En esta base de datos, cada fila o registro es un SKU facturado por algún cliente. De esta forma, cada factura se diferencia por un número de documento, donde un número de documento puede aparecer en varias filas o registros, lo que implica que en una misma factura se incluyeron diversos SKU's. Dentro de las características importantes de este tablón se encuentra el código de local del cliente que facturó, el código del SKU facturado, el número de documento de la factura, la fecha en que se facturó, la cantidad de kilogramos facturados, el precio por kilogramo del SKU y el valor total facturado por dicho SKU. Es importante destacar que el número de documento de factura es completamente diferente al número de documento de pedido.

Luego, para facilitar y digerir de mejor forma la información que entregan estas tablas, se construyen 2 tablas adicionales, en donde se realiza el cruce, en primer lugar, entre la maestra de pedidos con los clientes (referida desde ahora como PedidosxClientes), y en segundo lugar, entre la maestra de facturas con los clientes (referida desde ahora como FacturasxClientes). De esta forma, se logran generar 2 nuevos tablones, tanto de pedidos como de facturas, en las cuales se encuentra información únicamente de clientes pertenecientes a *Foodservice*.

Así, se utilizan estas 4 maestras y 2 tablones para generar las variables que se utilizarán para el entrenamiento y validación de los modelos.

5.2.2. Estructuración de la base de datos

En lo que sigue, se procede a estructurar la base de datos que se construirá para el entrenamiento y validación de los futuros modelos predictivos.

La estructura general que se sigue para la construcción del *dataset* corresponde a la mostrada en la sección 2.1.1 del estado del arte, en la cual se explica la metodología *multiple-time slicing* 2.2. En palabras sencillas, esta metodología implica utilizar múltiples ventanas de tiempo pasadas, desfasadas entre si temporalmente, para realizar el entrenamiento y validación de los modelos predictivos, en conjunto con una ventana de tiempo actual para la validación *out time*. Esta metodología permite generar una base de datos más robusta, al utilizar diversas ventanas para su construcción, entregando información del comportamiento temporal de los clientes.

Por otro lado, la validación *out time* permite evaluar el rendimiento real de los modelos construidos en cuanto a su capacidad de clasificación, ya que se está validando con un mes que no fue utilizado para el entrenamiento, por lo que esta validación permite ver la estabilidad del modelo en el tiempo.

Dicho esto, para el presente proyecto se consideran 6 ventanas de tiempo pasadas para el entrenamiento y validación de los modelos predictivos. Cada ventana de tiempo contempla un periodo de 12 meses para la construcción de las variables o características, con un mes de evaluación para construir la etiqueta de los datos. De esta forma, se generarán 6 ventanas de tiempo donde a cada una de estas se le calcularán las características explicativas de fuga utilizando los 12 meses de historia. Finalmente, se deja una ventana de tiempo más actual para la validación *out time*. La estructura general se ve de la siguiente forma:

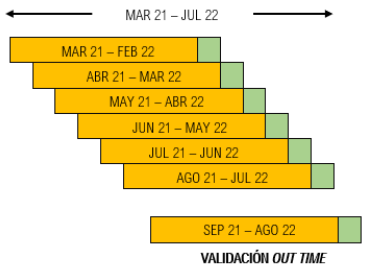

Figura 5.1: Estructura de construcción de ventanas de tiempo.

Luego de realizar el cálculo de las variables a cada ventana de tiempo, se realiza una segmentación en base al comportamiento transaccional de cada cliente, y un filtro en base a la cantidad de pedidos que tiene en los últimos 3 meses, el cual se explicará con más detalle en las secciones posteriores. Esto permitirá obtener los clientes más relevantes para la empresa, donde se dejarán a los clientes más recurrentes, fieles y estables en cuanto a su comportamiento de frecuencia de pedidos. Luego de esto, se procede a etiquetar a los clientes como fugados o no fugados, con el fin de poder entrenar los modelos futuros. Esta etiqueta se realiza utilizando el mes de evaluación indicado en color verde claro de cada ventana de tiempo en la figura 5.1. Una vez realizados estos procesos, se juntan las 6 ventanas de tiempo en una única base de datos la cual queda de la siguiente forma:

Figura 5.2: Estructura de unión de ventanas de tiempo en el *dataset* final.

En donde finalmente se obtienen 13.691 clientes en total, donde 12.098 son etiquetados como **no fugados** y 1.593 son etiquetados como **fugados**. Es importante mencionar que es altamente probable que exista una númerosa cantidad de clientes que se repiten en más de una ventana de tiempo. No obstante, es algo intencional, ya que en una ventana de tiempo el mismo cliente puede estar fugado por diversos motivos, y en otra no estarlo, y ambos datos nos entregarán información del por que se fugó en un momento y por que no lo está en otro, puesto que es relevante recordar que la fuga en este contexto no se define como el cliente que jamás volvió a comprar, si no más bien el que dejó de hacerlo por un periodo de tiempo excepcionalmente prolongado.

5.2.3. Descripción y construcción de variables independientes

A continuación se especifica el desarrollo de la construcción de variables, construidas para cada ventana de tiempo mostrada en la figura 5.1, utilizadas posteriormente para el entrenamiento y validación de los modelos predictivos. Como se mencionó previamente, se generó una lista de variables preliminares en la etapa de entendimiento del negocio, de la cual se desprenderán las variables que finalmente serán construidas y utilizadas.

Es importante mencionar que para la mayoría de las características construidas, a modo de entrenamiento, se consideró la mediana de los últimos 12 meses como la variable a utilizar, no obstante, a la hora de validar *out time* (y pasar a producción), se utilizará únicamente el valor del último mes. Se tomó la mediana ya que se quiere obtener el comportamiento normal de un cliente, y esta variable representa justamente el comportamiento regular de un local, a diferencia del promedio, el cual es muy sensible a los valores atípicos, que pueden suceder frecuentemente en este mercado.

La primera variable que se construye es la de **pedidos mensuales**. Para esta variable, se

utiliza únicamente la tabla PedidosxClientes. Lo que se realizó fue tomar, para cada cliente, la cantidad de pedidos que tiene al mes, durante un año, tomando la cantidad de números de documento de pedido diferentes (ya que un pedido puede tener diversos registros en la tabla). De esta forma, se tiene en cada ventana de tiempo, para cada cliente, 12 valores, donde cada valor es la cantidad de pedidos que realizó cada mes de dicha ventana de tiempo. Luego, a ese 'vector' de 12 valores, se le calculó mediana, rango y promedio. Para efectos del entrenamiento del modelo, se consideró la mediana como la variable a utilizar.

De manera análoga a la construcción de la variable anterior, correspondiente a pedidos mensuales, se construyen las variables de pedidos mensuales por omnicanalidad, lo que significa que se obtiene la cantidad de pedidos mensuales por cada canal de venta de la empresa, las cuales para el caso de *Foodservice* son principalmente 3. La primera de estas variables es **pedidos por venta móvil**, los cuales son pedidos que realiza el cliente mediante el preventa de manera presencial. La segunda corresponde a **pedidos *online***, los cuales son pedidos realizados a través de la página web de la empresa. La tercera y última variable es **pedidos por *call center***, cuyos pedidos se realizan a través de llamadas con ejecutivos de venta de la empresa. Al igual que la variable anterior, se considera como variable la mediana de los pedidos mensuales de los últimos 12 meses (para cada ventana de tiempo).

Siguiendo con la construcción de las variables, se obtiene la variable de **kilogramos mensuales pedidos**. De forma análoga a las variables anteriores, esta se obtiene a partir de la tabla de PedidosxClientes, donde esta vez se consideran los kilogramos de productos pedidos por cliente. En esta variable, para cada cliente, para cada mes de la ventana de tiempo en la que se encuentra, se obtiene la cantidad de kilogramos pedidos totales, sumando los kilogramos pedidos por cliente en cada mes, y se toma la mediana de los 12 valores como variable final.

La siguiente variable que se construye corresponde a la de **cantidad de pedidos devueltos**. Esta variable se obtiene del tablón de facturas, la cual tiene un campo que indica si un pedido fue devuelto o no, indicando en este mismo campo que factura fue la devuelta, donde una factura devuelta se denota con un valor monetario negativo, siendo ese el valor reembolsado hacia el cliente. Para esta variable, se toma el total de pedidos devueltos por cliente, tomando la suma total de número de documento de facturas únicos que tengan dicho campo completo. A diferencia de las variables anteriores, esta se toma de manera total, no mensual.

A continuación, se construye la variable correspondiente a **cumplimiento de pedidos**. Esta variable denota, en porcentaje, el cumplimiento que tuvo Agrosuper con respecto a los pedidos que realizó cada cliente. Para esta variable, se utilizan tanto los tablones de FacturasxClientes como PedidosxClientes. De esta forma se toma, en primer lugar, la cantidad total de pedidos hechos por cliente y la cantidad de facturas hechas por clientes. Luego, la variable se construye como la división entre las facturas y los pedidos hechos por cada cliente. Así, esta variable explica que tanto ha cumplido Agrosuper con la entrega de pedidos con sus clientes, donde si tienen un porcentaje bajo, pocos de los pedidos que ha realizado el cliente han sido efectivamente entregados, y viceversa.

Luego, se realiza la construcción de **meses de antigüedad**. Esta variable indica, en meses, cuanto tiempo llevan haciendo negocios con Agrosuper. Para generar esta variable,

se utiliza el tablón de clientes, en donde simplemente, para cada cliente de *Foodservice*, se obtiene el tiempo transcurrido entre la fecha actual, con la fecha de creación (día en que el cliente comenzó a comprar por primera vez) en meses.

Posteriormente, se construye la variable de **flexibilidad con respecto al precio**. Esta variable denota que tan flexible, o que tanta es la disposición de un cliente a comprar un SKU sin importar los cambios en su precio. Esta variable entrega información sobre que tanto valora, un cliente, la calidad del producto por sobre su precio, donde si un cliente tiene mayor flexibilidad, este compra un SKU a pesar de que suba de precio, y por lo tanto, es evidentemente más 'leal' a la empresa. Para la construcción de esta variable, en primer lugar, por cada cliente, se toman los n SKU's tal que constituyan el 50 % de los kilos totales pedidos por dicho cliente. De esta forma se obtienen los principales SKU's que cada cliente suele pedir. Una vez se tienen los n SKU's principales para cada cliente, se obtienen los rangos de precios (diferencia entre precio máximo que ha pagado por ese SKU y precio mínimo que ha pagado por ese SKU) para cada SKU principal. Luego, al tenerse la lista de rangos de precios para cada cliente, se obtiene el promedio ponderado de dichos rangos, ponderando por la cantidad de kilos que ha pedido por cada SKU. Así, para cada cliente se tiene la flexibilidad con respecto al precio, obtenida de los SKU's que más piden.

La siguiente variable que se construye corresponde a la del ***fillrate* de kilogramos**. Esta indica en porcentaje, de manera similar a la variable de cumplimiento de pedidos, que tanto cumplió la empresa con los pedidos de los clientes en cuanto a kilogramos de producto pedido. En este tipo de mercado, no es poco común que un pedido se entregue con menor cantidad de producto que lo solicitado, y esto sucede mayoritariamente debido a la falta de *stock* producto de la asignación y priorización de este, puesto que siempre existe más demanda que oferta. Es por eso que esta variable es capaz de transmitir el grado de cumplimiento de la cantidad de producto solicitado por el cliente, donde se espera que los clientes con menor *fillrate* sean más propensos a cambiarse a la competencia que los que reciben un mayor cumplimiento de cantidad de kilogramos de producto pedido. La construcción de la variable se realiza de manera análoga a la de cumplimiento de pedidos, tomando de manera mensual, la división entre la cantidad de kilogramos facturados y la cantidad de kilogramos pedidos. Luego, se toma la mediana de los 12 valores mensuales obtenidos como variable final. Para este fin, fue necesario realizar una modificación al tablón de facturas, puesto que los pedidos realizados entre el 27 y el 30 (o 31 dependiendo del mes) eran facturados al mes siguiente, por lo que afectaba los valores de cumplimiento de pedidos y *fillrate* de kilogramos, ya que habían facturas que pertenecían a pedidos del mes anterior, por lo que dichas facturas se pasaron al mes anterior, y posterior a eso se calcularon ambas variables como se mencionó.

Continuando con la construcción de las variables, se genera la variable **locales asociados**. Esta variable indica si un local (o cliente) tiene más locales asociados al mismo rut de empresa o no. La idea es lograr caracterizar a los clientes en cuanto a su tamaño o expansión dentro del mercado, donde si tienen un solo local asociado al rut de empresa, es posible que sea negocio particular o emprendimiento, mientras que si tienen múltiples locales, posiblemente sean negocios más grandes o cadenas. Para la construcción de esta variable, se utiliza el tablón de clientes, en donde en primer lugar, se genera una tabla auxiliar en la cual para cada rut de empresa, se toma cuantas veces está repetido ese mismo rut dentro del tablón. Luego, a cada cliente de *Foodservice*, se toma el rut de empresa asociado a dicho local, y se obtiene la cantidad de locales asociados a dicho rut de empresa mediante la tabla recién

construida, obteniéndose la cantidad de locales asociados al mismo rut de empresa.

Luego, se procede a construir la variable de **localización del cliente**. Esta variable permite caracterizar al cliente en cuanto a la zona donde se encuentra establecido. La variable consta de 5 localizaciones, la cual está codificada de forma *one hot encoding*. Esto último significa que solo una de las 5 columnas de localizaciones será de valor 1, mientras que las otras serán cero. Las localizaciones consideradas como variables son zona sur, zona centro-sur, zona centro o santiago, zona centro-norte y zona norte. Así entonces, estas 5 serán variables del modelo, en donde, por cada cliente, solo una columna de dichas 5 tendrá valor unitario (la zona en donde está ubicado el local) y las otras 4 tendrán valor nulo.

Siguiendo con la construcción del *dataset*, se procede a generar la variable **diversificación**. Esta variable permite caracterizar al cliente en cuanto que tan diverso es habitualmente a la hora de realizar pedidos. La variable se construye de manera mensual, en la que, para cada mes, se obtiene la cantidad de SKU's diferentes que pidió de la tabla de PedidosxClientes. Luego, la variable final será la mediana de los 12 valores mensuales obtenidos para la ventana de tiempo en la que se está construyendo la variable.

La siguiente variable que se construye corresponde a la de **diferencia porcentual de kilogramos pedidos**. Tal como lo dice su nombre, esta variable indica, de manera porcentual, la diferencia promedio mensual que tiene un cliente en cuanto a kilogramos pedidos totales, con respecto al mes anterior. De esta forma, a modo de ejemplo, si un cliente compra 100 kilogramos un mes, y al mes siguiente compra 10, la diferencia porcentual de ambos meses será -90 %. Luego, de los 11 valores (12 meses, 11 diferencias porcentuales), se obtiene la mediana. Luego, esta variable permite caracterizar al cliente en cuanto a su comportamiento transaccional con la empresa el último año, donde si la mediana es negativa, significa que a lo largo del último año ha ido disminuyendo su compra con nosotros, y viceversa.

Para finalizar la construcción de variables, se genera la llamada **visitas mensuales**. Esta variable indica cuantas visitas mensuales ha recibido cada cliente el último año. Como se mencionó anteriormente, los preventa son encargados de realizar visitas a los clientes, normalmente una vez por semana, en donde se toman reclamos, quejas, pedidos, etc. Estas visitas tienen como objetivo mantener al cliente satisfecho mediante una interacción más personal, en la cual el cliente se siente escuchado. Esta variable se genera a través de una tabla auxiliar, la cual contiene la información de dichas visitas realizadas por cada preventa, en donde se especifica a cuantos metros del local el preventa realizó una acción relacionada al local, en su dispositivo móvil. Es importante contextualizar que en esta aplicación, los preventa realizan pedidos, marcan asistencia, entre otros. Luego, todas las acciones que realice el preventa en su dispositivo quedarán guardadas en dicha tabla. Así, si un cliente le manda un mensaje por interno a un preventa para que realice un pedido por el, y este no está cerca del local, la distancia de dicha acción, en la tabla, será sumamente alta, y por lo tanto es posible filtrar distancias menores a cierto umbral para asegurar que fue realmente una visita. Este umbral en la empresa se suele considerar 500 metros, donde si se registra un pedido fuera de dicho radio, se descarta como pedido móvil.

Luego, estas 15 variables son construidas para cada una de las 6 ventanas de tiempo definidas en la figura 5.1. La estructura de la tabla de características construidas se muestra en el anexo del presente informe, en la figura B.5.

5.2.4. Segmentación y filtros realizados

Una vez construidas las variables explicitadas en la sección 5.2.3, para cada ventana de tiempo, se procede a realizar la segmentación de clientes en base a su comportamiento transaccional, con el fin de poder aplicar filtros y escoger el segmento más relevante de clientes para la empresa y modelarlo.

Para esto, en primer lugar se realiza el análisis estadístico del comportamiento transaccional de cada cliente de *Foodservice*. Esto se realiza tomando, para cada cliente, las fechas de cada uno de sus pedidos históricos con Agrosuper. Posteriormente, se calculan los días transcurridos entre cada pedido, obteniéndose así una lista con la frecuencia de pedidos históricos de cada cliente. Luego, se dejan únicamente los clientes que tengan 5 o más pedidos realizados históricamente, para evitar información no representativa (como un cliente que únicamente compró 3 veces de manera oportunista). Finalmente, a dicha lista se le obtienen 3 métricas de evaluación: la mediana, el promedio y el rango. El rango consta de el máximo valor menos el mínimo, por lo que, en este contexto, indica que tanto el cliente se escapa de su frecuencia regular de pedido, donde un rango 0 indica que jamás ha comprado con una frecuencia de compra diferente a la que acostumbra, y una rango alto implica que el cliente se ha escapado notablemente de su comportamiento regular de compra en algún momento. Para el rango, se obtiene el máximo como el percentil 90, y el mínimo como el percentil 10 de la lista de frecuencia de pedidos, con el fin de eliminar casos *outliers* de pedidos de clientes que por alguna razón en específico se demoraron más de lo normal en realizarlos. Así, se obtiene finalmente una tabla donde, para cada cliente, se tiene su mediana, promedio y rango de frecuencia de pedidos históricos.

Luego, para una de las 6 ventanas de tiempo a las cuales se les calculó las variables independientes, se les realizan 2 filtros. el primero, corresponde a dejar a los clientes que tengan al menos 3 pedidos en los últimos 3 meses. Este filtro permite obtener a los clientes recientemente activos de cada ventana de tiempo. El segundo filtro que se realiza es el de escoger el segmento de clientes en base a su comportamiento transaccional, obteniéndose de la tabla recién construida. Para esto, se tomarán los clientes de cada ventana de tiempo cuyo rango sea menor o igual a 7 días y su mediana de frecuencia de pedidos se encuentre entre 4 y 7 días. Este segmento corresponde a los clientes más recurrentes, estables y por lo tanto, fieles de Agrosuper, por lo que se escoge este segmento para el modelamiento.

Así, para cada ventana de tiempo, se filtran los datos dejando a los clientes más recurrentes, estables y activos, utilizando el análisis de comportamiento transaccional de estos.

5.2.5. Descripción y construcción de la variable respuesta

Continuando con la construcción del *dataset*, se procede a construir la variable respuesta, lo que implica etiquetar a los clientes que se obtuvieron de los filtros y segmentación, como fugados o no fugados. Para este fin, es importante, en primer lugar, definir o estandarizar un periodo de inactividad, con el cual evaluar la fuga de cada cliente. Como ya se mencionó anteriormente, no existe una instancia que asegure la fuga de un cliente en este mercado, debido a su naturaleza no contractual, por lo que se definirá un periodo de inactividad

estándar para todos los clientes en el segmento escogido anteriormente.

Para ver el impacto de las diferentes definiciones de fuga con respecto a la densidad o cantidad de clientes fugados y no fugados, se realiza un análisis de densidad en base a diferentes *flags* de fuga. Para esto, en primer lugar se toma la fecha del último pedido realizado dentro de la ventana de tiempo (si la ventana de tiempo es de marzo 2021 a febrero 2022, esta fecha corresponde al último pedido que realizó dentro de este periodo de tiempo). En segundo lugar, se toma la fecha del primer pedido realizado en el mes de evaluación (siguiendo con el ejemplo, sería el primer pedido realizado en marzo 2022). Luego, se calcula la cantidad de días transcurridos entre ambas fechas, obteniéndose la cantidad de días transcurridos desde el último pedido que realizó en la ventana de tiempo, la cual llamaremos días totales. Luego, para cada cliente de esta ventana de tiempo, se calcula la variable de *ratio* total, construida dividiendo la variable días totales sobre su mediana de frecuencia de pedidos obtenida anteriormente.

La variable *ratio* total entregará información sobre cuanto se demoró el cliente en pedir con respecto a lo que normalmente se demora en base a su comportamiento transaccional histórico. De esta forma, un cliente que tiene un ratio total de 2 significará que el cliente se demoró 2 veces lo que normalmente se demora en realizar un pedido (según su mediana). Luego, para definir que *ratio* total es el mejor para definir la fuga de un cliente, se realizó un gráfico de densidad de clientes en base al *ratio* total, con el fin de ver como se distribuye la cantidad de clientes a medida que aumenta esta variable. El gráfico en cuestión se ve de la siguiente forma:

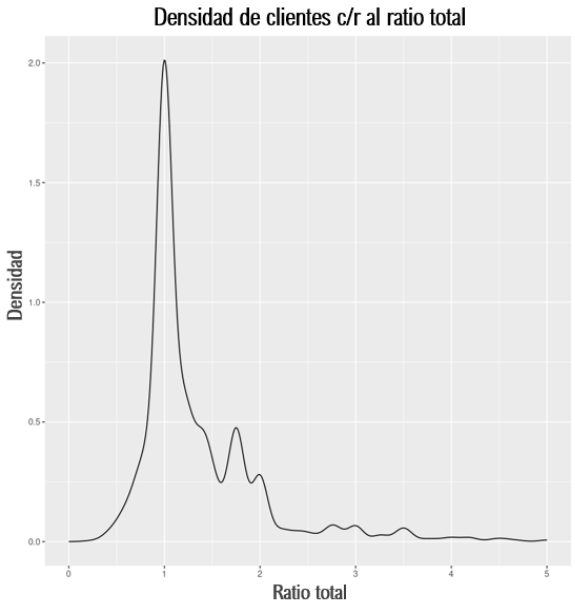

Figura 5.3: Densidad de clientes en base al *ratio* total.

El gráfico 5.3 permite entender la distribución de la cantidad de clientes en relación al *ratio* total, o bien, que tan anormal fue el tiempo que se demoraron en realizar el siguiente pedido en comparación con su comportamiento regular. Al observar el gráfico obtenido, se evidencian 2 claros declives de densidad de clientes, el primero siendo aproximadamente en el ratio 1.6, y el segundo aproximadamente en el ratio 2.2.

Luego, es importante darle sentido de negocio y no perder contexto. Como ya se mencionó anteriormente, la gran mayoría de los clientes de esta unidad de negocio realizan sus pedidos a través del preventa, quien se encarga de visitar los negocios, y dentro de todas sus responsabilidades, se encuentra tomar pedidos para los clientes. Luego, es común que el preventa no sea capaz de realizar la visita por algún imprevisto en particular, como puede ser que no se encontraba el encargado de pedidos a la hora de la visita, o el preventa no alcanzó por tiempo, por mencionar algunas. Esta situación significa que, si el cliente no pide por su propia cuenta a través de otro canal, como *call center* o la página web, tendrá que esperar a la siguiente visita del pre-venta para realizar otro pedido. Dicho esto, si un cliente tiene una mediana de frecuencia de compra de 7 días, y el pre-venta no realiza el pedido, significa que el cliente tendrá que esperar 14 días para hacer un pedido, lo que se traduce a un *ratio* total de 2, pero no necesariamente tuvo intenciones de fugarse.

Al ser este caso muy común dentro de esta unidad de negocio, se consideró el segundo punto de declive, correspondiente a un *ratio* total de 2.2. Dicho esto, se toma el punto de fuga como todos los clientes que tengan un *ratio* total igual o mayor a 2.5, mientras que se etiquetarán como no fugados los que tengan un *ratio* total menor a 2.5. Luego, para generar un modelo más certero y capaz de diferenciar entre ambas clases de mejor forma, no se consideran los clientes cuyo *ratio* total esté entre 2 y 2.5, con el fin de mejorar el entrenamiento y la capacidad de diferenciación del modelo.

De esta forma, se estandariza la definición de fuga para este segmento de clientes, en primer lugar obteniendo la diferencia de días entre el último pedido que realiza en la ventana de tiempo y el primer pedido que realiza en el mes de evaluación. Luego, se construye la variable de *ratio* total, la cual finalmente se utiliza para etiquetar al cliente, donde si esta variable es igual o mayor a 2.5, se etiquetará como fugado, y en caso de ser menor o igual a 2, se etiquetará como no fugado, donde los clientes entre 2 y 2.5 no se considerarán para los entrenamientos de los modelos con el fin de mejorar la capacidad de diferenciación del modelo.

Finalmente, se unen las 6 ventanas de tiempo en una sola gran base de datos que se utilizará para entrenar y validar los modelos predictivos, generada a partir de diferentes temporalidades pasadas, siguiendo la lógica del método *multiple-time slicing* definido anteriormente en la sección 2 (figura 2.2).

5.3. Análisis univariado

Siguiendo con el desarrollo del proyecto, se procede a la realización de los análisis univariados de las variables. Esta parte del proyecto permite, en primera instancia, entender la data, donde se revisan diversas medidas de distribución y tendencia de las variables, como también representaciones gráficas de distribución como histogramas y diagramas de cajón, donde es posible entender la forma en la que se distribuyen los datos, los cuartiles, la cantidad de *outliers* que existen, desviación de la mediana, por mencionar algunos.

Dentro de las métricas que se estudian para cada variable, se encuentran la mediana, desviación estándar, el promedio, el primer y tercer cuartil, la asimetría, la curtosis, entre

algunos otros. Estas métricas permiten estudiar y entender la distribución de los clientes con respecto a cada variable.

Posterior al análisis y entendimiento de la distribución de cada variable, se realiza la transformación de estas, en caso de ser necesario. La transformación que se realiza dependerá exclusivamente de la distribución inicial de los datos, por lo que es sumamente importante el estudio y análisis de las métricas de distribución mencionadas anteriormente, ya que aportan a tomar una decisión sobre que transformación realizar para mejorar la distribución de cada variable. A modo de ejemplo, se muestra a continuación el histograma de la variable de flexibilidad del cliente con respecto al precio, como también el diagrama de caja, previo a la transformación de datos.

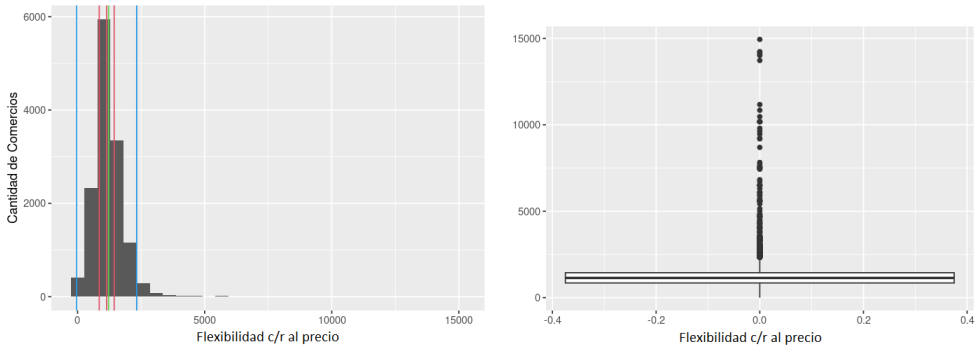

Figura 5.4: Histograma y diagrama de caja para la variable de flexibilidad con respecto al precio previo a la transformación univariada.

Ambos gráficos de la figura 5.4 permiten ver la distribución de la variable de flexibilidad del cliente con respecto al precio, en donde en el histograma se evidencia la tendencia a una disminución exponencial de clientes a medida que aumenta el valor, a partir del promedio; y en el diagrama de caja se observa la gran cantidad de datos *outliers*.

Este análisis entrega la información necesaria para tomar la decisión de realizar una transformación logarítmica a la variable, en conjunto con un truncamiento inferior, con el fin de normalizar la distribución de datos, reduciendo también la cantidad de *outliers*. Luego de dicha transformación, el histograma y diagrama de caja quedan de la siguiente forma:

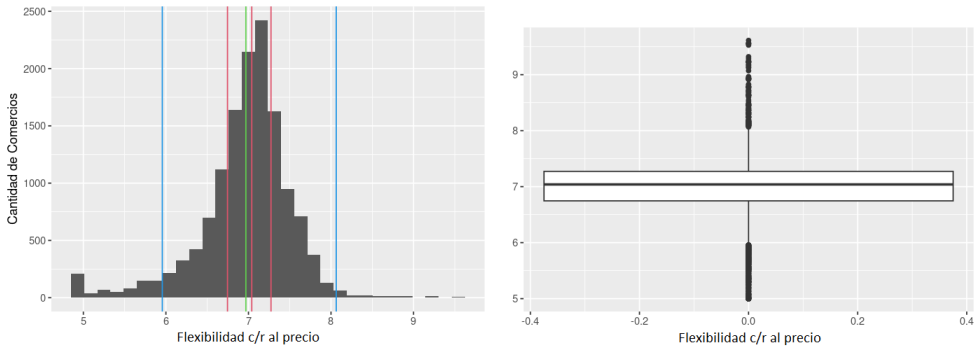

Figura 5.5: Histograma y diagrama de caja para la variable de flexibilidad con respecto al precio posterior a la transformación univariada.

Luego, las métricas de distribución de la variable, tanto previo a la transformación como posterior a la transformación, se muestran en la siguiente tabla comparativa:

Tabla 5.1: Tabla comparativa de métricas de distribución para la variable de flexibilidad de cliente con respecto al precio antes y después de la transformación univariada.

	Sin transformación	**Con transformación**
Mediana	1140	7.040
Desv. estándar	740	0.556
Promedio	1219	6.969
1° Cuartil	850	6.746
3° Cuartil	1440	7.273
Asimetría	5.681	-0.927
Curtosis	74	5.841

De esta forma, si bien varias métricas no son realmente comparables, puesto que se cambia la escala completamente a una logarítmica, es posible ver que la curtosis (en palabras simples, que tan pronunciada es la curva) disminuye, por lo que se 'aplana' la distribución, mientras que la simetría se acerca a 0 luego de la transformación, lo que implica que la distribución se volvió más homogénea en comparación con la distribución inicial.

Así, a cada variable se le realiza un análisis estadístico, y se le realiza una transformación en caso de ser necesaria, como se ejemplificó con la flexibilidad con respecto al precio. Los histogramas y diagramas de caja previo y posterior a la transformación de las variables transformadas en este proyecto son adjuntas en el anexo de este informe, en la sección A

5.4. Análisis multivariado

Posterior al análisis univariado y transformación estadística de las variables, se sigue con el análisis multivariado de los datos. Como bien dice su nombre, esta parte del proyecto permite analizar y estudiar la relación entre las variables, ya sea estudiar la linealidad de las variables predictivas con la variable respuesta, como la relación entre las variables predictivas, donde se pretende tratar la multicolinealidad de estas.

Dicho esto, lo primero que se realiza en esta etapa del proyecto es verificar y asegurar la linealidad de las variables con respecto a la variable respuesta, por lo tanto, se realiza una transformación a los datos utilizando *weight of evidence*, el cual se explica en la sección 3.1 de este informe. Es importante recalcar que **la transformación *WOE* de los datos se realiza exclusivamente para los datos que se utilizarán en el futuro para entrenar y validar los modelos de regresión logística.** La razón de esto último es debido a la construcción de las variables *WOE*, las cuales, además de asegurar la linealidad de los datos, son transformadas a una escala logarítmica, por lo que la regresión logística se beneficia notablemente de esta transformación, lo que la hace una técnica estadística muy popular para el tratamiento de datos en este tipo de modelos.

De esta forma, lo primero que se realiza, para los datos que se utilizarán para entrenar y

validar los modelos de regresión logística, es realizar la transformación *WOE* de cada variable. Para esto, se definen puntos de corte en los que se definirán los intervalos, de tal forma que se preserve la relación lineal entre cada variable independiente y la variable respuesta. Esto se hace escogiendo puntos de corte que definan umbrales tal que la probabilidad de evento positivo en cada umbral aumente o disminuya linealmente a medida que se aumente el umbral. Por otro lado, al escoger los puntos de corte, es importante tener en cuenta la cantidad de clientes en cada umbral definido, donde idealmente se debe mantener una cantidad de clientes similar en cada umbral definido por los puntos de corte.

Para ejemplificar la transformación *WOE* realizada, se presenta el caso de la variable de flexibilidad del cliente con respecto al precio, donde los umbrales generados y las transformaciones *WOE* quedan de la siguiente forma:

Tabla 5.2: Transformación WOE de la variable de flexibilidad del cliente con respecto al precio.

	Umbral	**No fugados**	**Fugados**	**Probabilidad**	**WOE**
1	$[-\infty$,6.4)	1314	275	17,31	0.463
2	[6.4,6.7)	1332	225	14.45	0.249
3	[6.7,6.9)	1456	213	12.76	0.105
4	[6.9,7.1)	2641	314	10.63	-0.102
5	[7.1, ∞)	5355	566	9.56	-0.219

De esta forma, la tabla 5.2 muestra los umbrales definidos por los puntos de corte escogidos, como también la cantidad de casos positivos y negativos (fugados y no fugados respectivamente) en cada umbral, en conjunto con la probabilidad de evento positivo y el valor *WOE* construido. Como se explica en la sección 3.1, se observa que el valor *WOE* del primer umbral es el mayor de todos, lo que implica que tiene una mayor capacidad de separar eventos positivos de los negativos. También, es posible evidenciar la linealidad con la que disminuye la probabilidad de evento positivo a medida que se aumenta el rango. Esto permite entender la relación de esta variable con la variable respuesta, como también la capacidad predictiva de esta variable, donde mientras más flexible es el cliente con respecto al precio de los productos que acostumbra a pedir, existe una menor probabilidad de que este se fugue, lo que tiene sentido a nivel de negocio.

A continuación de la transformación *WOE*, se procede a analizar la multicolinealidad de los datos, lo que significa revisar la correlación que existe entre las variables. Esto se realiza con el fin de eliminar información redundante, o bien, que no aporta a la varianza de la base de datos, puesto que si dos variables están altamente correlacionadas, probablemente ambas explican el mismo comportamiento, por lo que no es necesario tener ambas en la base de datos.

Para este fin, en primer lugar, se obtiene la matriz de correlación entre las variables que componen la base de datos, la cual se muestra a continuación:

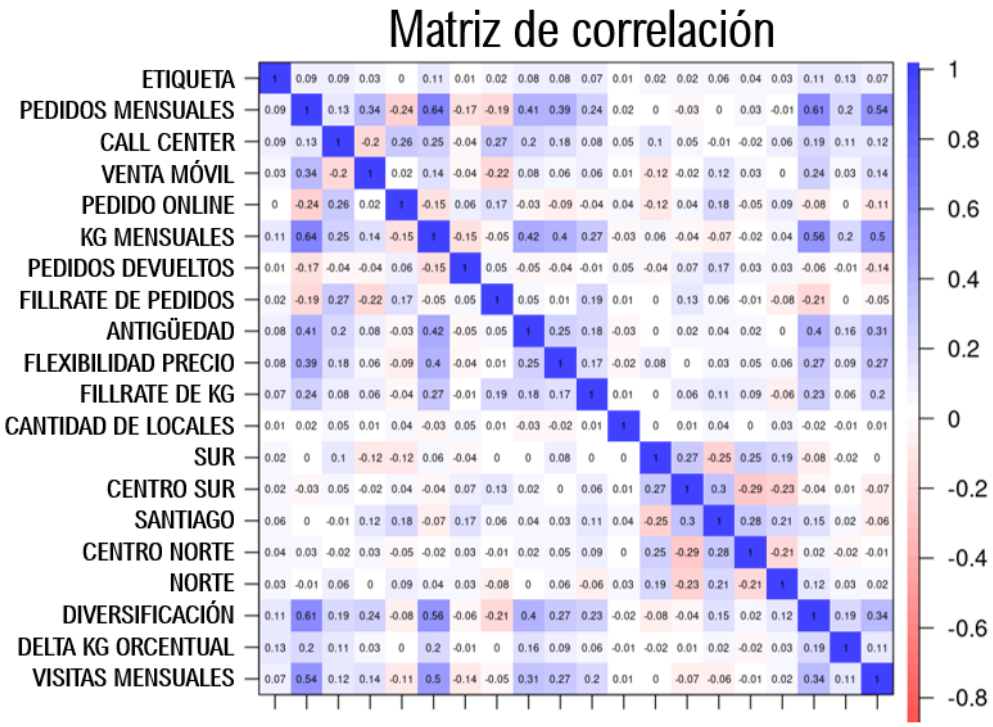

Figura 5.6: Matriz de correlación de las variables.

Luego, la matriz mostrada en la figura 5.6 permite identificar las variables que poseen la mayor correlación entre ellas. Si bien no existe ningún par de variables que posean una correlación extremadamente alta, las variables más correlacionadas son los pedidos mensuales, los kilogramos pedidos mensuales, la diversificación del cliente a la hora de comprar, la antigüedad del cliente y la cantidad de visitas mensuales.

Luego de identificar las variable más correlacionadas, estas se agrupan en una pequeña base de datos a la cual se le aplica *PCA*. Este algoritmo de reducción de dimensionalidad (explicado en la sección 3.2.1), permite reducir la dimensionalidad de un *dataset*, minimizando la pérdida de información o varianza y eliminando la multicolinealidad existente entre las variables a las que se le aplica este algoritmo, puesto que las nuevas variables, llamadas componentes principales, resultan de una combinación de las variables en cuestión.

Siguiendo con esto, luego del *PCA* realizado a las 5 variables mencionadas anteriormente, se procede a obtener las 3 principales componentes, que explican el 90 % de la varianza de los datos. De esta forma, además de reducir la dimensionalidad del *dataset* en 2 variables, se logra eliminar la correlación entre estas 5 variables, reemplazándolas finalmente en el *dataset* por las 3 componentes principales recién obtenidas.

Para finalizar el análisis multivariado, se realiza una estandarización del nuevo *dataset*, donde se obtiene la base de datos que se utilizará para entrenar y validar los modelos futuros.

5.5. Desarrollo, entrenamiento y validación de los modelos

Una vez finalizado tanto el análisis univariado y transformación estadística de la data como el análisis multivariado y tratamiento de multicolinealidad, se transita a la última etapa del proyecto, correspondiente al desarrollo, entrenamiento y validación de los modelos predictivos.

Para esto, se sigue una serie de pasos generales para los 3 modelos, el cual se realiza por n iteraciones dependiendo del modelo entrenado, con el fin de generar múltiples instancias de entrenamiento y asegurar la obtención del mejor modelo posible. Lo primero que se realiza, para el entrenamiento de un modelo, es dividir la data proveniente del análisis multivariado (12.098 clientes no fugados y 1.593 clientes fugados), **de manera aleatoria**, en 80/20, lo que significa que el 80 % de la data original será destinada a entrenar el modelo (9.678 clientes no fugados y 1.274 clientes fugados), mientras que el restante 20 % se destinará a la validación de este (2.420 clientes no fugados y 319 clientes fugados). Es sumamente importante validar con un conjunto aparte, ya que el modelo aprende de los datos de entrenamiento, por lo que calcular métricas de desempeño en base al conjunto de entrenamiento no representa el rendimiento real del modelo en lo absoluto.

Una vez se separa la data en 80/20, se realiza un proceso de *downsampling* en el conjunto de entrenamiento. *Downsampling* es una técnica que se utiliza para balancear las clases antes de entrenar. Es importante tener un balance de clases al momento de entrenar, ya que de lo contrario, el modelo se sobreentrena con datos de una clase en particular, lo que se traduce en un rendimiento usualmente bajo a la hora de validar, ya que el modelo tiende a sesgarse, prediciendo eventos de la clase mayoritaria por sobre la minoritaria. Luego, *downsampling* reduce la clase mayoritaria (en este caso, clientes no fugados) eliminando de manera aleatoria, datos de la clase mayoritaria (pasando de 9.678 a 1.274 clientes no fugados). Así, al conjunto de entrenamiento conformado por el 80 % de la data original, se le realiza un *downsampling*, obteniéndose un nuevo conjunto de entrenamiento con la misma cantidad de clases (misma cantidad de clientes fugados y no fugados).

5.5.1. Ajuste de intercepto - regresión logística

Una vez hecho el *downsampling*, y obtenido el nuevo conjunto de entrenamiento, se procede a entrenar los modelos predictivos. Para este fin, es relevante separar el caso el caso de la regresión logística con respecto a los otros modelos, la cual posterior al entrenamiento, pasa por un proceso de **ajuste de intercepto**.

Este proceso se realiza puesto que al ser entrenado con la misma cantidad de casos positivos y negativos (posterior al *downsampling* del conjunto de entrenamiento), y debido a la naturaleza del modelo de regresión, este termina el entrenamiento "pensando" que normalmente el 50 % de los datos son fugados y el otro 50 % son casos de clientes no fugados, lo cual en la realidad es erróneo, ya que existen muchos más casos de clientes no fugados que clientes que si lo están. Para corregir esto, se realiza el ajuste de intercepto.

Para esto, en primer lugar se entrena un modelo de regresión logística con el *dataset* de entrenamiento modificado (*downsampling*), con lo cual se obtienen las variables significativas del modelo y sus coeficientes asociados, donde las variables significativas son las características que, según el entrenamiento, son las más útiles para la predicción de una clase, y los coeficientes asociados representan el nivel de importancia que tiene cada variable en la decisión de la clase de un cliente. Luego, se obtiene el primer coeficiente, correspondiente al intercepto de la regresión recién entrenada, y se modifica el intercepto siguiendo la formula a continuación:

$$\beta^* = \beta_0 - \log\left(\frac{N_0}{N_1}\right) \tag{5.1}$$

donde B_0 es el intercepto que resulta del entrenamiento inicial de la regresión logística, B^* es el intercepto ajustado, N_0 es la cantidad de clientes etiquetados como no fugados y N_1 es la cantidad de clientes etiquetados como fugados (en el *dataset* original). Luego, la probabilidad de clase π_j del cliente j quedará definida como:

$$\pi_j = \frac{1}{1 + exp(-(\beta^* + \sum_{i=1}^{n} \beta_j X_{ij}))} \tag{5.2}$$

donde B_j son los coeficientes asociados a las variables significativas X_{ij} del j-ésimo cliente. De esta forma, una vez ajustado el intercepto, se tiene el modelo entrenado final.

5.5.2. Análisis del umbral de decisión

Volviendo con el desarrollo y entrenamiento de los 3 modelos estadísticos, una vez se entrenaron estos, lo primero que se realiza es obtener las probabilidades de predicción del conjunto de validación. Luego, se realiza un análisis de punto de corte, para definir cual es el umbral de decisión óptimo que maximiza el rendimiento del modelo.

El umbral de decisión representa la probabilidad desde la cual se etiquetará a un cliente como fugado. Para ejemplificar, si el umbral de decisión es 75 %, y el modelo entrenado, para cierto cliente, predice que un cliente es fugado con un 80 % de probabilidad, será etiquetado como fugado, pero si otro cliente es predicho como fugado con una probabilidad de 65 %, este último será etiquetado como no fugado. En palabras sencillas, representa la tolerancia de fuga.

Para escoger el mejor umbral posible, dada una lista de probabilidades predichas, se construye una función auxiliar, la cual recibe la lista de probabilidades predichas por el modelo y la lista de etiquetas reales del conjunto de validación. Luego, se recorren todos los umbrales posibles desde el 0 % al 100 % avanzando en 1 %. Para cada umbral, se etiquetan las probabilidades de cada cliente obtenidas por el modelo de la siguiente forma:

$$Y_j \rightarrow \begin{cases} \text{Fugado} & \text{si } \pi_j > U \\ \text{No fugado} & \text{si } \pi_j > U \end{cases} \tag{5.3}$$

Donde Y_j corresponde a la etiqueta del j-ésimo cliente, π_j la probabilidad predicha para el j-ésimo cliente y U el umbral de decisión.

Luego, al tenerse la lista con predicciones del conjunto de validación, y la lista con las etiquetas reales de validación, se obtiene el valor AUC (por sus siglas en inglés *area under the curve*). Cada vez que el valor AUC mejore, se guarda el umbral utilizado. De esta forma, para una lista de probabilidades predichas del modelo, se utilizan 100 umbrales diferentes, donde para cada umbral, se etiquetan los datos, se comparan con las etiquetas reales, se calcula el AUC, y se obtiene el umbral que maximiza dicha métrica.

Es importante mencionar que, para este proyecto, siempre se buscará maximizar la métrica de desempeño AUC, ya que esta representa la media entre la sensibilidad y especificidad del modelo, donde la sensibilidad representa la tasa de clasificación de verdaderos positivos y la especificidad representa la tasa de clasificación de verdaderos negativos. En palabras simples, es una métrica que representa el balance entre que tan bien clasifica los clientes fugados y que tan bien clasifica los clientes no fugados.

A diferencia de otros proyectos de predicción y clasificación, en este no se busca maximizar el *accuracy*, puesto que no es representativo del rendimiento real del modelo, esto debido al alto desbalance de clase que existe en la muestra de validación. Para entender esto de mejor manera, supongamos que se tiene una muestra de validación con 800 clientes etiquetados como no fugados, y 200 clientes etiquetados como fugados. Un modelo que clasificara todos los clientes como no fugados, tendría un *accuracy* de 80 %, lo cual evidentemente no representa el desempeño real del modelo, que solo aprendió a etiquetar clientes no fugados. Este modelo tendrá un AUC sumamente bajo, puesto que si bien la tasa de clasificación de verdaderos negativos fue alta (100 %), la tasa de clasificación de verdaderos positivos fue nula, por lo que tendrá un AUC de 50 %, lo cual equivale a la probabilidad de etiquetar a un cliente en base a la cara de una moneda.

5.5.3. Validación de los modelos e iteración

Finalmente, una vez se obtuvo el umbral de decisión que maximiza el rendimiento del modelo entrenado, se etiquetan las probabilidades predichas el modelo para cada cliente del conjunto de validación, y se comparan con las etiquetas reales del conjunto de validación, con el fin de obtener las métricas de validación.

En cuanto a las métricas que se obtienen, se encuentra en primer lugar el valor AUC ya explicado, en conjunto el valor KS. La métrica KS (Kolmogórov-Smirnov) es una prueba de bondad de ajuste que permite verificar si las puntuaciones de la muestra siguen o no una distribución normal. Luego, se obtiene la matriz de confusión, la cual evidencia la cantidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos obtenidos en el conjunto de validación. De esta matriz se desprenden métricas como el *accuracy*, *F1-score*, sensibilidad, especificidad, *recall*, por mencionar algunos. Finalmente, se obtiene la curva *ROC* del modelo. Este conjunto de métricas permiten evaluar y comparar el desempeño de los diferentes modelos entrenados, y permiten tomar una decisión en cuanto a cual es el mejor modelo, o bien el que más se adecua a las necesidades del problema.

Como se mencionó anteriormente, en este proyecto, dada la naturaleza de la cardinalidad de las clases, se enfatiza en la mejora del AUC. Al comenzar los entrenamientos de los modelos, se instancia un AUC inicial igual a cero (iteración 1). De esta forma, al llegar a este punto del proceso de validación de los modelos entrenados, se compara el AUC obtenido con el guardado, y si el recién obtenido es mejor que el guardado anteriormente, se guarda dicho AUC, en conjunto con el modelo que obtuvo dicho rendimiento, de lo contrario, simplemente se vuelve a iterar. Esto permite obtener el mejor modelo posible al final de las n iteraciones.

5.5.4. Validación *out time*

Finalmente, una vez se evalúa si el AUC mejoró o no, guardando el modelo en el caso positivo, se pregunta si se sigue iterando o no. Como se mencionó al principio de esta sección, esta serie de pasos se realiza n veces, con el fin de desarrollar y entrenar múltiples modelos para cada tipo de modelo, y asegurar la obtención de aquel que maximice el rendimiento, o bien, maximice el AUC.

En caso de seguir iterando, se vuelve a la división aleatoria de los datos, donde un modelo vuelve a ser entrenado y evaluado con un *dataset* aleatorio diferente cada vez. En caso de terminarse las iteraciones, se procede a la validación *out time*, que permitirá evaluar la estabilidad temporal de los modelos obtenidos, y de esta forma, evaluar el desempeño real de estos.

La validación *out time*, como se ve en la figura 5.1, corresponde a una ventana de tiempo actual, absolutamente independiente de la base de datos utilizada para el entrenamiento y validación de los modelos creados, la cual permite evaluar la estabilidad temporal y el correcto funcionamiento de los modelos. Al ser una ventana de tiempo independiente de las otras, permite visualizar cual será el desempeño de los modelos a la hora de utilizarlos en el dia a dia, ya que si bien la validación normal entrega información de la efectividad de los modelos, al estarse validando con datos de los mismos meses con los que fue entrenado, es posible que el desempeño que se obtiene en este conjunto no sea precisamente representativo con respecto a como se desempeñará en la realidad. Además, al ser una ventana actual también entrega información sobre como funcionará el modelo hoy en día, y no hace un par de meses atrás. Por estas razones es que la validación *out time* es sumamente importante, ya que es el real indicador de si los modelos son implementables en el futuro o no.

Para la construcción del conjunto de validación *out time*, es importante mencionar que la mayoría de las variables no fueron construidas a partir de los 12 meses de historia, si no del último mes. Para ejemplificar esto, la variable de pedidos mensuales, a diferencia de la base de datos de entrenamiento y validación, no corresponde a la mediana de pedidos mensuales del ultimo año, si no más bien a los pedidos del último mes. Esto se realizó para obtener el comportamiento actual de los clientes, y no el comportamiento promedio del último año, principalmente por dos motivos: permite caracterizar a los clientes con respecto a su comportamiento más actual y la alimentación del modelo, una vez implementado, será de esta forma, tomando los datos del último mes para predecir el siguiente.

Finalmente, la construcción del conjunto de validación *out time* sigue exactamente la misma metodología que el conjunto de entrenamiento, en donde se reutilizan los *WOE* y las

transformaciones estadísticas aplicadas a los datos del conjunto de entrenamiento, puesto que los datos deben tener las mismas escalas y formato para ser consistentes con la predicción. Se construyen las características, se filtran y etiquetan los clientes, se realizan las mismas transformaciones univariadas que a las variables del conjunto de entrenamiento, se utiliza el mismo *WOE* que el conjunto de entrenamiento, se aplica *PCA* a las mismas variables que el conjunto de entrenamiento, y finalmente se estandarizan los datos.

De esta forma, se utiliza este conjunto de validación *out time*, que permitirá conocer la estabilidad en cuanto al desempeño de los mejores modelos obtenidos, y de esta forma elegir al modelo más estable y con mejor rendimiento de los 3.

El diagrama que representa el proceso de desarrollo, entrenamiento y validación de los modelos se muestra a continuación:

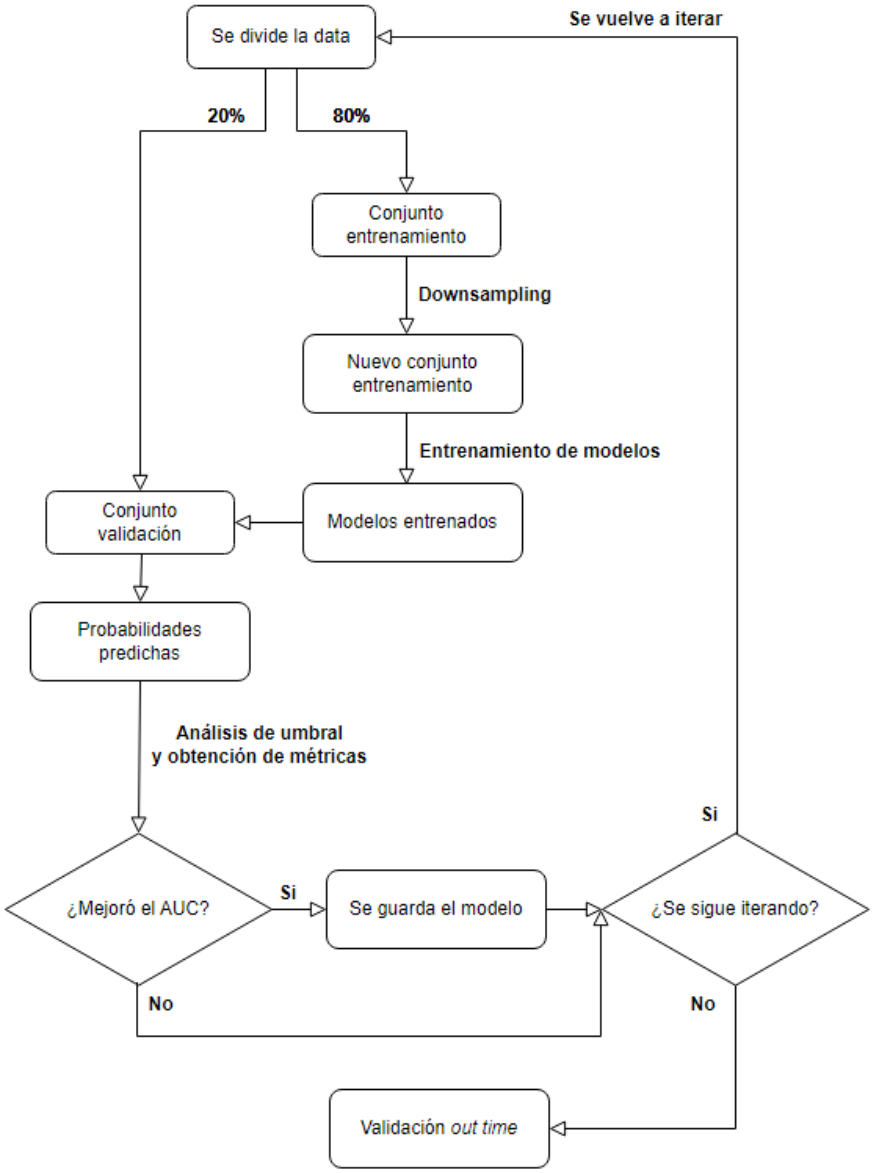

Figura 5.7: Diagrama de desarrollo, entrenamiento y validación de los modelos predictivos.

Capítulo 6

Resultados y análisis

A continuación, se mostrarán y analizarán los resultados obtenidos para cada uno de los mejores modelos desarrollados y entrenados en este proyecto, correspondiente a los modelos de regresión logística, *random forest* y *support vector machine*.

6.1. Regresión logística

Los primeros resultados que se obtienen son los correspondientes al mejor modelo obtenido para la regresión logística. Para este modelo, se desarrollan y entrenan 600 modelos, obteniéndose el que maximizó el AUC en validación.

Para el modelo de regresión obtenido, se tiene una función de probabilidad de clase mostrada en la ecuación 5.2, donde las variables significativas y coeficientes asociados a estas se muestran en la siguiente tabla:

Tabla 6.1: Variables significativas y coeficientes asociados para la función de probabilidad de clase de la mejor regresión logística obtenida.

Variable significativa X_i	**Coeficiente asociado β_i**
Diferencia porcentual de kg	0.7503
Pedidos *call center*	0.6664
Componente N°1 PCA	0.6353
Pedidos venta móvil	0.4716
Zona sur	0.4564
Zona Santiago	0.4428
Flexibilidad c/r al precio	0.3569
Zona norte	-0.2017
Intercepto	**-3.7605**

Es importante mencionar que la variable significativa llamada "Componente N°1 PCA" mostrada en la tabla 6.1 hace referencia a la primera componente principal obtenida al aplicar PCA a las variables de pedidos mensuales, kilogramos pedidos mensuales, diversificación del

cliente, antigüedad del cliente y la cantidad de visitas mensuales, tal como se menciona en la subsección 5.4 de análisis multivariado

De la tabla de variables y coeficientes 6.1, es posible evidenciar el impacto que tienen las diferentes variables significativas en la predicción de la permanencia de un cliente. Esto es una característica distintiva de la regresión logística, la cual permite entender de manera más directa que otros modelos, cuales son las variables que más influyen en la predicción de una clase, y en específico, cuanto influyen, lo que a su vez permite darle un enfoque de negocio más directo al modelo, a diferencia de otros como *support vector machine* por ejemplo, que es evidentemente más complejo y definitivamente menos interpretable.

Es posible evidenciar en la tabla 6.1 que la variable que más impacta en la probabilidad de predicción de fuga de la regresión logística obtenida es la variable de diferencia porcentual mensual de kilogramos pedidos. A nivel de negocio, esto tiene sentido, ya que esta variable caracteriza al cliente en base a cuantos kilogramos de producto pidió este mes en comparación con el anterior, por lo que un porcentaje alto indica que su comportamiento anual, en mediana, fue ir pidiendo cada mes más kilogramos que el anterior, lo que se puede traducir en una baja probabilidad de fuga (se ha ido fidelizando con la empresa). Por el contrario, si un cliente tiene una tendencia anual de cada mes ir pidiendo menos producto (porcentaje negativo), es más propenso a fugarse, ya que es probable que los kilogramos que está dejando de pedir, los está comprando a la competencia.

Por otro lado, se puede ver en la tabla 6.1 que la segunda variable que más impacta en la predicción de clase de un cliente corresponde a la mediana de pedidos mensuales por el canal *call center*. La razón de esto puede ser que los clientes que realizan pedidos por este canal son más constantes y recurrentes para pedir, ya que no deben esperar al encargado de pre-venta para realizar pedidos de forma presencial (canal venta móvil), si no que los pedidos los realizan ellos mismos mediante el canal *call center*, por lo que no sufren contratiempos que retrasan los tiempos de sus pedidos, como lo puede ser el incumplimiento de visita por parte del preventa.

De manera análoga, es posible ver que la variable que menos impacto tiene, dentro de las variables significativas que considera el mejor modelo de regresión logística entrenado, es la variable dicotómica zona norte, la cual indica si un cliente pertenece a la zona norte o no. Acá es sumamente interesante notar que, si bien tiene un impacto relativamente menor en relación con las otras, además tiene un valor negativo. Esto significa directamente que esta variable aumenta la probabilidad de fuga, en caso de ser positiva (zona norte igual a 1), lo que significa que, para este modelo, que un cliente pertenezca a la zona norte aumentará la probabilidad de fuga de este. Esto se explica por como se construye la probabilidad de predicción de clase en la ecuación 5.2. Al tener un coeficiente negativo, y ser una variable dicotómica, si el valor de esta variable es 1, el exponencial se hace más grande, y por lo tanto, la probabilidad de fuga aumenta.

Siguiendo con los resultados obtenidos para el modelo de regresión logística, se obtienen tanto la matriz de confusión de este modelo para el conjunto de validación, como la matriz de confusión porcentual para este mismo conjunto, las cuales se muestran a continuación:

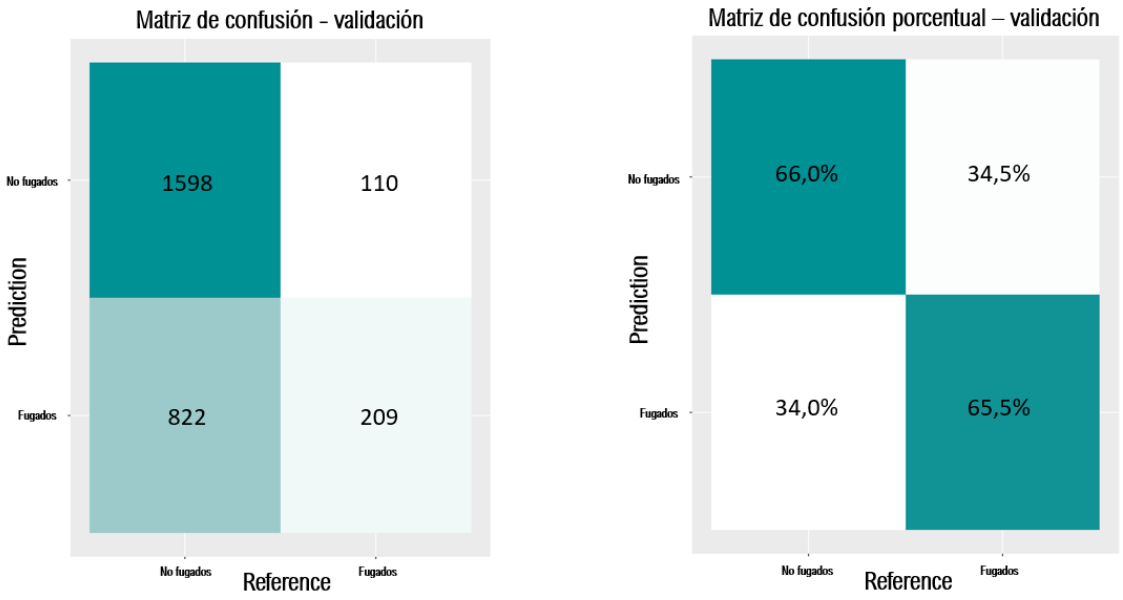

Figura 6.1: Matrices de confusión para la regresión logística en el conjunto de validación

En la matriz de confusión de la figura 6.1, es posible evidenciar la capacidad de predicción del modelo para cada tipo de clase, donde se explicita la cantidad de clientes etiquetados correctamente para cada clase (verdaderos positivos y verdaderos negativos), como también la cantidad de clientes etiquetados erróneamente (falsos positivos y falsos negativos). Para este caso en particular, es posible notar que el modelo resultó bastante consistente para ambas clases, donde la sensibilidad y especificidad del modelo en el conjunto de validación resultaron sumamente similares, con valores de 65,5 % y 66,0 % respectivamente.

Es importante recordar que la sensibilidad corresponde a la tasa de acierto de verdaderos positivos, lo que se traduce en palabras sencillas como la capacidad de predecir correctamente clientes fugados, mientras que la especificidad corresponde a la capacidad de predecir correctamente clientes que no se fugaron, o bien, verdaderos negativos. Dicho esto, la regresión, para el conjunto de validación, demostró un rendimiento bastante por sobre la aleatoriedad (50 % dado que es una clase binaria), que es naturalmente el mínimo esperado para un modelo de esta naturaleza.

Por otro lado, existe una pequeña diferencia entre la sensibilidad y especificidad del modelo, lo que se atribuye netamente en la aleatoriedad del entrenamiento, donde es probable que el modelo haya aprendido a diferenciar levemente mejor a los clientes no fugados de los fugados gracias al conjunto aleatorio con el que fue entrenado.

Luego, se evidencia la importancia del ajuste del intercepto y el análisis del umbral de decisión, donde en caso de no realizarse, la regresión hubiese adquirido una tendencia a la predicción de clientes no fugados por sobre los fugados, lo que a su vez se hubiese traducido en una alta especificidad y una baja sensibilidad del modelo de regresión.

Luego, nuevamente se muestran ambas matrices de confusión, para el conjunto de validación *out time*:

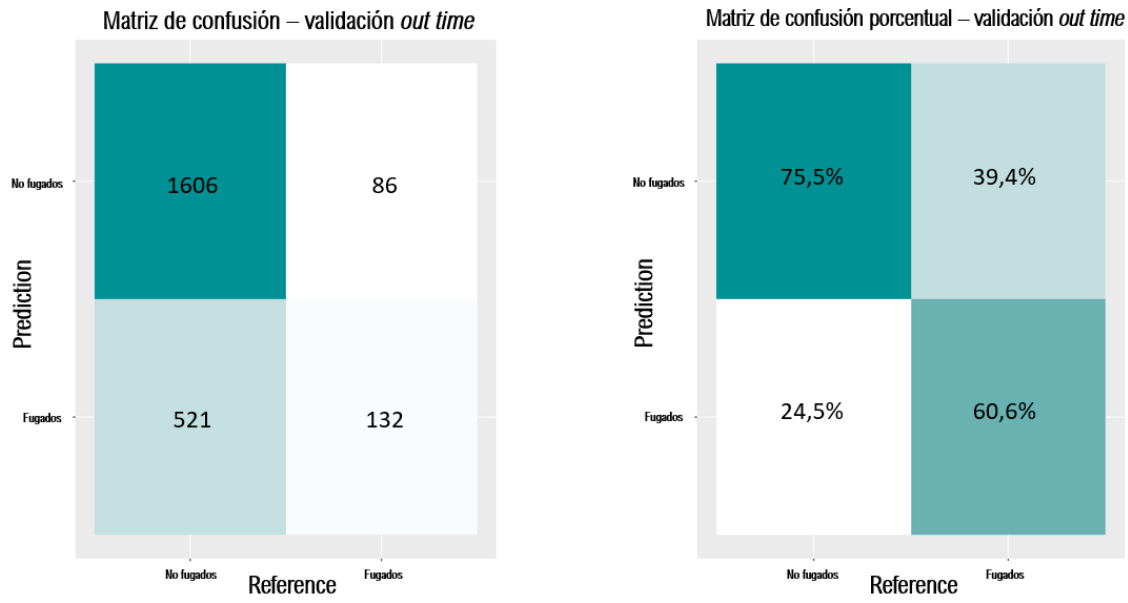

Figura 6.2: Matrices de confusión para la regresión logística en el conjunto de validación *out time*

En las matrices de la figura 6.2, es posible ver que para el caso de la validación *out time*, comparando con las matrices de validación de la figura 6.1, el modelo mejoró en un 9,5 % la capacidad de clasificación de verdaderos negativos (especificidad), mientras que en cuanto a su capacidad de clasificación de verdaderos positivos (sensibilidad), esta disminuyó en un 4,9 %. Una de las principales razones de estos cambios puede ser la manera en que se construyeron los datos de validación *out time*, donde para múltiples variables, como ya se explicó en la sección 5.5.4, se consideró únicamente el comportamiento del cliente en el último mes, con el fin de obtener el panorama actual del cliente y poder predecir su permanencia en la empresa en base a sus últimos comportamientos.

En cuanto al rendimiento general de la regresión en este conjunto, se esperaba que este disminuyera en comparación con el conjunto de validación, ya que como se explicó anteriormente, el conjunto de validación *out time* es construido en base a una ventana de tiempo totalmente diferente a las ventanas de tiempo que se utilizan para entrenar el modelo, y por lo tanto es información que el modelo nunca ha visto. No obstante, es posible ver que, si bien disminuyó un poco la sensibilidad, lo cual estaba contemplado, la especificidad aumentó casi en un 10 %, lo cual es señal de un modelo robusto y estable en el tiempo.

Por último, se obtienen las curvas *ROC* del modelo tanto para el conjunto de validación como el conjunto de validación *out time*. las cuales se muestran a continuación:

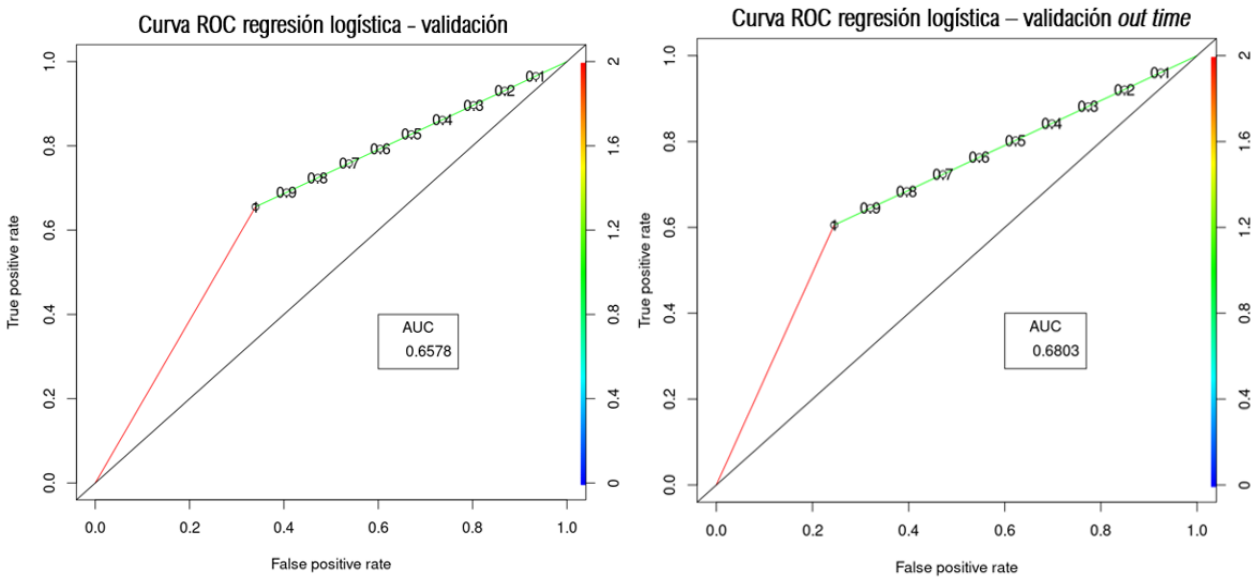

Figura 6.3: Curvas ROC para la regresión logística tanto en validación como en validación *out time*

Las curvas mostradas en la figura 6.3 transmiten la mejora que se logra en cuanto al AUC de ambos modelos desde el conjunto de validación al conjunto de validación *out time*. El valor AUC representa el área bajo la curva ROC, y es precisamente el promedio aritmético entre la especificidad y la sensibilidad del modelo. Como se mencionó en un principio, para la naturaleza del modelo y el desbalance de clases existente, se busca mejorar esta variable, puesto que no se ve alterada por el desbalance de clases, y por lo tanto permite entender el rendimiento real de los modelos evaluados sin ser perjudicado por la diferencia de cardinalidad.

Luego, es posible ver en la figura 6.3 que el AUC del conjunto de validación *out time* aumenta aproximadamente en un 2,2 % con respecto al conjunto de validación, lo que corresponde al balance porcentual entre el aumento de 9,5 % de la especificidad y la baja de 4,9 % de la sensibilidad entre la validación y la validación *out time*. Este aumento, como ya se mencionó, ratifica la estabilidad temporal de la regresión logística obtenida, abriendo la oportunidad a ser implementada a futuro.

6.2. *Random forest*

Los siguientes resultados que se muestran corresponden a los del modelo *random forest*. Para este modelo, al igual que la regresión logística, se realizaron 600 iteraciones de entrenamiento y validación, en la cual se obtuvo el mejor modelo en base al AUC de validación obtenido.

La importancia de las variables, según el mejor modelo *random forest* obtenido, son las siguientes:

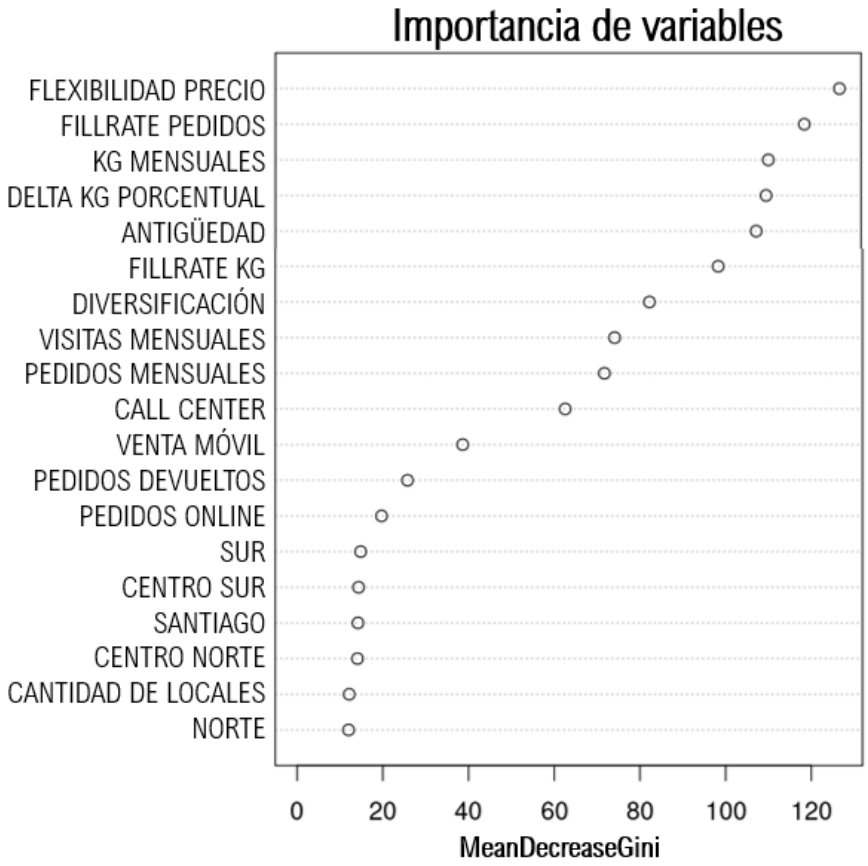

Figura 6.4: Importancia de las variables según mejor *random forest* obtenido.

Luego, se puede observar en la figura 6.4 que la variable más importante a la hora de etiquetar a un cliente como fugado o no fugado es la flexibilidad con respecto al precio. Esto tiene sentido de negocio, ya que esta variable indica que tan dispuesto está un cliente a pagar más por un producto que compra recurrentemente, por lo que es natural pensar que un cliente más flexible al precio, y por lo tanto más dispuesto a comprar un mismo producto más caro, es un cliente fidelizado que prioriza la empresa y la calidad de sus productos por sobre el precio de este, por lo que es menos probable que se cambie a la competencia.

De la misma forma, la segunda variable más significativa para este tipo de modelo es el *fillrate* de pedidos. Esta variable explica el porcentaje de cumplimiento que tiene la empresa Agrosuper con los pedidos de sus clientes, donde un porcentaje bajo significa que la minoría de los pedidos que realizó un cliente realmente le llegaron. Al realizar la ruta con el preventa (sección 5.1), se logró ver que una de las causas más común de fuga que mencionan los clientes tiene que ver con la gestión de los pedidos, y con esto se refieren a pedidos que llegan en mal estado, tarde o simplemente no llegan. Dicho esto, es esperable que un cliente sea más propenso a fugarse si tiene un *fillrate* de pedidos bajo, ya que o bien debe comprar sus productos de una u otra forma puesto que no tiene capacidad de refrigeración, o bien perdió la confianza con la empresa y decidió cambiarse a la competencia.

Por otro lado, las variables que menos explican la fuga de un cliente, o las variables menos importantes según el modelo *random forest* obtenido, son principalmente las 5 localizaciones de los clientes, las cuales son variables dicotómicas *one hot encoded* que indican a que zona

pertenece cada cliente, siendo 1 el caso positivo y 0 el caso negativo. El modelo de *random forest* indica que no existe una fuerte relación entre estas variables y el hecho de que un cliente vaya a fugarse o no.

Luego, las matrices de confusión normal y porcentual para el conjunto de validación de este modelo se muestran en la siguiente figura:

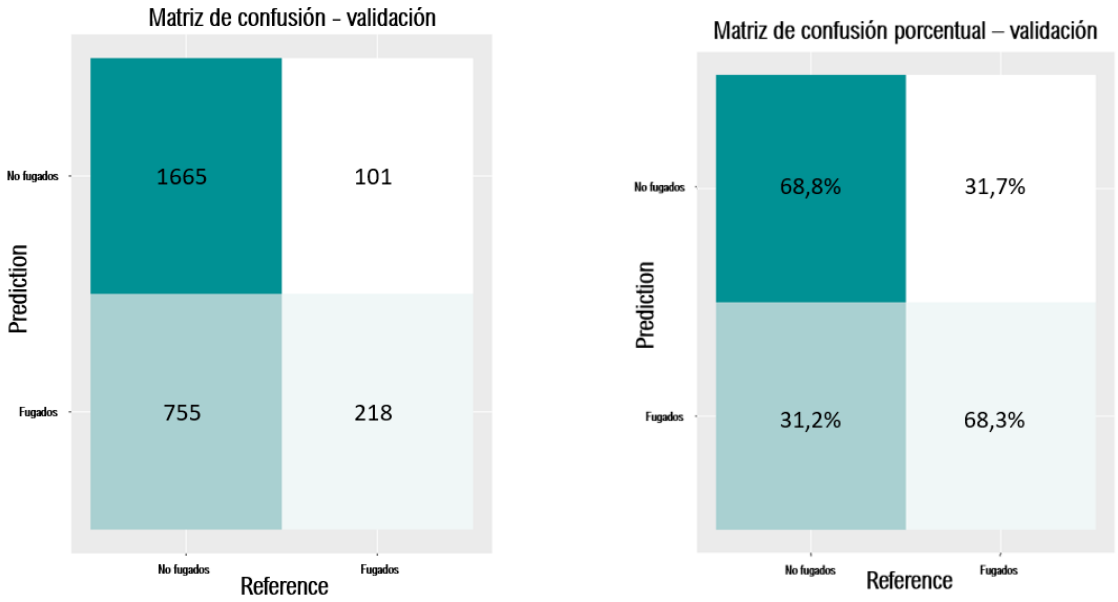

Figura 6.5: Matrices de confusión para *random forest* en el conjunto de validación

En este caso, es posible ver en la figura 6.5 que al igual que para la regresión logística, se tiene un mejor desempeño en cuanto a la tasa de clasificación de verdaderos negativos (especificidad) por sobre la tasa de clasificación de verdaderos positivos, superando a esta última por solo 0,5 %. Nuevamente, esta pequeña diferencia se puede atribuir a la aleatoriedad del conjunto de entrenamiento con el que se obtuvo el mejor modelo de *random forest*.

Por otro lado, es importante ver que el mejor modelo de *random forest*, en el conjunto de validación, tiene un mejor desempeño que el mejor modelo de regresión logística, tanto en especificidad como sensibilidad. En este caso, no es atribuible a la aleatoriedad de los datos, puesto que para ambos tipos de modelo se realizaron 600 entrenamientos con *datasets* diferentes, y el mejor modelo de *random forest* fue superior que el mejor modelo regresión logística.

Es posible que este modelo sea capaz de diferenciar con mejor desempeño que el modelo de regresión debido a como se entrena el modelo en si, donde para *random forest* se genera un proceso llamado *bagging*, lo que permite entrenar una gran cantidad de árboles de decisión, cada uno enfocado a clientes diferentes, ya que utiliza *bootstrap* (figura 3.1) para generar diversos *datasets* de entrenamiento con data repetida, y cada uno enfocado a diferentes características, ya que a cada árbol se le escogen características al azar para ser entrenado. Luego, el dato predicho será el voto mayoritario de los n árboles entrenados. Esto entrega un modelo sumamente robusto y de bajo sesgo, lo que puede verse favorecido al clasificar datos provenientes del mismo *dataset* de donde se obtuvo el *dataset* de entrenamiento, como lo es el de validación.

No obstante, es importante mencionar que la consistencia entre la especificidad y la sensibilidad del modelo, en el conjunto de validación, permite entender en este caso la indiferencia que presentan los modelos de *random forest* al desbalance de clases, donde para este modelo no fue necesario ajustar ningún parámetro para obtener los resultados que se obtuvieron.

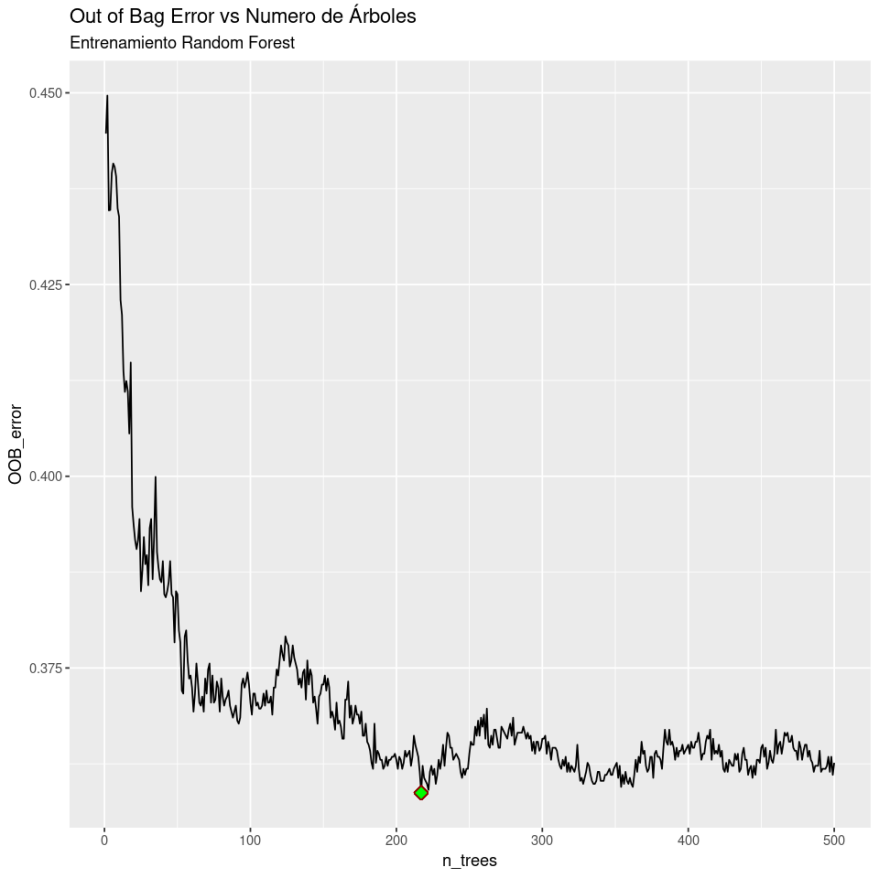

Figura 6.6: Error *out of bag* en función de la cantidad de árboles entrenados para el mejor modelo de *random forest*.

Luego, la figura 6.6 permite observar que la cantidad óptima de árboles de decisión a considerar, para la obtención del mejor modelo, son aproximadamente 220 árboles (punto verde). Esta cantidad de árboles es la que genera, para un mismo *dataset* de entrenamiento, el menor error *out of bag*. A modo de contextualización, este error dice, en palabras sencillas, como se desempeña el modelo, en promedio, a la hora de predecir datos fuera de su entrenamiento o *out of bag data*. Para cada árbol, como se mencionó anteriormente, se toma un *dataset* aleatorio con reposición del *dataset* de entrenamiento original (*bootstrap*). Por lo tanto, para cada árbol existen datos que se quedan fuera del entrenamiento. El error *out of bag* consiste en calcular el error de predicción de cada árbol con los datos que se quedaron fuera de su entrenamiento.

Luego, las matrices de confusión del mejor modelo de *random forest* para el conjunto de validación *out time* se muestran a continuación:

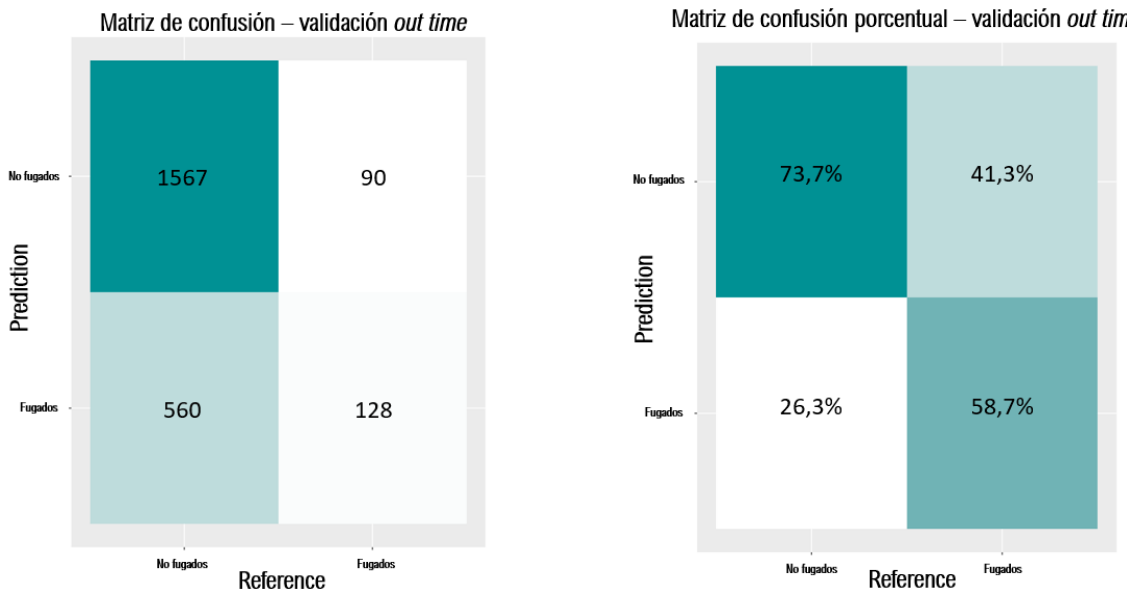

Figura 6.7: Matrices de confusión para *random forest* en el conjunto de validación *out time*

En la figura 6.7, en primer lugar, es posible evidenciar que existe un aumento de la especificidad del modelo de 4,9 % y una baja de la sensibilidad de un 9,6 % con respecto al conjunto de validación, lo cual, de manera análoga al análisis de las matrices de confusión de validación *out time* de la regresión logística, era un evento totalmente esperable. Como se menciona anteriormente, al estarse validando con un conjunto creado a partir de una ventana de tiempo diferente a con las que fue entrenado el modelo, se espera una baja de desempeño a la hora de clasificar.

Es posible notar que se produce un fenómeno sumamente similar al caso de la regresión logística, donde en ambos casos, existe un aumento de la especificidad del modelo, y una baja en la sensibilidad de este con respecto al conjunto de validación. No obstante, el rendimiento de la regresión logística es superior al del *random forest*, tanto en especificidad como en sensibilidad. Si bien el modelo de *random forest* obtuvo un desempeño superior a la regresión en el conjunto de validación, a la hora de validar *out time*, esto se invirtió. La razón de esto puede deberse a que, si bien *random forest* genera un modelo sumamente robusto y de bajo sesgo por la forma en la que se entrena y compone, es probable que este modelo sea así de robusto principalmente para datos obtenidos de la misma ventana de tiempo, es decir, no mantiene su desempeño en problemáticas donde entra en juego la temporalidad, como si lo hace la regresión logística.

Finalmente, se muestran las curvas *ROC* del modelo de *random forest*, tanto para el conjunto de validación como el conjunto de validación *out time*:

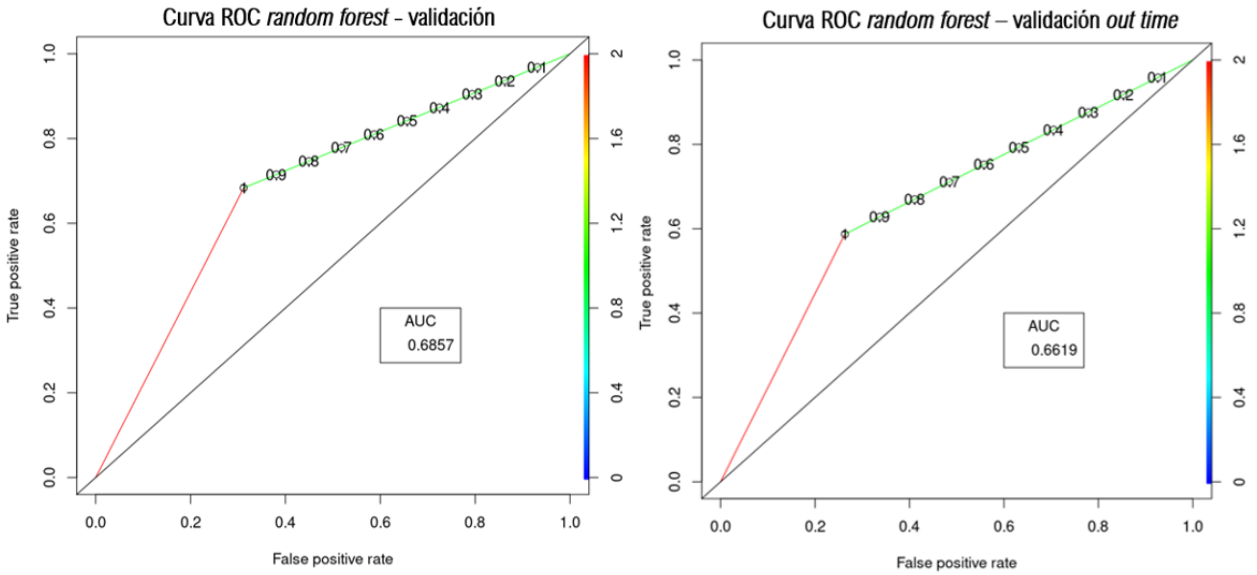

Figura 6.8: Curvas ROC para *random forest* tanto en validación como validación *out time*

De esta forma, las curvas *ROC* dejan en evidencia la baja general del rendimiento del modelo en el conjunto de validación *out time* con respecto al conjunto de validación, donde el área bajo la curva, que corresponde a la media entre la especificidad y la sensibilidad del modelo, disminuye en un 2,4 % aproximadamente. Si bien existe una baja, como se mencionó anteriormente, esto era absolutamente esperable. Una baja de 2,4 % es relativamente bajo cuando se trata de una validación *out time*, y de igual forma se considera que este modelo es estable en el tiempo y absolutamente implementable. No obstante, es importante destacar que el modelo de *regresión logística* obtuvo un mejor desempeño en el conjunto de validación *out time*, obteniendo un área bajo la curva 1,8 % mejor que el *random forest*, siendo el modelo a implementar por el momento.

6.3. *Support vector machine*

Para finalizar, se muestran los resultados del mejor modelo de *support vector machine*. Para este modelo, se realizaron 300 iteraciones de entrenamiento y validación, en la cual se obtuvo el mejor modelo en base al AUC de validación obtenido. Se realizó la mitad de iteraciones que los otros 2 modelos implementados debido al costo computacional que requería entrenar un modelo de esta naturaleza.

Las matrices de confusión del conjunto de validación se muestran a continuación:

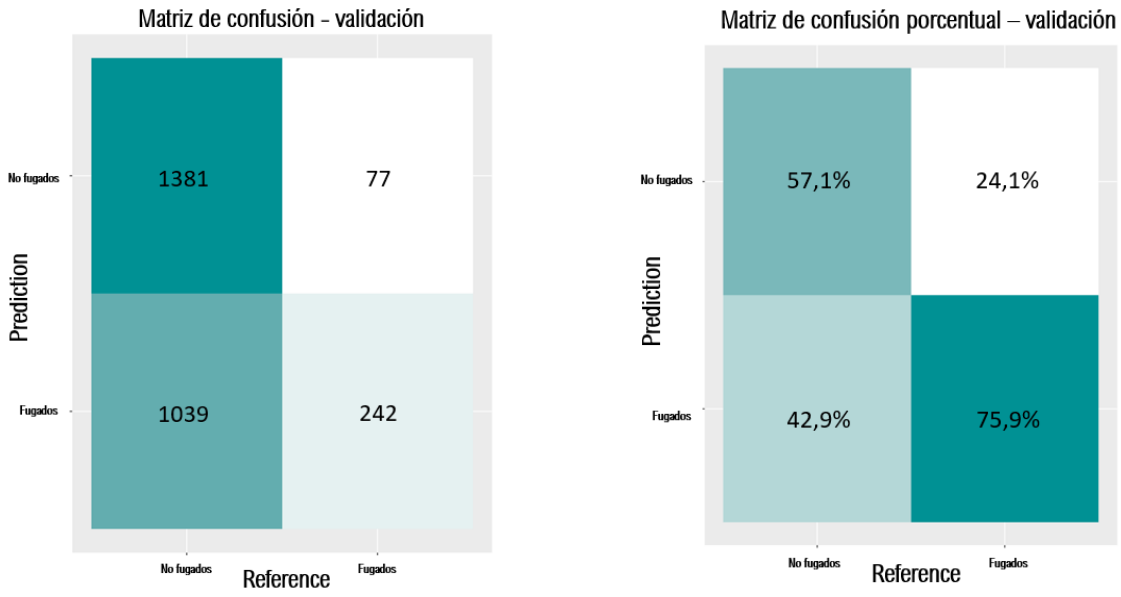

Figura 6.9: Matrices de confusión para *support vector machine* en el conjunto de validación

Luego, de la figura 6.9 se desprende, en primer lugar, el altísimo valor de la sensibilidad del modelo, alcanzando un 75,9 % de certeza de predicción en clientes fugados. No obstante, la sensibilidad del modelo se ve levemente opacada por la baja especificidad del modelo, alcanzando un bajo 57.1 %. La razón de esta gran diferencia posiblemente se deba a la manera en la que se construye un modelo de esta naturaleza, en donde, como se explica en la sección 3.3.2, se busca el clasificador que maximice el margen entre ambas clases. Luego, es probable que los datos que se están utilizando en este proyecto estén lejos de ser linealmente separables, por lo que el error siempre existirá. No obstante, la gran diferencia entre la especificidad y la sensibilidad del modelo puede deberse específicamente a como están distribuidos los clientes, donde es posible que los clientes fugados estén más concentrados que los no fugados, y al momento de escoger el hiperplano separador o clasificador, este sea capaz de separar con gran desempeño a los datos positivos (fugados) y con peor desempeño a los datos negativos (no fugados).

Dicho esto, es evidente que este modelo es el que posee la mejor sensibilidad de los 3 modelos desarrollados en el conjunto de validación, mientras que tiene la peor especificidad de los 3 modelos también. A diferencia de los otros dos modelos desarrollados, los cuales presentan una pequeña diferencia entre sus especificidades y sensibilidades, este modelo es poco consistente. Si bien tiene una gran tasa de clasificación de fugados, es importante tener armonía entre la clasificación de fugados y no fugados para ser considerado un modelo útil, ya que de lo contrario, no se tiene suficiente certeza de la predicción que realiza.

Siguiendo con los resultados del modelo de *support vector machine*, se tienen las siguientes matrices de validación *out time*:

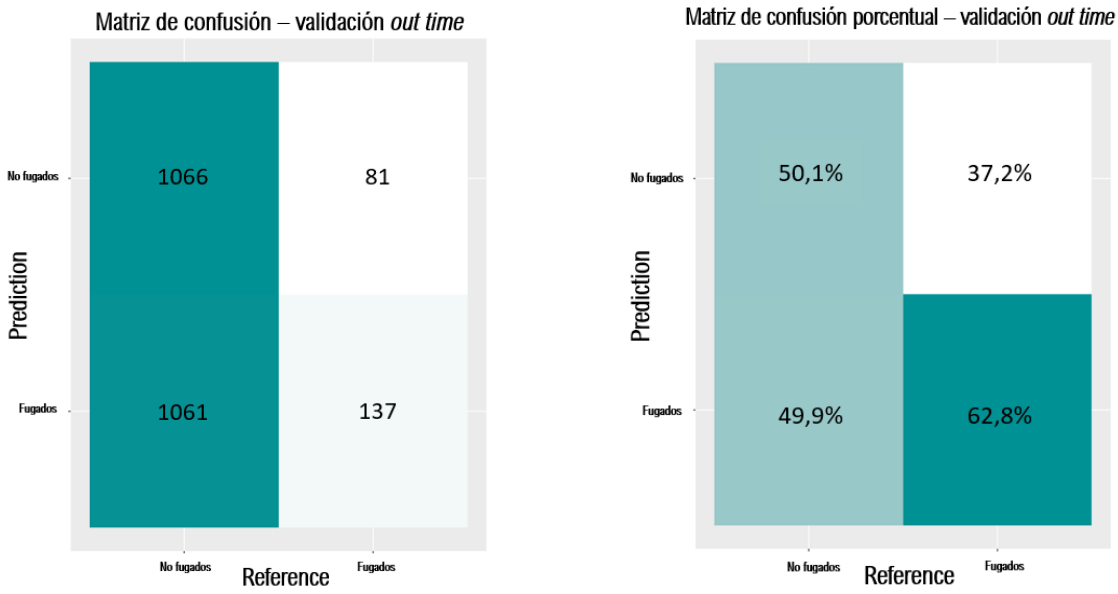

Figura 6.10: Matrices de confusión para *support vector machine* en el conjunto de validación *out time*

De la figura 6.10 se evidencia la gran disminución tanto de especificidad como de sensibilidad del modelo de *support vector machine* en el conjunto de validación *out time*, con respecto al conjunto de validación. Es posible ver que la especificidad baja de un 57,1 % a un 50,1 %, lo cual probabilísticamente es similar a predecir un verdadero positivo tirando una moneda al aire. Por otro lado, la sensibilidad del modelo, que alcanzó un 75.9 % en el conjunto de validación, disminuyó 13.1 %, quedando en 62.8 %.

Esta baja rotunda en el desempeño del modelo de *support vector machine* en el conjunto de validación *out time* deja en evidencia la baja estabilidad temporal del modelo. La explicación detrás de este fenómeno, se debe a la construcción del clasificador. Como se mencionó anteriormente, el clasificador de *support vector machine* busca maximizar el margen entre las clases, escogiendo el hiperplano que haga esta tarea. No obstante, una gran característica de este tipo de modelos, es que se asume que los datos a los que se les realizará una predicción provienen de la misma distribución que los datos de entrenamiento. Esto explica la evidente baja en el desempeño del modelo con respecto al conjunto de validación, puesto que en este último, los datos que se estaban prediciendo provenían del mismo *dataset* y la misma temporalidad que los datos con los que se entrenó y se definió el hiperplano separados. No obstante, al tomarse una ventana de tiempo diferente, los datos a predecir dejaron de comportarse de la misma forma, haciendo que el hiperplano separados de clases de máximo margen deje de clasificar con el mismo desempeño que antes.

Esto permite entender que el modelo *SVM* no es un modelo útil a la hora de ser utilizado en problemáticas que involucren temporalidad, puesto que el clasificador que se construye con este modelo es muy ajustado a los datos de una misma distribución, y por lo tanto no es estable en el tiempo. Este tipo de modelo podría ser útil en un problema de clasificación de tumores, en donde todos los datos provienen de una misma distribución, por mencionar algún ejemplo.

Finalmente, se tienen las curvas *ROC* del modelo de *support vector machine* para los

conjuntos de validación y validación *out time*, las cuales se muestran a continuación:

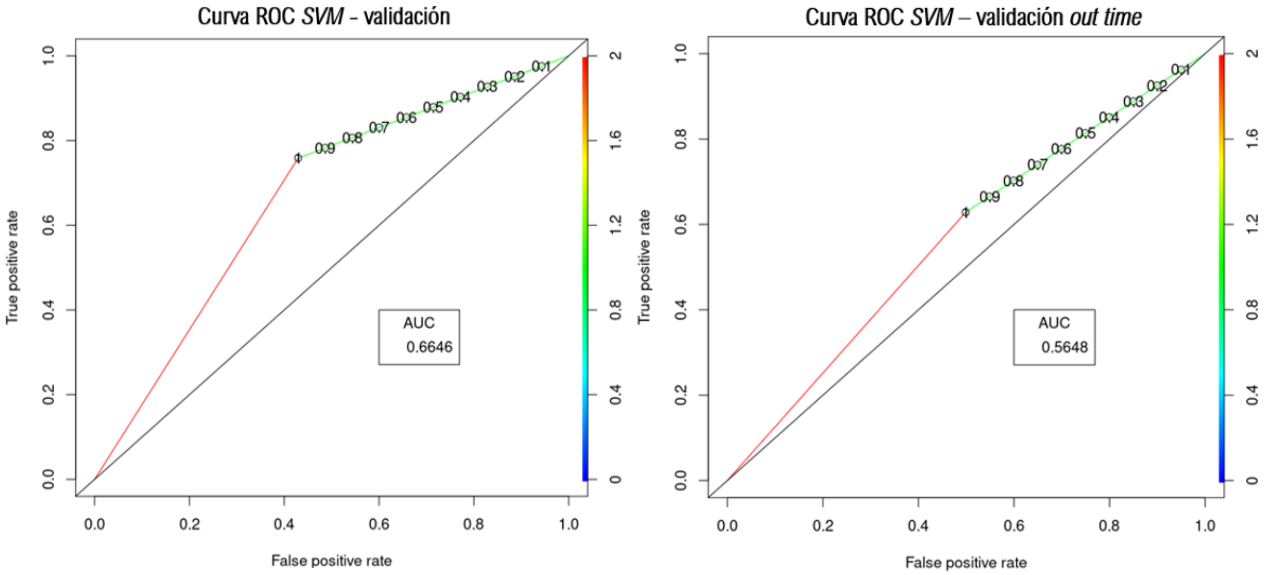

Figura 6.11: Curvas ROC para *support vector machine* tanto en validación como validación *out time*

Finalmente, las curvas ROC de este modelo de la figura 6.11 muestran la baja de desempeño que se produce al comparar el rendimiento en el conjunto de validación con el rendimiento en el conjunto de validación *out time*, donde se evidencia un aplanamiento brusco de la curva ROC en el segundo conjunto, generándose una reducción de AUC de un 10 % aproximadamente, en donde, si bien se mencionó que es esperable una baja de rendimiento al validar *out time*, este caso se escapa de lo presupuestado.

De esta forma, queda evidenciada la imposibilidad de implementación futura de este modelo en específico, el cual demostró no ser lo suficientemente estable en el tiempo como para ser utilizado mes a mes de manera recurrente.

6.4. Tablas comparativas

Para finalizar la sección de resultados y análisis, se tabulan las principales métricas de cada modelo, tanto para el conjunto de validación como el conjunto de validación *out time*. De esta forma, es posible ver y comparar los resultados obtenidos en cada uno de los mejores modelos desarrollados de manera más directa.

La primera tabla que se construye es la del conjunto de validación, en la cual se muestra el valor AUC, KS, sensibilidad, especificidad y *accuracy* de cada uno de los 3 modelos desarrollados:

Tabla 6.2: Tabla comparativa de métricas de desempeño para los 3 modelos desarrollados en validación

	AUC	KS	Sensibilidad	Especificidad	Accuracy
Regresión logística	0.658	0.316	0.655	0.660	0.660
Random Forest	0.686	0.371	0.683	0.688	0.688
SVM	0.665	0.329	0.759	0.571	0.593

Luego, en la tabla 6.2 se destaca que el modelo con mejor desempeño en este conjunto es el *random forest*, obteniendo el mejor AUC de los 3 modelos desarrollados, seguido del modelo *support vector machine*, terminando con la regresión logística en tercer lugar. No obstante, si bien SVM posee un AUC relativamente bueno, posee una inconsistencia en cuanto a la tasa de clasificación de verdaderos positivos y negativos, alcanzando una sensibilidad de 75.9 %, siendo 7.6 % superior al modelo que lo sigue en rendimiento, correspondiente al *random forest*. Esto se ve compensado con el bajo desempeño a la hora de clasificar clientes no fugados, teniendo una especificidad de 57.1 %, siendo 8.4 % más bajo que el modelo que lo sigue, correspondiente a la regresión logística.

Luego, la tabla comparativa de los 3 modelos en el conjunto de validación *out time* se muestra a continuación:

Tabla 6.3: Tabla comparativa de métricas de desempeño para los 3 modelos desarrollados en validación *out time*

	AUC	KS	Sensibilidad	Especificidad	Accuracy
Regresión logística	0.680	0.361	0.606	0.755	0.741
Random Forest	0.662	0.324	0.587	0.737	0.723
SVM	0.565	0.130	0.628	0.501	0.513

Para finalizar, la tabla 6.3 permite entender que el mejor modelo corresponde a la regresión logística, la cual si bien no tiene el mejor desempeño en el conjunto de validación, obtiene el mejor rendimiento al validarse *out time*, que es de mucha mayor relevancia, puesto que como ya se explicó anteriormente, esta validación permite ver el rendimiento real de los modelos y su estabilidad temporal. Este modelo alcanza un AUC de 68 %, mejorando en un 2.2 % su AUC con respecto al conjunto de validación, mejorando su especificidad en un 9.5 % y disminuyendo su sensibilidad en un 4.9 %, lo cual como ya se mencionó, era esperable.

En segundo lugar y no muy lejos, se tiene el modelo de *random forest*, alcanzando un AUC solamente 2.2 % menor que la regresión logística, disminuyendo solamente 2.4 % con respecto al conjunto de validación. Si bien no es el mejor modelo de los 3, sigue siendo uno robusto y estable, demostrando su capacidad predictiva en el conjunto de validación *out time*.

En tercer y último lugar, se tiene al modelo SVM, el cual demostró no ser estable temporalmente debido a su construcción y lógica de clasificación, quedando en evidencia a la hora de validar de manera *out time*, alcanzando un AUC de solo 56.5 %, disminuyendo un 10 % su rendimiento. Esto hace destacar la importancia de la validación *out time*, que permite identificar que modelos son realmente capaces de rendir en condiciones diferentes a las de entrenamiento, y por lo tanto, son estables.

Capítulo 7

Conclusiones y trabajo futuro

A modo de conclusión, se cumple tanto el objetivo principal como los objetivos específicos del trabajo, lográndose diseñar y desarrollar tres modelos predictivos de fuga de clientes de diferente naturaleza, los cuales se entrenan, validan y comparan escogiéndose el mejor de estos para la futura implementación. El modelamiento de estos algoritmos se enfoca a un segmento de clientes relevante para la empresa, correspondiente a los clientes más estables y recurrentes según su comportamiento transaccional. Se estudian, analizan y transforman los datos tanto de forma univariada como multivariada, en donde se aplican técnicas como transformaciones estadísticas, tratamientos de multicolinealidad y reducción de dimensionalidad de los datos.

En cuanto a la metodología que se siguió en el desarrollo de este proyecto (5.2), basada en la metodología clásica CRISP-DM (3.4), se destaca la profundidad con la que se comienza el desarrollo del trabajo, específicamente en cuanto al entendimiento del negocio, en donde se logró identificar los dolores de los clientes no solo a nivel informativo, si no que de forma personal, realizando incluso entrevistas presenciales a estos junto con el encargado de preventa en la comuna de Colina, con el fin de escuchar sus dolores y lograr plasmarlos en los datos que se utilizaron después.

Con respecto a los resultados de los modelos desarrollados, el que demostró el mejor rendimiento, y por lo tanto tuvo el mayor desempeño en el conjunto de validación *out time* fue el modelo de regresión logística, alcanzando un AUC de 68 %, demostrando ser lo suficientemente estable para una futura implementación. El modelo con peor rendimiento resultó ser el *support vector machine*, el cual disminuyó su área bajo la curva en casi 10 % terminando con un AUC de 56 % en validación *out time*, demostrando su inestabilidad temporal e inconsistencia de predicción fuera del conjunto de entrenamiento y validación.

Se destaca la importancia de la validación *out time*, puesto que permite evaluar el rendimiento real de los modelos predictivos, poniendo a prueba su estabilidad temporal y capacidad de predicción en conjuntos independientes de los utilizados para el entrenamiento de estos. Esto se evidencia en el caso del *support vector machine*, en el cual, de no haberse validado *out time*, no se hubiese evidenciado su mal rendimiento en la clasificación de clientes en meses actuales, puesto que obtiene un gran rendimiento en validación, pero se cae 10 % al validarse en el conjunto *out time*, lo que inmediatamente imposibilita su implementación en el futuro.

Si bien a lo largo de las validaciones siempre se buscó maximizar el área bajo la curva, es importante no dejar de lado la interpretación de falsos positivos y falsos negativos, puesto que ambos son errores que implican diferentes costos, donde es importante tener en cuenta que los falsos negativos (etiquetar como no fugado cuando lo si lo está) conlleva a perder clientes probablemente por un periodo de tiempo debido a la falta de retención, y falsos positivos (etiquetar como fugado cuando no lo está) conlleva perdida innecesarias de recursos (tiempo y capital) en retención mediante propuestas de valor como descuentos, ofertas, envíos gratis, por mencionar algunos.

Finalmente, se recalca la necesidad y lo provechoso de utilizar herramientas de *machine learning* en el mundo industrial, puesto que, como se transparenta en este trabajo, permite tomar decisiones importantes con tiempo, lo que ayuda a optimizar diversos procesos dentro de los negocios. En particular, un modelo de predicción de fuga en un negocio de relación no contractual como lo es Agrosuper, entrega información y las herramientas necesarias para identificar a los clientes más propensos a dejar de comprar, permitiendo actuar con tiempo para retenerlos y así aumentar la tasa de permanencia y satisfacción de los clientes, minimizando la pérdida de capital por la fuga de estos.

Como trabajo a futuro de este proyecto se considera, en primer lugar, armar el proceso almacenado, que implica crear una función cuyos parámetros se actualicen de manera automática periódicamente, donde para el caso del modelo desarrollado será cada 1 mes. Este proceso se ejecutará de manera mensual, obteniendo las características de los clientes (utilizando esta vez los valores del último mes a diferencia de la mediana anual para evidenciar los cambios de comportamiento en los clientes), realizando los filtros, las transformaciones pertinentes y prediciendo la fuga de los clientes del mes. Posterior al proceso almacenado, se incorporará a los sistemas de Agrosuper, en donde la predicción mensual de este modelo permitirá desencadenar diversos procesos que se consideren necesarios, como pueden ser alertas a los encargados de preventa, *mailing* con descuentos a los clientes propensos a fugarse, ofertas, encuestas, por mencionar algunos. Naturalmente, el último paso corresponde a monitorear y validar el modelo una vez ya lleve algún tiempo en producción.

Bibliografía

[1] C.-A. Azencott. *"Introduction au machine learning.* Dunod, 2019.

[2] Alexandru Niculescu-Mizil ; Rich Caruana. *Predicting Good Probabilities With Supervised Learning.* Cornell University, United States, 2005.

[3] Ali Tamaddoni Jahromi ; Mohammad Mehdi Sepehri ; Babak Teimourpour ; Sarvenaz Choobdar. *Modeling customer churn in a noncontractual setting: the case of telecommunications service providers.* Tarbiat Modares University , Tehran, I.R. Iran, 2010.

[4] Elsa Mireya Guadarrama Tavira, Enrique; Rosales Estrada. *Marketing relacional: valor, satisfacción, lealtad y retención del cliente. Análisis y reflexión teórica.* Santo Domingo, República Dominicana, 2015.

[5] Andrés Martínez ; Claudia Schmuck ; Sergiy Pereverzyev Jr. ; Clemens Pirker ; Markus Haltmeier. *A machine learning framework for customer purchase prediction in the noncontractual setting.* University of Innsbruck, Austria, 2020.

[6] Daniel Ringbeck ; Dmitry Smirnov ; Arnd Huchzermeier. *Proactive Retention Management in Retail: Field Experiment Evidence for Lasting Effects.* Otto Beisheim School of Management, Germany, 2019.

[7] Robert Tibshirani J. F. Trevor Hastie. *The elements of statistical learning.* Springer, 2009.

[8] Cadima J Jolliffe IT. *Principal component analysis: a review and recent developments.* Royal Society, 2016.

[9] Mónica Lizares Castillo. *Comparación de modelos de clasificación: regresión logística y árboles de clasificación para evaluar el rendimiento académico.* Lima, Perú, 2017.

[10] T.; Stefanovic D.; Anderla A.; Gracanin D Mirkovic, M.; Lolic. *Customer Churn Prediction in B2B Non-Contractual Business Settings Using Invoice Data.* University of Novi Sad, Serbia, 2022.

[11] Joaquín Amat Rodrigo. *Regresión logística simple y múltiple.* Berlin, 2nd edition, 2016.

[12] Tolga Kaya S. Nazlı Günesen; Necip Sen ; Nihan Yıldırım(B). *Customer Churn Prediction in FMCG Sector Using Machine Learning Applications.* Istanbul Technical University, Istanbul, Turkey, 2021.

[13] Theresa Gattermann-Itschert; Ulrich W. Thonemann. *Proactive costumer retention management in a non-contractual B2B setting based on churn prediction with random forests*. University in Cologne, Germany, 2021.

[14] Theresa Gattermann-Itscherta ; Ulrich W. Thonemanna. *How training on multiple time slices improves performance in churn prediction*. University of Cologne, Germany, 2021.

[15] Felipe Tobar. *Aprendizaje de Máquinas*. Santiago, Chile, 2022.

[16] Pablo Estévez V. *Apuntes Inteligencia Computacional*. Santiago, Primavera, 2020.

[17] W. Vorhies. *"Crisp-dm – a standard methodology to ensure a good outcome*. 2016.

Anexos

Anexo A

Análisis univariado

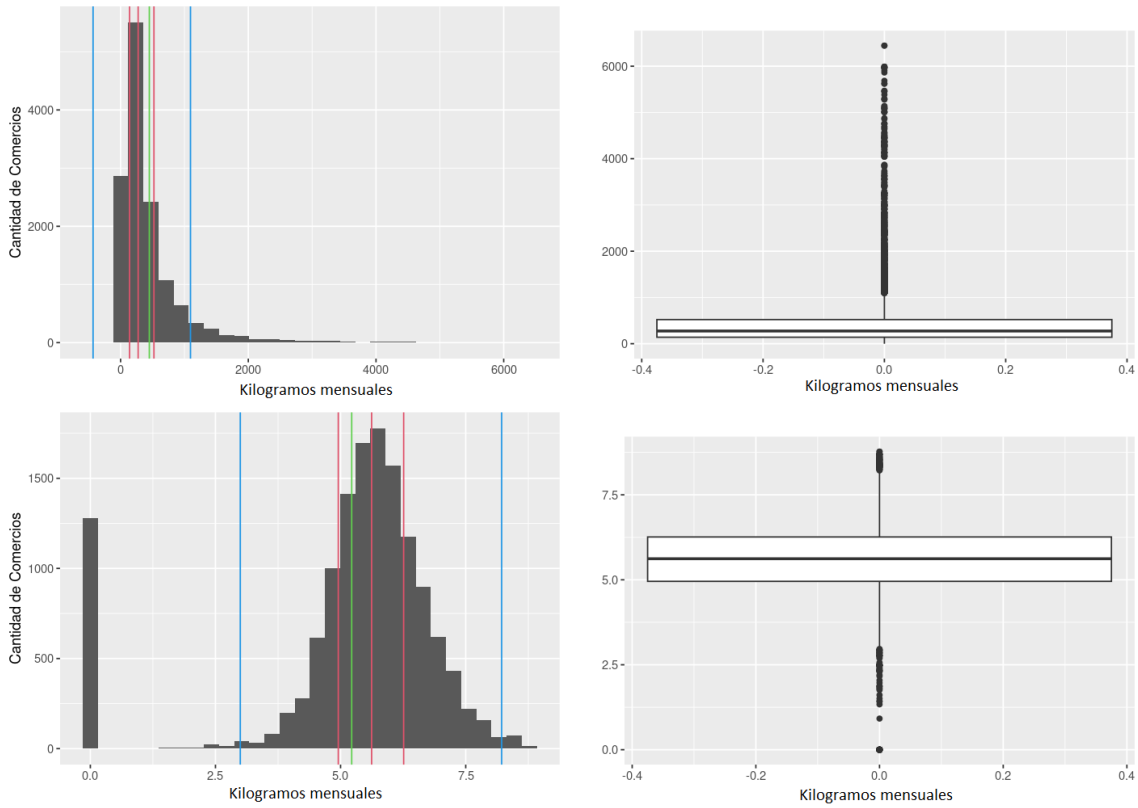

Figura A.1: Transformación univariada de la variable de kilogramos mensuales pedidos.

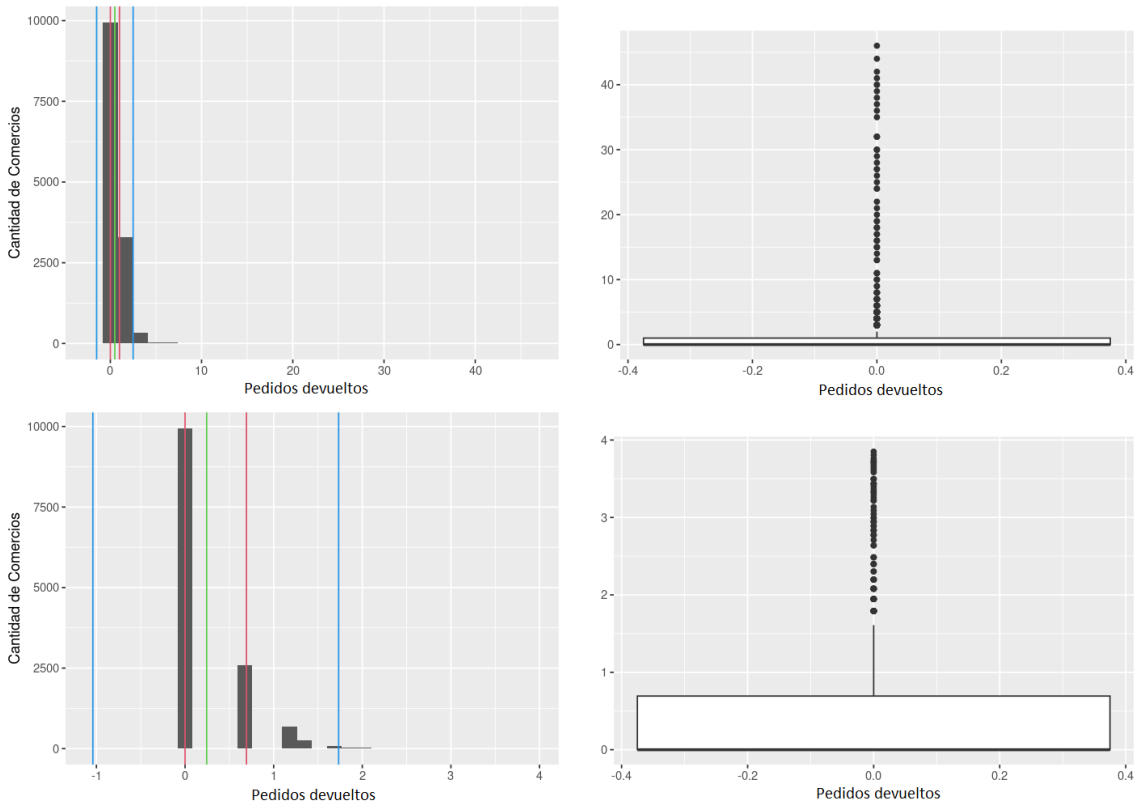

Figura A.2: Transformación univariada de la variable de pedidos mensuales devueltos.

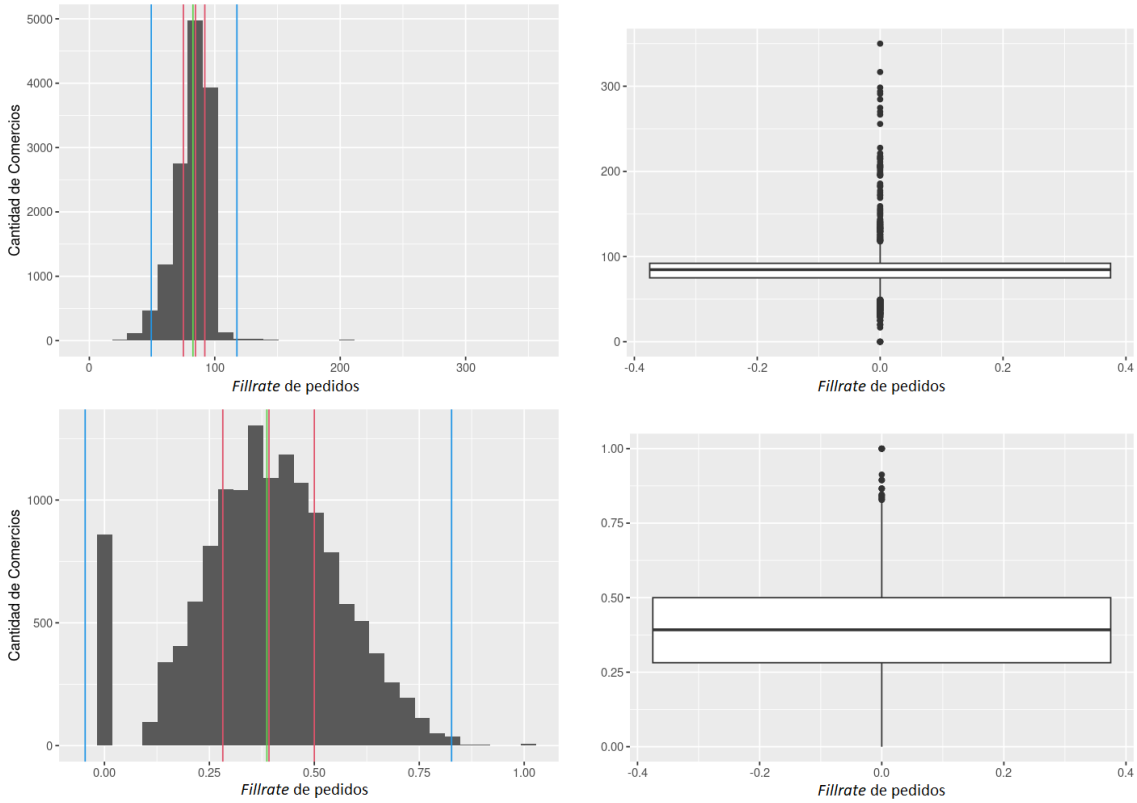

Figura A.3: Transformación univariada de la variable de *fillrate* de pedidos.

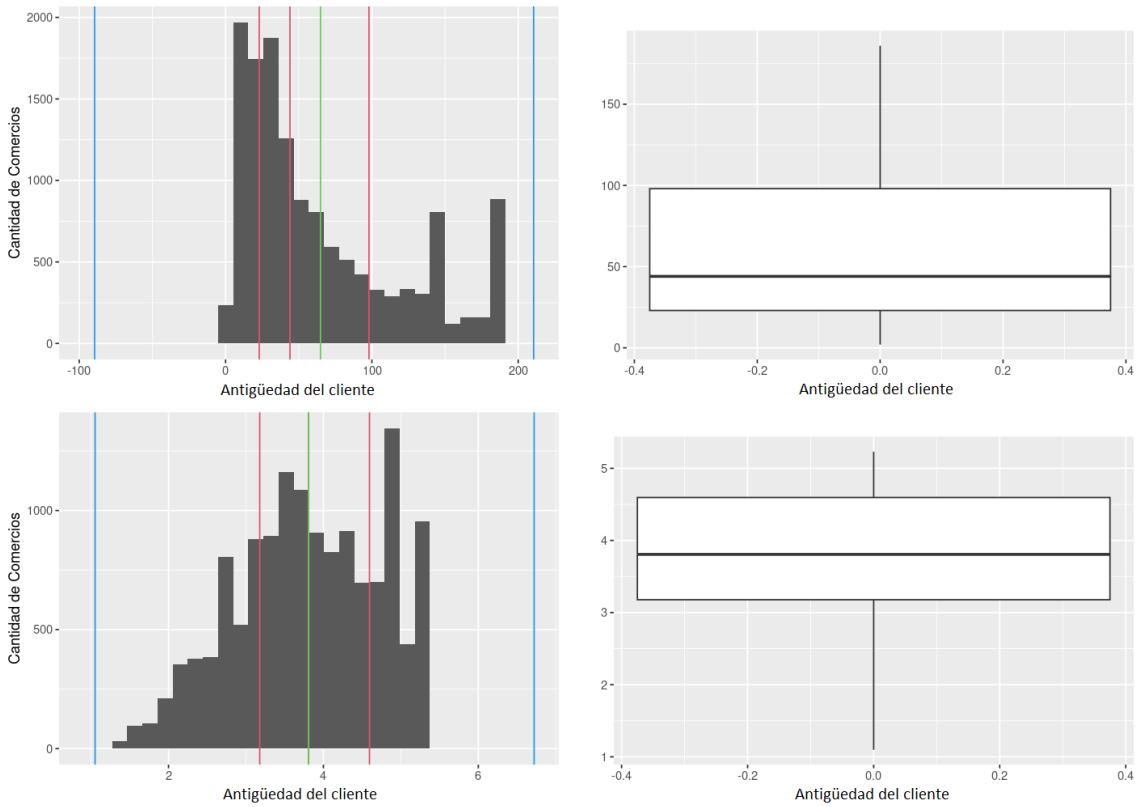

Figura A.4: Transformación univariada de la variable de antigüedad del cliente.

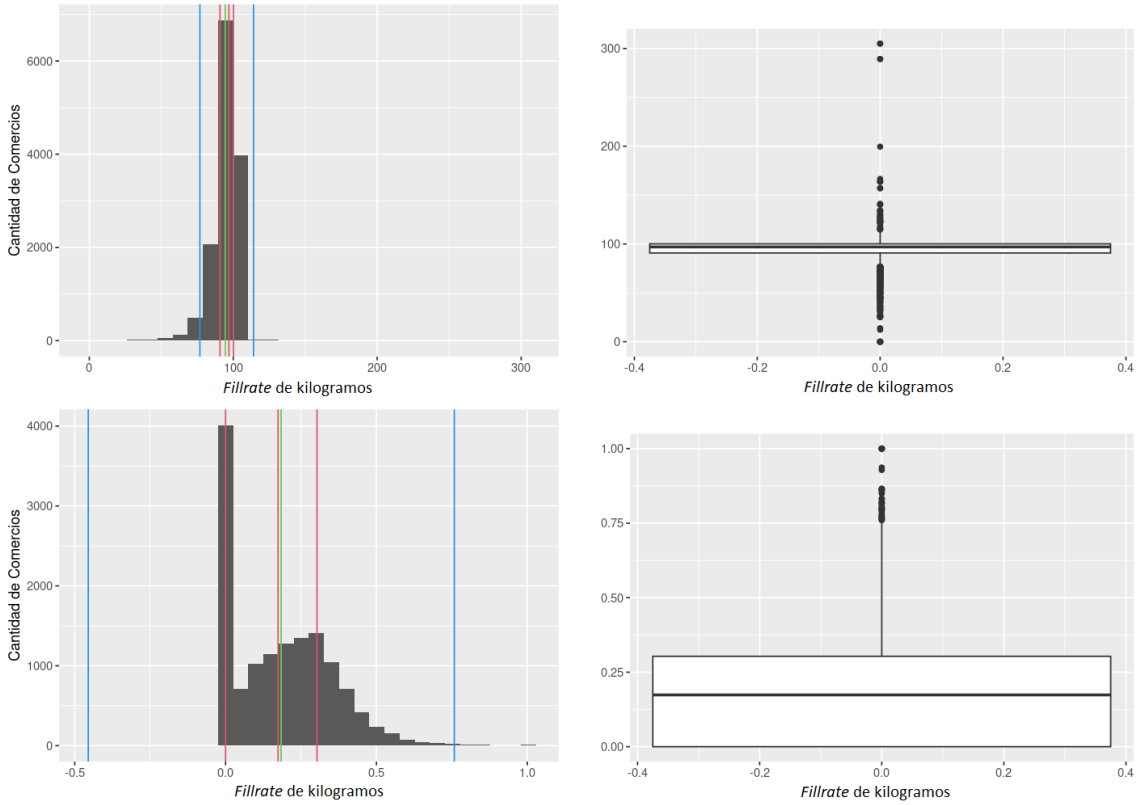

Figura A.5: Transformación univariada de la variable de *fillrate* de kilogramos pedidos.

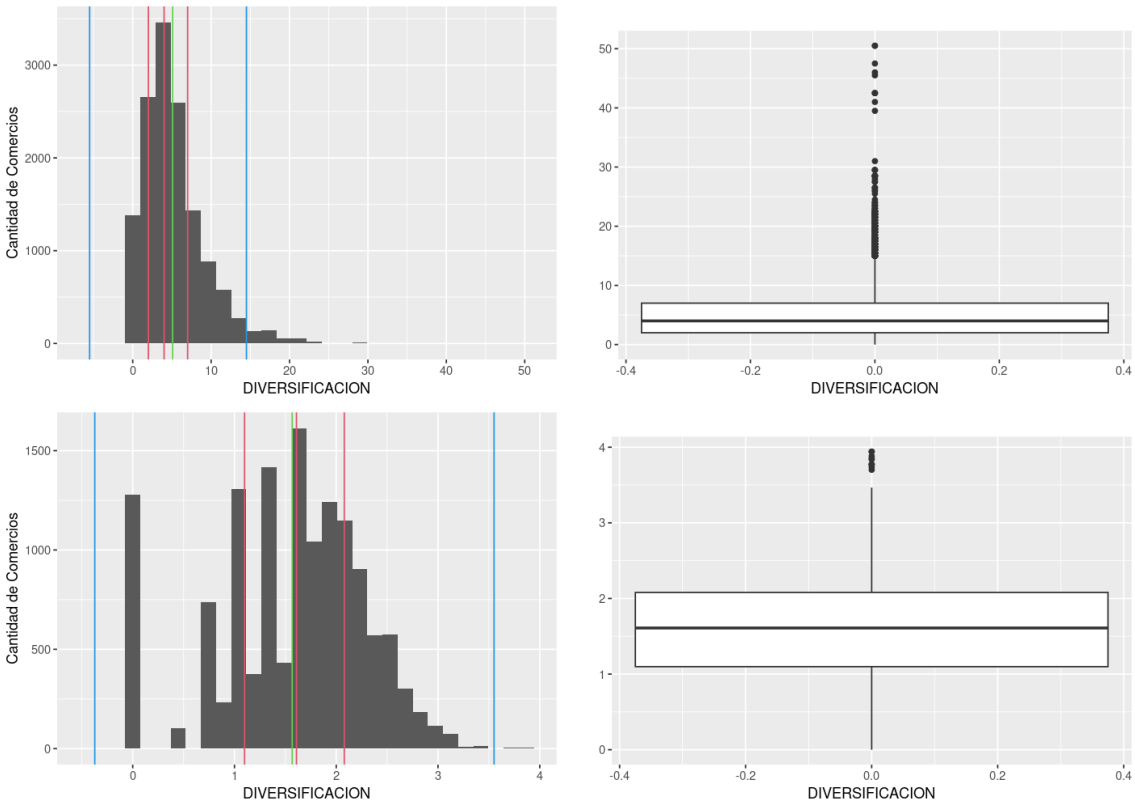

Figura A.6: Transformación univariada de la variable de diversificación del cliente.

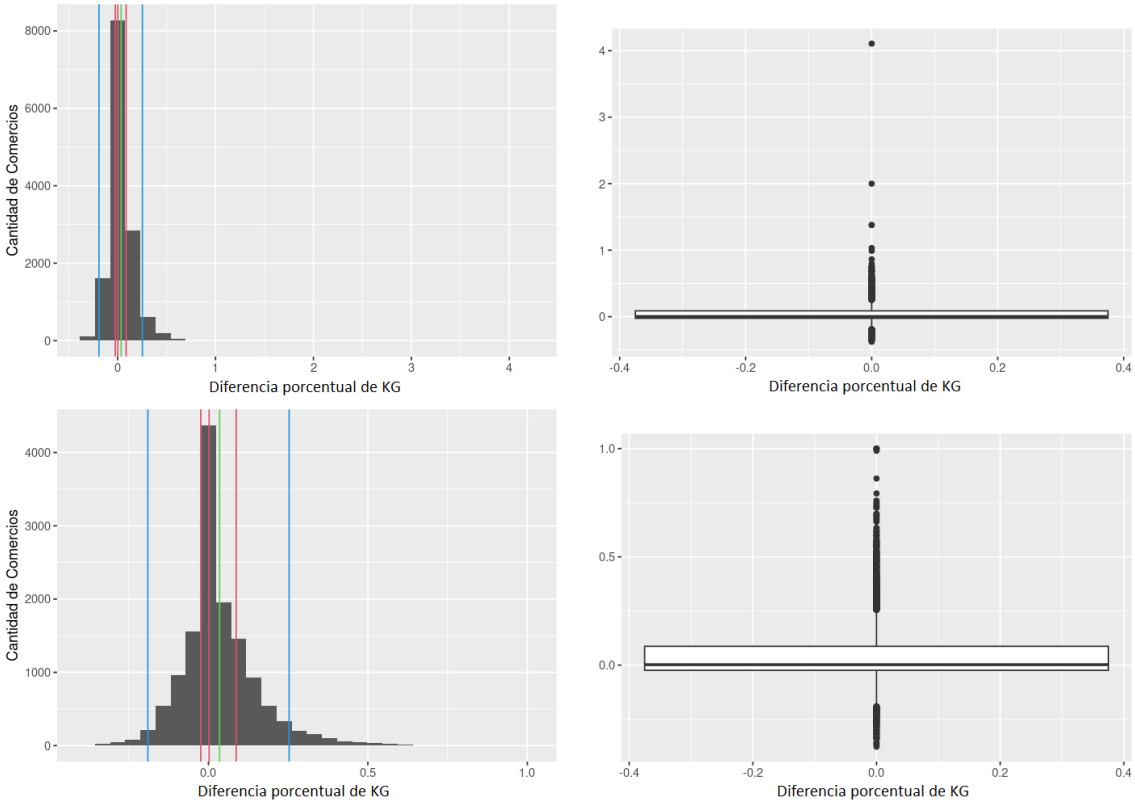

Figura A.7: Transformación univariada de la variable de diferencia porcentual de kg pedidos.

Anexo B

Construcción de características

Tabla B.1: Estructura de maestra de clientes.

Campo	Descripción	Ejemplo
RUT_EMPRESA	Rut de la empresa dueña del local	76194115
COD_LOCAL	Código único del local	3200092610
NOMBRE	Nombre al cual está asociado	Quality Service Ltda
DIRECCION	Dirección donde se encuentra	Julio Troncoso S/N Olivar
COD_TIPOLOC	Código del tipo de local	A3
DES_TIPOLOC	Que tipo de local es	Casino Local
COMUNA	Comuna en la que está ubicado	Olivar
CIUDAD	Ciudad en la que está ubicado	Cachapoal
REGION	Numero de la región en la que está ubicado	06
DES_REGION	Región en la que está ubicado el local	BERNARDO O'HIGGINS
LATITUD	Latitud de la ubicación del local	34.358914
LONGITUD	Longitud de la ubicación del local	70.848941
FECHA_CREAC	Fecha en la que comenzó a ser cliente	20140801
SALES_DIST	Es el código de la zona de ubicación del local	NA0102
DESSALES_DIST	Es la zona de ubicación del local	SUBG CENTRO SUR

Tabla B.2: Estructura de maestra de facturas.

Campo	Descripción	Ejemplo
DOC_NUMBER	Número de identificación de factura	955179105
COD_LOCAL	Código único del local	3200007193
COD_SKU	Código de identificación del producto facturado	1012181
DIA_FACTURA	Fecha en la que se generó la factura	20220418
PRECIO_XKG	Precio por KG del producto facturado en pesos chilenos	942
KG_FACT	KG de productos facturados	10.8
PRECIO_FACT	Precio total facturado	10170
REF_DOC_NO	Número único de referencia de la factura emitida	83736378
BICZASGFAC	Permite evaluar que factura se anula cuanto se solicita una nota de crédito	

Tabla B.3: Estructura de maestra de pedidos.

Campo	Descripción	Ejemplo
DOC_NUMBER	Número de identificación del pedido	242090222
COD_LOCAL	Código único del local que realizó el pedido	3200002197
COD_SKU	Código de identificación del producto pedido	1120425
CALDAY	Fecha en la que se realizó el pedido	20210424
DOC_TYPE	Tipo de documento que especifíca por que canal se realizó el pedido	ZSCN
PRECIO_XKG	Precio por KG del producto pedido en pesos chilenos	2500
PESO_NETO	KG de producto pedido	14.0
PRECIO_FACT	Precio total del pedido realizado en pesos chilenos	35000

Tabla B.4: Estructura de maestra de materiales.

Campo	Descripción	Ejemplo
COD_SKU	Código de identificación del producto pedido	1011408
DESCRIPCION_MAT	Descripción del material o producto	PchDeh tumbleada elaborado# Bj 20k SP
COD_N4	Código de identificación del cuarto nivel del producto	1011025090
COD_N3	Código de identificación del tercer nivel del producto	1011025
DES_N3	Descripción del tercer nivel del producto	Pechuga Desh s/Piel s/grasa s/filete
COD_N2	Código de identificación del segundo nivel del producto	1011
DES_N2	Descripción del segundo nivel del producto	Pechuga Desh
SECTOR	Código del sector al que pertenece el producto	01
DESSECTOR	Sector al que pertenece el producto	POLLO
FECHA_CREA	Fecha en la que el producto se comenzó a comercializar	20110630
ESTADO_REFRI	Estado de refrigerado (refrigerado, no refrigerado o congelado)	REFRIGERADO
ESTADO_ENVAS	Estado de envasado (granel, laminado, envasado, etc)	GRANEL
ESTADO_CRUDO	Estado de crudo (crudo o procesado)	CRUDOS

Tabla B.5: Estructura de tablón de características previo al análisis univariado.

Campo	Descripción	Tipo	Tabla(s) de procedencia
COD_LOCAL	Código de identificación del local	Double	PedidosxClienteFS
PED_MENS	Cantidad de pedidos mensuales realizados	Double	PedidosxClienteFS
PED_CC	Cantidad de pedidos mensuales realizados por call center	Double	PedidosxClienteFS
PED_VM	Cantidad de pedidos mensuales realizados por venta móvil	Double	PedidosxClienteFS
PED_PO	Cantidad de pedidos mensuales *online* realizados	Double	PedidosxClienteFS
KG_MENS	Cantidad de kg mensuales pedidos de producto	Double	PedidosxClienteFS
PED_DEV	Cantidad de pedidos devueltos	Bigint	FacturasxClientesFS_devueltas
FILLRATE_PED	Porcentaje de cumplimiento de pedidos	Double	PedidosxClienteFS y FacturasxClientesFS
ANTIG_MES	Cantidad de meses que lleva siendo cliente con AS	Double	ClientesFS
FLEX_PRECIO	Elasticidad del cliente c/r al precio	Double	PedidosxClienteFS
FILLRATE_KG	Porcentaje de cumplimiento de kg pedidos	Double	PedidosxClienteFS y FacturasxClientesFS
CANT_LOCALES	Cantidad de locales asociados a la misma empresa	Bigint	ClientesFS
ZONA_SUR	Si el local se encuentra en la zona sur	Double	ClientesFS
ZONA_CENTRO_SUR	Si el local se encuentra en la zona centro sur	Double	ClientesFS
ZONA_STGO	Si el local se encuentra en la zona centro	Double	ClientesFS
ZONA_CENTRO_NORTE	Si el local se encuentra en la zona centro norte	Double	ClientesFS
ZONA_NORTE	Si el local se encuentra en la zona norte	Double	ClientesFS
DIVERSIFICACION	Cuantos SKU's diferentes compra al mes	Double	PedidosxClienteFS
DELTA_KG_POR	Diferencia porcentual de kg pedidos entre cada mes	Double	PedidosxClienteFS
VISITAS_MENS	Cantidad de visitas mensuales	Double	VisitasFS