Trabajo Fin de Máster

Modelo de propensión al abandono en el canal Horeca

José Piñeiro Abal

Máster en Técnicas Estadísticas

Curso 2018-2019

II

Propuesta de Trabajo Fin de Máster

Título en galego: Modelo de propensión ao abandono na canle Horeca
Título en español: Modelo de propensión al abandono en el canal Horeca
English title: Abandonment propensity model in the Horeca channel
Modalidad: Modalidad B
Autor: José Piñeiro Abal, Universidade de Santiago de Compostela
Director: Alberto Rodríguez Casal, Universidade de Santiago de Compostela;
Tutor: Jorge López Muñíz, Estrella Galicia;
Breve resumen del trabajo: Obtención de la tasa mensual de abandono de cerveza de los clientes de Estrella Galicia en una región determinnada.
Recomendaciones:
Otras observaciones:

Índice general

Resumen

Resumen en español

En este trabajo comentaremos el proceso de obtención de la tasa de abandono mensual de cerveza de los locales que son clientes de Estrella Galicia en una determinada región. Para realizar ésto, antes de emplear las técnicas de predicción, será necesario transformar y limpiar el conjunto de datos. Una vez preparado el conjunto de datos, se comprobó el buen funcionamiento de diversas técnicas de predicción en el entorno de programación R, como puede ser Random Forest, XGBoost, Regresión logística, Redes Neuronales, etc., y de plataformas de pago, como la herramienta Machine Learning de Amazon Web Service. Para ello han sido testados los diversos métodos realizando las predicciones de los meses de julio, agosto y septiembre de 2018. En este caso, el método Random Forest ha sido el que mejor resultado nos ha dado. Además también se trabajó en la extensión del método pudiendo obtener la tasa de abandono semanal tanto de cerveza como de cualquier otro producto de los que Estrella Galicia es propietaria.

English abstract

In this paper we will comment on the process of obtaining the monthly beer abandonment rate of the premises that are Estrella Galicia customers included in a certain region. To do this, before using the prediction techniques, it will be necessary to transform and clean the data set. Once the data set was prepared, the correct functioning of various prediction techniques in the R programming environment was verified, such as Random Forest, XGBoost, Logistic Regression, Neural Networks, etc., and payment platforms, such as Machine Learning tool from Amazon Web Service. To do this, the different methods have been tested, making the predictions for the months of July, August and September 2018. In this case, the Random Forest method has been the one that has given us the best result. In addition, we also worked on the extension of the method, being able to obtain the weekly abandonment rate for both beer and any other product that Estrella Galicia owns.

Introducción

Uno de los aspectos vitales dentro del sector económico es poder detectar cuándo un cliente va a dejar de comprar un determinado producto con el fin de poder anticiparse a este suceso, identificar su causa y buscar una solución en el caso de ser posible.

El hecho de que un cliente deje de comprar un cierto producto lo definiremos como abandono. Por otra parte, denominaremos tasa de abandono a la probabilidad de que un determinado cliente finalmente acabe abandonando, es decir, que deje de ser cliente atendiendo al criterio especificado para cada caso.

El estudio de la tasa de abandono está en pleno auge, principalmente en las compañías telefónicas, debido fundamentalmente a que se considera que atraer a un nuevo cliente es alrededor de 8 veces más costoso que mantener a un antiguo cliente. Por tanto, poder anticiparse al abandono es un aspecto clave en el sector empresarial, ya que nos permite identificar el cliente que puede dejar de comprar nuestro producto y trabajar en la manera de ponerle solución.

Este trabajo se desarrolló en la corporación Hijos de Rivera S.A.U., empresa dedicada al sector alimentario, cuyo negocio se centra en la producción y venta de bebidas. Su presencia es destacable, ya que es una de las empresas de bebidas más importante del país, por tanto, crear un método que permita detectar qué clientes, en nuestro caso locales, abandonen o vayan a la competencia, es clave.

El objetivo de este trabajo consistirá en la obtención de un algoritmo de predicción que nos permita detectar qué locales van a dejar de comprar cerveza de la marca Estrella Galicia en una cierta región. La idea principal es que a partir de una cierta probabilidad de abandono, se envíe un aviso a los promotores de ventas informándoles de que con una cierta probabilidad un local va a dejar de comprar cerveza al mes siguiente, pudiendo con ello anticiparse en la búsqueda de una solución.

Para construir el método disponemos del historial mensual de ventas que realizan los distribuidores de Estrella Galicia a los hoteles, restaurantes y cafeterías, tanto de cerveza como de agua y distribución. Se entenderá como distribución todos los productos ajenos a la empresa Hijos de Rivera, compañía a la cual pertenece Estrella Galicia, pero que se encarga de distribuir esta empresa. Como agua, será entendida cualquiera de las marcas de agua perteneciente a Hijos de Rivera S.A.U (Cabreiroá, Agua de Cuevas y Fontarel) y como cerveza a cualquiera de los productos de cerveza propiedad de Estrella Galicia (Estrella Galicia Especial, 1906, Red Vintage,...). Este conjunto de datos pertenece al canal HORECA (acrónimo de Hoteles, Restaurantes y Cafeterías) y se encuentran almacenados dentro de la base de datos de la compañía en el programa SQL Server, el cual es una herramienta de tratamiento de datos.

Así pues, para construir el modelo que nos permita obtener la tasa de abandono, en el Capítulo 1 una vez extraído el histórico de datos del canal HORECA se presentan los datos y posteriormente se pondrá énfasis en observar su comportamiento y en analizar sus propiedades. Una vez realizado esto, se transforman y depuran los datos para llevar a cabo las técnicas de predicción que nos permitan detectar qué locales van a abandonar.

Tras la depuración de los datos, en el Capítulo 2 nos centraremos en explicar las diversas técnicas que han sido empleadas para la construcción del modelo, sobre el cual realizaremos las predicciones para obtener la tasa de abandono.

Posteriormente en el Capítulo 3 se comenta con detenimiento los mejores resultados obtenidos por el modelo para las predicciones de los meses de julio, agosto y septiembre de 2018.

Para finalizar, en el Capítulo 4 enunciaremos brevemente las conclusiones obtenidas a lo largo del trabajo y se comentarán los pasos posteriores que servirán para mejorar el modelo e implementarlo en la práctica en el futuro.

Capítulo 1

Datos

Los datos de los que disponemos es el historial de ventas mensuales dentro de la península ibérica (España y Portugal), tanto de productos de la marca Estrella Galicia como de distribución de otras marcas, en el periodo comprendido entre Enero de 2010 y Noviembre de 2018 dentro del canal HORECA.

Dichos datos se englobarán en tres grupos:

- **Agua**: En este apartado clasificaremos las ventas mensuales de agua de las cuales Estrella Galicia es propietaria.

- **Cerveza**: Este grupo comprende el sector principal de Estrella Galicia. Nuestro objetivo será predecir la tasa de abandono de dicho grupo por parte de un local.

- **Distribución**: En esta categoría se reúnen todos los productos ajenos a Estrella Galicia pero ante los cuales se encarga de su distribución, principalmente en Galicia. Este apartado abarca una amplia gama de bebidas (refrescos, zumos, etc.) de diversas marcas.

Estos datos son recogidos por los distribuidores encargados de la entrega del producto en cada uno de los locales. En la actualidad estos pedidos se realizan de manera digital a través de una aplicación denominada Handy (diseñada por Hijos de Rivera S.A.U.) de modo que los datos ya quedan almacenados automáticamente de manera electrónica. A continuación estos datos se procesan y se depuran a diario para la posterior subida a la base de datos afincada en SQL Server.

Nuestro objetivo, por tanto, será obtener los datos de dicha plataforma para analizarlos, depurarlos y aplicar los métodos apropiados para estimar la probabilidad de abandono de un local. Para la extracción de los datos al entorno de programación R, fue necesario conectar R al sistema de manejo de datos SQL Server debido a la imposibilidad de transferir la gran cantidad de datos de manera indirecta. Esta conexión se realizó a través de la librería de R, RODBC (Ripley, 2017).

Tras conectar R con SQL Server mediante la función *odbcDriverConnect*, procedemos a la extracción de los datos mediante la función *sqlQuery* en la cual introducimos el código SQL que nos permita extraer los datos deseados. Tras realizar este proceso obtenemos un total de 10942945 filas y 15 columnas de modo que cada fila de los datos representa el pedido mensual de un local sobre cada uno de los pedidos de agua, distribución o cerveza y las columnas muestran las características del pedido como puede ser litros del pedido, población del local, etc. En nuestro caso, nos quedamos únicamente con los datos de una determinada región.

A continuación en el Cuadro 1.1 se muestra un pequeño ejemplo de los datos obtenidos tras la extracción:

MES	POBLACION	PROVINCIA	AGRUPACION MARCA	AGRUPACION ENVASE	AGRUPACION NEGOCIO	AÑO	INSTALACION
201001	X	Y	EG	Cerveza	101	2010	0
201001	X	Y	1906	Cerveza	101	2010	0
201001	X	Y	Zuvit	Distribucion	103	2010	1
201001	X	Y	Cola-Cao	Distribucion	103	2010	1
201001	X	Y	Schweppes	Distribucion	103	2010	1
201001	X	Y	EG	Cerveza	101	2010	0

MARCA DESGLOSE	TIPO INSTALACION	ACTIVIDAD LOCAL	MES SIMPLE	VENTASBRUTASACUMMESLITROS	VENTASBRUTASACUMMESEUROS	LOCAL ID
EG ESPECIAL	Barril	Sin Especificar	01	31.68	62.800	2
1906	Barril	Sin Especificar	01	7.92	18.900	2
Zuvit	Barril	Restaurante	01	4.80	15.970	6
Cola-Cao	Barril	Restaurante	01	5.46	16.360	6
Schweppes	Barril	Restaurante	01	19.20	69.552	6
EG ESPECIAL	Sin instalación	Cervecería	01	118.80	232.020	39

Cuadro 1.1: Ejemplo de los datos extraídos.

1.1. Variables

Veamos con detenimiento el significado de cada una de las variables extraídas del conjunto de datos:

- **MES**: Esta variable nos proporciona el mes y el año en el que se realizó el pedido. Consta de 6 dígitos donde los cuatro primeros indican el año y los dos últimos el mes en el que el pedido fue realizado.
- **POBLACION**: Indica la población en la cual se realizó el pedido.
- **PROVINCIA**: Indica la provincia en la cual se realizó el pedido.
- **AGRUPACION MARCA**: Proporciona la marca general sobre la cual se realizó el pedido (EG, Cola-Cao, CocaCola, etc.)
- **AGRUPACION ENVASE**: Indica si el producto del pedido consiste en cerveza, agua o distribución.
- **AGRUPACION NEGOCIO**: Indica dentro del canal HORECA si se trata de un hotel, restaurante o cafetería.
- **INSTALACION ESTABLECIMIENTO**: Muestra si el establecimiento tiene grifos de Estrella Galicia o no.
- **MARCA DESGLOSE**: Desglosa los productos dentro de la marca. Por ejemplo, la marca 1906 se desglosa en: 1906, Red Vintage y Black Coupage
- **TIPO INSTALACION**: Informa si el local dispone de barril, tanque o ninguna de las dos opciones anteriores.

- **ACTIVIDAD LOCAL**: Señala el tipo de local que es. Por ejemplo si se trata de una cafetería, cervecería, local nocturno etc.
- **LOCAL ID**: Le asigna a cada local una identidad numérica.
- **VENTASACUMMESLTS**: Venta mensual en litros de un determinado producto.
- **VENTASBRUTASACUMMESEUROS**: Venta mensual en euros de un determinado producto.

1.2. Características de los datos

Antes de proceder a tratar los datos es esencial conocer previamente su comportamiento. A continuación comentaremos las principales características observadas en el conjunto de datos.

1.2.1. Estacionalidad

En primer lugar, para hablar de la estacionalidad de los datos, nos centraremos en el ejemplo concreto de venta mensual de cerveza en litros en una ciudad que hemos considerado como estándar. Como se puede observar en la Figura 1.1, la cual representa la venta mensual de litros de cerveza en esa ciudad, nuestros datos tienen un fuerte comportamiento estacional ya que como podemos observar se consume más cerveza en los meses de verano que en invierno.

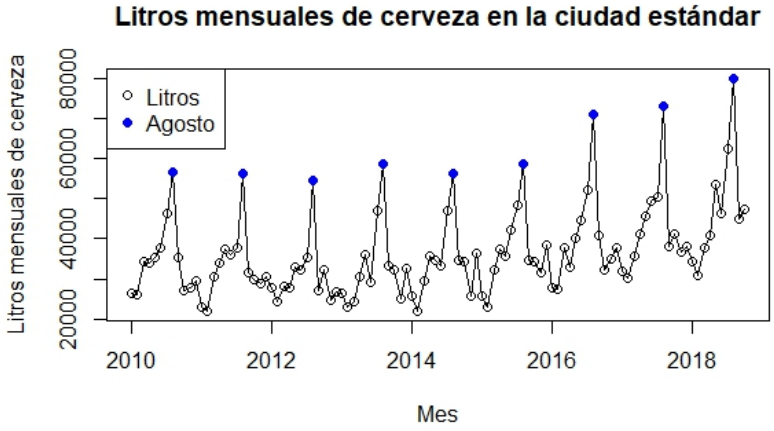


Figura 1.1: Ventas mensuales de litros de cerveza en la ciudad estándar.

Ahora surge el problema de cómo tratar estos datos, debido a que una bajada de agosto a septiembre debe ser considerada normal, mientras que una bajada de junio a julio no. Por tanto, nuestro modelo deberá tener en cuenta el mes en el cual se realizó el pedido.

Importancia de dividir los locales en zonas

Esta estacionalidad también varía dependiendo de la zona en la que nos encontremos. Para ver esto de manera más clara se comparará la Figura 1.1, la cual muestra los litros de cerveza de la ciudad que hemos considerado como éstandar, con otros dos tipos de ciudades: una ciudad en la cual conviven tanto estudiantes como turistas (turístico-estudiantil) mostrada en la Figura 1.2 y una ciudad en la cual predomina el turismo, Figura 1.3.

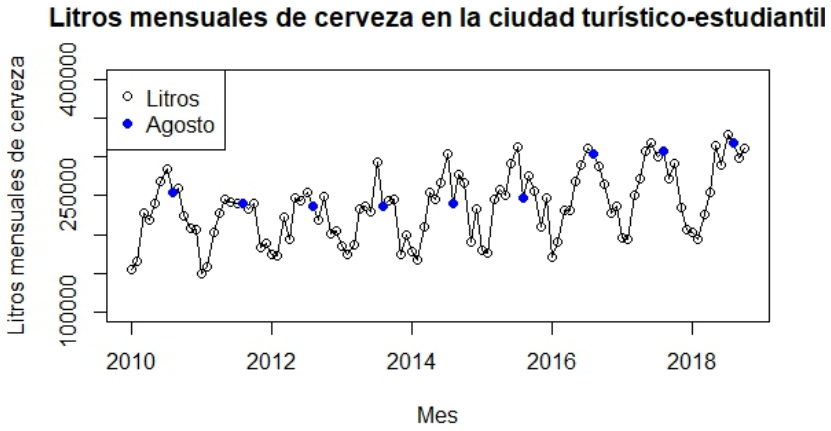


Figura 1.2: Ventas mensuales de litros de cerveza en la ciudad turístico-estudiantil.

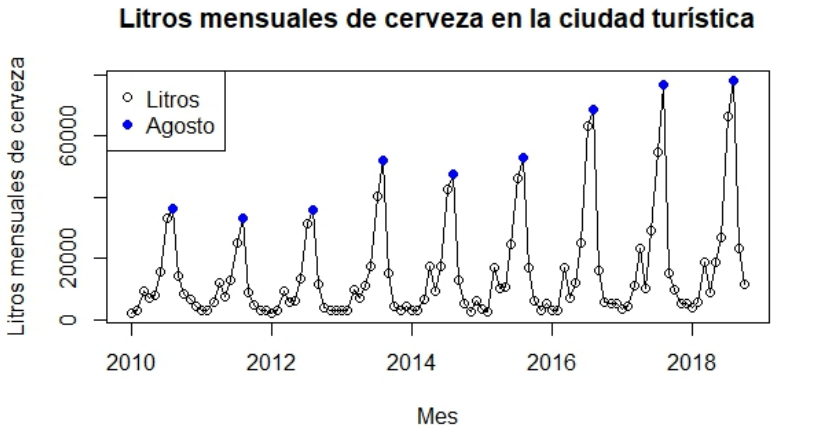


Figura 1.3: Ventas mensuales de litros de cerveza en la ciudad turística.

Como podemos observar, las tres ciudades presentan estacionalidad pero su comportamiento es bastante diferente entre sí. Mientras que, por ejemplo, en la Figura 1.2 se observa un aumento y disminución más moderado que en las dos otras gráficas, en la Figura 1.3 se observa un pico muy pronunciado en los meses de verano. Por este motivo, además de considerar la estacionalidad temporal que sufren los locales, también se deberá tener en cuenta su localización. Como solución a este problema, además de considerar la población en la que se encuentra el local, también se procederá a crear una nueva variable que mida la diferencia entre el comportamiento del local y su entorno.

Esta nueva variable consistirá en una estandarización de los litros y euros de cerveza, agua o distribución de un local según su población. Es decir, supongamos que la media de litros vendidos de cerveza en septiembre de 2018 en una ciudad es 70 y la desviación típica de los locales de esa población respecto a la media es 0,5. A continuación consideraremos el caso de un local de dicha población que ese mes compró 60 litros de cerveza. Pues nuestra nueva variable consistirá en restar a los 60 litros la

media poblacional, para posteriormente dividirla entra la desviación típica, es decir:

$$\frac{60-70}{0{,}5}$$

.

Variación de clientes

Otro de nuestros intereses será conocer la variación de clientes mensuales, es decir, el número de locales mensuales que realizan un pedido de cerveza. En la Figura 1.4 se muestra el número de clientes mensuales en la población que hemos considerado como estándar; en ella podemos observar cómo el número de clientes tiende a mantenerse estable, pero aún así se puede observar que presenta cierta estacionalidad, debido a que en verano hay un mayor número de clientes. Por este motivo crear una variable que informe al modelo acerca del mes en el que se realizará la predicción, podría corregir esta pequeña estacionalidad. En el caso de las otras dos ciudades, observamos un comportamiento bastante similar.

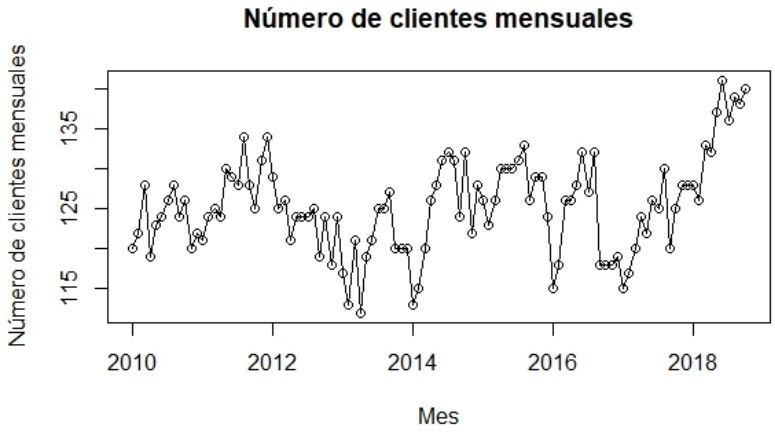


Figura 1.4: Número mensual de clientes en la ciudad estándar.

Estacionalidad en los datos de Agua

En el caso de ventas de agua como era de suponer también observamos que presenta estacionalidad, tal y como se muestra en la Figura 1.5.

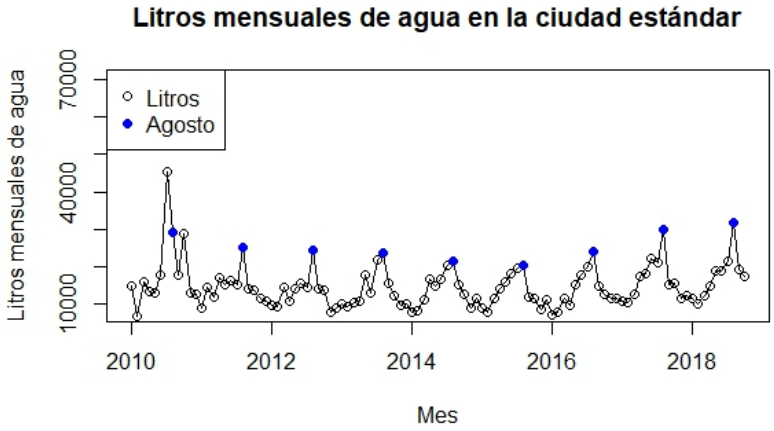


Figura 1.5: Ventas mensuales en litros de agua en la ciudad estándar.

Variación en los datos de Distribución

En la Figura 1.6 se muestran los datos de ventas mensuales de distribución en la ciudad que hemos considerado como estándar. Como podemos observar, a partir de 2017 se aprecia un ascenso notorio en las ventas de distribución, lo que se debe a un gran acuerdo con una de las multinacionales de bebidas más importantes, a través del cual Estrella Galicia se encarga de la distribución de sus productos desde enero de 2017 en algunos territorios. Este hecho también debe ser tomado en consideración en la construcción del modelo, ya que no es lo mismo un pedido de distribución en 2013 que en 2017.

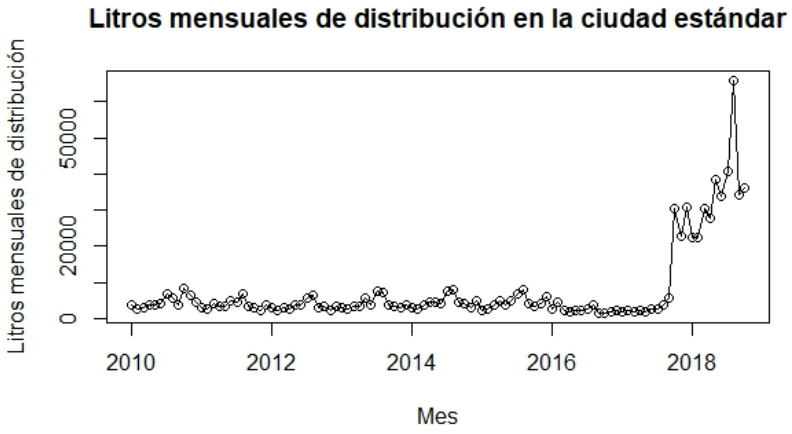


Figura 1.6: Ventas mensuales de litros de distribución en la ciudad estándar.

1.3. Tratamiento de los datos

1.3.1. Diagnóstico de abandono

Uno de los retos que hemos de afrontar en este trabajo es poder diagnosticar cuando un local deja de comprar cerveza, lo cual denominaremos como abandono.

El criterio que se estableció a lo largo de este trabajo es que si un local lleva dos meses seguidos sin comprar cerveza, se clasifica como abandono. Sin embargo, si un local compra sólo un mes y no vuelve a comprar, o sólo deja de comprar un mes y después sigue comprando lo consideraremos como un atípico y no como que volvió a comprar o dejó de comprar respectivamente.

Ejemplo práctico

Para ver esto con más detalle consideraremos el siguiente caso práctico, en el cual se muestra un vector dicotómico que toma el valor 0 si el local no realizó una compra de cerveza ese mes y 1 en el caso contrario.

1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0

En nuestro caso, aplicando el criterio anterior, consideraríamos que hubo dos intervalos de compra en ese local, los cuáles aparecen representados en color azul.

Preparación, limpieza y depuración de los datos

Una vez observado el comportamiento de los datos y definido lo que entendemos por abandono, estamos ante una serie de problemas y es que no conocemos qué locales han sufrido abandono ni en qué momento dejaron de ser clientes ni que locales en la actualidad se encuentran abiertos o cerrados. Además nuestro objetivo es obtener la probabilidad de que un local vaya a abandonar y simplemente disponemos de un historial mensual de pedidos.

Debido a esto, antes de centrarnos en las técnicas que nos permitirán calcular la tasa de abandono, nuestro objetivo será preparar, limpiar y depurar el conjunto de datos quedándonos sólo con la información que nos interesa. Esta preparación es necesaria debido a que los datos que disponemos es el historial mensual de ventas de todos los locales, por tanto, la preparación de los datos es un aspecto clave. Además se realizan tareas de limpieza y depuración de datos debido a que cuanta más información nos aporte el conjunto de datos y más depurados estén, los métodos nos proporcionarán un mejor ajuste y precisión. Entre los procesos de depuración y limpieza de datos está la supresión de valores NA's (valores no válidos) debido a errores de los distribuidores, como puede ser falta de datos en el pedido y de datos nulos, es decir, pedidos de 0 litros y 0 euros.

La preparación de los datos se basa en el siguiente procedimiento:

1. Separamos los datos en locales.

2. Localizamos qué locales sufrieron abandono y en qué momento.

3. Modificamos los datos de modo que cada fila represente el historial de un local desde que empezó a comprar cerveza hasta que deja de comprar (abandona) o hasta dos meses antes del mes que queremos predecir, quedándonos sólo con la información que nos interesa.

Tomando como ejemplo el caso anteriormente comentado en la Sección 1.3.1, en el conjunto de datos original los datos de ventas del primer local constan de 126 filas o pedidos y 15 columnas. Nuestro objetivo será reducir considerablemente esta gran cantidad de datos mediante la construcción de una

nueva tabla, la cual nos proporcionará la información necesaria para la creación del algoritmo que queremos desarrollar.

Como observamos en el ejemplo anterior, el local presentó dos intervalos en los que fue cliente antes de que abandonase. Por tanto nuestro propósito es crear una tabla nueva de datos la cual consistirá de 2 filas en dónde cada fila represente el intervalo en el que el local fue cliente, almacenando la información que puede ser interesante para el desarrollo del trabajo.

En esta tabla, cada columna nos proporcionará la siguiente información:

- **LOCAL ID, POBLACION, PROVINCIA, TIPO INSTALACION, ACTIVIDAD LOCAL, TIPO NEGOCIO**: Explicadas en el apartado anterior.
- **ABANDONO**: Nos indica si en el local en ese periodo abandonó ("SI") o no ("NO").
- **SIGUIO SEGMENTADO**: Nos permite identificar aquellos locales que dejaron de comprar cerveza, ABANDONO, pero que compraron el mes siguiente algún producto de agua o de distribución. Esto nos permitirá realizar una división entre posibles bares que cerraron y posibles bares que van a la competencia. Esta variable presenta tres valores: "NO ABANDONO" si el local no presenta abandono, "COMPETENCIA" si al mes siguiente al abandono el local realizó alguna compra de agua, distribución o los dos y "CERRO" si en el mes siguiente al abandono el local no realizó ninguna compra a los distribuidores.
- **SIGUIO**: Ampliación de la variable SIGUIO SEGMENTADO en la cual los locales que se clasificaron como COMPETENCIA se dividen en "AGUA" si al mes siguiente del abandono se realizó exclusivamente un pedido de agua, "DISTRIBUCION" si al mes siguiente al abandono se realizó exclusivamente un pedido de distribución o "DOS" en el caso de que al mes siguiente se hayan realizado los dos pedidos.
- **COMUNIDAD, PAIS, PROVINCIA**: Nos proporciona la provincia, comunidad autónoma y el país en el cual se sitúa el local, ya que dentro del canal HORECA tenemos los datos de la península ibérica, es decir, España y Portugal.
- **DURACION**: Número de meses en los que fue cliente hasta el abandono o hasta los dos meses anteriores al mes que queremos predecir.
- **VENTAS LITROS PRIMER MES, ULTIMO MES Y PENULTIMO MES CERVEZA**: Nos muestran las ventas en litros de cerveza del mes en el cual el local realizó el primer pedido y los litros de los dos últimos meses del local.
- **VENTAS EUROS PRIMER MES, ULTIMO MES Y PENULTIMO MES CERVEZA**: Nos muestran los ingresos de cerveza del mes en el cual el local realizó el primer pedido y los ingresos de los dos últimos meses.
- **VENTAS LITROS PRIMER MES, ULTIMO MES Y PENULTIMO MES AGUA**: Nos muestran las ventas en litros de agua del mes en el cual el local realizó el primer pedido y los litros de los dos últimos meses.
- **VENTAS LITROS PRIMER MES, ULTIMO MES Y PENULTIMO MES DISTRIBUCIÓN**: Nos muestran las ventas en litros de distribución del mes en el cual el local realizó el primer pedido y los litros de los dos últimos meses antes de abandonar.
- **MES EMPEZO**: Nos indica el mes y el año en el cual el local se convirtió en cliente. A su vez creamos las variables MES EMPEZO SIMPLE y ANO EMPEZO SIMPLE, de modo que cada variable nos suministra el mes y el año respectivamente.

- **MES ACABO**: Nos informa del mes y año que se produjo el abandono en el caso de producirse. A su vez creamos dos variables MES ACABO SIMPLE y ANO ACABO SIMPLE, de modo que cada variable nos suministra el mes y el año respectivamente.
- **MESES VENDIO EG, 1906 Y DESGLOSE**: Meses totales en los que mientras fue cliente realizó pedidos de EG, 1906 y desglosados.
- **DISTRIBUCIÓN**: Nos indica si en el periodo en el que fue cliente el local, se realizó al menos un pedido de distribución.
- **AGUA**: Nos indica si en el periodo en el que fue cliente el local, se realizó al menos un pedido de agua.
- **LOCALES SITIO**: Nos indica el número de clientes diferentes que hubo hasta fecha de hoy en ese establecimiento.
- **LOCALES POBLACIÓN**: Nos indica el número de locales que hay en la población del establecimiento.
- **VENTA LITROS PRIMER, PENULTIMO Y ULTIMO MES DE CERVEZA, AGUA y DISTRIBUCIÓN ESTANDARIZADO**: Debido a que nuestros datos se ven influenciados por la zona y el mes a causa de que presentan estacionalidad, creamos esta variable como idea para estandarizar estas dos cosas, la cual consiste en coger los litros vendidos de agua, distribución o cerveza, restarle la media de ventas mensuales de la población en la que ésta el local y dividirlo entre la desviación típica de todos los residuos.
- **VENTA LITROS PRIMER, PENULTIMO Y ULTIMO MES DE CERVEZA, AGUA y DISTRIBUCIÓN MEDIA**: En este caso realizamos lo mismo que en el caso anterior, pero sin dividir entre la desviación típica.
- **POBLACIÓN**: Conjunto de 4 variables que muestran qué locales se encuentran en una población que supere los 10000, 20000, 30000 y 40000 habitantes. Esta variable se seleccionó debido a que se observó que mientras que los locales que más abandonan están en poblaciones turísticas, los locales que clasificamos que van a la competencia suelen ser locales que están en grandes poblaciones, por ejemplo en este caso, las 7 poblaciones en las que más locales se fueron a la competencia coinciden con las 7 ciudades más grandes.

Tras aplicar este procedimiento al conjunto de datos, pasamos de más de 10 millones de filas a un total de 53717 filas y 64 columnas.

Ejemplo práctico

Tras aplicar el tratamiento de datos, en el caso del primer local, los nuevos datos tendrían el aspecto que se muestra en el Cuadro 1.2, dónde aparecen representadas algunas de las variables comentadas anteriormente. Debido a problemas de tamaño no introducimos todas las variables.

LOCAL ID	POBLACION	PROVINCIA	MES EMPEZO	MES ACABO	EUROS PENULTIMO MES CERVEZA	LITROS PENULTIMO MES CERVEZA	DURACION
2	X	Y	201001	201210	35.5	15.84	34
2	X	Y	201702	201801	0.2942	0.2946	12

ABANDONO	TIPO INSTALACION	MESES VENDIO EG	MESES VENDIO 1906	ACTIVIDAD	TIPO NEGOCIO	EUROS PRIMER MES CERVEZA	EUROS ÚLTIMO MES CERVEZA
SI	Barril	31	22	Sin Especificar	101	81.7000	96.5200
SI	Barril	12	10	Sin Especificar	101	122.9788	86.8258

LITROS PRIMER MES CERVEZA	LITROS ÚLTIMO MES CERVEZA	MESES EG ESPECIAL	MESES 1906 SIMPLE	MESES SIN ALCOHOL	MESES RED	MESES SHANDY	MESES BODEGA
39.60	39.60	31	22	0	0	0	0
45.84	31.68	12	10	0	0	0	0

Cuadro 1.2: Ejemplo de los datos obtenidos tras la transformación.

1.4. Nuevo conjunto de datos

Una vez realizada la transformación de los datos, la cual requiere de una gran carga computacional, obtenemos una nueva tabla la cual consta de 53717 filas (locales) de modo que cada fila representa el historial desde que el local es cliente hasta el abandono o la actualidad y 64 columnas (características del local).

Los datos transformados presentan las siguientes características:

1.4.1. Propiedades del nuevo conjunto de datos

- De los 53717 locales, 15047 (28 %) no sufrieron ABANDONO, mientras que 38670 (72 %) SI. Esto es debido a que estamos en un historial desde que el local empezó a ser cliente hasta que se produjo el abandono o la actualidad en la cual sigue siendo cliente. Aún así, esta proporción de abandono no es realista en comparación con los datos que queremos predecir, es decir, para realizar la predicción de julio de 2018 de la cual hablaremos posteriormente con detenimiento, tan sólo un 3 % de los locales abandonaron.

- Mientras que en los locales que no abandonaron, la correlación de litros de cerveza del último mes de compra con los litros de cerveza del penúltimo mes es de 0.86, en los locales que abandonaron la correlación es 0.40.

- De los locales que se clasificaron como ABANDONO, 33546 al mes siguiente no compraron ni agua ni distribución, es decir, consideramos como que cerró el local. Mientras que 5124 se clasificaron como que fueron a la competencia, por tanto, al mes siguiente del abandono realizaron al menos un pedido de agua o distribución. Esto representa un 13.25 % de los locales que abandonan.

- En la Figura 1.7 se muestra el número de locales que abandonaron en cada mes. Podemos observar como los meses de agosto, septiembre, octubre y diciembre suelen ser los meses en los cuales tienden a abandonar un mayor número de locales.

- También podemos observar en la Figura 1.8 el histograma y la función de densidad (color rojo) de los meses de duración de los locales que abandonaron. Podemos ver como a medida que van avanzando los meses, menos locales tienden a abandonar. Tras observar el histograma, se comprobó tanto de manera gráfica como analítica que no seguía una distribución exponencial, Weibull o chi-cuadrado.

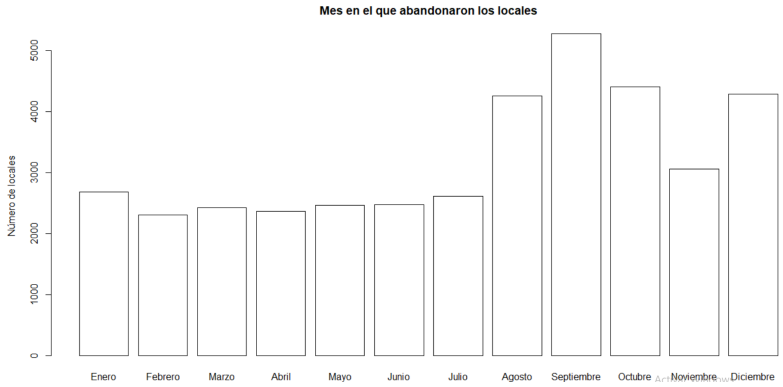


Figura 1.7: Diagrama de barras de los meses en los cuales abandonaron los locales.

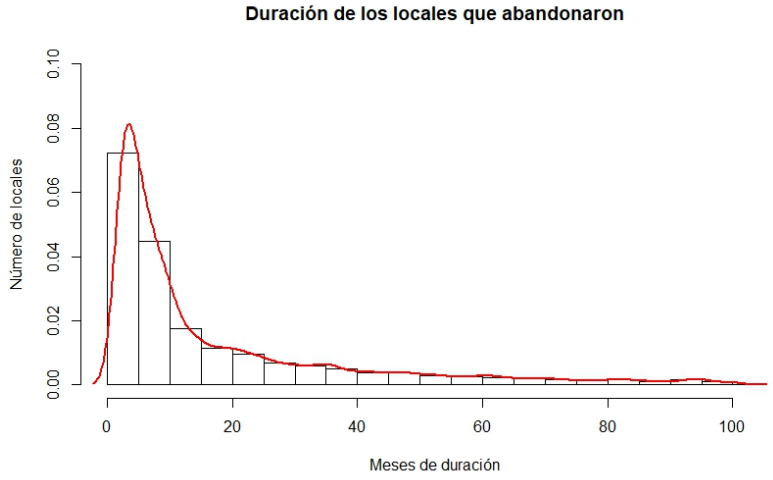


Figura 1.8: Histograma que nos muestra la duración en meses del local antes del abandono.

- Para el caso de los locales que se han ido a la competencia, se han obtenido unos resultados bastante similares a los obtenidos en las Figuras 1.7 y 1.8.
- En total disponemos de un total de 4305 locales que llevan siendo clientes desde enero de 2010 hasta la actualidad, los cuales se distribuyen en: 2352 en la provincia de A Coruña, 391 en la provincia de Lugo, 408 en la provincia de Ourense y 1154 en la provincia de Pontevedra.
- Observemos en el Cuadro 1.3 la diferencia de media de litros de cerveza vendida entre el penúltimo y último mes de cerveza entre los locales que no abandonaron, locales que abandonaron y locales que fueron a la competencia. Mientras que en los locales que abandonaron se nota un descenso de compra de cerveza entre el penúltimo y el último mes, en los locales que fueron a la competencia pasa lo opuesto, es decir, observamos un aumento. En el caso de los locales que no abandonaron, volvemos a detectar otro problema, y es debido a que al tratarse de un historial de ventas desde que el local empieza a ser cliente hasta que abandona o hasta la actualidad (mayo de 2018), los locales que no abandonaron están formados exclusivamente por locales que han realizado su último pedido en abril o mayo de 2018. Por tanto es normal este aumento debido a que son meses en los que el consumo de cerveza va aumentando.

Media	Locales abandonaron	Locales no abandonaron	Locales competencia
Penúltimo mes	158.83	340.80	98.80
Último mes	115.71	396.80	112.49

Cuadro 1.3: Comparación de la media en litros de cerveza entre el penúltimo y último mes.

- En el caso de la comparación entre los litros estandarizados de cerveza del último y penúltimo mes, se presenta un comportamiento bastante similar al que acabamos de comentar. Cabe destacar que en los bares que abandonan la media es negativa, mientras que en los que finalmente no se produce el abandono es positiva como podemos ver en el Cuadro 1.4.

Media	Locales abandonaron	Locales no abandonaron
Penúltimo mes	-0.124	0.016
Último mes	-0.213	0.007

Cuadro 1.4: Comparación de la media en litros de cerveza estandarizado entre el penúltimo y último mes.

1.5. Creación del conjunto de datos sobre el que realizaremos las predicciones.

Como hemos observado en la sección anterior, el conjunto de datos que hemos creado presenta una serie de problemas, y es que la proporción de locales que abandonan no se asemeja al conjunto de datos sobre los cuales realizaremos la predicción. Además debido a que se basa en un historial de ventas hasta mayo de 2018, los locales que no han abandonado han realizado su último pedido de cerveza exclusivamente en los meses de abril o mayo de 2018.

Para solucionar esto crearemos un nuevo conjunto de datos de modo que se asemeje al conjunto de datos sobre el cual realizaremos la predicción. Por ejemplo, el conjunto de datos del mes de julio 2018 sobre el que queremos predecir el abandono estaría formado por el historial de los locales que realizaron un pedido de cerveza en julio de 2018 y nuestro objetivo será obtener la probabilidad sobre cada local de que los dos meses posteriores no realice ningún pedido de cerveza, y por tanto, abandone.

Este nuevo conjunto de datos estaría formado por 12 subconjuntos, es decir, debido a que podemos obtener el historial de los locales que realizaron pedido los meses anteriores y además conocemos si al final acabaron abandonando o no, consideramos los datos que se utilizarían para realizar las predicciones entre los meses de junio 2017 y mayo 2018, los unimos y este sería nuestro nuevo conjunto de datos el cual utilizaremos para obtener la tasa de abandono.

Esta creación del nuevo conjunto de datos, requiere de un gran coste computacional (dado que se necesitan hacer 12 transformaciones independientes) y tarda alrededor de 20 horas de computación, pero mejora considerablemente las predicciones obtenidas.

Debido al elevado coste computacional que conlleva realizar estos 12 subconjuntos y ya que se pueden compilar de manera totalmente independientes entre sí, se trabajó en paralelizar el proceso. Esto

permite acelerar la obtención del conjunto de entrenamiento aprovechando al máximo las características de nuestra computadora o máquina virtual. Para este trabajo, se empleó una máquina virtual con 8 núcleos y 24 gigas de RAM.

En la Figura 1.9 se muestra de manera gráfica como se construiría el nuevo conjunto de datos que será el que utilicemos para la construcción del modelo.

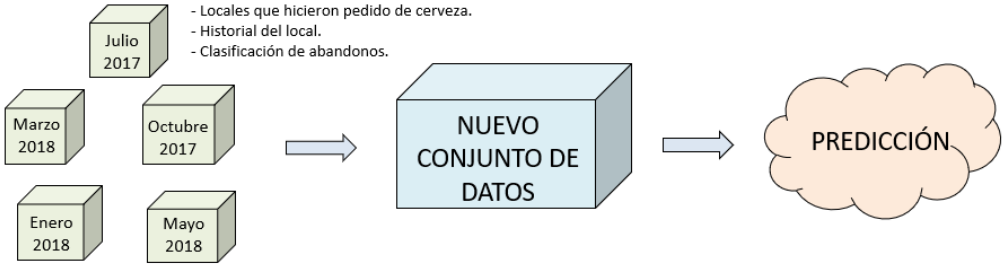


Figura 1.9: Ejemplo gráfico del nuevo conjunto de datos.

1.5.1. Propiedades del nuevo conjunto de datos

Este nuevo conjunto de datos, consta de un mayor número de filas, 178895, de modo que cada fila representa el historial de un local hasta uno de los meses anteriores y 64 columnas (características).

Además este nuevo conjunto de datos, corrige los "errores" que encontramos en el conjunto de datos creado en la sección 1.3. Es decir:

- De los 178895 locales, 173527 (97 %) no sufrieron ABANDONO, mientras que 5368 (3 %) SI.
- La diferencia de media entre el penúltimo y último mes de los locales que no abandonaron se estabiliza, mientras que en los locales que abandonaron o fueron en la competencia se siguen comportando en la misma línea.

Media	Locales abandonaron	Locales no abandonaron	Locales competencia
Penúltimo mes	158.83	340.80	98.80
Último mes	115.71	396.80	112.49

Cuadro 1.5: Comparación de la media en litros de cerveza entre el penúltimo y último mes.

Capítulo 2

Técnicas a emplear

Una vez creado el conjunto de datos sobre el cual realizaremos la predicción, nuestro objetivo será obtener la tasa de abandono.

Para ello, denotaremos como X el conjunto de datos que acabamos de crear, y como $X_1, ..., X_p$ el conjunto de variables explicativas, como pueden ser litros de cerveza del último mes, mes en el cual realizamos la predicción, etc. Nuestro objetivo es poder predecir la variable respuesta Y, en nuestro caso ABANDONO, en función de los resultados obtenidos por las variables explicativas. Para lograr este objetivo se han probado diversas técnicas como pueden ser la regresión logística, redes neuronales, árboles de decisión, Random Forest, etc. obteniendo con el método Random Forest los mejores resultados.

A lo largo de este capítulo comentaremos las técnicas empleadas, observando como a partir de una idea bastante sencilla e intuitiva como son los árboles de decisión CART (acrónimo de Classification And Regression Trees) introducido por Breiman en el año 1984, dicha idea ha ido evolucionando tanto por Breiman (Breiman, 1996) y (Breiman, 2001) como por otros autores a lo largo de los años, derivando en técnicas más eficientes y complejas como son las técnicas bagging y boosting.

2.1. Árboles de Decisión

Uno de los métodos utilizados para la predicción de una variable son los Árboles de Decisión. Dependiendo de la categoría de la variable respuesta que queremos predecir, clasificamos los árboles de decisión en dos grandes conjuntos: Árboles de Regresión (cuando la variable respuesta es continua) y Árboles de Clasificación (cuando la variable respuesta es discreta). En nuestro trabajo, como el objetivo es predecir el ABANDONO, nos centraremos en los Árboles de Clasificación.

Tanto los árboles de clasificación como los de regresión se basan en la sencilla idea de desarrollar el modelo en forma de árbol, tal y como se muestra en la Figura 2.1. En primer lugar se empieza con todo el conjunto de datos (en nuestro caso sería el nuevo conjunto de datos que acabamos de crear) y, mediante divisiones binarias, se va dividiendo el conjunto de datos a través de ramificaciones sobre alguna de las variables explicativas (como puede ser litros de cerveza del último mes, población en la que se sitúe el local, etc.) hasta alcanzar una de las condiciones de parada de las cuales hablaremos posteriormente. Una vez alcanzada la condición de parada, el árbol dejará de ramificarse y el conjunto de datos que queda en ese nodo se denominará nodo terminal. Por ejemplo, en la Figura 2.1, los nodos terminales serían los nodos representados en color azul, mientras que los cuadrados blancos, los denominaremos nodos o ramas.

Esta distinción entre árboles de clasificación y regresión (CART) (Breiman, 1984) es necesaria debido a que utilizan distintos criterios para la división del árbol. Mientras que en los árboles de regresión se realiza la división que minimice la suma de residuos al cuadrado (RSS), en los árboles de clasificación, ante la imposibilidad de calcularla, existen múltiples alternativas con el objetivo de

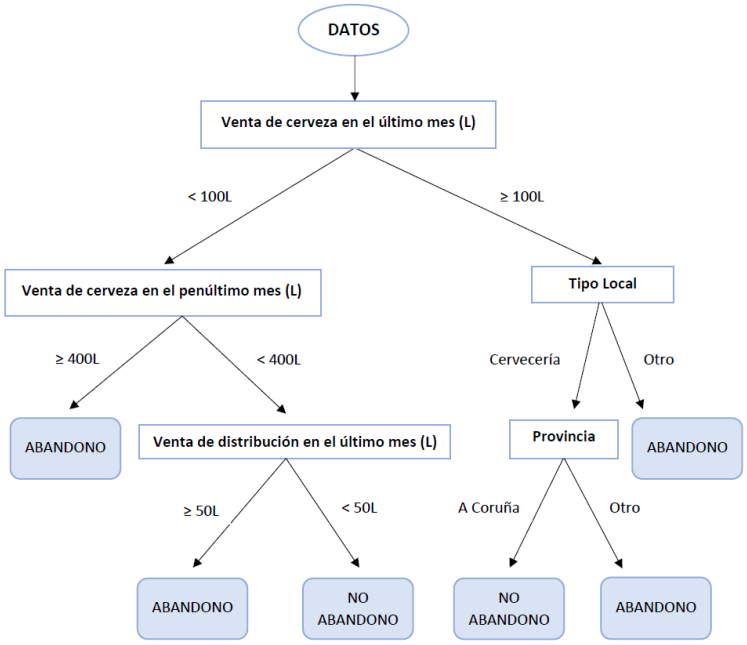


Figura 2.1: Ejemplo gráfico de un Árbol de Decisión.

encontrar las mejores divisiones posibles, es decir, ir separando mediante las divisiones más adecuadas las diferentes clases de la variable respuesta.

Acompañaremos la explicación de las siguientes medidas mediante un ejemplo práctico:

Imaginemos que disponemos de 50 locales los cuales se dividen en 3 clases de la siguiente manera: 30 siguen siendo clientes, 8 acaban cerrando y 12 van a la competencia. Sobre este conjunto se utiliza la variable litros de cerveza en el último mes para la división, dando dos ramas según el punto de corte. La rama de la izquierda, que denominaremos rama A tiene 2 locales que siguen siendo clientes, 4 que cierran y 8 que van a la competencia, y la rama de la derecha, que denominaremos como rama B, consta de 28 locales que siguen siendo clientes, 4 que cierran y 4 que finalmente van a la competencia. Este ejemplo lo podemos ver de manera gráfica en la Figura 2.2.

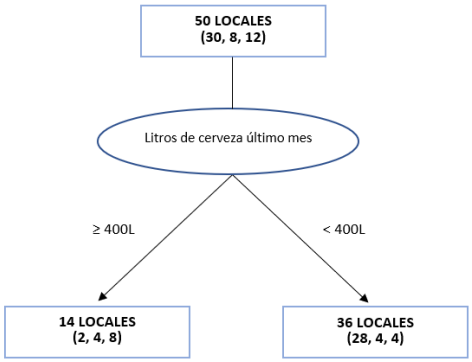


Figura 2.2: Representación gráfica del ejemplo.

Las medidas más empleadas como criterios de división para los árboles de clasificación son:

Classification error rate: Se basa en el porcentaje de observaciones que no pertenece a la clase más común del nodo, considerando como clase cada uno de los posibles grupos de la variable respuesta Y.

$$E_m = 1 - max_k(\hat{p}_{mk}) \tag{2.1}$$

Siendo max_k la clase a la cual pertenece la mayoría de los datos y $\hat{p}_{mk}$ la proporción de observaciones dentro de una rama m que pertenece a la clase k.

En nuestro ejemplo, el clasiffication error rate de la rama A se calcularía de la siguiente manera:

$$1 - (8/14) = 0.42$$

.

Siendo 8 la clase mayoritaria y 14 el conjunto total de muestras de la rama A.

Mientras que para la rama B sería:

$$1 - (28/36) = 0.22$$

.

En el caso de la rama B, la clase mayoritaria tiene 28 miembros de los 36 que hay en total

Como podemos observar, nuestro objetivo es obtener el classification error rate más bajo.

Índice de Gini: El índice de Gini es una métrica que mide la frecuencia con la que un elemento elegido al azar se identifica incorrectamente. Como podemos ver en la fórmula (2.2), nuestro objetivo será obtener un índice de Gini bajo debido a que nos indica una mayor pureza del nodo. Este método de división es el empleado por técnicas como el Random Forest (Breiman, 2001) sobre el cual nos centraremos más adelante.

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2.2}$$

Denotaremos como K al número de clases de la variable respuesta Y.

En nuestro ejemplo, para la rama A obtendríamos:

$$G_A = \frac{8}{14} * \left(1 - \frac{8}{14}\right) + \frac{4}{14} * \left(1 - \frac{4}{14}\right) + \frac{2}{14} * \left(1 - \frac{2}{14}\right) = 0.57$$

De la misma forma, para la rama B obtendríamos:

$$G_B = \frac{4}{36} * \left(1 - \frac{4}{36}\right) + \frac{4}{36} * \left(1 - \frac{4}{36}\right) + \frac{28}{36} * \left(1 - \frac{28}{36}\right) = 0.37$$

Entropía: La entropía es una manera de contabilizar la dispersión de cada clase. Si una división presenta valores de una sola clase se considerará que es una división pura y por tanto, su entropía es cero; mientras que si la frecuencia de cada clase es la misma, el valor de la entropía alcanza el valor máximo de 1. Por tanto nuestro objetivo es crear reglas que permitan diferenciar las distintas clases de la variable respuesta obteniendo un valor bajo de entropía. El algoritmo C5.0. (Quinlan, 1997) del cual hablaremos con más detalle a posteriori, utiliza esta medida.

$$D_m = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}) \tag{2.3}$$

En nuestro ejemplo, para la rama A tendríamos:

$$D_A = -\frac{8}{14} * \log\left(\frac{8}{14}\right) - \frac{4}{14} * \log\left(\frac{4}{14}\right) - \frac{2}{14} * \log\left(\frac{2}{14}\right) = 0.95$$

Igualmente, para la rama B:

$$D_B = -\frac{4}{36} * \log\left(\frac{4}{36}\right) - \frac{4}{36} * \log\left(\frac{4}{36}\right) - \frac{28}{36} * \log\left(\frac{28}{36}\right) = 0.68$$

En la práctica, para realizar la división del árbol de clasificación se recomienda utilizar estas dos últimas medidas debido a que el classification error rate no es suficientemente sensible para la creación de buenos árboles (James et al., 2013).

Por desgracia, debido a que por motivos computacionales no es factible en la práctica comprobar todas las posibilidades de división, se recurre a un método conocido como "recursive binary splitting", el cual no evalúa todas las posibles divisiones pero si que alcanza un buen equilibrio computación-resultado. Caso similar a como actúa la selección de predictores (AIC, BIC) en regresión lineal múltiple.

El objetivo de este método es encontrar en cada división del árbol una variable explicativa X_j, $j \in 1, \cdots, p$, por ejemplo, en nuestro caso podría ser los meses que el local realizó un pedido de red vintage, y un valor c tal que si se realiza la división $\{X|X_j < c\}$ y $\{X|X_j \geq c\}$, ésta consiga la menor entropía o índice de Gini en el caso de árboles de clasificación, o el menor RSS en el caso de árboles de regresión. El algoritmo de división del árbol de clasificación funciona de la siguiente manera:

1. El proceso se inicia en lo más alto del árbol, donde todas las observaciones pertenecen a la misma región. En la práctica se suele tomar alrededor del 70 % de las observaciones para la construcción del árbol (conjunto de entrenamiento) y el 30 % restante de los datos se utiliza para comprobar el buen funcionamiento del árbol (conjunto de validación). Es decir, creamos el árbol a partir de la muestra de entrenamiento y, una vez creado, comprobamos su eficacia mediante el conjunto de validación. Esto se debe a que el árbol puede funcionar bien en los datos usados para la construcción del árbol pero sin embargo funcionar mal para realizar predicciones (problema conocido como sobreajuste).

2. Se identifican todos los posibles puntos de corte c sobre los que se dividirá el árbol para cada una de las variables respuesta $(X_1, \cdots, X_p)$. En el caso de de variables cualitativas los posibles puntos de corte son cada uno de sus niveles, mientras que para valores continuos, se ordenan de menor a mayor sus valores muestrales y se selecciona el punto medio entre cada par de valores como punto de corte.

3. Se calcula la medida deseada, ya sea índice de Gini o la Entropía, para los dos conjuntos que se formarían tras aplicar la división, los cuales definimos como A y B y se ponderaría de la siguiente manera:

$$p_A * \text{pureza A} + p_B * \text{pureza B} \tag{2.4}$$

   Siendo p_A y p_B el promedio de observaciones que van a los nodos A y B respectivamente. En el caso del ejemplo anterior, de 50 locales que teníamos antes de la división, 14 fueron a la rama A, por tanto $p_A = \frac{14}{50} = 0.28$, mientras que 36 acabaron en la rama B de modo que $p_B = \frac{36}{50} = 0.72$.

Volviendo entonces a nuestro caso práctico, considerando el índice de Gini para la división, la medida ponderada sería:

$$0.28 * 0{,}57 + 0.72 * 0{,}37 = 0.42$$

4. El menor valor obtenido es seleccionado como división óptima y por tanto se realiza la división.

5. Se repite el proceso para cada una de las regiones que se han creado en la división anterior hasta alcanzar alguna norma de parada.

2.1.1. Sobreajuste

Como comentamos anteriormente, la construcción de un árbol de decisión se realiza sobre un conjunto de entrenamiento debido a que en la construcción del árbol hay un problema denominado sobreajuste. Es decir, el modelo se ajusta demasiado bien a los datos de entrada conllevando consigo un empeoramiento de las predicciones. Esto se debe a que si en el modelo no le introducimos unas condiciones de parada, éste crecerá dividiéndose hasta por ejemplo alcanzar tantos nodos terminales como datos hayan entrado en el árbol.

Para solucionar este problema disponemos de dos alternativas:

La primera es mediante la selección de unos criterios de parada escogidos previamente a la construcción del árbol por nosotros, denominados hiperparámetros, entre los que destacan:

- **Nodesize**: Número mínimo de datos que debe haber en un nodo terminal.
- **Profundidad máxima del árbol**: Número máximo de divisiones de la rama más larga.
- **Número máximo de nodos terminales**: Número máximo de nodos terminales que puede tener un árbol.
- **Reducción mínima del error**: Reducción mínima de error que debe tener una división para llevarse a cabo.

La otra alternativa es la poda del árbol, la cual consiste en la construcción de un árbol de decisión sin prácticamente condiciones de parada para posteriormente seleccionar el mejor sub-árbol, entendiendo como el mejor sub-árbol la partición del árbol de decisión que consigue un error más bajo en el test.

2.1.2. Predicción

Una vez creado y ajustado el árbol de regresión o clasificación, procedemos a la predicción de los nuevos conjuntos de datos. Esta predicción se realiza empezando por el principio del árbol y descendiendo mediante los criterios de división hasta llegar a un nodo terminal. En el caso de árboles de regresión el valor de la predicción está formado por la media de la variable cuantitativa, mientras que en los árboles de clasificación está formado por la moda de la variable cualitativa. Por este motivo los árboles de decisión suelen funcionar mejor como método de clasificación debido a que discretizan la variable continua perdiendo parte de su información. En el caso de la clasificación puede aportarse el porcentaje de cada clase en el nodo terminal, lo que aporta información sobre la confianza de la predicción. Este porcentaje será nuestro objetivo, ya que nos permitirá conocer la tasa de abandono de un local.

2.2. Técnicas de ensemble: Bagging y Boosting

Como en todo estadístico, los árboles de predicción sufren el problema de equilibrio entre el sesgo y la varianza. Mientras que el sesgo nos indica la diferencia en promedio entre las predicciones y los valores reales, la varianza nos informa de la variación de las estimaciones dependiendo de la muestra empleada en el entrenamiento.

Observemos el ejemplo gráfico mostrado en la Figura 2.3 en la que se muestra como afecta el sesgo y la varianza a las predicciones del modelo. Nuestro objetivo será obtener un modelo con poca varianza y poco sesgo.

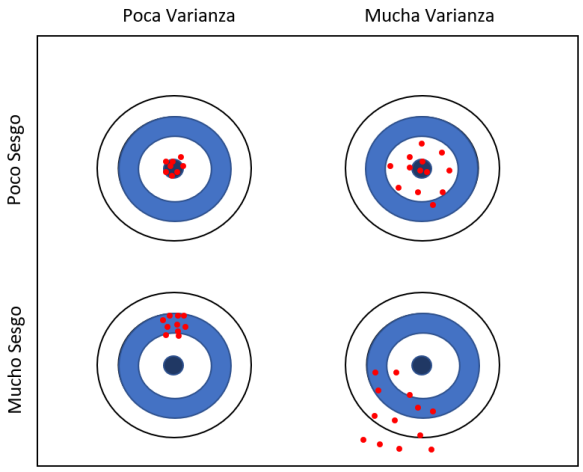


Figura 2.3: Ejemplo gráfico relación Sesgo-Varianza.

Como comentamos anteriormente, a medida que la complejidad del modelo aumenta, es decir, trabajamos con árboles más ramificados, disponemos de menos sesgo debido a que el árbol cada vez se asemejará más al conjunto de entrenamiento. Sin embargo, tendremos una mayor varianza debido a que el árbol funcionará mal ante futuras predicciones (sobreajuste). Por otra parte, si construimos árboles sencillos, no se representará bien la combinación entre las variables, con lo cual tendremos un alto sesgo ya que las predicciones poco se asemejarán al conjunto de entrenamiento, pero tendremos poca varianza debido a que al tratarse de un árbol más sencillo, no cobrará demasiada importancia el conjunto de datos tomado como conjunto de entrenamiento. El objetivo de las técnicas conocidas como ensemble, basadas en la creación de múltiples árboles, se centran en la reducción del error del modelo (sesgo y varianza).

Entre las técnicas de ensemble destaca el bagging y el boosting. Aunque estas dos técnicas presenten la misma finalidad y se basen en la construcción de un gran número de árboles B, trabajan de manera prácticamente opuesta.

Mientras que el método bagging se basa en la creación de B árboles independientes sin prácticamente condiciones de parada, es decir, árboles con poco sesgo pero mucha varianza, el boosting consiste en la creación de B árboles sencillos, con poca varianza pero mucho sesgo, llamados weak learners, en el que cada árbol va aprendiendo de los errores del anterior.

2.3. Bagging

El método Bagging (Breiman, 1996) parte de la sencilla idea de que si disponemos de n observaciones independientes $Z_1, \cdots, Z_n$ de modo que $Var(Z_1) = Var(Z_2) = \cdots = Var(Z_n) = \sigma^2$, entonces la

varianza de su media $\bar{Z}$ es $Var(\bar{Z}) = \frac{\sigma^2}{n}$.

Por tanto el método bagging (contracción de boostrap aggregation) se basa en la realización de los siguientes pasos:

1. Se crean B pseudomuestras independientes de modo que cada pseudomuestra estaría formada por 2/3 de los datos del conjunto de entrenamiento. Esto se realiza mediante técnicas de remuestreo bootstrap.

2. De manera totalmente independiente se crean B árboles (uno para cada una de las B pseudomuestras obtenidas) los cuales crecerán prácticamente sin condiciones de parada y no se someterán a poda (pruning), por tanto, estos árboles presentarán poco sesgo pero mucha varianza.

3. Para realizar la predicción de un conjunto de datos se realiza una predicción individual sobre cada uno de los B árboles creados. El resultado final es la media de las B predicciones en el caso de variables continuas y la moda de las B predicciones en variables categóricas. Como en el caso de los árboles de decisión, las técnicas bagging también nos proporcionan una probabilidad numérica a la hora de realizar la predicción.

La Figura 2.4 nos muestra un ejemplo práctico de como funciona el procedimiento bagging.

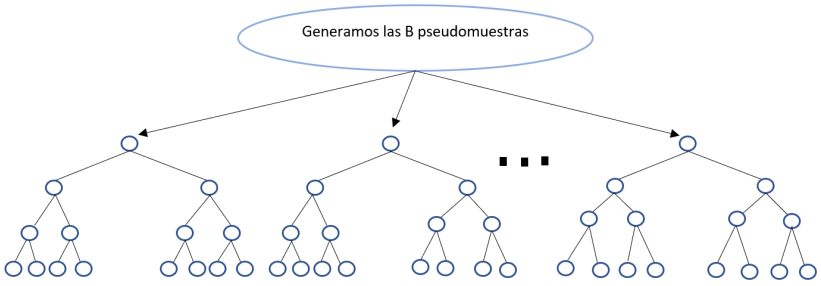


Figura 2.4: Ejemplo Bagging.

Out-of-bag error

Como comentamos antes, en la construcción de cada árbol mediante bagging, cada remuestra bootstrap sólo utiliza 2/3 de los datos del conjunto de entrenamiento. Definimos el out-of-bag (OOB) del árbol j-ésimo como las observaciones del conjunto de entrenamiento que quedaron fuera de la remuestra bootstrap del árbol j-ésimo. Es decir, los datos que no se utilizaron para la construcción del árbol j-ésimo se aprovechan para realizar una validación interna de éste. Por tanto, el out-of-bag error se basa en el tercio de datos restantes que no seleccionó la remuestra boostrap, los cuales se utilizan para predecir ese árbol y con ello calcular el OOB-clasification-error (para árboles de clasificación) o el OOB-mean-square (para árboles de regresión).

Breiman (Breiman, 1996 b) proporciona una prueba empírica de que el estimador out-of-bag es tan preciso como un conjunto de validación externo. Ésto nos ofrece la posibilidad de poder ajustar los hiperparámetros sin necesidad de realizar un conjunto de validación externo con el ahorro de coste computacional que ello conlleva.

Importancia de los predictores

Uno de los inconvenientes de utilizar técnicas bagging comparado con los árboles de decisión es que su interpretación gráfica es más complicada, debido a que mientras los árboles de decisión está formado

por un sólo árbol, los métodos bagging están formados por la combinación de B árboles. Sin embargo, disponemos de dos medidas que nos proporcionan la importancia de cada variable en la construcción del proceso.

- **Incremento del MDA**

  El método funciona de la siguiente manera:

  1. Creamos los B árboles mediante bagging sobre los datos de entrada.
  2. Calculamos el out-of-bag error del modelo. A este valor lo denominaremos como mda_o.
  3. Transformamos los datos de entrada del proceso bagging aleatorizando los valores de la variable X_j, $j \in \{1, \cdots, p\}$ cuya importancia queremos medir.
  4. Como en el caso anterior creamos los B árboles mediante bagging sobre los datos modificados.
  5. Calculamos el out-of-bag error del nuevo modelo. A este valor lo denominaremos como mda_j.
  6. Se calcula el incremento del error debido a la aleatorización de los valores del predictor j.

  $$\%IncMDA = \frac{(mda_j - mda_o)}{mda_o} * 100$$

  Veamos esto con un sencillo ejemplo. Consideremos que queremos medir la importancia de los meses en los que los locales hacen pedidos de agua, el local 1 realizó 2 pedidos de agua, el local 2 realizó 5 pedidos, y el local 3 realizó 12 pedidos. A continuación se aplican las técnicas bagging y se obtiene el out-of-bag error del modelo (mda_o).

  Tras esto, se procede a la aleatorización de los valores de meses en los se realizó pedidos de agua, de modo que por ejemplo, el local 1 ahora realizó 12 pedidos, el local 2 realizó 2 y el local 3 realizó unicamente 5 pedidos. Se vuelve a aplicar las técnicas bagging esta vez con los datos de meses de agua falseados, obteniendo el out-of-bag error del nuevo modelo (mda_j).

  Tras obtener los valores (mda_j) y (mda_o) calculamos el incremento del error que se muestra en el paso 6.

  El objetivo tras la aleatorización es crear valores falsos en la variable que queremos predecir y por tanto, si la variable es importante, el out-of-bag error del nuevo modelo en comparación con el del original debería ser bastante mayor.

- **Incremento de la pureza de los nodos**: Se calcula el valor promedio de medida de pureza empleada, bien sea índice de Gini o entropía en los árboles de clasificación, o MSE en los árboles de regresión, que sufrió el árbol cuando se seleccionó la variable X_j, $j \in \{1, \cdots, p\}$ para la división.

  Cabe destacar que estas medidas nos proporcionan información sobre la influencia de los predictores en el modelo. No confundir la influencia de los predictores en el modelo con la influencia que tienen estas variables sobre la variable respuesta, en nuestro caso ABANDONO.

2.4. Random Forest

Como hemos comentado anteriormente, el bagging se basa en la reducción de la varianza a través de la creación de B árboles creados de manera independiente, pero cuando los árboles están bastante correlados entre sí, la reducción de la varianza puede ser prácticamente nula. Ésto sucede en el caso de que una de las variables respuesta $X_1, ..., X_p$ sea mucho más influyente que el resto de variables, lo que hace que todos o casi todos los árboles B creados estean formados a partir del mismo predictor y por tanto sean muy parecidos entre sí. En nuestro caso, la variable litros de cerveza en el último

mes, es una variable bastante influyente y por tanto, casi todos los árboles utilizan esa variable para la división. Además, se reduciría con ello la fuerza de variables que pueden parecer poco influyentes pero que pueden resultar clave para realizar una buena construcción del modelo.

El Random Forest evita este problema de correlación entre los árboles mediante una selección aleatoria de m, $m \in \{1, ..., p\}$ predictores antes de evaluar cada división, en la práctica se suele utilizar $m << p$. De esta forma un promedio de $\frac{p-m}{m}$ veces no se contemplará al predictor influyente para realizar la división, permitiendo así que otros predictores puedan ser seleccionados añadiendo con ello más diversidad al modelo. Este valor m, recibe el nombre de *mtry*.

Sólo con añadir este paso extra con respecto al bagging se consigue decorrelacionar los árboles, por lo que su implementación consigue una mayor reducción de la varianza.

La diferencia entre bagging y Random Forest consiste básicamente en que en el Random Rorest, antes de cada división, se seleccionan automáticamente m predictores. La diferencia en el resultado dependerá del m escogido. Por tanto, si tomamos $m = p$ el Random Forest es equivalente al bagging.

En gran medida éste método presenta gran éxito debido a su fácil implementación y adaptabilidad a una gran variedad de campos, como pueden ser la medicina, economía, etc. Además funciona bastante bien en muestras de gran dimensión como las que tratamos en este trabajo.

El Random Forest presenta las siguientes características:

- Su precisión es tan buena o incluso mejor que algunas técnicas boosting más complejas computacionalmente, como en el caso del AdaBoost (Breiman, 2001).
- Es relativamente robusto a datos atípicos y a variables de ruido.
- Es más rápido que aplicar técnicas boosting o bagging.
- Nos proporciona un estimador interno (out-of-bag error) que nos permite medir tanto el error como la potencia, correlación e importancia de las variables (Incremento del MDA e Incremento de la pureza de los nodos) sin necesidad de recurrir a validadores externos.
- Permite paralelizar el proceso de manera sencilla debido a que se basa en la creación de B árboles independientes.

2.4.1. Selección de hiperparámetros

En el caso de realizar el Random Forest es muy importante ajustar adecuadamente los hiperparámetros (valores prefijados anteriormente por nosotros antes de la construcción del árbol). Mediante el out-of-bag error realizaremos este ajuste.

Los hiperparámetros que debemos ajustar en el Random Forest son los siguientes:

Elección del parámetro B

Gracias al Teorema 1.2 (Breiman, 2001), el cual demuestra mediante la ley de los grandes números que al aumentar el número de árboles B el out-of-bag error converge, sabemos que seleccionar un número de árboles B, $B \in \mathbb{N}$, no es un hiperparámetro crítico, debido a que tomar un valor de B elevado no nos lleva a un problema de sobreajuste. Sin embargo, dado que llegado a un número elevado de árboles el out-of-bag error tiende a estabilizarse, debemos seleccionar el menor número árboles posibles en el que se alcanza la estabilidad, reduciendo con ello el tiempo de compilación. Por defecto en el paquete RandomForest (Liaw and Wiener, 2018) se considera $B = 500$.

Elección del parámetro mtry

Denominamos como *mtry*, $mtry \in \{1, \cdots, p\}$ al número de variables explicativas consideradas en cada división. Por defecto el valor que se utiliza en el caso de regresión es $p/3$ mientras que para

clasificación se considera $\sqrt{p}$. Por un lado este tamaño puede ser demasiado pequeño, especialmente en presencia de un gran número de parámetros que aportan ruido, ya que por ejemplo algunos árboles podrían estar formados exclusivamente mediante la división de variables de ruido y por tanto crear árboles imprecisos. Por el contrario, si el valor es demasiado grande las variables con efectos moderados, pero que resultan claves para crear una buena predicción, podrían tener menos fuerza en el modelo. Por tanto, además de seleccionar *mtry*, seleccionar las variables claves que entrarán en el modelo es uno de los aspectos más importantes.

Elección del parámetro nodesize

Definimos como nodesize el número mínimo de datos que debe haber en cada nodo terminal. En la práctica se recomienda utilizar un tamaño de *nodesize* pequeño, por defecto se utiliza $nodesize = 1$ para clasificación y $nodesize = 5$ para regresión.

2.5. Boosting

La idea principal de las técnicas boosting consiste en ir ajustando de manera individual B árboles sencillos, los cuales presentan mucho sesgo pero poca varianza, de modo que cada árbol aprenda de los errores del árbol anterior y con ello ir mejorando iteración a iteración. Los principales algoritmos boosting son AdaBoost (Freud et al., 1999), Gradient Boosting (Friedman, 2001), Stochastic Gradient Boosting (Friedman, 2002) y XGBoost (Chen, 2016).

En el caso de las técnicas boosting mencionadas, AdaBoost y Gradient Boosting, no disponen de conjuntos de validación internos como ocurría en el caso de las técnicas bagging.

Además los algoritmos boosting necesitan de más hiperparámetros para la construcción del modelo, debido a que no se cumple el Teorema 1.2 propuesto por (Breiman, 2001). El boosting puede sufrir sobreajuste si el número de árboles B es excesivamente alto. Para evitar esto se necesita emplear un término de regularización conocido como learning rate.

Learning rate (λ): Controla el ritmo de "aprendizaje" de los modelos. Suelen recomendarse valores de 0,01 o 0,001 aunque la elección correcta puede variar dependiendo del problema. Cuanto menor sea el valor λ más árboles se necesitarán para la obtención de buenos resultados, puesto que el aprendizaje se realiza de manera más lenta pero disminuye con ello el riesgo de sobreajuste.

La Figura 2.5 es un ejemplo gráfico del funcionamiento del boosting.

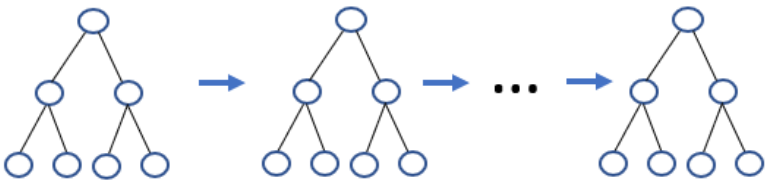


Figura 2.5: Ejemplo Boosting.

2.6. AdaBoost

El algoritmo AdaBoost (abreviación de Adaptive Boosting) (Freund and Schapire, 1999) se basa en la idea anteriormente comentada de las técnicas boosting, las cuales se basan en la creación de B árboles simples de manera que cada árbol vaya aprendiendo de los errores del árbol anterior.

El algoritmo AdaBoost funciona de la siguiente manera:

1. Se crea el weak learner (árbol con pocas ramificaciones) en el cual todos los datos del conjunto de entrenamiento presentan el mismo peso.

2. Se observa qué datos han sido mal clasificados por el modelo, aumentando así su peso para la clasificación del siguiente árbol.

3. Se asigna un peso total al weak learner de manera que mida su influencia en el conjunto de forma proporcional al número de aciertos. Este peso nos ayudará a la hora de realizar la predicción de un nuevo conjunto de datos ya que AdaBoost tendrá más en cuenta a la hora de realizar la predicción a los weak learners que han sido más efectivos.

4. Se vuelve a realizar el proceso B veces creando un weak learner sobre el conjunto de entrenamiento, pero esta vez sobre los pesos actualizados.

La predicción se lleva a cabo de manera que el conjunto de datos realizará una **predicción individual ponderada** sobre cada uno de los B weak learners, es decir, se dará más fuerza a las predicciones de los weak learners que han cometido un menor número de equivocaciones.

2.7. Gradient Boosting

El método Gradient Boosting (Friedman, 2001) se basa en minimizar la función de costes mostrada en la Sección 4.6 de *Greedy funnction approximation: A gradient boosting machine* (Friedman, 2001) a través del método de descenso por gradiente. El método de descenso por gradiente consiste a su vez en ir descendiendo de manera iterativa sobre la función de costes hasta alcanzar su mínimo. En la Figura 2.6 se muestra un ejemplo gráfico del método de descenso por gradiente.

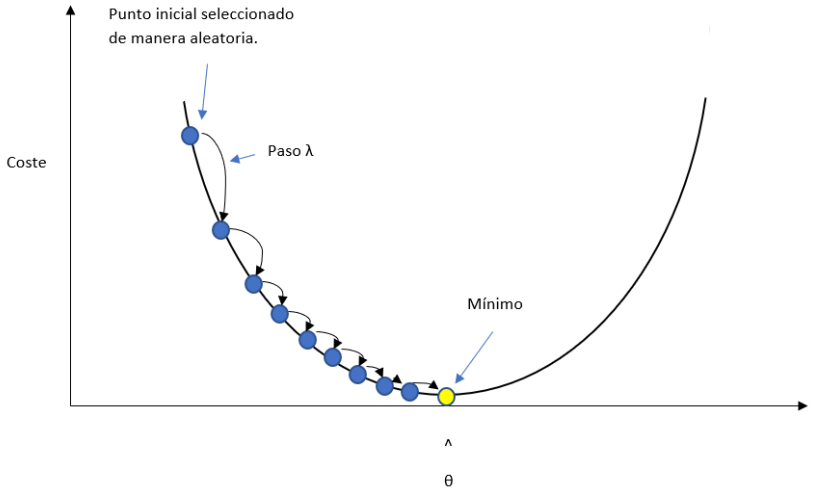
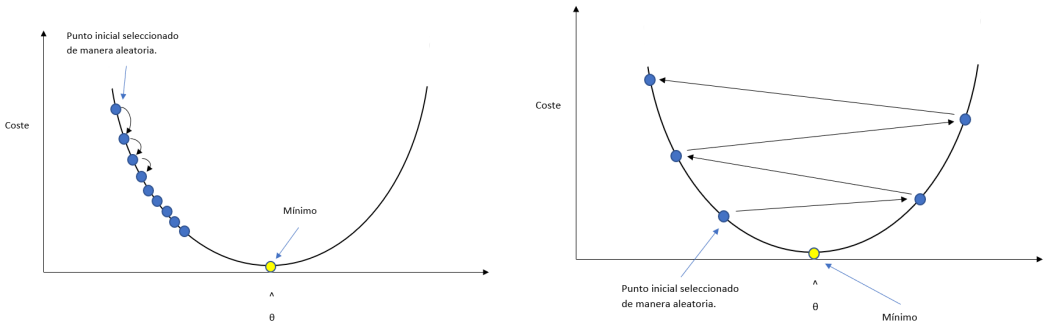


Figura 2.6: Ejemplo gráfico del método de descenso gradiente.

Un parámetro importante del descenso por gradiente es el tamaño de los pasos, cuyo valor es el learning rate que hemos comentado anteriormente. Dicho valor debe ser seleccionado con cuidado, dado que si elegimos un valor demasiado pequeño, el algoritmo necesitará realizar muchas iteraciones para encontrar el mínimo. Por otro lado, si elegimos un valor demasiado alto, puede atravesar el mínimo y acabar más lejos de dónde empezamos. Esto lo vemos representado en la Figura 2.7.



(a) Caso λ pequeño (b) Caso λ grande

Figura 2.7: Problema de elegir adecuadamente el parámetro λ.

De todos modos, el ejemplo mostrado en la Figura 2.6 es un caso bastante sencillo, pueden haber funciones de coste las cuales tengan mínimos locales y dónde encontrar el mínimo global sea bastante difícil. Para mejorar este problema, con el fin de encontrar el mínimo global, surge el método Stochastic Gradient Boosting.

2.8. Stochastic Gradient Boosting

El método Stochastic Gradient Boosting (Friedman, 2002) surge para detectar de manera más efectiva el mínimo global en el caso de haber mínimos locales. Esto se realiza seleccionando para la construcción de cada árbol una submuestra de manera aleatoria del conjunto de datos de entrenamiento (generalmente sin reemplazo) con el objetivo de proporcionarle una cierta aleatoriedad al descenso de gradiente. Aunque esta aleatorización del conjunto de entrenamiento no asegura que se alcance el mínimo global absoluto, puede ayudar a que el algoritmo salte de los mínimos locales y las mesetas y se acerque al mínimo global.

2.8.1. Selección de los hiperparámetros

En el caso de Gradient Boosting y Stochastic Gradient Boosting, los hiperparámetros más importantes son:

- Número de árboles: Debido a que a diferencia del Random Forest, si seleccionamos un números elevado de árboles estos métodos tienden a sobreajustarse, es esencial encontrar el número de árboles óptimo, para lo que se emplearía validación cruzada.

- Número máximo de ramificaciones: Número máximo de ramas que puede tener cada weak learner desde que empezamos con el conjunto de datos de entrenamiento hasta el nodo terminal. En la práctica se suele emplear un valor comprendido entre 1 y 10.

- Learning rate (λ): Controla el ritmo al que van avanzando los dos métodos. Como comentamos anteriormente, seleccionar un valor óptimo λ ayuda a acelerar el proceso.

- Submuestra: Exclusivo del Stochastic Gradient Boosting, este valor consiste en el porcentaje del conjunto de datos de entrenamiento que se utilizará en la construcción de cada weak learner.

2.9. XGboost

El algotimo XGBoost (Chen, 2016) acrónimo de "Extreme Gradient Boosting", es una implementación más rápida (10 veces más rápida que cualquiera de los otros métodos anteriormente comentados) y eficaz que las técnicas Gradient Boosting y Stochastic Gradient Boosting que acabamos de comentar. En la actualidad suele ser uno de los métodos más eficaces. XGBoost presenta un problema y es que sólo trabaja con variables numéricas, por tanto, se deben convertir las variables categóricas en numéricas, mediante la construcción de variables dummy. Debemos tener cuidado con esto, ya que en nuestro caso trabajamos con variables categóricas con muchos niveles como puede ser la población del establecimiento, el mes en el que se realiza el pedido, etc.

2.10. C5.0

El algoritmo C5.0, sucesor de C4.5 (ambos publicados por Quinlan), dependiendo de nuestro objetivo nos permite tanto calcular árboles de decisión como árboles mediante técnicas boosting (utilizando un procedimiento semejante a AdaBoost) (Kuhn, 2018).

Como hemos comentado, el algoritmo C5.0 utiliza la entropía como medida de división de los árboles. Una de las funcionalidades de esta técnica, es que además de combinar árboles de decisión y técnicas boosting, también nos permite asignar diferentes pesos a los errores de clasificación. Por ejemplo, en nuestro caso, no tiene la misma importancia equivocarse en la clasificación de un local que no abandonó que uno que finalmente abandono.

Además el algoritmo C5.0 incorpora una estrategia para la selección de predictores importantes para la creación del modelo denominada Winnowing. Este procedimiento actúa de la siguiente manera:

1. Antes de la creación del árbol se separa el conjunto de datos de entrenamiento en dos muestras proporcionales (50 % cada una).

2. En una de las dos muestras obtenidas se ajusta un árbol, el cual se donominará como árbol winnowing. La muestra restante se utilizará para medir el error del árbol winnoning.

3. Las variables explicativas que no han sido utilizadas para la creación del árbol winnoning se consideran como variables no útiles y se desechan.

4. A continuación de manera iterativa se crean diversos árboles winnoning eliminando las variables explicativas (una a una) que se han utilizado para la creación del árbol winnoning. Si debido a la eliminación de esa variable, el error del nuevo árbol winnoning disminuye, se clasifica como un predictor no útil.

5. Una vez clasificadas todas las variables explicativas, se reajusta el árbol winnoning sólo con los predictores útiles y observamos mediante el conjunto de validación si disminuye el error. Si disminuye, se aplica el algoritmo C5.0 unicamente considerando las variables útiles para la creación de los árboles, en caso contrario, se desecha y se crea un C5.0 convencional.

2.11. Otras técnicas empleadas

Además de realizar la predicción sobre las técnicas basadas en la construcción de árboles anteriormente comentadas, también se trabajó con otros métodos:

- **Regresión logística** (Hosmer, 1989). En este caso, en vez de crear el modelo de clasificación a partir de árboles de decisión, se creó mediante un análisis de regresión. En este caso, Se obtuvieron peores resultados.

- **Redes Neuronales** (Zurada, 1992) Esta técnica también fue empleado obteniendo resultados negativos.
- **Plataforma Machine Learning (Amazon Web Services)** (AWS, 2019). Amazon, a través de su plataforma de pago Amazon Web Service (AWS), nos permite obtener la tasa de abandono de un local mediante técnicas Machine Learning. Simplemente introducimos dos archivos .csv, el primero, el cual será el conjunto de datos de entrenamiento sobre el cual crearemos el modelo, y el segundo archivo el cual consistirá de los datos que queremos predecir. En la práctica se probó este método consiguiendo peores resultados a los obtenidos previamente aplicando Random Forest.

Capítulo 3

Resultados

A lo largo de este capítulo comentaremos los resultados obtenidos al realizar la predicción de la tasa de abandono y de competencia para los meses de julio, agosto y septiembre de 2018 tras aplicar el método Random Forest, el cual nos proporcionó un mejor resultado, sobre el conjunto de entrenamiento creado en la Sección 1. 5.

De este conjunto de entrenamiento sólo se ha trabajado con cierto número de variables, aquellas que han dado mejores resultados tras diferentes pruebas de depuración con el conjunto completo. Para ello realizaremos la predicción sobre los modelos que comentaremos a continuación y veremos en cuales de ellos obtenemos mejores soluciones.

3.1. Aplicación de los métodos a nuestro conjunto de datos

Una vez creado nuestro conjunto y comentado las técnicas a emplear, procederemos a estimar para los meses de julio, agosto y septiembre de 2018, la tasa de abandono y la tasa de abandono a la competencia.

Los datos que vamos a predecir se obtienen de la misma manera que el nuevo conjunto de datos que hemos explicado, con la única condición de que el local debe haber realizado al menos una compra de cerveza en el mes que queremos predecir. Es decir, consistiría en el historial de ventas de los locales desde que estos empezaron a comprar hasta el mes que queremos predecir. Por tanto nuestro objetivo será obtener qué locales en los dos próximos meses a la predicción no realizarán una compra de cerveza.

3.1.1. Conjunto de datos a predecir

Tras aplicar el proceso de depuración a los meses de julio, agosto y septiembre de 2018, obtenemos los siguientes resultados:

- **Datos de Julio 2018:** Consta de 15547 locales, de los cuales 350 sufrieron abandono. De los locales que sufrieron abandono, 88 se clasificaron como que fueron a la competencia, es decir, al mes siguiente realizaron al menos un pedido de agua o distribución.
- **Datos de Agosto 2018:** Consta de 15631 locales, de los cuales 772 sufrieron abandono. De los locales que sufrieron abandono 102 se clasificaron como locales que se han ido a la competencia. Como hemos comentado, debido a la estacionalidad en el número de clientes, abandonaron en agosto más del doble que en el mes de julio, sin embargo esto parece no afectar a la tasa de locales que abandonan para irse a la competencia. Nótese que sólo se van 14 más respecto al mes anterior.
- **Datos de Septiembre 2018:** Consta de 15301 locales, de los cuales 731 sufrieron abandono. De los locales que sufrieron abandono, 129 al mes siguiente compraron agua o distribución.

A la hora de aplicar cualquiera de los métodos comentados anteriormente, además de la depuración y tratamiento de los datos, también es clave seleccionar correctamente el conjunto de datos sobre el que vamos a realizar la predicción. Para este fin, aplicamos las técnicas anteriormente comentadas sobre 6 modelos diferentes:

- **Modelo 1**: Este modelo consiste en realizar cada uno de los métodos sobre la muestra original para intentar predecir el ABANDONO.
- **Modelo 2**: Análogo al Modelo 1, pero en este caso nuestro objetivo será predecir qué locales acabarán yéndose a la competencia mediante la variable SIGUIO SEGMENTADO.
- **Modelo 3**: Debido a que tenemos clases desbalanceadas, probamos a aplicar los métodos anteriores para predecir el ABANDONO sobre muestras proporcionales del conjunto de entrenamiento, de modo que un 50 % de los datos serán locales que abandonaron y el otro 50 % no.
- **Modelo 4**: En este caso, procedemos a realizar la predicción de la variable SIGUIO SEGMENTADO sobre los datos del conjunto de entrenamiento balanceados, es decir, un 33.33 % de los locales abandonaron, un 33.33 % de los locales se fueron a la competencia, es decir, compraron al mes siguiente agua, distribución o los dos y un 33.33 % de los locales que abandonoran no compraron ningún producto los dos meses posteriores.
- **Modelo 5**: Este modelo actuaría de la misma manera que el Modelo 4, pero como nuestro objetivo es realizar la predicción del mes de julio, agosto o septiembre y no "validar" el conjunto de entrenamiento, para que el algoritmo disponga de más datos introducimos todos los datos de la muestra sin necesidad de un conjunto de entrenamiento.
- **Modelo 6**: Este escenario surge debido a que se observó que tanto los Modelos 4 y 5 funcionan bien en la predicción de aquellos locales que se van a la competencia y que realizaron compra al mes siguiente tanto de agua como de distribución. Teniendo por tanto, problemas en clasificar los locales que al mes siguiente del abandono compraron sólo agua o distribución. Para que el modelo detecte mejor el abandono a la competencia de los locales, se construye el modelo prediciendo sobre la variable SIGUIO.

Además, como hemos comentado en la Sección 2.4.1, seleccionar las variables claves que entrarán en el modelo es uno de los aspectos más importantes. Por tanto se eliminan variables demasiado correladas, como puede ser el caso entre litros y euros de venta de cerveza en el último mes. Como ambas variables aportan la misma información se elimina la variable euros de cerveza. En total, para realizar las predicciones, disponemos de un total de 34 variables.

3.2. Resultados

A lo largo de este trabajo se han aplicado diversas técnicas: Regresión logística, C5.0, Random Forest, Adaboost, Gradient Boosting, Stochastic Gradient Boosting, XGBoost, Redes Neuronales y las herramientas de Machine Learning disponibles en Amazon Web Service sobre los dos conjuntos de datos que hemos creado y sobre los 6 modelos que acabamos comentar. Por motivos de espacio no es viable mostrar todos los resultados (en total serían más de 1000 tablas). Nos centraremos por tanto en comentar con detenimiento el método que mejores resultados nos ha proporcionado, en nuestro caso, el Random Forest.

La construcción del Random Forest se hizo en el entorno de programación R a través de dos librerías: RandomForest (Liaw and Wiener, 2018) y h2o (LeDell, 2018). Esto lo hacemos debido a que la librería RandomForest no permite introducir variables categóricas con más de 53 niveles, por tanto, no podemos introducir variables tan importantes como ocurre con la población del local. Sin embargo, la librería h2o sí que nos permite introducir esta variable pudiendo incluso paralelizar el proceso, disminuyendo con ello el tiempo de compilación. Como comentamos anteriormente, el método

Random Forest es fácilmente paralelizable debido a que se basa en la creación de B árboles de manera independiente.

Debido a que R es mononúcleo y a la gran cantidad de tiempo que lleva realizar la creación y optimización de los árboles, paralelizar el proceso aprovechando toda la capacidad posible del ordenador es un aspecto vital. Entendiendo como paralelizar la división del trabajo en los distintos núcleos que tiene el ordenador o máquina virtual.

De todos modos, los mejores resultados los obtenemos con la librería RandomForest tras la optimización de los hiperparámetros *nodesize*, *mtry* y número de árboles B (Sección 2.4.1). Esto puede ser debido al gran número de niveles que presenta la variable población, en total, 3241 niveles (poblaciones) diferentes.

3.3. Predicción Julio 2018

Una vez transformado y depurado el conjunto de datos que hemos creado para realizar la predicción vamos a ver el comportamiento de los diferentes modelos para predecir el mes de julio de 2018. Nuestro objetivo será identificar los locales que van a abandonar para enviar una alerta al promotor de ventas. En el caso de julio de 2018, abandonaron un total de 350 locales y 88 han sido clasificados como que fueron a la competencia, es decir, no realizaron ningún pedido de cerveza en agosto y septiembre y sin embargo, en agosto de 2018 han realizado al menos un pedido de agua o distribución.

3.3.1. Modelo 1

Como hemos comentado anteriormente, en este modelo empezamos con todo el conjunto de datos que hemos creado sobre el cual obtendremos la tasa de abandono para el mes de julio de 2018.

Primeramente, antes de crear el Random Forest a partir del conjunto de datos del Modelo 1, se ajustarán los hiperparámetros *mtry*, *nodesize* y número B de árboles. Para la optimización de los hiperparámetros, lo que haremos será observar el valor del hiperparámetro que minimice el out-of-bag error (Breiman 1996, b).

En la Figura 3.1 se muestra como varía el out-of-bag error al modificar el valor *mtry* (número de variables explicativas que se consideran para cada división). Observamos que el valor que minimiza el out-of-bag error es 4. De todos modos, el out-of-bag error del valor *mtry* considerado por defecto en la función randomForest (RandomForest), que es $mtry = 5$, es prácticamente idéntico.

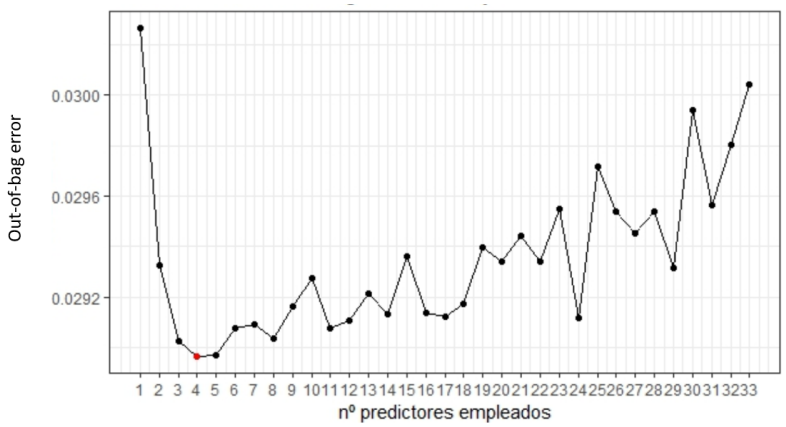


Figura 3.1: Out-of-bag error dependiendo del valor mtry.

A continuación vamos a obtener el valor óptimo de *nodesize*, es decir, el número mínimo de datos en cada nodo terminal. En la Figura 3.2 se muestran los resultados obtenidos, donde observamos que

el valor óptimo es 7. Como en el caso anterior, vemos que tampoco hay mucha diferencia con los parámetros seleccionados por defecto, $nodesize = 1$, por la función randomForest (Liaw and Wiener, 2018).

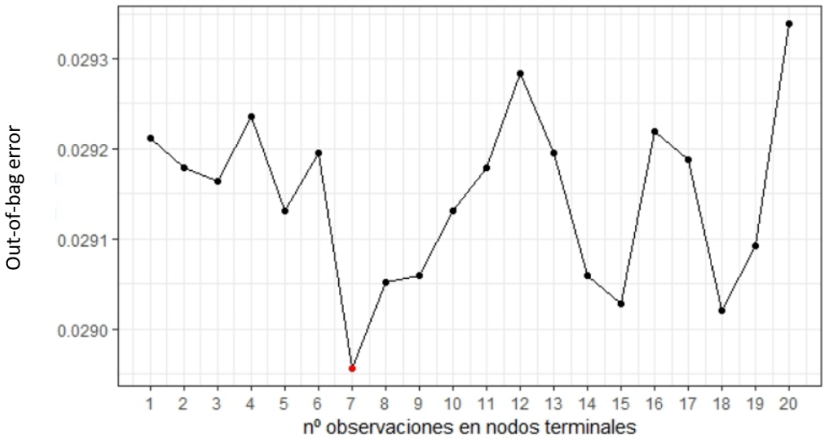


Figura 3.2: Out-of-bag error dependiendo del valor nodesize.

Tras el ajuste de los hiperparámetros procederemos a la construcción del Modelo 1 aplicando Random Forest. En el Cuadro 3.1 se muestra la clasificación del Modelo 1 sobre el conjunto de entrenamiento, en dónde las filas nos muestran lo que clasifica el modelo y en las columnas, lo que realmente sucedió. Además, el error nos indica la proporción de datos que ha clasificado incorrectamente. Podemos observar asimismo, como en este caso el modelo respecto a los datos de entrenamiento clasifica bastante bien los locales que no abandonaron. Sin embargo, aunque en el caso de clasificación de locales que han abandonado tiene bastante precisión, es decir, clasifica como que abandonaron 568, equivocándose sólo en 156, esto solo representa una pequeña proporción (11.75 %) de los locales que realmente han abandonado del conjunto de entrenamiento.

		Realidad		
	Modelo	NO	SI	Error
Clasificación	NO	121235	3423	0.12 %
	SI	156	412	89.25 %

Cuadro 3.1: Clasificación del Modelo 1 sobre el conjunto de entrenamiento.

En la Figura 3.3, se muestra la evolución del out-of-bag error al aumentar el número de árboles, dónde se puede observar en color verde el out-of-bag error de los locales que abandonan, en color rojo los que no y en negro el total. Podemos ver como a partir de un pequeño número de árboles (alrededor de 20) el out-of-bag error se estabiliza.

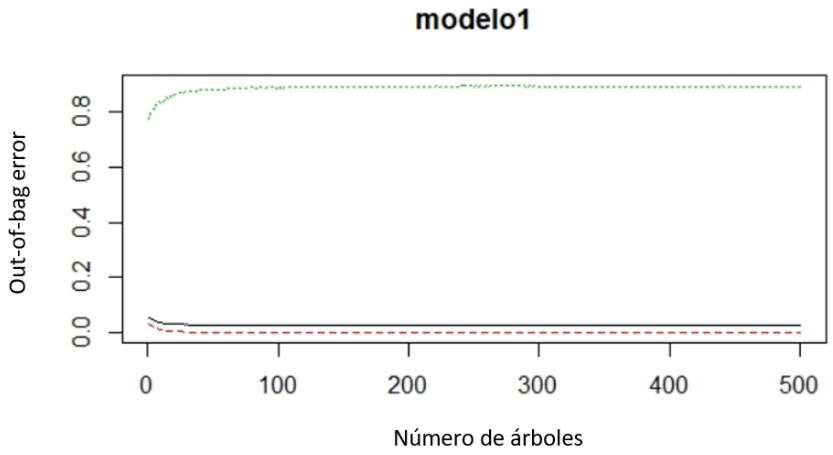


Figura 3.3: Representación del out-of-bag error con el número de árboles.

Además el RandomForest también nos permite ver las variables más importantes para la construcción del modelo, como se muestra en la Figura 3.4. En este caso se muestran las dos medidas comentadas en la Sección 2.3, el incremento del MDA y el incremento de la pureza de los nodos.

Observamos como los litros que vendió el local de cerveza el primer, penúltimo y último mes tienen una gran importancia en la construcción del modelo. Esto es debido a que, como hemos observado, se aprecia un descenso de ventas en litros de cerveza el último mes en los locales que abandonan. Además el mes en el que nos encontramos (variable MES ACABO) también tiene bastante importancia.

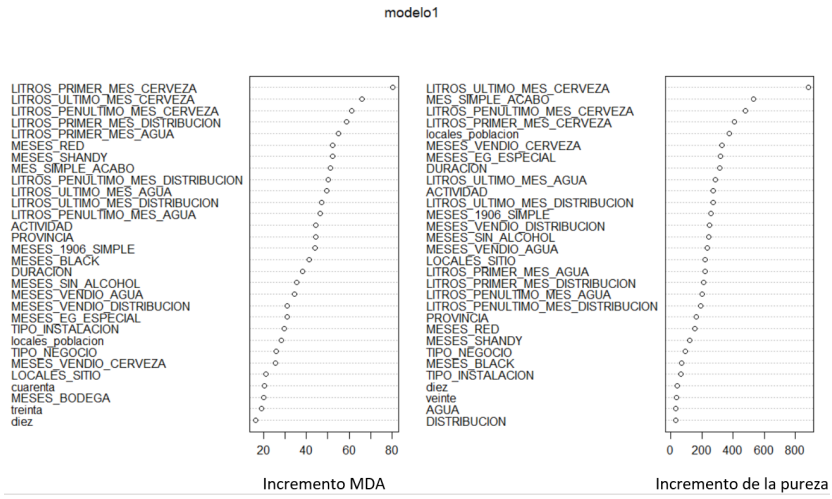


Figura 3.4: Importancia de las variables para la construcción del Modelo 1.

Una vez construido el modelo procedemos a realizar la predicción para el mes de julio 2018. En el Cuadro 3.1 se observa la clasificación de la predicción con una probabilidad de abandono del 50 %. Podemos observar como la predicción se comporta de manera similar a los resultados obtenidos para el conjunto de entrenamiento, es decir, clasifica pocos locales como que abandonaron, clasificando 37 correctamente de los 350 que realmente han abandonado, pero con bastante precisión (acierta 37 equivocándose solamente en 12).

Predicción	NO	SI
NO	15185	313
SI	12	37

Cuadro 3.1: Predicción para Julio de 2018 del Modelo 1.

A continuación observamos en el Cuadro 3.2 cómo clasifica el modelo cuando aumentamos o disminuimos la probabilidad de abandono. Se aprecia como a medida que aumentamos la probabilidad de que el local abandone (tasa de abandono) el modelo es más preciso. Por ejemplo, si consideramos los locales que el modelo ha clasificado con un 70 % o más de probabilidad de que abandonen, obtendríamos 11 locales, los cuales finalmente abandonan.

Modelo 1	25 %		50 %		70 %		90 %	
Abandono	No	Si	No	Si	No	Si	No	Si
Clasificación	179	75	12	37	0	11	0	3

Cuadro 3.2: Probabilidades de la predicción de abandono del Modelo 1.

3.3.2. Modelo 2

Para la creación del Modelo 2 partimos del mismo conjunto de datos de entrenamiento del Modelo 1, pero en este caso nuestro objetivo será identificar los locales que se van a la competencia.

Como en el Modelo 1, lo primero que haremos será optimizar los hiperparámetros *mtry*, *nodesize* y número B de árboles.

Como podemos observar en la Figura 3.5 el valor *mtry* que minimiza el out-of-bag error es 8, pero como en el caso anterior, no hay demasiada diferencia con los otros valores, excepto en el caso que se seleccionase $mtry = 1$.

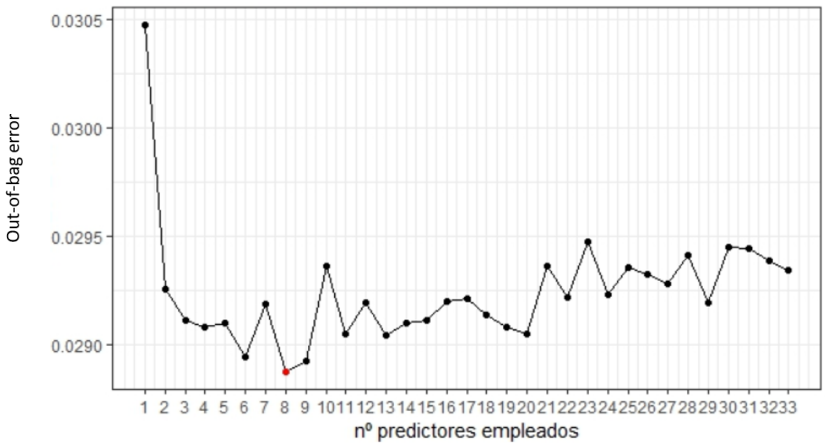


Figura 3.5: Out-of-bag error dependiendo del valor mtry.

En la Figura 3.6 se muestra que el mínimo de datos que debe haber en cada nodo terminal que

minimiza el out-of-bag error es 9, sin embargo, tampoco habría demasiada diferencia con el valor seleccionado por defecto en la construcción del modelo, que es 5.

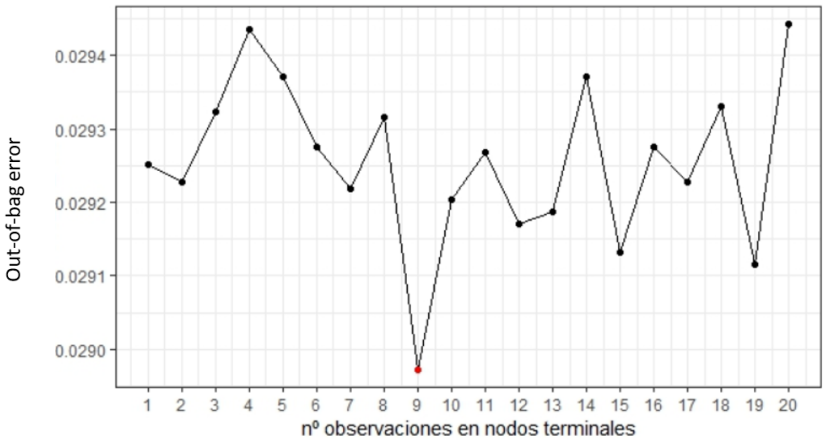


Figura 3.6: Out-of-bag error dependiendo del valor nodesize.

Tras realizar la optimización de los hiperparámetros procederemos a la construcción del Modelo 2. En el Cuadro 3.3 se muestra la clasificación del Modelo 2 sobre el conjunto de entrenamiento para 3 variables. "No abandono" indica que el cliente no abandonó, "Cerró" que el local sufrió abandono pero al mes siguiente no realizó un pedido de agua o distribución y "Competencia" que al mes siguiente del abandono, realizó al menos un pedido de agua o distribución. Observamos como el Modelo 2 no clasifica bien los locales que van a la competencia, lo que puede deberse a que disponemos de pocos datos en el conjunto de entrenamiento (exclusivamente 630 de más de 120000 datos). En el caso de "Cerró" y "no abandonó", observamos un comportamiento similar al Modelo 1.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	121278	2846	614	0.00 %
CERRO	113	359	16	88.79 %
COMPETENCIA	0	0	0	100 %

Cuadro 3.3: Clasificación del Modelo 2 sobre el conjunto de entrenamiento.

En la Figura 3.7 se muestra como varía el out-of-bag error al aumentar el número de árboles. Observamos en color rojo la representación del out-of-bag error de los locales que no abandonan, en color verde de los que abandonan pero no van a la competencia y en color azul de los locales que van a la competencia. En color negro observamos la evolución del out-of-bag error total. Como en el modelo anterior, con pocos árboles B el out-of-bag error se estabiliza.

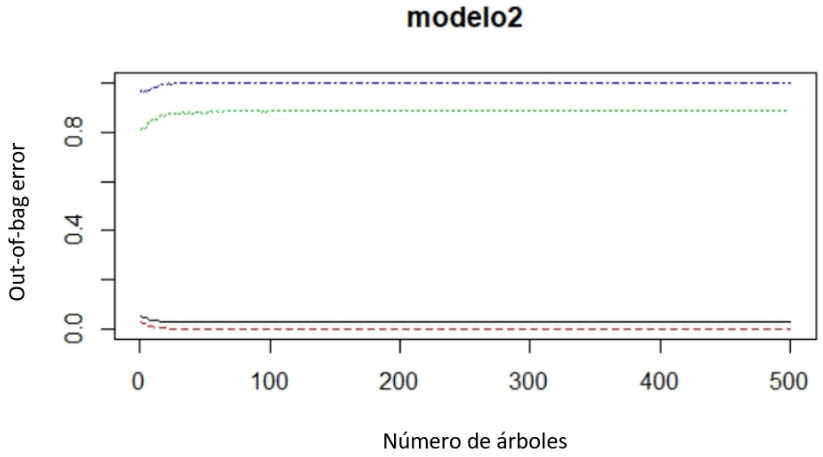


Figura 3.7: Representación del out-of-bag error con el número de árboles.

También podemos observar en la Figura 3.8 la importancia de las variables en la construcción del modelo y cómo es prácticamente idéntico a lo obtenido en el Modelo 1.

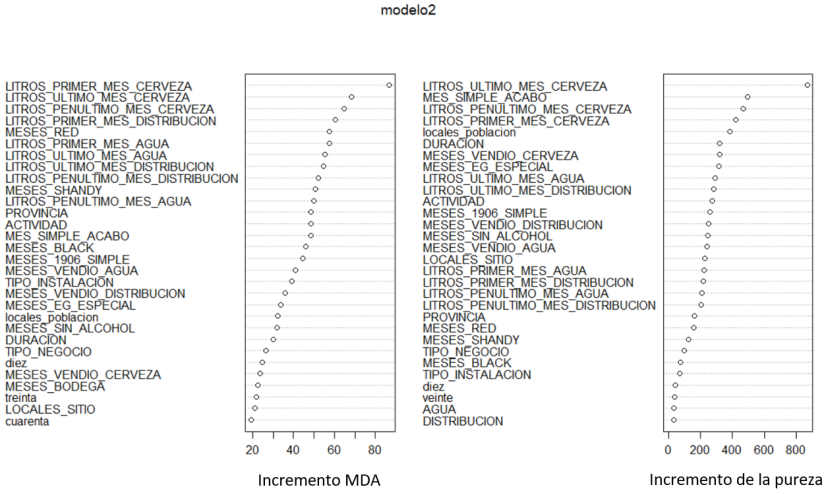


Figura 3.8: Importancia de las variables para la construcción del modelo.

Por último en el Cuadro 3.4 se muestra la predicción que realiza el Modelo 2 sobre el mes de julio de 2018. Podemos observar como los resultados son bastante parecidos al Modelo 1.

Predicción	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	15189	232	83
CERRO	8	31	4
COMPETENCIA	0	0	0

Cuadro 3.4: Predicción para Julio de 2018 del Modelo 2.

3.3.3. Modelo 3

Como hemos observado tanto en el Modelo 1 como en el Modelo 2, al realizar las predicciones, los modelos son precisos pero no clasifican muy bien, es decir, identifica los locales que abandonan y que van a la competencia con precisión, pero clasifica muy pocos. Esto es debido a que los datos están demasiado desproporcionados: un 97 % de los locales no abandonan, mientras que solamente un 3 % sí. Por tanto vamos a considerar una muestra balanceada del conjunto de entrenamiento, es decir, una muestra en la cual el 50 % de los locales no han sufrido abandono, mientras que el 50 % restante sí.

Como en los casos anteriores, lo primero que haremos será realizar el ajuste de los hiperparámetros *mtry*, *nodesize* y número de árboles B.

En la Figura 3.9 se muestra la evolución del out-of-bag error al variar el número de predictores evaluados para cada división. Observamos que el valor *mtry* que minimiza el out-of-bag error es 14.

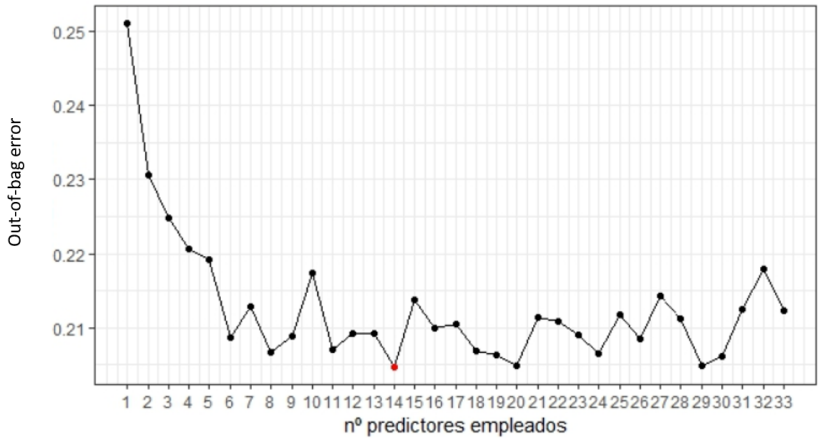


Figura 3.9: Out-of-bag error dependiendo del valor mtry.

En el caso de *nodesize*, el valor que nos proporciona un menor out-of-bag error es también 14, como se puede observar en la Figura 3.10.

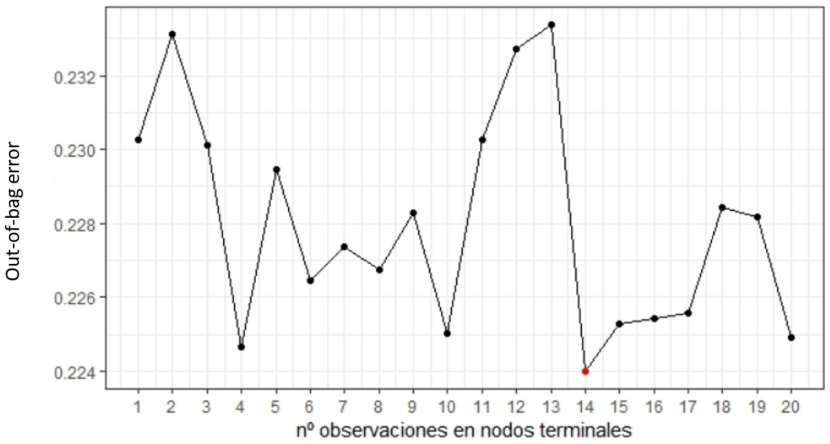


Figura 3.10: Out-of-bag error dependiendo del valor nodesize.

Una vez establecidos los hiperparámetros óptimos, creamos el modelo. El Cuadro 3.5 nos muestra la clasificación del conjunto de entrenamiento con una probabilidad del 50 %. En este caso, el Modelo

3 reduce considerablemente el error de clasificación de abandono, pasando de un 89.25 % en el Modelo 1 a un 19.37 % en el Modelo 3.

Modelo	NO	SI	Error
NO	3061	743	20.18 %
SI	774	3092	19.37 %

Cuadro 3.5: Clasificación del Modelo 3 sobre el conjunto de entrenamiento.

También vemos en la Figura 3.11 como el out-of-baf error necesita un mayor número de árboles para estabilizarse (próximo a 200). En dicha gráfica aparece representado en color verde el out-of-bag error de los locales que abandonaron, en color rojo de los locales que no abandonaron y en color negro el total.

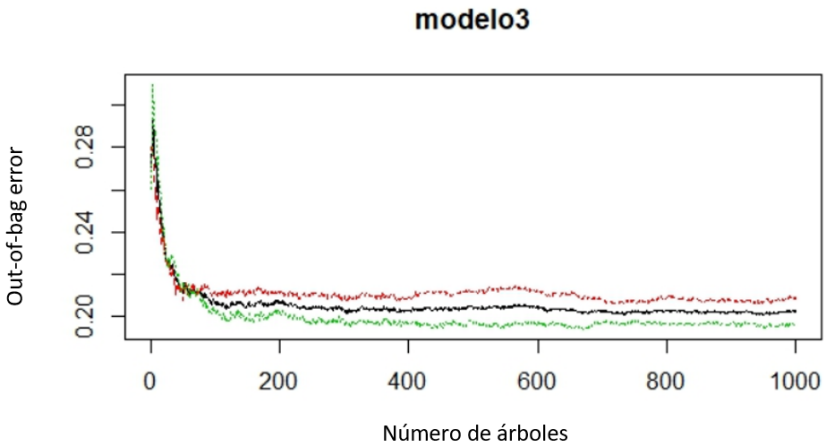


Figura 3.11: Representación del out-of-bag error con el número de árboles.

Respecto a la importancia de las variables, vemos como en la Figura 3.12 la venta de litros del penúltimo y último mes de cerveza, el mes en el que nos encontramos y el tiempo que el local lleva siendo cliente adquieren gran importancia en la creación del Modelo 3.

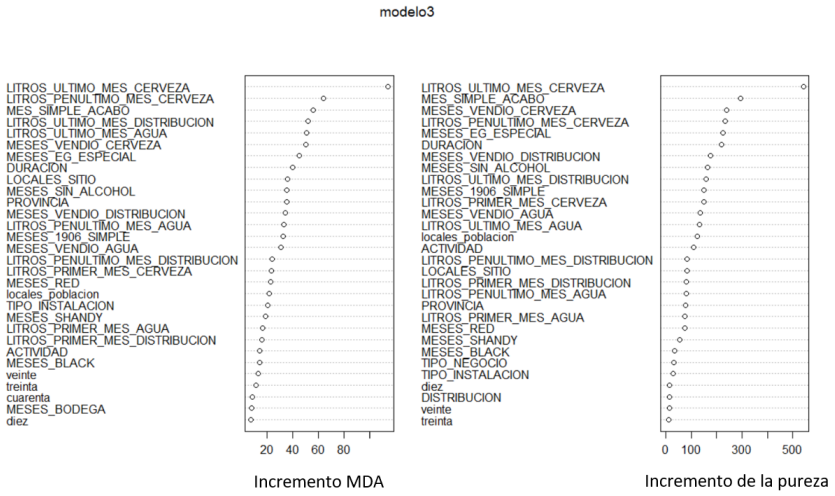


Figura 3.12: Importancia de las variables para la construcción del modelo.

A continuación analizaremos el comportamiento del Modelo 3 para realizar la predicción de julio de 2018. En el Cuadro 3.6 se muestra la predicción del Modelo 3 sobre julio de 2018 para una probabilidad del 50 %. Observamos como clasifica correctamente un mayor número de locales que abandonan, conllevando consigo también más fallo. Es decir, el modelo clasifica como que abandonan con una probabilidad mayor o igual al 50 % un total de 2377 locales, de los cuales 242 efectivamente abandonaron.

Predicción	NO	SI
NO	13062	108
SI	2135	242

Cuadro 3.6: Predicción para Julio de 2018 del Modelo 3.

En el Cuadro 3.7 vemos como funciona la predicción del modelo a medida que aumentamos la probabilidad de abandono. Podemos ver como en el caso de una probabilidad de abandono superior o igual al 90 %, se encuentran 201 de los que 57 locales que realmente abandonaron.

Modelo 3	25 %		50 %		70 %		90 %	
Abandono	No	Si	No	Si	No	Si	No	Si
Clasificación	5966	324	2134	242	608	150	144	57

Cuadro 3.7: Probabilidades de la predicción de abandono del Modelo 3.

3.3.4. Modelo 4

En este caso, partiendo como en el Modelo 3 de una muestra balanceada sobre el conjunto de entrenamiento (33.33 % de los locales abandonaron, 33.33 % de los locales se fueron a la competencia y un 33.33 % de los locales que abandonoran no compraron ningún producto los dos meses posteriores),

nuestro objetivo será poder predecir los locales que se van a la competencia de una manera más efectiva que en el Modelo 2. Para ello, empezaremos por ajustar los hiperparámetros.

En la Figura 3.13 se muestra el out-of-bag error al variar el parámetro *mtry*. Observamos que cuando $mtry = 13$ obtenemos el menor out-of-bag error.

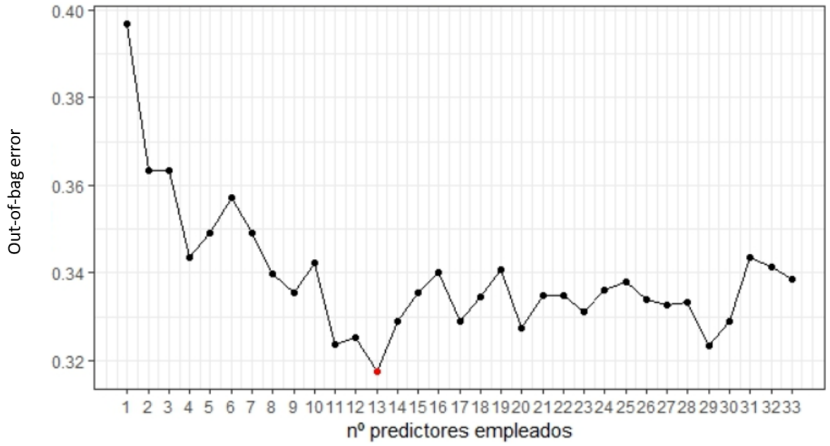


Figura 3.13: Out-of-bag error dependiendo del valor mtry.

En el caso del hiperparámetro *nodesize*, obtenemos que el mínimo out-of-bag error se alcanza cuando $nodesize = 2$, Figura 3.14.

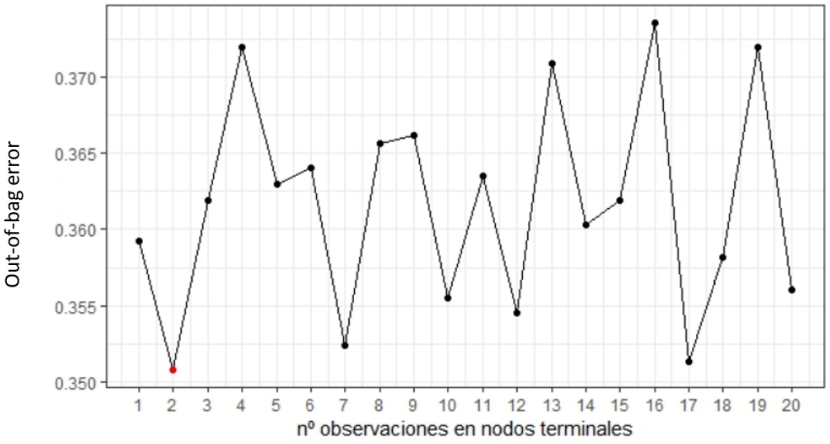


Figura 3.14: Out-of-bag error dependiendo del valor nodesize.

Tras ajustar los hiperparámetros críticos, construimos el Modelo 4. Se obtiene, a diferencia del Modelo 2, una clasificación más o menos precisa de los locales que van a la competencia comprendidos en el conjunto de datos de entrenamiento, Cuadro 3.8.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	448	106	86	28.88 %
CERRO	112	436	121	30.79 %
COMPETENCIA	70	88	423	32.85 %

Cuadro 3.8: Clasificación del Modelo 4 sobre el conjunto de entrenamiento.

En la Figura 3.15 se muestra la evolución del out-of-bag error al aumentar el número B de árboles, observamos como a partir de 400 árboles, el out-of-bag error tiende a estabilizarse. Es normal que el out-of-bag error necesite de un número de árboles mayor para estabilizarse debido a que estamos considerando una variable respuesta con un mayor número de niveles.

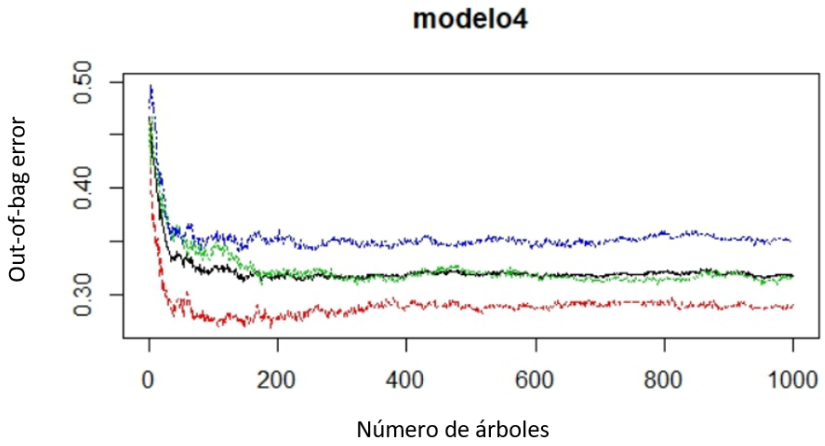


Figura 3.15: Representación del out-of-bag error con el número de árboles.

También observamos en la Figura 3.16 las variables más importante para la creación del modelo. En este caso notamos como además de las variables importantes en el Modelo 3, los litros de agua o distribución cobran una mayor importancia.

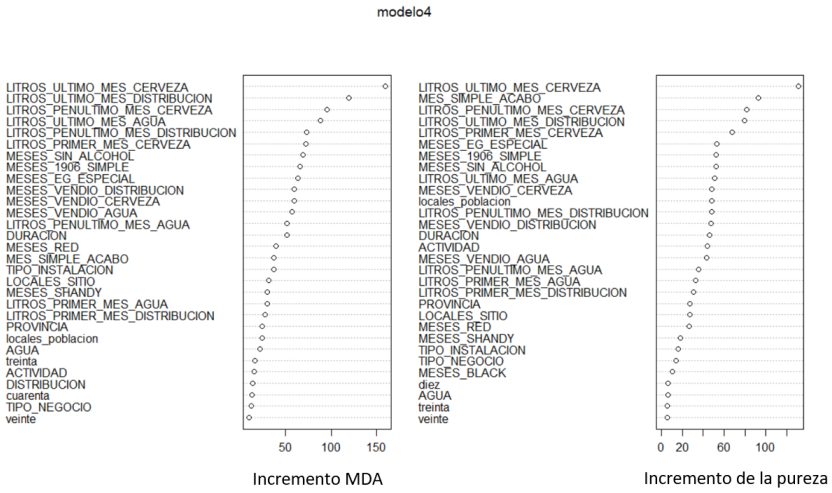


Figura 3.16: Importancia de las variables para la construcción del modelo.

Una vez creado el Modelo 4 procedemos a realizar la predicción sobre el mes de julio de 2018. En el Cuadro 3.9 podemos ver la predicción que realiza el Modelo 4 sobre el mes de julio 2018. Como ocurrió para el conjunto de entrenamiento, el Modelo 4 clasifica de manera aceptable los locales que han ido a la competencia.

Predicción	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	11664	59	19
CERRO	1374	144	14
COMPETENCIA	2159	60	54

Cuadro 3.9: Predicción para Julio de 2018 del Modelo 4.

A continuación en el Cuadro 3.10 observamos qué ocurre cuando aumentamos la probabilidad, viendo por ejemplo, que con una probabilidad igual o superior al 90 % acierta 4 y falla solamente 11.

Modelo 4	25 %			50 %			75%			90 %		
Siguio	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA
CERRO	4002	207	34	629	97	11	9	9	1	0	0	0
COMPETENCIA	4499	136	72	1291	35	41	249	5	19	11	0	4

Cuadro 3.10: Probabilidades de la predicción de abandono a la competencia del Modelo 4.

3.3.5. Modelo 5

Debido a que nuestro objetivo no es comprobar cómo de bien se ajusta el modelo a los datos pasados sino a la realización de predicciones futuras las cuales están en continuo cambio y además disponemos de pocos datos de abandono. Crearemos el Modelo 5 con una muestra balanceada, pero esta vez sobre todo el conjunto, es decir, mientras que en el Modelo 4 se realizó una muestra balanceada del 70 % de los datos, en este caso consideraremos su totalidad.

Centrándose en el ajuste de hiperparámetros, en la Figura 3.17 observamos que en este caso el valor *mtry* que minimiza el out-of-bag error es 9.

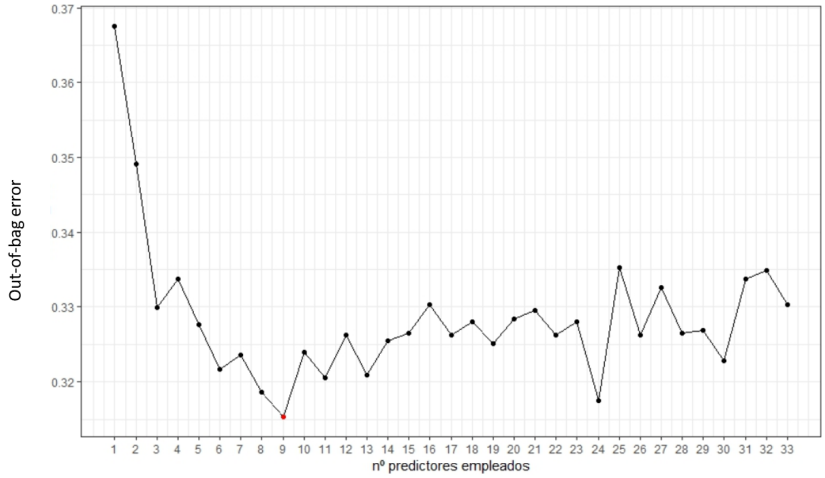


Figura 3.17: Out-of-bag error dependiendo del valor mtry.

En el caso de *nodesize* obtenemos el out-of-bag error más bajo cuando $nodesize = 3$, como podemos apreciar en a Figura 3.10.

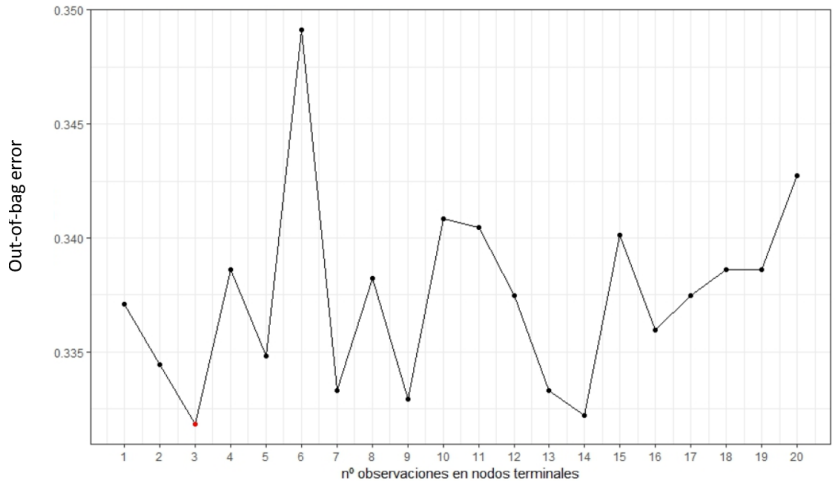


Figura 3.18: Out-of-bag error dependiendo del valor nodesize.

Una vez establecidos los dos hiperparámetros pasamos a construir el Modelo 5, el cual nos permite identificar los locales que se van a ir a la competencia. En el Cuadro 3.11 se puede observar la clasificación que realiza el Modelo 5 sobre los datos sobre los que se ha construido el modelo. Observamos un error similar al error obtenido en el Modelo 4.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	651	170	125	26.52 %
CERRO	148	581	172	34.42 %
COMPETENCIA	87	135	589	33.52 %

Cuadro 3.11: Clasificación del Modelo 5 sobre el conjunto de entrenamiento.

En la Figura 3.19 se aprecia la evolución del out-of-bag error según vamos aumentando el número de árboles B. Observamos como a partir de más o menos 400 árboles el out-of-bag error se estabiliza, necesitando de un mayor número de árboles para estabilizarse que el Modelo 4 debido a que consta de un mayor número de datos.

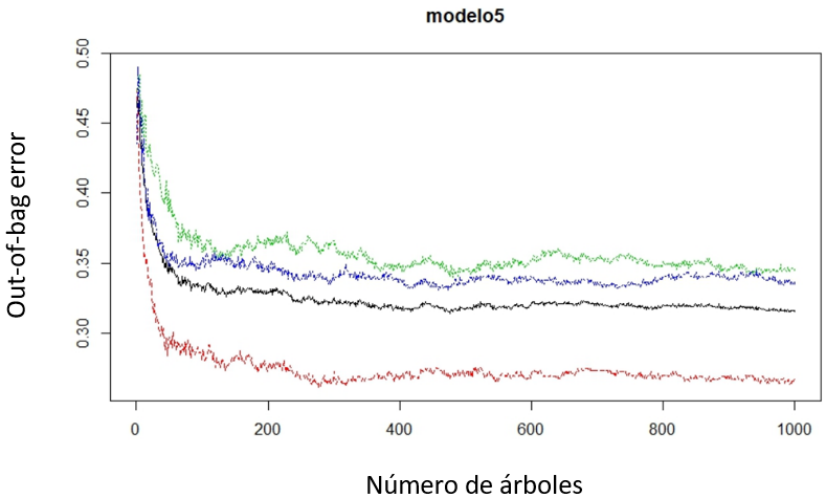


Figura 3.19: Representación del out-of-bag error con el número de árboles.

En la Figura 3.20 observamos que las variables más importantes para la construcción del modelo son prácticamente idénticas a las obtenidas por el Modelo 4.

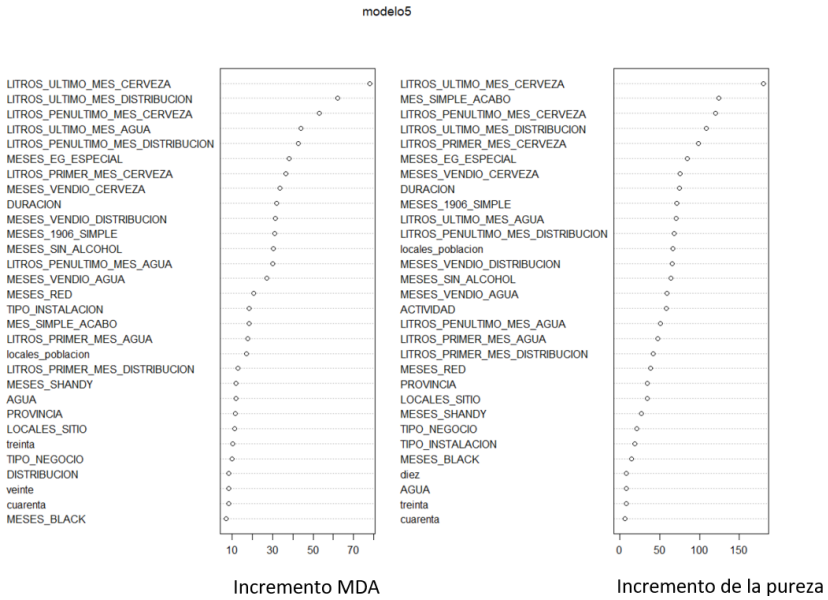


Figura 3.20: Importancia de las variables para la construcción del modelo.

Una vez construido el Modelo 5 vamos a ver su comportamiento respecto a la predicción del mes de julio de 2018. Observamos como en el caso de locales que se fueron a la competencia obtenemos una mejor predicción que la proporcionada en el Modelo 4, debido a que tenemos los mismos aciertos pero menos fallos.

Predicción	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	11451	57	18
CERRO	1726	152	15
COMPETENCIA	2020	54	54

Cuadro 3.12: Predicción para Julio de 2018 del Modelo 5.

A continuación en el Cuadro 3.13 se muestra qué ocurre cuando aumentamos la probabilidad. En este caso vemos como presenta un comportamiento similar al Modelo 4 pero a medida que aumentamos la probabilidad parece que el Modelo 4 afina más. De todos modos antes de decantarnos por un modelo en concreto, veremos su comportamiento ante la predicción de los meses de agosto y septiembre de 2018.

Modelo 5	25 %			50 %			75 %			90 %		
Siguio	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA
CERRO	4361	219	38	957	111	10	81	12	0	0	0	0
COMPETENCIA	4417	130	69	1206	34	41	288	7	17	56	1	6

Cuadro 3.13: Probabilidades de la predicción de abandono a la competencia del Modelo 5.

3.3.6. Modelo 6

Debido a que tanto el Modelo 4 como el Modelo 5 detectan bien los locales que se han ido a la competencia pero que al mes siguiente compran tanto agua como distribución, dividimos esta variable de modo que nos permita identificar qué locales de los que se han ido a la competencia al mes siguiente han comprado sólo agua, sólo distribución o los dos.

Lo primero que haremos será realizar el ajuste de hiperparámetros. En las Figuras 3.21 y 3.22 se observan como los valores *mtry* y *nodesize* que minimiza el out-of-bag error es en ambos casos 8.

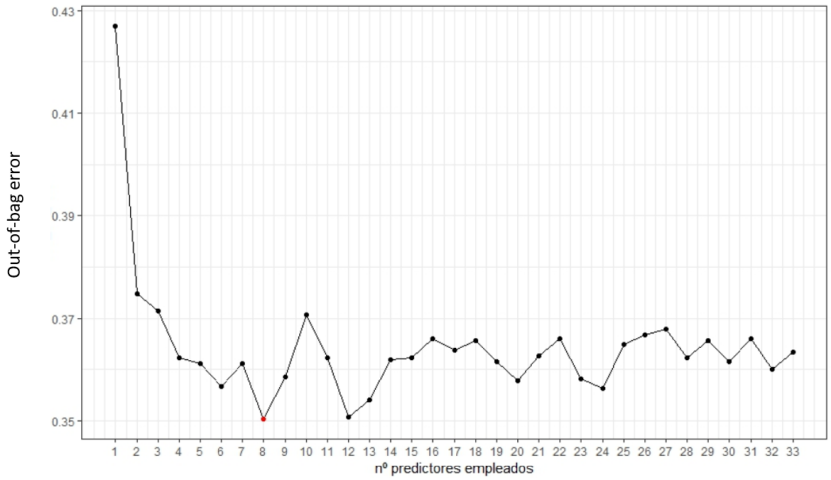


Figura 3.21: Out-of-bag error dependiendo del valor mtry.

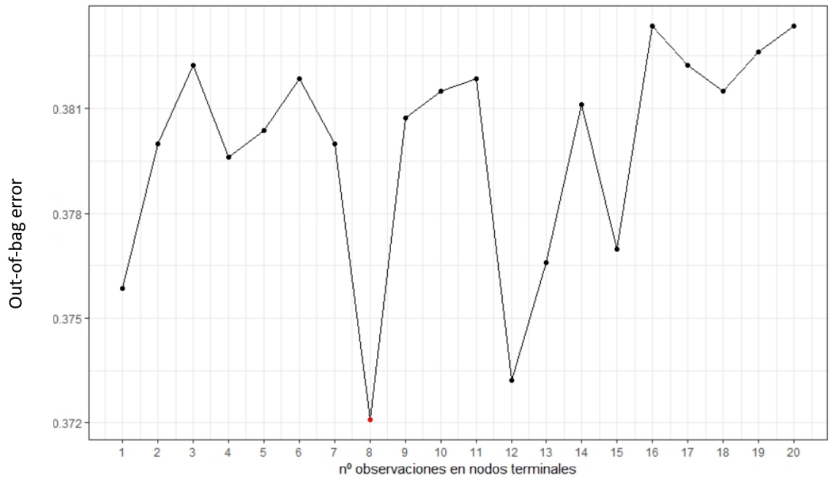


Figura 3.22: Out-of-bag error dependiendo del valor nodesize.

Una vez construido el Modelo 6 observamos en el Cuadro 3.14 como clasifica el modelo sobre el conjunto de entrenamiento. Apreciamos un mayor error, pero vemos cómo clasifica de manera aceptable.

Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO	ERROR
AGUA	131	2	13	27	42	53.21 %
DISTRIBUCIÓN	4	235	24	34	46	47.30 %
DOS	13	11	69	2	6	56.87 %
NO ABANDONO	39	79	33	648	183	26.86 %
CERRO	93	119	21	175	609	31.26 %

Cuadro 3.14: Clasificación del Modelo 6 sobre el conjunto de entrenamiento.

Para el número B de árboles percibimos en la Figura 3.23 cómo a partir de 200 árboles el out-of-bag error tiende a estabilizarse.

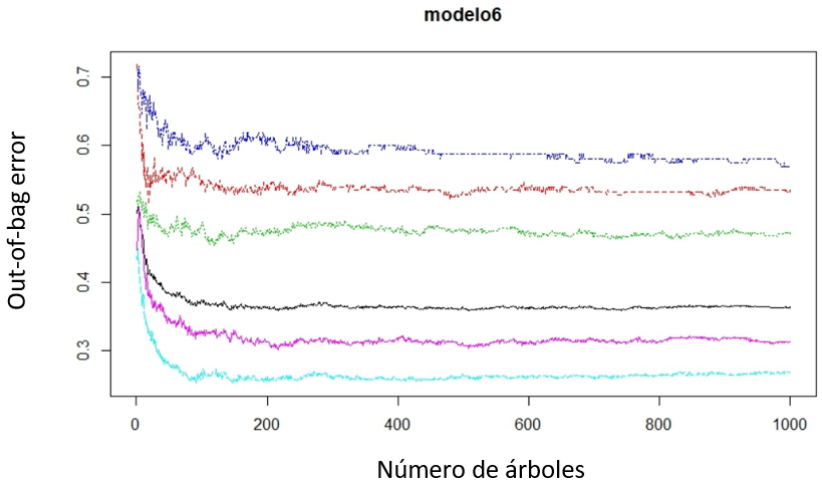


Figura 3.23: Representación del out-of-bag error con el número de árboles.

En el caso de la importancia de las variables, observamos en la gráfica 3.24 como cobran bastante fuerza los litros vendidos tanto de cerveza como de agua y distribución.

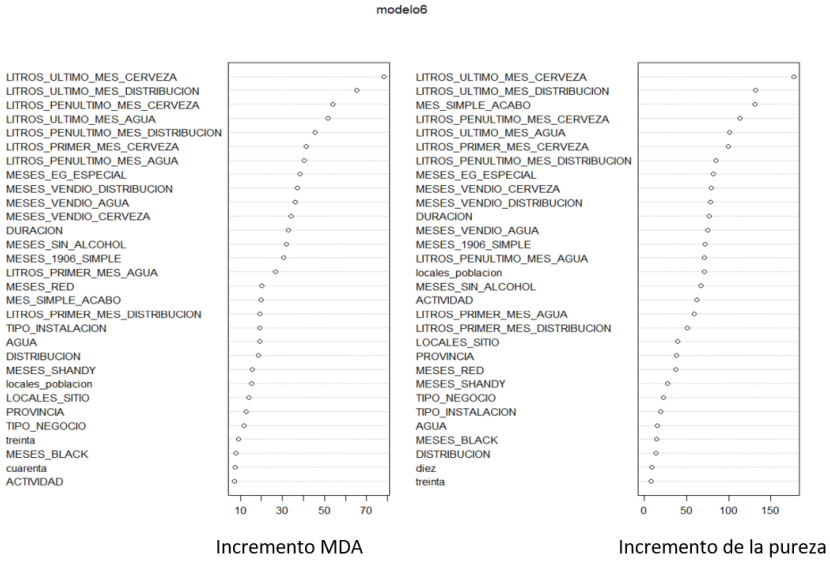


Figura 3.24: Importancia de las variables para la construcción del modelo.

Ahora realizaremos la predicción del mes de julio de 2018. Observamos en el Cuadro 3.15 los resultados, viendo como el Modelo 6 distingue bastante bien los casos en los que abandona a la competencia comprando únicamente agua, distribución o los dos. Además se observa bastante precisión.

Predicción	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	23	0	3	201	13
DISTRIBUCIÓN	1	26	1	890	32
DOS	3	1	13	185	3
NO ABANDONO	5	2	7	10693	70
CERRO	21	15	8	2601	484

Cuadro 3.15: Predicción de julio de 2018 del Modelo 6.

A continuación en el Cuadro 3.16 veremos su comportamiento cuando vamos aumentando la probabilidad. Observamos como en el caso que tenemos una probabilidad igual o mayor al 75 %, clasifica como que van a ir a la competencia 142 locales de los cuales 13 finalmente abandonan.

Probabilidad	25 %				
Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	32	0	7	368	43
DISTRIBUCIÓN	5	33	4	1705	81
DOS	3	6	18	235	3
CERRO	34	19	9	5349	549

Probabilidad	50 %				
Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	16	0	0	70	8
DISTRIBUCIÓN	0	13	0	384	15
DOS	0	1	6	45	0
CERRO	15	8	1	1174	377

Probabilidad	75 %				
Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	2	0	0	2	0
DISTRIBUCIÓN	0	10	0	118	4
DOS	0	0	1	4	0
CERRO	1	3	0	176	124

Cuadro 3.16: Probabilidades de la tasa de competencia para julio de 2018 del Modelo 6.

3.4. Predicción Agosto 2018

Para ver si el comportamiento de los diferentes modelos se mantiene estable para todos los meses, además de realizar la predicción para el mes de julio, también haremos la predicción para los meses de agosto y de septiembre de 2018. Se seleccionaron estos tres meses debido a que es cuando más variaciones sufren los locales y es cuando mejor podemos comprobar la eficacia de los modelos.

Para llevar a cabo la predicción actualizamos el conjunto de datos de entrenamiento, es decir, mientras que para la predicción de julio 2018 el conjunto de datos de entrenamiento estaba formado por los locales que abandonaron entre junio de 2017 a mayo 2018, este nuevo conjunto irá de julio de 2017 a junio 2018.

3.4.1. Modelo 1

Siguiendo el procedimiento llevado a cabo para realizar la predicción del mes de julio de 2018, obtenemos que los hiperparámetros que minimizan el out-of-bag error son $mtry = 2$ y $nodesize = 11$. Una vez establecidos los dos hiperparámetros, creamos el Modelo 1. Como podemos observar en el Cuadro 3.17 el Modelo 1 tiene un comportamiento bastante similar respecto a los datos de entrenamiento obtenidos para el mes anterior.

Modelo	NO	SI	Error
NO	121132	3330	0.11 %
SI	145	390	89.51 %

Cuadro 3.17: Clasificación del Modelo 1 sobre el conjunto de entrenamiento.

También observamos como a partir de unos 20 árboles el out-of-bag error tiende a estabilizarse.

En el caso de la predicción de agosto de 2018 se aprecia un comportamiento practicamente idéntico al obtenido al realizar la predicción de julio 2018: el modelo clasifica pocos datos como que abandonan pero los pocos que clasifica los tiende a clasificar bien, como se puede observar en los Cuadros 3.18 y 3.19.

Predicción	NO	SI
NO	14831	731
SI	28	41

Cuadro 3.18: Predicción para Agosto de 2018 del Modelo 1.

Modelo 1	25 %		50 %		70 %		90 %	
Abandono	No	Si	No	Si	No	Si	No	Si
Clasificación	491	233	28	41	1	5	0	0

Cuadro 3.19: Probabilidades de la predicción de abandono del Modelo 1.

3.4.2. Modelo 2

Para la creación del Modelo 2, los hiperparámetros que minimizan el out-of-bag error son $mtry = 2$ y $nodesize = 10$. Como en el caso de julio de 2018, podemos observar en el Cuadro 3.20 como no clasifica ningún local en la competencia en el conjunto de entrenamiento.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	121173	2738	621	0.08 %
CERRO	104	344	17	88.83 %
COMPETENCIA	0	0	0	100 %

Cuadro 3.20: Clasificación del Modelo 2 sobre el conjunto de entrenamiento.

En el caso de la predicción de agosto 2018 se puede ver en el Cuadro 3.21 como el resultado es prácticamente idéntico aunque quizás un poco menos preciso al obtenido para julio de 2018.

Predicción	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	14842	638	100
CERRO	17	32	2
COMPETENCIA	0	0	0

Cuadro 3.21: Predicción para Agosto de 2018 del Modelo 2.

3.4.3. Modelo 3

En la predicción de julio de 2018, observamos como este modelo clasifica más cantidad de abandonos que en el Modelo 1, veremos si esto ocurre de manera similar para la predicción de agosto de 2018.

En el ajuste de hiperparámetros, obtenemos que los valores *mtry* y *nodesize* que minimizan el out-of-bag error son: $mtry = 16$ y $nodesize = 9$. Una vez optimizado los hiperparámetros observamos en el Cuadro 3.22 como la clasificación del Modelo 3 sobre el conjunto de entrenamiento se comporta de manera bastante similar a lo sucedido para el mes de julio.

Modelo	NO	SI	Error
NO	2971	722	20.13 %
SI	749	2998	19.40 %

Cuadro 3.22: Clasificación del Modelo 3 sobre el conjunto de entrenamiento.

Una vez creado el modelo y viendo que se comporta de manera similar al modelo creado en julio de 2018, vamos a realizar la predicción del mes de agosto de 2018. En los Cuadros 3.23 y 3.24 podemos ver un comportamiento parecido a la predicción de julio de 2018 en la clasificación y al ir aumentando la probabilidad de abandono, observamos sobre todo, como el modelo detecta que abandonan un mayor número de locales que en el mes de julio.

Predicción	NO	SI
NO	11580	249
SI	3279	523

Cuadro 3.23: Predicción para Agosto de 2018 del Modelo 3.

Modelo 3	25 %		50 %		70 %		90 %	
Abandono	No	Si	No	Si	No	Si	No	Si
Clasificación	7235	678	3279	523	1395	379	414	160

Cuadro 3.24: Probabilidades de abandono a la competencia de Agosto de 2018 del Modelo 3.

3.4.4. Modelo 4

De manera similar al Modelo 3, se verá cómo se comporta el Modelo 4 para clasificar de manera adecuada los locales que se van a la competencia. Al tratar el ajuste de hiperparámetros, obtenemos que los valores $mtry = 9$ y $nodesize = 15$ son los que minimizan el out-of-bag error.

En el Cuadro 3.25 observamos como igual que en los casos anteriores, la clasificación del modelo respecto a los datos de entrenamiento se comporta de manera prácticamente idéntica a lo ocurrido en julio.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	455	116	103	28.68 %
CERRO	108	412	126	35.42 %
COMPETENCIA	75	110	409	35.89 %

Cuadro 3.25: Clasificación del Modelo 4 sobre el conjunto de entrenamiento.

A continuación en el Cuadro 3.26 se muestra la clasificación del Modelo 4 para la predicción del mes de agosto de 2018. Como para el mes de julio, agrupa la mayoría de los locales que se van a la competencia, en nuestro caso, 70 de 102 en simplemente 2000 locales de los 15000 sobre los que realizamos la predicción. Cabe destacar la dificultad de predecir qué locales van a acabar yendo a la competencia sin tan siquiera disponer de qué locales están abiertos o cerrados, o qué locales de los cuales han dejado de ser clientes finalmente han ido a la competencia.

Predicción	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	10823	211	11
CERRO	2036	354	21
COMPETENCIA	2000	105	70

Cuadro 3.26: Predicción para Agosto de 2018 del modelo 4.

En el Cuadro 3.27 observamos como a medida que aumentamos la probabilidad de que un local cierre o vaya a la competencia, el modelo es cada vez más preciso.

Modelo 4	25 %			50 %			75%			90 %		
Siguio	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA
CERRO	5048	542	55	1052	261	14	168	76	1	0	0	0
COMPETENCIA	4736	270	87	944	60	58	162	9	16	21	0	3

Cuadro 3.27: Probabilidades de la predicción de abandono a la competencia del Modelo 4.

3.4.5. Modelo 5

Para la predicción de julio de 2018 observamos un comportamiento bastante similar entre el Modelo 4 y el Modelo 5. Veremos que ocurre para la predicción del mes de agosto de 2018. En el ajuste de hiperparámetros, obtenemos $mtry = 9$ y $nodesize = 2$ como el conjunto de hiperparámetros que minimizan el out-of-bag error.

En el Cuadro 3.28 observamos la clasificación del Modelo 5 sobre el conjunto de entrenamiento. Apreciamos como los errores son más bajos que para los del Modelo 4, lo que se debe a que disponemos de un mayor número de datos.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	689	162	134	25.83 %
CERRO	146	625	187	32.72 %
COMPETENCIA	94	142	608	34.55 %

Cuadro 3.28: Clasificación del Modelo 5 sobre el conjunto de entrenamiento.

En los Cuadros 3.29 y 3.30 se muestra la predicción del Modelo 5 sobre el mes de agosto de 2018, aunque los resultados son bastante similares a los obtenidos en el Modelo 4, parece que se comporta un poco mejor. Por ejemplo, en el caso de una probabilidad igual o superior al 90 % de que el local se vaya a la competencia, el Modelo 4 clasifica 24 locales de los que acierta 3, mientras que el Modelo 5 clasifica 47 locales de los que acierta 8.

Predicción	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	10707	196	10
CERRO	2104	368	18
COMPETENCIA	2048	106	74

Cuadro 3.29: Predicción para Agosto de 2018 del modelo 5.

Modelo 5	25 %			50 %			75 %			90 %		
Siguio	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA
CERRO	5141	554	55	1158	271	13	160	75	4	1	1	0
COMPETENCIA	4333	263	87	1129	63	62	274	12	20	38	1	8

Cuadro 3.30: Probabilidades de la predicción de abandono a la competencia del Modelo 5.

3.4.6. Modelo 6

Ahora pasaremos a comentar la predicción del Modelo 6 sobre agosto de 2018. Primeramente ajustaremos los hiperparámetros, en este caso, el mejor ajuste de hiperparámetros es $mtry = 14$ y $nodesize = 4$. Una vez optimizados estos, observamos en el Cuadro 3.31 el comportamiento del Modelo 6 sobre el conjunto de datos de entrenamiento. Como es lógico, debido a que tenemos un mayor número de categorías sobre las que realizar la predicción, obtenemos un mayor error que en los otros modelos.

Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO	ERROR
AGUA	126	3	12	15	39	54.18 %
DISTRIBUCIÓN	4	249	25	38	45	48.83 %
DOS	15	16	77	2	7	55.49 %
NO ABANDONO	39	86	32	708	166	23.78 %
CERRO	39	45	7	164	674	27.44 %

Cuadro 3.31: Clasificación del Modelo 6 sobre el conjunto de entrenamiento.

En los Cuadros 3.32 y 3.33 se muestran las predicciones del Modelo 6 sobre el mes de agosto de 2018. Como en el caso de la predicción de julio, el modelo distingue bastante bien entre los locales que se van a la competencia comprando al mes siguiente solo agua, solo distribución o los dos productos. A medida que aumentamos la probabilidad notamos una mayor precisión, pero no tan buena como la obtenida en el Modelo 5.

Predicción	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	14	0	1	294	29
DISTRIBUCIÓN	0	33	3	824	38
DOS	0	1	8	177	6
NO ABANDONO	4	7	1	11087	201
CERRO	12	14	4	2477	396

Cuadro 3.32: Predicción para Agosto de 2018 del Modelo 6.

Probabilidad	25 %				
Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	17	0	3	538	61
DISTRIBUCIÓN	0	35	4	1466	66
DOS	0	3	8	243	8
CERRO	22	19	4	5042	542

Probabilidad	50 %				
Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	11	0	0	122	15
DISTRIBUCIÓN	0	23	0	412	14
DOS	0	0	2	46	3
CERRO	5	8	0	1099	260

Probabilidad	75 %				
Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	3	0	0	22	6
DISTRIBUCIÓN	0	9	0	99	5
DOS	0	0	0	6	0
CERRO	2	2	0	182	77

Cuadro 3.33: Probabilidades de la predicción de Agosto de 2018 del Modelo 6.

3.5. Predicción Septiembre 2018

Para finalizar, realizaremos también la predicción de los 6 modelos sobre el mes de septiembre de 2018. En este mes abandonan un gran número de locales debido a que es cuando finaliza el verano.

3.5.1. Modelo 1

En el caso del Modelo 1, tras optimizar los siguientes hiperparámetros : $mtry = 2$ y $nodesize = 18$. Observamos en el Cuadro 3.34 el comportamiento del Modelo 1 sobre el conjunto de entrenamiento. Como en los casos anteriores, este método presenta gran precisión pero clasifica como abandono muy pocos locales.

Modelo	NO	SI	Error
NO	120991	3363	0.11 %
SI	139	393	89.53 %

Cuadro 3.34: Clasificación del Modelo 1 sobre el conjunto de entrenamiento.

En los Cuadros 3.35 y 3.36 observamos como el modelo detecta el mes en el cual estamos clasificando con bastante precisión el abandono, predominando en la construcción del modelo el mes en el que estamos y la duración del local.

Predicción	NO	SI
NO	14546	584
SI	24	147

Cuadro 3.35: Predicción para Septiembre de 2018 del Modelo 1.

Modelo 1	25 %		50 %		70 %		90 %	
Abandono	No	Si	No	Si	No	Si	No	Si
Clasificación	423	356	24	147	0	43	0	0

Cuadro 3.36: Probabilidades de la predicción de abandono del Modelo 1.

3.5.2. Modelo 2

A continuación veremos si el Modelo 2 no válido para predecir los locales que se van a la competencia o, sin embargo, para un mes como puede ser septiembre de 2018, sí lo es. En este caso tomamos $mtry = 4$ y $nodesize = 14$. Como podemos observar en los Cuadros 3.37 y 3.38 sigue clasificando de manera ineficaz los locales que van a la commpetencia, por tanto, el Modelo 2 no es un modelo fiable para localizar los locales que se van a la competencia.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	121028	2750	665	0.08 %
CERRO	102	329	12	89.31 %
COMPETENCIA	0	0	0	100 %

Cuadro 3.37: Clasificación del Modelo 2 sobre el conjunto de entrenamiento.

Predicción	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	14555	475	126
CERRO	15	127	3
COMPETENCIA	0	0	0

Cuadro 3.38: Predicción para Septiembre de 2018 del Modelo 2.

3.5.3. Modelo 3

Vamos a observar como funciona para la predicción de septiembre de 2018 el caso en el que los locales están balanceados, es decir, 50 % de los datos son abandonos y el otro 50 % no. Tras la optimización de los hiperparámteros obtenemos $mtry = 23$ y $nodesize = 11$. En el Cuadro 3.39 se muestra la clasificación del Modelo 3 respecto al conjunto de entrenamiento, conllevando un comportamiento bastante similar al de los meses anteriores.

Modelo	NO	SI	Error
NO	2969	787	20.95 %
SI	679	3077	18.07 %

Cuadro 3.39: Clasificación del Modelo 3 sobre el conjunto de entrenamiento.

En el caso de la predicción del mes de septiembre de 2018 del Modelo 3, se observa en los Cuadros 3.40 y 3.41 una buena clasificación tal y como ocurría en julio y agosto de 2018, llevando por ejemplo con una probabilidad de abandono igual o superior al 90 % a acertar 265 abandonos equivocándose solamente en 264.

Predicción	NO	SI
NO	11014	89
SI	3556	642

Cuadro 3.40: Predicción para Septiembre de 2018 del Modelo 3.

Modelo 3	25 %		50 %		70 %		90 %	
Abandono	No	Si	No	Si	No	Si	No	Si
Clasificación	7794	707	3556	642	1214	503	264	265

Cuadro 3.41: Probabilidades de la predicción de abandono del Modelo 3.

3.5.4. Modelo 4

Tanto en este apartado como en el siguiente veremos si el Modelo 4 y 5 son buenos para clasificar los locales que va a la competencia.En primer lugar empezaremos por el ajuste de los hiperparámetros, obteniendo $mtry = 18$ y $nodesize = 8$ como los hiperparámetros que minimizan el out-of-bag error. En el Cuadro 3.42 se muestra el comportamiento del Modelo 4 sobre el conjunto de datos de entrenamiento. Observamos que, igual que en los casos anteriores, el modelo sí permite localizar los posibles locales que se van a la competencia.

En los Cuadros 3.43 y 3.44 podemos observar la predicción para el mes de septiembre de 2018. Vemos como presenta un comportamiento similar a los meses anteriores demostrando la estabilidad de este modelo a lo largo de los meses.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	494	95	88	27.03 %
CERRO	115	459	103	32.20 %
COMPETENCIA	111	129	437	35.45 %

Cuadro 3.42: Clasificación del Modelo 4 sobre el conjunto de entrenamiento.

Modelo	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	10631	78	10
CERRO	1883	411	33
COMPETENCIA	2056	113	86

Cuadro 3.43: Predicción para Septiembre de 2018 del Modelo 4.

Modelo 4	25 %			50 %			75%			90 %		
Siguio	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA
CERRO	4861	538	56	976	353	24	152	132	5	3	6	0
COMPETENCIA	5101	269	114	1106	49	72	150	6	35	14	0	7

Cuadro 3.44: Probabilidades de la predicción de abandono a la competencia del Modelo 4.

3.5.5. Modelo 5

Otro de nuestros objetivos con la predicción del mes de septiembre de 2018 es observar qué modelo se comporta mejor, si el Modelo 4 o el Modelo 5. Para ello empezaremos optimizando los hiperparámetros, en este caso, $mtry = 5$ y $nodesize = 2$. En el Cuadro 3.45 observamos un menor error y una mayor precisión a lo obtenido en el Modelo 4.

Modelo	NO ABANDONO	CERRO	COMPETENCIA	Error
NO ABANDONO	726	155	131	24.68 %
CERRO	146	658	203	31.74 %
COMPETENCIA	92	151	630	34.64 %

Cuadro 3.45: Clasificación del Modelo 5 sobre el conjunto de entrenamiento.

En cuanto a la predicción del mes de septiembre de 2018 se puede observar en las gráficas 3.46 y 3.47 los resultados obtenidos. También vemos como en la gráfica 3.46 funciona mejor el Modelo 5 que la clasificación del Modelo 4, acertando más y equivocándose menos. De todos modos, los dos comportamientos son bastante similares.

Modelo	NO ABANDONO	CERRO	COMPETENCIA
NO ABANDONO	10311	62	11
CERRO	2308	450	31
COMPETENCIA	1951	90	87

Cuadro 3.46: Predicción para Septiembre de 2018 del Modelo 5.

Modelo 5	25 %			50 %			75%			90 %		
Siguio	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA	NO ABANDONO	CERRO	COMPETENCIA
CERRO	5580	555	67	1189	385	26	151	131	2	3	5	0
COMPETENCIA	4501	271	112	1054	47	74	236	4	36	21	1	8

Cuadro 3.47: Probabilidades de la predicción de abandono a la competencia del Modelo 5.

3.5.6. Modelo 6

Por último, para terminar, echemos un vistazo el comportamiento del Modelo 6. Tras el ajuste de hiperparámetros obtuvimos los valores $mtry = 6$ y $nodesize = 3$. En el Cuadro 3.48 se muestra la clasificación del Modelo 6 sobre el conjunto de datos de entrenamiento. Se aprecia un comportamiento bastante similar a los meses de julio y agosto de 2018.

Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO	ERROR
AGUA	117	4	12	36	104	57.14 %
DISTRIBUCIÓN	2	266	14	89	147	48.24 %
DOS	13	27	77	33	27	56.49 %
NO ABANDONO	14	38	10	744	158	22.82 %
CERRO	30	58	6	171	699	27.48 %

Cuadro 3.48: Clasificación del Modelo 6 sobre el conjunto de entrenamiento.

En cuanto a la predicción de septiembre de 2018, observamos los resultados obtenidos en los Cuadros 3.49 y 3.50. Aunque obtenemos bastante precisión, los resultados son peores que los obtenidos tanto por el Modelo 4 como por el Modelo 5.

Modelo 6	AGUA	DISTRIBUCIÓN	DOS	NO ABANDONO	CERRO
AGUA	23	0	3	201	13
DISTRIBUCIÓN	1	26	1	890	32
DOS	3	1	13	185	3
NO ABANDONO	5	2	7	10693	70
CERRO	21	15	8	2601	484

Cuadro 3.49: Predicción para Septiembre de 2018 del Modelo 6.

Porcentaje	25%				
Modelo 6	Agua	Distribución	Dos	No cerró	Nada
Agua	32	0	7	368	43
Distribución	5	33	4	1705	81
Dos	3	6	18	235	3
No	34	19	9	5349	549

Porcentaje	50%				
Modelo 6	Agua	Distribución	Dos	No cerró	Nada
Agua	16	0	0	70	8
Distribución	0	13	0	384	15
Dos	0	1	6	45	0
No	15	8	1	1174	377

Porcentaje	75%				
Modelo 6	Agua	Distribución	Dos	No cerró	Nada
Agua	2	0	0	2	0
Distribución	0	10	0	118	4
Dos	0	0	1	4	0
No	1	3	0	176	124

Cuadro 3.50: Probabilidades de la predicción de competencia del Modelo 6.

Capítulo 4

Conclusiones

A continuación se destacan las conclusiones obtenidas a lo largo de este trabajo:

En primer lugar, se han probado multitud de métodos: regresión logistica, árboles de decisión, bagging, Random Forest, AdaBoost, Gradient Boosting, Stochastic Gradient Boosting, XGBoost, redes neuronales y las técnicas machine learning disponibles en la plataforma Machine Learning dentro de Amazon Web Service. De entre todos ellos, la técnica que mejor resultados proporciona es el Random Forest.

Tras aplicar las técnicas Random Forest para 6 modelos diferentes con el objetivo de obtener la probabilidad de que un local abandone o vaya a la competencia, observamos que para el caso de los locales que abandonan, mientras que el Modelo 1 nos proporciona una buena precisión, el Modelo 3 nos clasifica un mayor número de datos. El hecho de que el Modelo 1 y el Modelo 3 se comporten de forma diferente debe ser tenido en cuenta por el promotor de ventas para seleccionar el modelo dependiendo de si su interés es obtener una buena precisión o clasificación. Para el caso de obtener la probabilidad de que un local se vaya a la competencia, tanto el Modelo 4 como el Modelo 5 son modelos bastante eficaces en su clasificación.

Además de funcionar bien el Random Forest como un método para predecir el abandono, también funciona bien para predecir qué locales van a la competencia. Esto tiene cierto interés debido a que si se logra evitar la fuga de un cliente a la competencia, se consigue reducir la capacidad de atracción de clientela de un competidor.

Del mismo modo que se ha realizado este trabajo para obtener la tasa de abandono mensual de cerveza, también se puede obtener fácilmente para un periodo semanal y para otros productos como pueden ser agua, distribución, 1906, etc. cuyos resultados se muestran en los Anexos I, II y III. Por ejemplo, siendo este trabajo comentado con un promotor de ventas, uno de los indicios de que un local deja de comprar cerveza suele ser cuando dejan de comprar agua o la marca de cerveza 1906. Por tanto, obtener la tasa de abandono de estos productos puede resultar bastante interesante, permitiendo a la empresa ahorrar costes y aumentar sus beneficios.

Para el desarrollo de este trabajo, un aspecto destacado ha sido hacer frente a la gran carga computacional requerida, para lo que ha sido necesario emplear una máquina virtual con 8 núcleos y 24 gigas de RAM. En la actualidad, se están realizando las compilaciones en una máquina virtual más potente, con 15 núcleos y 122 gigas de RAM, agilizando considerablemente la obtención de los resultados.

Este trabajo ha proporcionado buenos resultados desde el punto de vista empresarial, pudiendo con ello enviar un aviso al promotor de ventas a fin de poder anticiparse tanto al abandono mensual

como semanal de los clientes que compran un determinado producto. La idea original del proyecto fue obtener la tasa de abandono, pero a medida que se fue trabajando en el mismo, se avanzó y se decidió probar a obtener la tasa de abandono de los locales que se han ido competencia obteniendo buenos resultados.

Bibliografía

[1] Amit, Y., Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural computation, 9(7), 1545-1588.

[2] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984). Classification and regression trees. Wadsworth Inc., Belmont,

[3] Breiman, L. (1996). Bagging predictors. Machine learning, 24 (2), 123-140.

[4] Breiman, L. (2001). Random forests. Machine learning, 45 (1), 5-32.

[5] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794). ACM.

[6] Dietterich, T. G. (1998). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. Machine learning, 32, 1-22.

[7] Freund, Y., Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In Icml (Vol. 96, pp. 148-156).

[8] Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 14 (771-780), 1612.

[9] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[10] Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4), 367-378.

[11] Ho, T:K:(1998). The random subspace method for constructing decision forest. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(5), 473-490

[12] Hosmer, D.W. and Lemeshow, S. (1989). Applied Logistic Regression. Wiley.

[13] Kuhn, M. and Quinlan, R. C50: C5.0 Decision Trees and Rule-Based Models. (2018). R package version 0.1.2.

https://CRAN.R-project.org/package=C50

[14] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: springer.

[15] LeDell, E. et al. (2018). h2o: R Interface for 'H2O' R package version 3.20.0.2

https://CRAN.R-project.org/package=h2o

[16] Quinlan, J. R. (1997). C5. 0 Data Mining Tool. RuleQuest Research.

[17] Fortran original by Leo Breiman and Adele Cutler, R port by Andy Liaw and Matthew Wiener (2018). Classification and Regression by randomForest. R package version 4.6-14

https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[18] Zurada, J. M. (1992). Introduction to artificial neural systems (Vol. 8). St. Paul: West publishing company.

[19] Ripley, B. and Lapsley, M. (2017). RODBC: ODBC Database Access R package version 1.3-15

https://cran.r-project.org/web/packages/RODBC/RODBC.pdf

[20] Amazon Web Service: Machine Learning: (Enero 2019).

https://aws.amazon.com/es/aml/

Anexo I: Resultados de abandono de cerveza sobre datos semanales

Además de obtener la tasa mensual de abandono y la tasa de locales que se van a la competencia, se trabajó en la obtención de estas tasas de manera semanal. Para el caso semanal se considera un local como cliente en el momento que lleva comprando cerveza durante un periodo igual o superior a 4 semanas, y diremos que un local abandona si a las dos semanas posteriores de la semana sobre la que queremos predecir no realiza ningún pedido de cerveza.

En este apéndice se tratará la predicción de la tasa de abandono de cerveza realizada para la semana 46 del año 2018. Esta semana consta de 8766 locales que se clasifican de la siguiente manera: 8289 locales siguen siendo clientes, mientras que 477 locales se clasifican como abandono. De estos 477 abandonos, 62 locales se han clasificado que han ido a la competencia, es decir, a la semana siguiente al abandono realizaron al menos un pedido de agua o distribución.

En el Cuadro 1 se muestra la evolución de la tasa de abandono tras realizar la predicción de la semana 46 de 2018 sobre el Modelo 3 (datos balanceados, Sección 3.1.1) tras el ajuste de los hiperparámetos. En nuestro caso obtuvimos, $mtry = 15$ y $nodesize = 18$. Vemos como a partir de la probabilidad del 75 % de abandono ya obtenemos resultados aceptables, puesto que el modelo acertaría 149 abandonos equivocándose en 378.

Modelo 3	25 %		50 %		75 %		85 %		90 %		95 %	
Abandono	No	Si	No	Si	No	Si	No	Si	No	Si	No	Si
Clasificación	5828	455	2043	341	378	149	134	75	55	43	32	37

Cuadro 1: Probabilidades de la tasa semanal de abandono en la venta de cerveza.

En el Cuadro 2 se muestra como varía la clasificación de los locales que van a la competencia cuando aumentamos o disminuimos la probabilidad en el Modelo 4 (datos balanceados para la competencia, Sección 3.1.1) tras optimizar los hiperparámetros $mtry = 6$ y $nodesize = 18$. Nos damos cuenta que con una probabilidad del 85 % de abandono a la competencia obtenemos un buen resultado, puesto que acierta 4 equivocándose solamente en 1.

Modelo 4	25%			50%			75%			85%		
Siguio	Cliente	Cerró	Competencia	Cliente	Cerró	Competencia	Cliente	Cerró	Competencia	Cliente	Cerró	Competencia
Cerró	3653	368	36	389	140	10	32	26	1	8	11	0
Competencia	2643	151	47	589	31	23	61	1	7	1	0	4

Cuadro 2: Probabilidades de la tasa semanal de competencia en la venta de cerveza.

Anexo II: Resultados de abandono de agua sobre datos semanales

En este anexo se comenta la predicción de la tasa de abandono semanal de agua sobre la semana 46 de 2018. En este caso la semana 46 de 2018 consta de 4047 locales, de los cuales no abandonaron 3699 y abandonaron 348. Para ello se aplicó el Modelo 3 (Sección 3.1.1) con los hiperparámetros optimizados $mtry = 6$ y $nodesize = 18$. Los resultados se muestran en el Cuadro 3, dónde se aprecia que al igual que ocurría en las probabilidades de abandono semanal de cerveza, a partir de una probabilidad del 75 % de abandono, obtenemos unos buenos resultados.

Modelo 3	25 %		50 %		75 %		85 %		90 %		95 %	
Abandono	No	Si	No	Si	No	Si	No	Si	No	Si	No	Si
Clasificación	2712	338	1136	246	148	84	33	22	7	8	1	0

Cuadro 3: Probabilidades de la tasa semanal de abandono en la venta de agua.

Anexo III: Resultados de abandono de 1906 sobre datos semanales

Por último, trataremos la tasa de abandono semanal sobre la marca de cerveza 1906 para la semana 46 de 2018. En este caso disponemos de un total de 3207 locales, de los que 2893 siguen siendo clientes y 314 abandonan. La obtención de la tasa semanal de abandono se realizó sobre el Modelo 3 (Sección 3.1.1) con los hiperparámetros $mtry = 22$ y $nodesize = 9$. Los resultados obtenidos se muestran el Cuadro 4. De la misma manera que en los otros dos anexos, observamos como a partir de una probabilidad del 75 % de abandono obtenemos unos resultados interesantes desde el punto de vista económico.

Modelo 3	25 %		50 %		75 %		85 %		90 %		95 %	
Abandono	No	Si	No	Si	No	Si	No	Si	No	Si	No	Si
Clasificación	2057	303	727	188	90	45	21	10	4	5	0	0

Cuadro 4: Probabilidades de la tasa semanal de abandono en la venta de 1906.