Universidad de Valladolid

Facultad de Ciencias

Trabajo Fin de Grado

Grado en estadística

Herramientas para el BigData y Machine Learning

Autor:

Rafael González-Iglesias González

Tutor:

Jesús Sáez Aguado

TABLA DE CONTENIDOS

Tabla de contenidos

1. Resumen

La academia, dentro del contexto del plan Bolonia, ha incrementado significativamente la puesta en práctica de los conocimientos. Sin embargo, la reducción temporal que ha supuesto, también ha acotado la cantidad de conocimientos adquiridos. Este trabajo pretende aportar un conocimiento más enfocado a la vida laboral. Se trata de un pequeño manual autobiográfico en el que se describirán conceptos y herramientas útiles en la práctica profesional, con la intención de incentivar la curiosidad en el lector y de aportar una guía al futuro recién graduado. Junto con descripciones generalistas en áreas no estrictamente del trabajo del estadístico se aportan ejemplos prácticos. En el caso de los conceptos propios del grado, se incluye un ejemplo desarrollado en mayor profundidad con datos de Kaggle.

The academia, within the framework of the Bologna Process, has significantly improved the hands-on knowledge and know-how. Nevertheless, time constraints included in the process have also shortened the amount of knowledge acquired. This study is intended to contribute with a knowledge focused on the professional field. It is a brief autobiographic manual which will describe useful concepts and tools useful for professional practice, in an attempt to stir the curiosity of its readers, and to provide a guide for future graduates. Along with generalist descriptions in not strictly statistician's matters, there are practical examples. In the case of those concepts more related to the degree, there is an example further developed, with Kaggle data.

2. Introducción

El BigData consiste en la recopilación de grandes volúmenes de información. La humanidad nunca ha dejado de recopilar información, sin embargo las capacidades y necesidades han variado a lo largo del tiempo, desde las primeras formas de escritura hasta los sistemas modernos de almacenamiento de datos. (1)

La historia del BigData se remonta hasta la Segunda Guerra Mundial. La intención de cuantificarar la información ha tenido todo tipo de fines, pero en los últimos 30 años ha cobrado una importancia mediática muy relevante debido a las nuevas fuentes de información y a la apreciación de las capacidades de tratamiento de estos datos almacenados.

Hoy en día el BigData es un término que hace referencia a volúmenes de datos pertenecientes tanto a grandes como pequeñas empresas multinacionales, cuando el volumen de datos supera una cantidad de varios Gb y los requisitos propios de actualización de la estructura de datos conllevan una complejidad que precisa equipos completos especializados. Normalmente los centros de almacenamiento de estos datos se denominan *Data Warehouses*.

Este campo se está convirtiendo en una herramienta tan necesaria como la seguridad o la contabilidad en cualquier empresa, y su conjunción con los medios sociales lo convierten en un potente elemento en la atención al cliente.

Veamos el ejemplo descrito por Erik Jensen, (2) que incide en cómo el BigData y la analítica pueden tener una influencia significativa a la hora de aumentar el compromiso de los clientes:

Como parte de una estrategia de relación empresa-cliente, JetBlue Airways (JBA) monitorea las redes sociales en busca de menciones al nombre de su empresa. Probablemente, este proceso está automatizado, y envía alertas a los empleados del departamento de atención al cliente (AC). Esto permite a los empleados una respuesta rápida frente a los cambios de la sensación que los clientes tienen de la empresa, incluso cuando no se comunican con la empresa de forma directa.

En este caso particular, un cliente se quejaba por Twitter sobre un vuelo retrasado. Primero, JBA respondió en menos de una hora demostrando al cliente (y a todo el que siguiera su queja) que JBA se preocupaba por su experiencia y su tiempo. Segundo, JBA pudo ofrecerle actualizaciones sobre el estado de su vuelo.

Es un escenario que no plantea grandes retos de lógica, el cliente no estará satisfecho del todo pero se minimiza una crítica negativa, pública y lícita hacia la compañía.

Si se tiene en cuenta que en 2014 se retrasaron 67.445 vuelos de JBA, con una media de 100 pasajeros por vuelo y el 5% de los clientes escribirán una crítica negativa en redes sociales por caso, habrá 337.225 tweets negativos acerca de JBA.

Según un estudio de American Express (3) el 59% de los clientes probarán una compañía con un mejor servicio de atención al cliente. Lo que supone una pérdida potencial de 100 millones de dólares si se asume que el precio del billete es de 500 dólares y que cada cliente realiza este viaje una vez al año.

Se estima que la compañía ahorra un millón de dólares por cada 1% de clientes sobre los que el servicio de atención al cliente ejerce una pequeña influencia si tenemos en cuenta a los clientes con intención de cambiar de compañía y a aquellos que potencialmente leen sus comentarios.

Por burdas que sean las aproximaciones, la suma es razonable y la gestión y coste de los servicios de Business Analytics para este proceso son mínimos frente al ahorro potencial.

En el ejemplo intervienen dos elementos principales que permiten la pronta actuación del departamento de atención al cliente:

1. Un elemento que recopile y procese la información de las redes sociales.
2. Un sistema rápido de almacenamiento y consulta donde volcar y procesar la información.

El primer elemento comúnmente contará con un lector automatizado de contenido web (parser) y un modelo estadístico que sea capaz de interpretar como buenos o malos los comentarios recogidos por el parser.

El segundo elemento será la base de datos, con capacidad para almacenar y recoger información de forma eficiente, y a la vez de realizar consultas en tiempo real.

La base de datos por lo tanto no será sólo un cajón de datos, sino un sistema automatizado que recoge la información. Está construido por arquitectos de bases de datos que permiten que la recogida sea lo más eficiente posible y a la vez esté indexada de tal manera que las consultas no supongan un esfuerzo eterno. El reto de diseñar este sistema se ve influido por cada pequeño detalle referente a los datos y a su uso.

El BigData y la Ingeniería Estadística son campos que van de la mano en su crecimiento; entender y apreciar las cualidades de los sistemas de almacenamiento provee al estadístico con mayor destreza en el desarrollo de sus capacidades.

3. Terminología

Arboles de decisión: modelos de decisión basados en grafos de estructura tipo árbol, en los que en cada rama se evalúa el resultado de segmentar el conjunto a estudio por una dicotomía de un caso por una variable independiente.

Boosting: método generalmente usado en la mejora de la precisión de cualquier algoritmo de aprendizaje. Se basa en el concepto de que es más fácil buscar reglas de predicción entre muchas reglas básicas, que buscar una sola regla única y válida para el conjunto de predicción. Es un método para encontrar reglas avanzadas de predicción a partir de otras más simples. (Schapire)

ETL: En computación es el término referido a Extract, Transform, Load (Extraer, transformar, cargar). Suele hacer referencia a aquellos entornos o paquetes de software que comunican unos programas o lenguajes con otros. Por ejemplo, el paquete MySQL de R es un ETL entre R y una base de datos MySQL.

Machine Learning: Se trata de un término que trata de abarcar un amplio conjunto de elementos. Estrictamente el Machine Learning se da en la práctica en pocos y raros casos, como en el caso de Watson IBM, o del robot de Honda ASIMO. Se trata de software capaz de generar instancias de reglas de forma autónoma, basados en sucesos de reconocimiento instantáneo. Generalmente, el uso que se hace de Machine Learning (o Aprendizaje automático), hace alusión al análisis de modelos de complejidad media-alta, por medio de algoritmos analíticos vorazes basados en algún criterio de mejora incremental de los resultados por medio de un alto consumo de procesamiento computacional. En la industria, el departamento que se encarga de diseñar estos modelos, suele llevar el nombre de Bussines Intelligence (BI), o Advanced Analytics. Generalmente este campo está estrechamente relacionado con el BigData, ya que en muchas ocasiones la complejidad del problema no proviene sólo del escollo resolutivo, sino de la ingente cantidad de información.

4. Herramientas

4.1 Software de BD

El conjunto de programas de manejo de bases de datos (DBMS) es amplio e inabarcable para el presente trabajo. Existen dos grandes diferencias en la estructura de las bases de datos con las que conviene estar familiarizado o al menos brevemente informado.

La base de datos es el concepto que hace referencia a cómo está organizada una información, que puede ser de estructura relacional o no relacional, e influye determinantemente en el tratamiento de los datos. A continuación se especifica de forma general qué supone cada una y a qué se debe su existencia:

1. El concepto de base de datos relacional es una estructura de datos que permite que la información esté relacionada entre sí, entre distintas 'tablas' o diferentes tipos de contenedores de datos.
   Un contenedor de datos debe contener una clave primaria que permita conectar de manera unívoca con cada instancia del contenedor. Desde otros contenedores se podrán crear instancias que apunten a las claves primarias de las anteriores tablas, teniendo siempre identificadores únicos.

2. Una base de datos no relacional almacena la información sin una estructura específica ni mecanismo de enlace de los datos de un contenedor a otro definido de base.

La diferenciación de estos dos tipos es más conocida como SQL frente a non-SQL, que proviene del término inglés *structural query language*.

En el caso de las grandes bases de datos, ambas estructuras son comúnmente usadas, aunque la tendencia en los DW es emplear las no relacionales. Esto se debe a que las bases de datos no relacionales permiten una relajación de la información.

Supóngase un caso práctico:

Un empleado del ayuntamiento de Simancas debe generar una nueva base de datos para recoger información sobre los habitantes del pueblo. En dicha ocasión el empleado recurre a un sistema estructurado en el que el identificador único de

cada habitante sea su DNI. Dentro de la tabla "HABITANTE" además almacena el domicilio y color de ojos. A la anterior tabla le sigue tabla que informa de las características del domicilio del habitante "DOMICILIO" teniendo como identificador único la dirección. Cada instancia en la última tabla puede estar repetida en la tabla "HABITANTE" ya que varias personas pueden convivir en un mismo hogar.

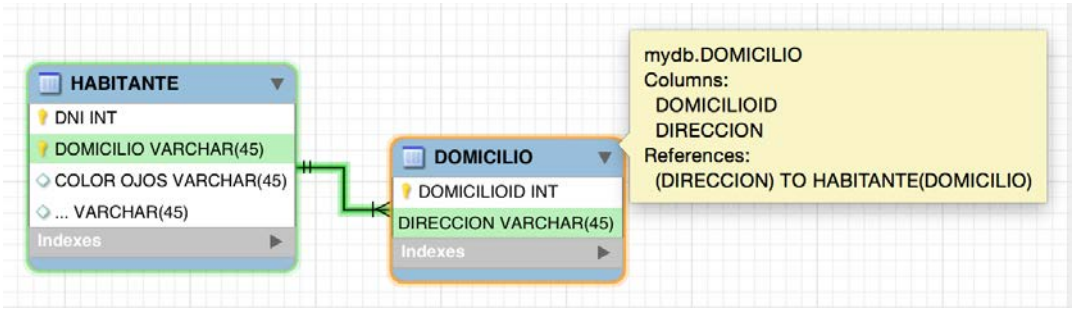

Figura 1: Esquema de BD

A lo largo del tiempo crece el número de tablas y la información en ellas, pero se mantiene la misma estructura inicial, cada habitante debe tener un domicilio, una fecha de nacimiento, cada domicilio un registro en hacienda, cada habitante una madre, etc., de forma que cada nueva instancia registrada no aparece en la base de datos hasta que se cumplan todas las condiciones de llenado de los campos dependientes entre sí. No se podrá registrar un domicilio sin dueño ni habitantes sin edad.

Por su trabajo, excepcionalmente realizado, es contratado por el Corte Inglés para desarrollar una tarea similar con sus clientes.

En esta ocasión le será mucho más complicado generar una estructura tan estricta de la base de datos, ya que los 10.5 millones de clientes (5) no están igualmente informados, debido a los cambios de políticas de empresa, los nuevos clientes generados en cada fecha eran encuestados sobre distinta información sin obligación a contestar. Además existen errores de anteriores arquitectos de bases de datos que aportan mayor complejidad al problema.

El trabajador recurre en esta ocasión a una estructura no relacional en la que no se suele invertir tanto esfuerzo en la definición de las propiedades de los datos, y que permiten mostrar información parcial o con campos ausentes. En dicha base,

aunque se imponga la condición de que cada cliente debe tener un nombre, se podrán almacenar instancias sin nombre que sean unívocamente diferenciables del resto.

Cuando el volumen y la calidad de los datos alcanzan niveles en los que el esfuerzo de generación de tablas SQL supera la inversión prevista, se recurre a estructuras non-SQL.

Las bases de datos, sin embargo, están inicialmente especializadas en el almacenaje y no en el manejo, extracción y transformación de estos. La conexión entre los procesadores de datos y las BD depende del software utilizado para su manejo (DBMS) (MySQL, mongoDB, …) y de las herramientas asociadas.

De entre las posibles herramientas, hay dos clases especialmente relevantes y que en ocasiones pueden ser confundidas con DBMS en sí mismos. Por un lado están los sistemas de bases de datos en paralelo y por otro el MapReduce (MR).

- A mediados de 1980 se implementó un paradigma de paralelización pionero que se basaba en un cluster de nodos de procesadores separados entre sí y conectados por una interconexión de alta velocidad. En esta estructura, una consulta implicaría una primera partición horizontal de la información BD SQL y a su vez de una segunda partición de la ejecución de la consulta, de forma que, con la partición horizontal se distribuyan las filas en el cluster de manera que se puedan procesar de forma independiente y paralela. La paralelización de procesos tiene muchos formatos comerciales, pero exige en muchos casos el uso de estructuras SQL.

- MR tiene como cualidad atractiva su simplicidad. Consiste en dos funciones básicas (Map y Reduce) que el usuario especifica para procesar pares de claves/valor. Este sistema fue popularizado por Google en 2004 y hoy en día existen varias opciones comerciales y de software libre como Hadoop o Spark.

Los sistemas en paralelo son especialmente eficaces en consultas de grandes BD y MR en analítica compleja. (5)

4.2 Software para el tratamiento de datos

En las tareas de análisis de datos se debe tener en cuenta siempre un profundo trabajo previo del problema y su entorno, y por otro lado la estadística puramente descriptiva. Posteriormente a este análisis, se podrá comenzar el proceso de estadística analítica.

Para el tratamiento de estadística puramente descriptiva existen muchas herramientas enfocadas a esta tarea, como Excel, Tableau, o Tibco Spotfire (de pago) y Google Docs o Ruby (software libre). Se trata de programas que permiten cierta flexibilidad en las consultas al usuario, pero especialmente enfocados a la generación de gráficos y muestra de estadísticas básicos de los datos. En su uso se deben tener en cuenta las capacidades de tratamiento de datos, generalmente se usan con muestras representativas de la BD y no con todas las disponibles.

Situado entre estos dos mundos se encuentra Orange (http://orange.biolab.si/), que permite realizar análisis estadísticos muy avanzados con gran capacidad de modificación por parte del usuario y con una interfaz intuitiva.

Para el análisis estadístico final están marcándose grandes diferencias entre los programas de software libre y gratuitos frente a los programas de pago. Por un lado, se encontrarían R y Phyton con las librerías NumPy y Panda con una gran comunidad científica e informática detrás. Por otro lado estarían programas como SAS, SPSS, StatGraphics, XPRESS Mossel, AMPL, etc. Son programas con licencias relativamente costosas para empresas medianas pero con mantenimiento y unas ciertas garantías. Generalmente Las empresas grandes utilizan software de pago, mientras que en las empresas pequeñas o en investigación se recurre a software libre. Sin embargo cada vez más empresas grandes utilizan software libre, ya sea de forma paralela o de manera exclusiva.

Dentro del enfoque de negocio, el trabajo de Business Inteligence debe compensar y equilibrar tanto la calidad del análisis, como la interpretabilidad y usabilidad de los resultados. Entender y comprender los problemas que se plantean siendo capaz de resolverlos, debe compaginarse con la lectura de más bajo nivel analítico de las soluciones para el resto del equipo de negocio.

Por ello hoy en día está apareciendo cada vez más software enfocado a la implementación del perfil más comercial de las empresas. Programas como Watson Analytics buscan mejorar la capacidad de consulta y de acceso a la información de los usuarios ajenos al análisis de datos, a cambio de restringir la maniobrabilidad. Aunque esto no debe desalentar al analista de su uso, cuanto mayor y más profundo sea el campo de conocimiento y la versatilidad, mayor será el posible rendimiento y oportunidades laborales de un analista.

Para la ejecución del ejemplo propuesto, así como en los enfoques de resolución, se utilizará R.

R es un programa muy completo y con una amplia comunidad detrás, que no debe desestimarse. Recursos como http://stackoverflow.com/, https://www.r-project.org/ o http://www.inside-r.org/ pueden ser instrumentos de gran utilidad. En la mayoría de los casos, los problemas que se encuentran en un análisis estadístico ya han sido encontrados, preguntados y respondidos en estos lugares.

En este trabajo se pretende aportar un conocimiento enriquecedor de ciertos paquetes que pueden servir para mejorar significativamente el rendimiento y el esfuerzo, y que han demostrado ser valiosos en la práctica diaria.

4.3 Paquetes

data.table: Entender el funcionamiento interno de R puede suponer un gran aumento en el rendimiento del usuario, pero existen implementaciones de calidad que facilitan en ocasiones la ausencia de conocimiento. La interesante historia de este paquete está marcada por el inconformismo de un joven estudiante de matemáticas y computación llamado Matt Dowle. Tras acabar la carrera en 1996 comenzó a trabajar en Lehman Brothers donde aprendió los entresijos de las BD SQL. En 1999 cambió la empresa por Salomon Brothers, donde le enseñaron la comodidad de trabajar con un lenguaje orientado al procesamiento de datos con S-PLUS. La intuitiva comprensión del funcionamiento de las estructuras Data.Frame sobre las que estaba construida S fue toda una revelación. A continuación se presenta la transcripción que él mismo hizo de la conversación con su nuevo jefe (Pat) al mostrarle por primera vez el funcionamiento de S-PLUS:

Pat: Es un conjunto de columnas. Todas las columnas son del mismo tamaño pero pueden ser de diferentes clases.

```
> DF <- data.frame( A = letters[1:3], B =  c(1,3,5) )
```

Matt: ¿Entonces, dataframe es como una tabla de una base de datos?

Pat: Sí.

Matt: Genial, lo pillo. ¿No tuviste que hacer CREATE TABLE primero y luego INSERT datos?

Pat: Correcto. Es un solo paso

Matt: ¡Enséñame más!

Matt: ¡Vaya! ¿No necesito crear una columna conteniendo el número de columnas como en SQL?

```
> DF[2:3,]
  A B
2 b 3
3 c 5
```

Pat: ¡No señor! El orden de las filas es como

se almacena en memoria. Por eso es
tan bueno para las series temporales.

(7)

Su fascinación frenó a la siguiente pregunta, cuando Matt planteó que si DF[2:3 , sum(B)] devolvía 8. Es fácil comprobar que hoy en día eso devuelve un error tanto en R como en S-PLUS.

Tiempo después, tras esta conversación, Matt encontró un error de software que el servicio técnico de S-PLUS no arreglaría hasta la salida de la nueva actualización del paquete, sin embargo frente a su insistencia, Pat le referenció el software R, que tenía el código abierto y sobre el que podría arreglar todas las faltas que encontrase por sí mismo.

Matt implementó el código de S-PLUS directamente en R y realizó la operación que habitualmente le llevaba una hora en un solo minuto. Rápidamente Matt descubrió las ventajas de R y decidió solucionar el mismo aquel problema que le planteó a Pat.

El resultado de esta inquietud de Matt fue el paquete data.table, que merece especial atención frente a los demás mencionados, al ser computacionalmente extremadamente más eficiente frente a las anteriores dataframe, y por su versatilidad de operaciones. Matt combinó en una estructura de datos intuitiva, la fuerza bruta de un DBMS, con la capacidad de incluir las consultas analíticas de R en las consultas. En la Figura 2 (7) se puede observar un test de velocidad de la estructura data.table frente a otro paquete de R comparable "dplyr" y la versión de Phyton para el análisis de datos.

La velocidad de procesamiento de este test con los data.tables sólo se ve superada por panda en el Test 4, y en ningún caso por dplyr. La capacidad de este paquete ha sido evaluada y agradecida en innumerables ocasiones por el autor de este trabajo, permitiéndole implementar soluciones ad hoc que procesan un 33% más rápido consultas a base de datos de 500Gb. La velocidad de indexación de campos frente a la obtenida con paquetes como MySQL da como resultado comparaciones de tiempo de instantes frente a incapacidad de proceso.

La estructura básica de un data.table consiste en tres piezas fundamentales:

DT [i, j, by]

i : Permite hacer una selección de campos de una tabla en base a una condición que se cumpla en alguna o varias de sus columnas o también una selección por fila de forma numérica.

j : Puede realizar tanto selecciones de columnas como aplicar cualquier función aplicable a un dataframe, sobre las columnas deseadas de la tabla, incluso habiendo hecho la selección previa en i.

by : Permite hacer agrupaciones por cualquier clasificación de una de las columnas.

Por ejemplo:

DT[clase %in% "mamíferos" & extinct == 1 , plot(fecha, cantidad), by = nombre]

Realizaría un gráfico histórico de la población, por cada especie animal que estuviese extinta y fuese mamífera de nuestra base de datos.

Como lectura recomendada podría comenzarse con la explicación de las ventajas del paquete que se pueden encontrar en (9).

caret: La función preProcess proviene del paquete "caret" (classification and regression training). El paquete caret fue creado para el aprendizaje automático bajo la filosofía de "no free lunch" ("no hay comida gratis") que establece en ausencia de conocimiento previo de un problema de predicción, no hay un método que pueda decirse que es mejor que otro.

Bajo esta filosofía, el paquete facilita la experimentación empírica para llevar a cabo el análisis. Hay cuatro facetas principales en las que este paquete trata de facilitar la tarea del analista:

Figura 2: Benchmark de funcionamiento para data.table, dplyr con data.frame y python con panda

Creación estándar de modelos: Provee una interfaz consistente con la gran mayoría de algoritmos en R para el entrenamiento de modelos sobre muestras de gran tamaño. La función createDataPartition extrae muestras con diferentes implementaciones (m.a.s., bootstrap, pesos…) de la variable objetivo.

Evaluar el efecto de los parámetros: Funciones como preProcess permiten realizar a la vez un análisis para transformación de Box-Cox, normalización, estandarización,

análisis de componentes principales, etc. Además es capaz de realizar imputación de datos por medio de diferentes algoritmos de imputación.

Selección del modelo: Aporta herramientas que evalúan la calidad de los modelos bajo diferentes criterios de la función objetivo, tales como el AUC, AIC, BIC, R2, etc.

Estimar la precisión del modelo y ajustar los parámetros: La conjunción de las funciones train y resample permite iterar con diferentes parámetros de un modelo. Por ejemplo, en un modelo LASSO podrían evaluarse las diferentes combinaciones de α y β. (9)

imputeR: En ocasiones los datos corruptos en un problema de grandes dimensiones pueden ser tratados por medio de la estimación de los posibles valores corruptos. Existen varias opciones para hacer frente a este problema, y la elección depende del analista y del caso. Se debe evaluar si los datos corruptos están focalizados en pocas variables y la posibilidad de prescindir de ellas, en otro caso podría intentar hacerse una imputación de variables que no afectase a la matriz de covarianzas, o finalmente una imputación basada en técnicas estadísticas. Para facilitar de nuevo la vida del experimentador está el paquete imputeR. (11) El interés de este paquete no queda restringido a su utilidad, sino a conocer las técnicas de los distintos modelos implementados en él. Entre los más destacables se encuentran modelos como:

- **Cubist:** Es un modelo de reglas, que resulta de la extensión del modelo M5 de Quinlan, en el que las hojas terminales de cada rama son modelos lineales de regresión. Estos modelos se basan en los predictores obtenidos por las ramificaciones de cada entramado del árbol. También hay modelos intermedios a cada nivel del árbol. La predicción se basa en el resultado de la regresión de los nodos terminales, pero está suavizada por los modelos previos intermedios, de forma iterativa. El árbol posteriormente es podado y reducido a un conjunto de reglas que inicialmente son recorridos desde el inicio del árbol hasta el final en una hoja. Por último se realiza una segunda simplificación, en este caso de las reglas, o bien por poda o por simplificación de reglas similares. (11)

- **Gbm**: Este modelo se está convirtiendo en un clásico del aprendizaje automático. Su nombre proviene de "generalized boosting regression model" implementado en R por Greg Ridgeway en 2012 basado en la máquina de gradient boosting de J. Friedman y en el modelo AdaBoost de Freund y Schapire. En estos modelos se implementa mediante una función de ganancia de información un doble muestreo bootstrap. Por un lado se genera una cantidad N de árboles de decisión, en los que cada árbol tiene una muestra distinta de observaciones, extraída del conjunto inicial. Por otro lado, en cada rama del árbol, sólo una muestra de las variables es evaluada mediante la función de ganancia de información. Por último se hace sistema por votación entre los resultados de todos los árboles en la predicción del modelo. El término "generalized" hace referencia a la capacidad del experimentador, de evaluar distintas funciones para las variables, como por ejemplo seleccionar la función de riesgos proporcionales de Cox, o AdaBoost, pasando por la función Gaussiana, Gamma, logística, etc. (12)Resultan de especial interés las distintas charlas impartidas por los creadores de estas funciones, que están disponibles en internet, como la que se puede encontrar en https://www.youtube.com/watch?v=wPqtzj5VZus.

- **Lasso**: "Least Absolute Shrinkage and Selection Operator" (Sesgo absoluto y selección de operador mínimos). En el algoritmo Lasso se utiliza un procedimiento similar al RIDGE, permitiendo un modelo sesgado, en el que la función a minimizar añade a la función del error mínimo cuadrático el término [lambda]:

$$\hat{\beta}^{L} = \arg\min\left(\sum_{i=1}^{N} (y_i - \alpha - \sum_j \beta_j x_{ij})^2 + \lambda \sum_j |\beta_j| \right),$$

  Al optimizar de esta manera las constantes de los estimadores de los parámetros, se permite que el valor de beta sea 0, con lo que sirve de método de selección de variables, a la vez que es capaz de utilizar distintas funciones de distribución para la variable objetivo. (14)

Por último este paquete además incluye funciones de evaluación de los datos imputados, somo SimImpute o SimEval.

RMySQL: Este paquete consiste en un ETL que hace fácil la comunicación directa entre R y una base de datos MySQL. De entre los paquetes anteriormente mencionados, este es el único que no tendrá un uso directo sobre el ejemplo. Sin embargo es destacable que en conjunción con data.table, este paquete alcanza y supera el funcionamiento propio de distribuciones como MySQL Workbench.

La funcionalidad de esta herramienta no reside en una simplificación del lenguaje con la BD. De hecho, mantiene la misma sintaxis que MySQL, pero convierte a R en un DBMS potente, con la capacidad propia de análisis de R. (15)

5. Caso práctico

A continuación se mostrará mediante un ejemplo práctico con los datos de una competición activa en Kaggle (15). Que consiste en predecir un problema del tipo "next product to buy" (el próximo producto a comprar).

5.1 Descripción del problema

En la sección anterior se comenta un hecho que es de especial relevancia en cualquier problema y al que en ocasiones, por falta de tiempo principalmente, no se le presta la atención necesaria. Comprender todos los aspectos que rodean un problema es fundamental y enriquecedor en su resolución.

El enunciado propuesto es el siguiente:

"Recruit Ponpare es el principal sitio web de cupones de Japón, que ofrece grandes descuentos en diversos productos, desde clases de yoga a altas temperaturas hasta sushi gourmet, pasando por un festival de conciertos de verano. Los cupones de Ponpare abren puertas para los compradores que ellos sólo habían soñado con traspasar. Pueden aprender habilidades difíciles de alcanzar, aventurarse en la inmensidad, y cenar como (y con) las estrellas.

Invertir en una nueva experiencia no es barato. Tememos gastar el tiempo y el dinero en un producto o servicio que no nos entretenga o no lleguemos a comprender completamente. Ponpare elimina ese elevado coste de la ecuación, y hace que para ti resulte sencillo dar el paso hacia tu primer salto base o hacia tu diamante para el anillo de compromiso.

Mediante el historial de compras y navegación del usuario, nuestra competición te pide que predigas qué cupones comprará un usuario en un periodo determinado de tiempo. El modelo resultante será usado para mejorar el sistema de recomendaciones de Ponpare para que pueda asegurarse de que sus clientes no se pierdan su próximo entretenimiento favorito."

Las reglas de entrega de la respuesta incluyen que cada usuario sobre el que se predice tenga la siguiente descripción:

USER_ID_hash,PURCHASED_COUPONS
0004901ba699a49fd93a3c6bb1768b8f,hash4
0006d6ac7c6ef3fc0ab0dc40deb3c960,hash1,hash2
00078d03b4dda619293c1793c251f783,
etc...

Donde cada PURCHASED_COUPONS puede contener 0 o varios próximos cupones de compra posibles.

5.2 Descripción de las tablas:

Se entrega un año de datos de las transacciones de 22.873 usuarios en la web ponpare.jp. El conjunto de entrenamiento abarca las fechas desde 2011-07-01 hasta 2012-06-23. El conjunto de test abarca la semana posterior al final del conjunto entrenamiento. La meta de la competición es recomendar una lista ranqueada de cupones para cada usuario que se encuentra en el dataset (user_list.csv). Las predicciones se compararán con las compras reales realizadas durante el conjunto de la semana de entrenamiento, de los 310 cupones.

- **user_list.csv** (1,6 Mb): La lista de los perfiles del conjunto de datos.

Column Name	Description	Type	Length	Decimal	Note
USER_ID_hash	User ID	VARCHAR2	32		
REG_DATE	Registered date	DATE			Sign up date
SEX_ID	Gender	CHAR	1		f = female m = male
AGE	Age	NUMBER	4	0	
WITHDRAW_DATE	Unregistered date	DATE			
PREF_NAME	Residential Prefecture	VARCHAR2	2		[JPN] Not registered if empty

- **coupon_list_train.csv** (671 Kb): La lista de cupones que se consideran parte del conjunto entrenamiento.
- **coupon_list_test.csv** (59 Kb): La lista de cupones que se consideran parte del conjunto test.

Column Name	Description	Type	Length	Decimal	Note
CAPSULE_TEXT	Capsule text	VARCHAR2	20		[JPN]
GENRE_NAME	Category name	VARCHAR2	50		[JPN]
PRICE_RATE	Discount rate	NUMBER	4	0	
CATALOG_PRICE	List price	NUMBER	10	0	
DISCOUNT_PRICE	Discount price	NUMBER	10	0	
DISPFROM	Sales release date	DATE			
DISPEND	Sales end date	DATE			

DISPPERIOD	Sales period (day)	NUMBER	4	0	
VALIDFROM	The term of validity starts	DATE			
VALIDEND	The term of validity ends	DATE			
VALIDPERIOD	Validity period (day)	NUMBER	4	0	
USABLE_DATE_MON	Is available on Monday	CHAR	1		
USABLE_DATE_TUE	Is available on Tuesday	CHAR	1		
USABLE_DATE_WED	Is available on Wednesday	CHAR	1		
USABLE_DATE_THU	Is available on Thursday	CHAR	1		
USABLE_DATE_FRI	Is available on Friday	CHAR	1		
USABLE_DATE_SAT	Is available on Saturday	CHAR	1		
USABLE_DATE_SUN	Is available on Sunday	CHAR	1		
USABLE_DATE_HOLIDAY	Is available on holiday	CHAR	1		
USABLE_DATE_BEFORE_HOLIDAY	Is available on the day before holiday	CHAR	1		
large_area_name	Large area name of shop location	VARCHAR2	30		[JPN]
ken_name	Prefecture name of shop	VARCHAR2	8		[JPN]
small_area_name	Small area name of shop location	VARCHAR2	30		[JPN]
COUPON_ID_hash	Coupon ID	VARCHAR2	32		

- **coupon_visit_train.csv**: la vista de las visitas de los usuarios que entran en la web y buscan cupones en la página web durante el conjunto entrenamiento.

Column Name	Description	Type	Length	Decimal	Note
PURCHASE_FLG	Purchased flag	NUMBER	1	0	0:Not purchased 1:Purchased
PURCHASEID_hash	Purchase ID	VARCHAR2	128		
I_DATE	View date	DATE			Purchase date if purchased
PAGE_SERIAL		VARCHAR2			
REFERRER_hash	Referer	VARCHAR2	4000		
VIEW_COUPON_ID_hash	Browsing Coupon ID	VARCHAR2	128		
USER_ID_hash	User ID	VARCHAR2	10		
SESSION_ID_hash	Session ID	VARCHAR2	128		

- **coupon_detail_train.csv**: La sesión de compras de aquellos usuarios que compraron cupones durante el periodo del conjunto entrenamiento.

Column Name	Description	Type	Length	Decimal	Note
ITEM_COUNT	Purchased item count	NUMBER	10	0	
I_DATE	Purchase date	DATE			
SMALL_AREA_NAME	Small area name	VARCHAR2	30		[JPN] User redidential area name

PURCHASEID_hash	Purchase ID	VARCHAR2	32
USER_ID_hash	User ID	VARCHAR2	32
COUPON_ID_hash	Coupon ID	VARCHAR2	32

- **documentation.zip**: un archivo de ficheros Excel que contienen un diagrama de entidades de relación y traducciones inglés-japonés.

En el archivo de relación de entidades se muestra el esquema relacional de las claves entre tablas. Lo que significa que se trata de un conjunto SQL de almacenamiento de datos en el que como se puede observar que hay dos claves primarias fundamentales, el identificador de usuario y del cupón.

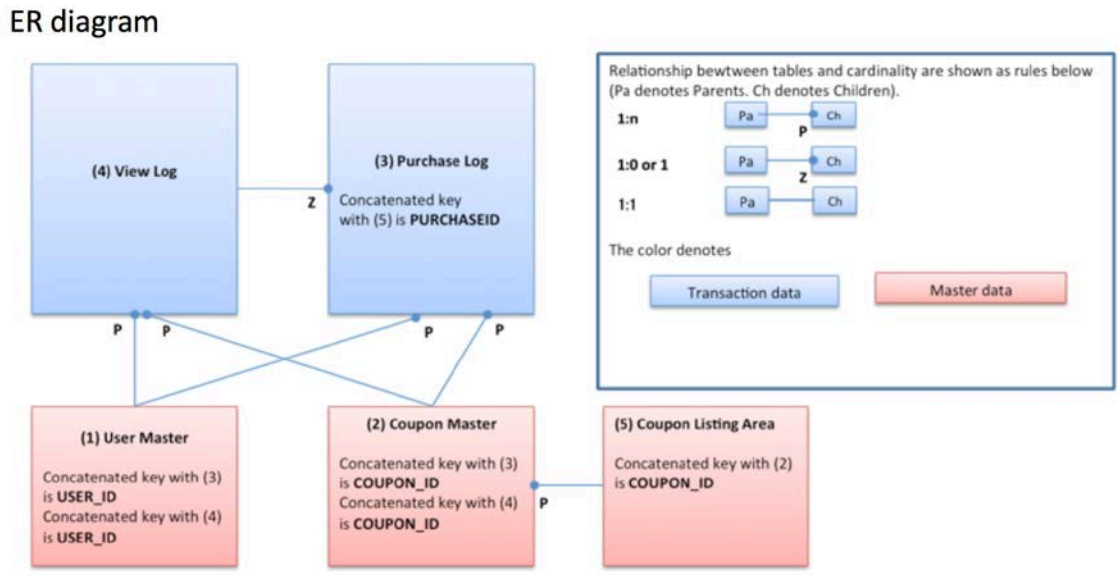

Figura 3: Esquema de BD

5.3 Entender el problema:

Para poder entender en profundidad cualquier problema la primera tarea posterior a la lectura inicial es documentarse. Por un lado conviene documentarse sobre las posibles soluciones de problemas similares y por otro, aprender al máximo el contexto.

5.3.1 Soluciones anteriores

En el artículo de Knott et al. (16) se aporta mucha información sobre cómo enfrentarse a este tipo de problemas. Como predictor más relevante se apunta que es el haber comprado con anterioridad el producto. Los modelos con los que mejor resultado se obtuvo fueron: regresión logística, regresión multinomial, análisis discriminante y redes neuronales. Sin embargo, comentan que ninguno era significativamente superior salvo el de redes neuronales, que implica un post análisis y comprensión del modelo más complejos.

Otros libros como "Retail Marketing and Branding: A Definitive Guide to Maximizing ROI" (17), foros especializados y el propio Kaggle pueden ser buenas fuentes de información.

5.3.2 Marco del problema

Para poder entender mejor qué buscar, primero habría que tratar de ver el problema desde la perspectiva de los propios usuarios sobre los que se modeliza. La perspectiva sociocultural de un analista no japonés puede sesgar el análisis. Las diferencias culturales pueden influir en la comprensión de los resultados y en la posterior toma de decisiones. Si un comprador medio de estos productos suele esperar mes y medio en volver a comprar el mismo producto, puede quedar oculto si el analista toma la decisión de que 3 semanas es el umbral máximo de influencia de una compra anterior, las correlaciones entre distintos productos pueden quedar ocultas. Existen infinitos matices que pueden restar calidad al análisis, ya que en definitiva, detrás de una máquina está un humano. El analista puede recurrir a técnicas de otras áreas como el método Stanislavski utilizado por los actores, entrar en la aplicación web, simular distintos tipos de compras o experimentarlas directamente. Entender las motivaciones sociales que influyen en las variables objetivo no sólo es enriquecedor para el análisis sino para el analista.

Por otro lado también quedaría investigar las características propias de la aplicación. Plantearse cuestiones tales como: ¿Cuántas campañas de autopromoción realizó en el último año y cuándo tuvieron lugar? ¿Ha variado su estrategia de mercado con el tiempo? ¿Son iguales los productos que vendía hace un año que los que vende al finalizar el periodo de entrenamiento?

5.3.3 Brainstorm

Es recomendable, antes de entrar a plantear el análisis, desligarse de cualquier idea preconcebida de comportamiento, incluso si esto choca con los anteriores párrafos. Todo conocimiento es útil, pero los resultados estarán sujetos a la realidad de los datos y puede o no que los resultados se ajusten a las ideas iniciales.

Sin embargo tampoco hay que dejar el ejercicio de documentación de lado. Para ello se puede recurrir a un brainstorm exhaustivo de todos los planteamientos previos que queramos realizar a los datos, tras haber procesado la información.

A modo de ejemplo, en las siguientes líneas se muestra un posible brainstorm, resultado de las breves búsquedas mostradas y con un reposo de información de un día:

- ¿Qué relaciones existen entre las distintas categorías?
- ¿Cómo son los clientes que se han ido incorporando en cada etapa de la empresa? ¿Existen agrupaciones?
- ¿Puede haber relación entre los clientes? Si una compra se realiza de forma simultánea por un grupo de usuarios, ¿Cuál será la dependencia grupal en la próxima compra? ¿Quién es el que toma las decisiones en el grupo?
- ¿Qué relación hay entre buscar y comprar? ¿Cómo de impulsivos son los clientes? ¿Clientes más impulsivos recurren a productos distintos de los menos impulsivos?
- ¿En qué fechas de la semana, mes o año compran más productos?
- ¿Cuánto varían las ofertas? ¿Dependen de la cantidad, la fecha u otras variables?
- ¿Cuáles son los productos estrella? ¿Cuáles los menos valorados? ¿Cuáles los más rentables?
- Si un cliente compra un producto, ¿es de esperar que exista estacionalidad diaria, semanal, mensual o anual?

- ¿Qué horas son las óptimas para recibir ofertas? ¿Qué horario laboral tienen los clientes? ¿Tienen trabajo? ¿Diferentes ofertas afectan más en diferentes horarios?
- ¿Cuál es el compromiso medio con la aplicación?
- ¿Influyen los anuncios de los programas televisivos más vistos, en el comportamiento de los clientes?
- ¿Cuántos perfiles de cliente hay?
- ¿Cuántos perfiles de comportamiento hay frente a cada tipo de compra?

5.4 Estadística descriptiva:

El primer análisis debe ser de la cantidad de información disponible.

Tamaño de las tablas usadas:

Nombre	Nº Filas	Nº de campos	NA's
User_list	22.873	6	
Coupon_visit_train	2.833.180	8	0
Coupon_detail_train	168.996	6	
Coupon_list_train	19.413	24	75.339
Coupon_list_test	310	24	

TABLA 1: TAMAÑO DE LAS TABLAS USADAS

Un cupón (producto objeto de estudio identificado por COUPON_ID_hash) consiste en una variable que tiene una fecha inicial disponible (DISPFROM) y una fecha final de compra (DISPEND) con una fecha de validez del cupón (VALIDFROM) y fecha de fin de la validez (VALIDEND), que comprende una cantidad de días (VALIDPERIOD). Además cada cupón tiene un descuento (PRICE_RATE), que implica un valor económico (DISCOUNT_PRICE) sobre el precio real del producto (CATALOG_PRICE), asociado a un campo de ventas (GENRE_NAME) de los 13 disponibles. El periodo comprende desde el 27 junio de 2011 hasta el 23 de junio del 2012.

Cada usuario (USER_ID_hash) puede o no comprar (PURCHASE_FLG), pero se registra su navegación asociada a la sesión en la que está (SESSION_ID_hash), junto con la fecha (I_DATE). El cupón visto (VIEW_COUPON_ID_hash) en caso de comprarse, queda registrado en la BD (PURCHASEID_hash). De los 22.873 usuarios, sólo 22.805 iniciaron una sesión en el último año. Lo que indica que puede ser bueno verificar la salud del número de usuarios de la web

De cada usuario (USER_ID_hash) se conoce la edad (AGE), el sexo (SEX_ID), la fecha en la que se registró (REG_DATE). Además para 938 usuarios se tiene su fecha de baja del sistema (WITHDRAW_DATE).

Como herramienta de estadística descriptiva se utiliza el software Orange, que permite obtener de forma automática por medio de distintos criterios de selección, las gráficas que mayor valor aportan. Por ejemplo en la Figura 4 hay un plot que se

obtiene de forma automática al seleccionar por medio de Partial Least Squares, las mejores proyecciones.

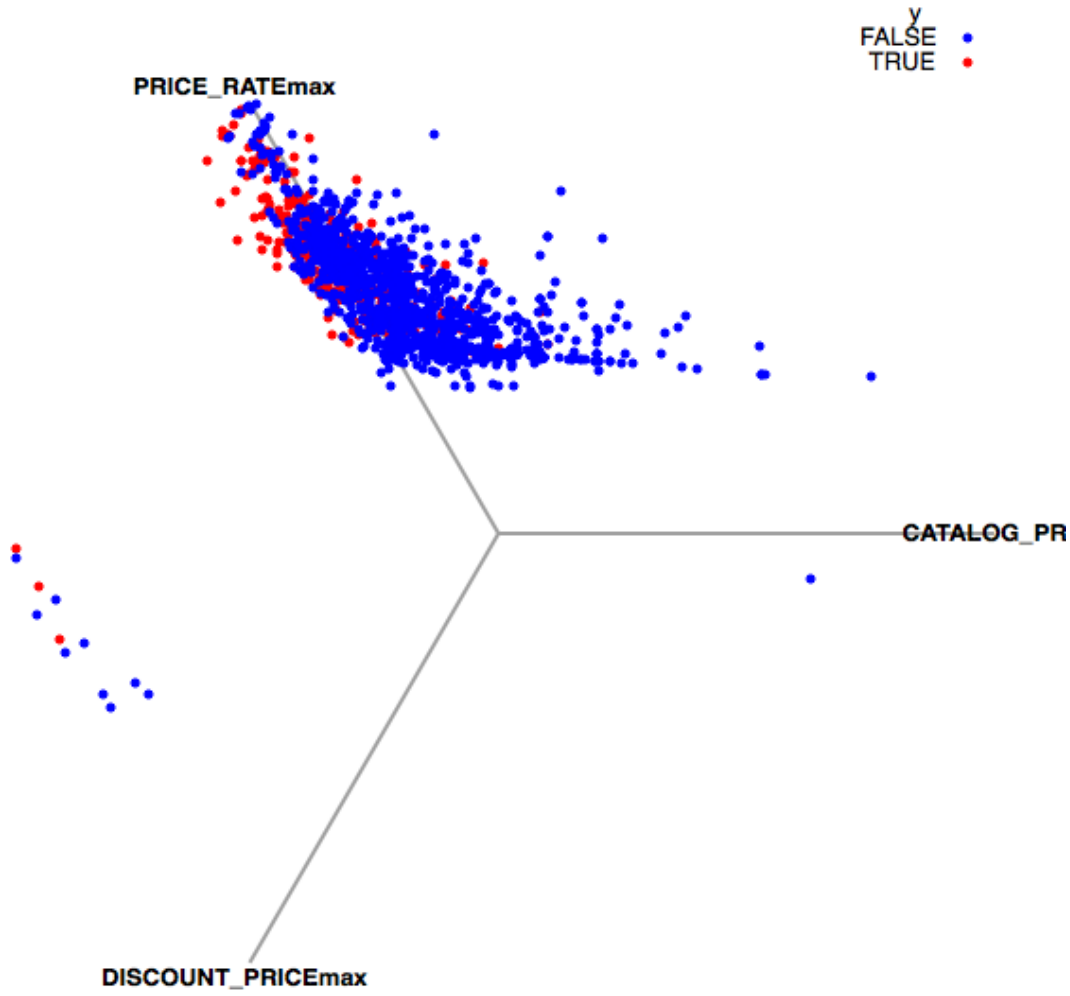

Figura 4: Proyección de las observaciones sobre 3 variables, desarrollada con Orange

Orange permite realizar gran variedad de gráficos, desde gráficos de mosaico, a gráficos de análisis de correspondencias. Por su simplicidad, es muy versátil, aunque consume mucha memoria en disco, lo que impide que en casos de BigData se pueda implementar su uso en el estado actual del software. Sin embargo es una gran herramienta de aprendizaje de las técnicas de Machine Learning.

5.4.1 Selección de una muestra

Para ejemplificar las funcionalidades se selecciona una muestra de la BD de forma que se seleccionan sólo la información referente a los cupones de comida. En esta muestra quedan 15.289 usuarios sobre los que estimar su propensión a consumir un cupón, semana a semana.

Las variables que se ven en la Figura 4 forman parte de un conjunto de variables generadas a partir de la base de datos, en la que se obtuvieron estadísticas del comportamiento de los usuarios en la búsqueda de cupones. Las variables extraídas son:

- Sexo: SEX_ID
- Edad: AGE
- Tasa de descuento mínima, media y máxima: PRICE_RATE
- Precio mínimo, medio y máximo sin cupón de la oferta: CATALOG_PRICE
- Descuento absoluto mínimo, medio y máximo: DISCOUNT_PRICE
- Periodo válido mínimo, medio y máximo del cupón: VALIDPERIOD
- Puede usarse antes de días festivos: USABLE_DATE_BEFORE_HOLIDAY
- Puede usarse en días festivos: USABLE_DATE_HOLIDAY
- Sólo tiene validez en días laborales: USABLE_DATE_WEEK

La variable objetivo se genera como una variable binaria en la que se considera 1 en el caso de que haya comprado al menos un cupón la semana anterior al periodo Test y 0 en caso contrario.

Por otro lado, también se extrae una variable que registra si el usuario ha consumido un cupón del mismo tipo en el periodo restante. Sin embargo esta variable no tiene variación. Es 1 para todos los usuarios.

5.5 Analítica

El desarrollo de esta sección se basa principalmente en los ejemplos descritos en un recomendable libro de cabecera de Max Kuhn y Kjell Johnson: "Applied Predictive Modeling"

5.5.1 Tratamiento de las variables

Una vez están obtenidos y organizados los datos, tras haber hecho el análisis descriptivo necesario, se comienza el tratamiento y preprocesado de las variables.

Para poder evaluar posteriormente los modelos se separa como primer paso un 20% de los datos, teniendo en cuenta la variable objetivo.

```
inTrain <- createDataPartition(X$y, p = .8)[[1]]
X <- X[ inTrain, ]
X_Test  <- X[-inTrain, ]
```

La transformación de BoxCox por ejemplo, es una de las múltiples utilidades que tiene la función preProcess. En la Figura 5 se ve una aplicación directa sobre la variable AGE.

```
xPP <- preProcess(X_Train, method = "BoxCox")
xPPTrans <- predict(xPP, X_Train)
```

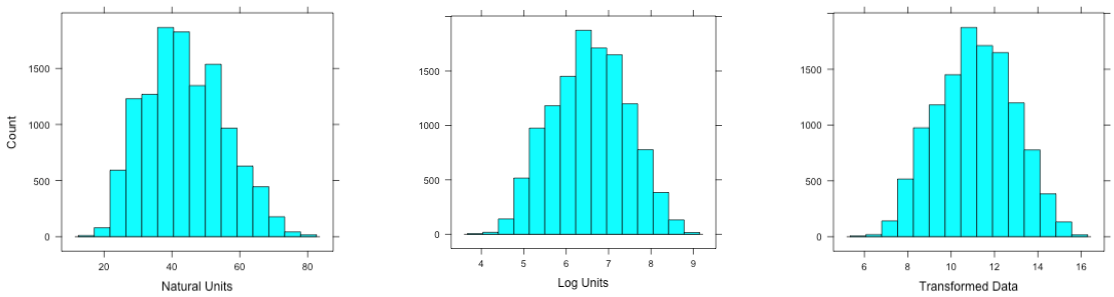

Figura 5: Transformación de la variable AGE, de izquierda a derecha, sin transformar, con la raíz cuadrada, transformación automática.

Muchos algoritmos no permiten su implementación sobre datos corruptos, infinitos o vacíos. Si escogiésemos hacer una imputación de variables, podrían rellenarse los 515 valores vacíos, tanto con la función Rimpute como con la función preProcess:

```
impX <- impute(X_Train, ImFun = "lassoR",cFun="glmboostR")
```

ó

```
method='medianImpute'

model_impute=preProcess(X_Train,method=method)
X_impute=predict(model_impute,X_Train)
```

Se puede hacer análisis de conmponentes principales sobre los datos con la función preProcess en conjunción con la función prcomp. En el siguiente gráfico (Figura 6) se puede ver el resultado, para el que se observa que la variable objetivo está concentrada en ciertas regiones, pero para la que sería complicado realizar una separación mediane una transformación lineal. Quizás un modelo de máquina de vector soporte sería capaz de discretizar y separar las dos categorías.

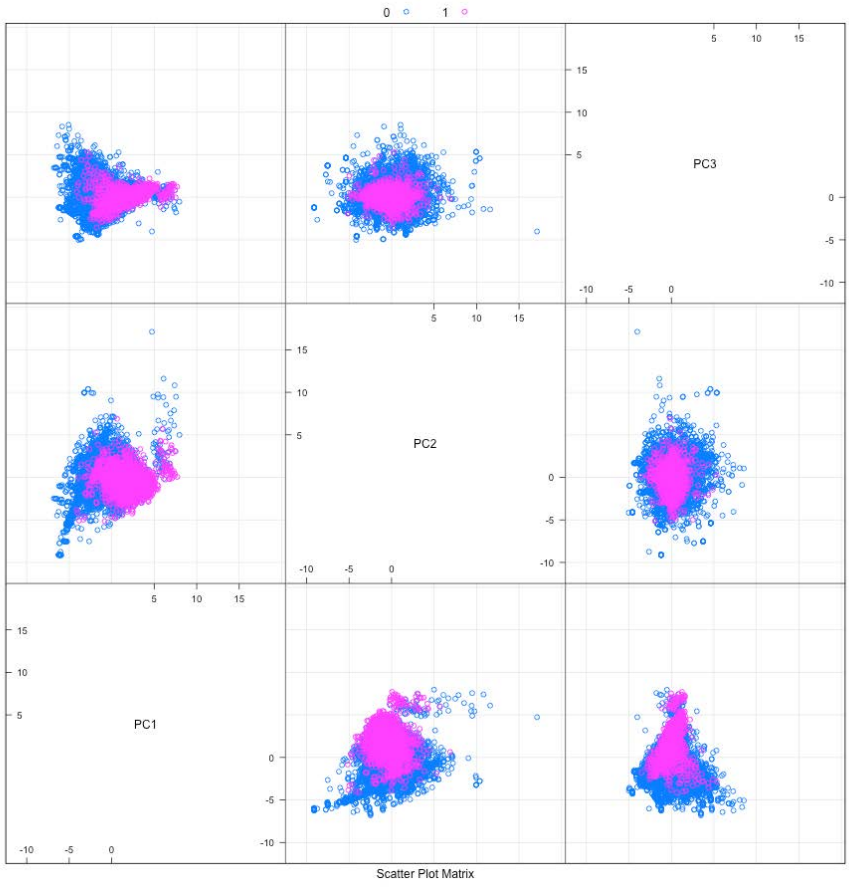

Figura 6: Proyección de la muestra en los 3 primeros componentes principales. Rosa='Sí' Azul='No'

Con la función corrplot se puede sacar un análisis visual rápido de la estructura de correlaciones. En la figura….. se puede apreciar que hay altas correlaciones entre las

variables CATALOG_PRIZE y DISCOUNT_PRIZE. También se observa correlación inversa alta entre el porcentaje de descuento y el precio de catálogo y el periodo de validez. Es decir, que cuanto mayor es el precio o el periodo de validez de un cupón, menor es el descuento asociado. A la vista de la Figura 7 se puede considerar eliminar las variables DISCOUNT_PRICE.

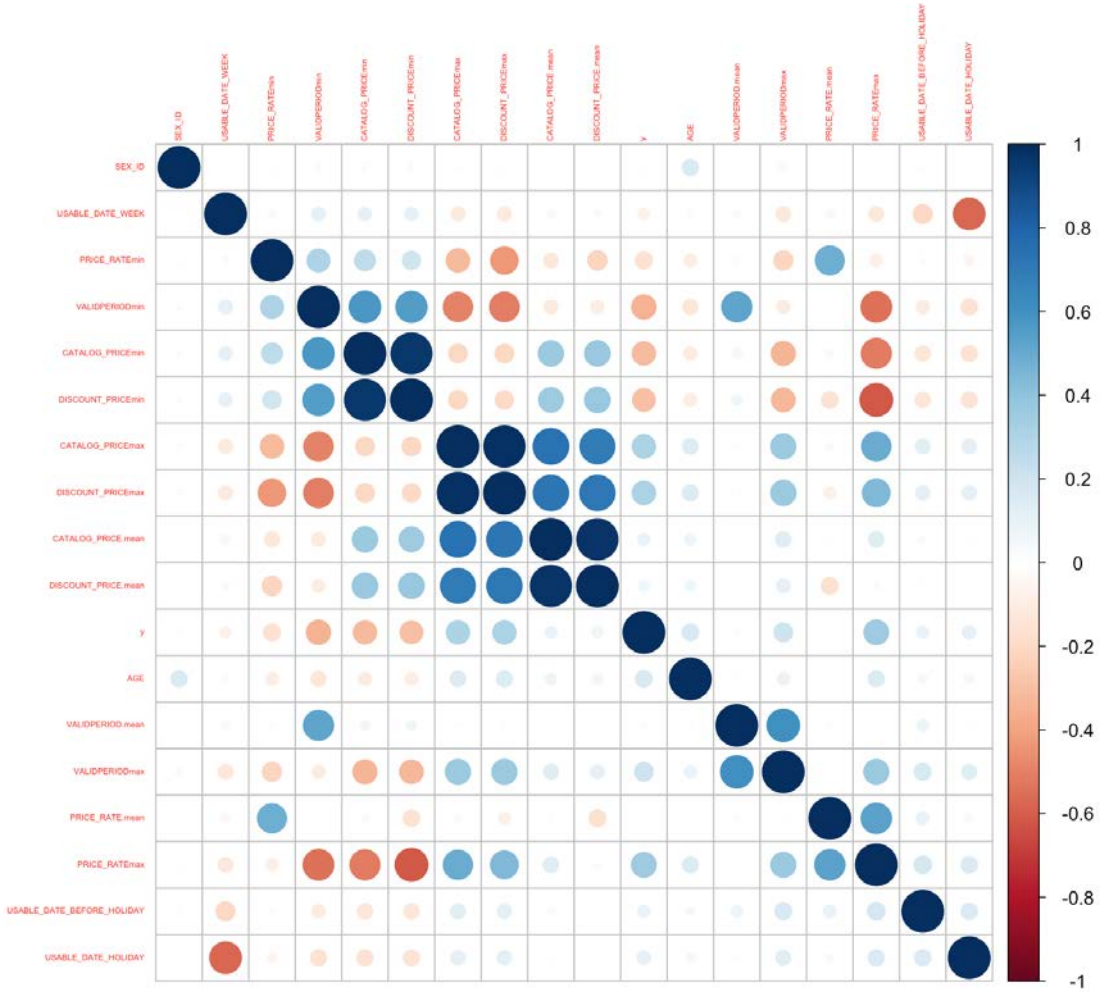

Figura 7: Plot de correlaciones

Tras eliminar las variables que podían dar problemas de multicolinealidad se aplican distinos modelos a los datos para evaluar su ajuste. Se evalúan varios modelos de clasificación, con un control del modelo de validación cruzada repetida. En la Figura 8 se ven los resultados obtenidos mediante la función resample, que selecciona muestras y evalúa sobre ellas, los modelos, repetidas veces.

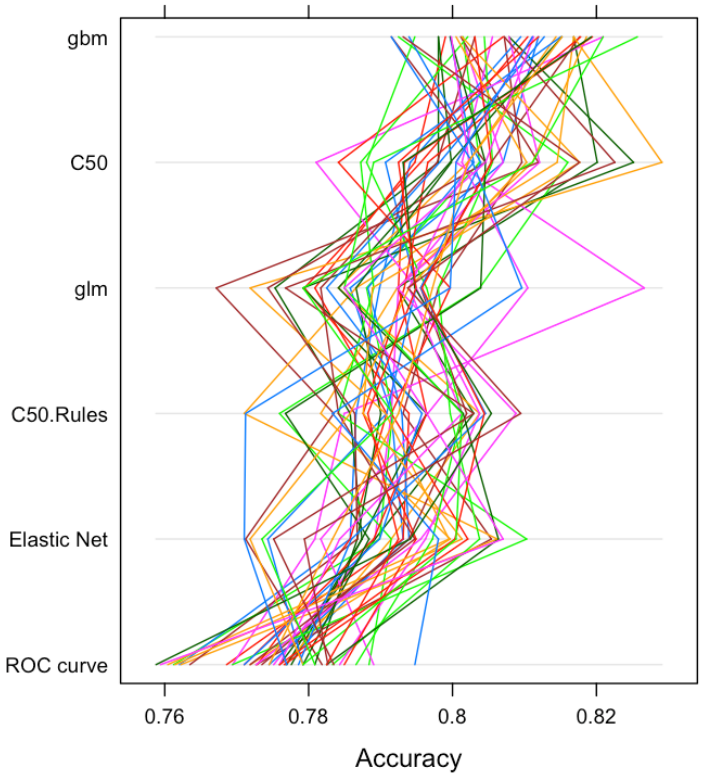

Figura 8: ParallelPlot de evaluación de los modelos: Gradient Boosting Machine, C5.0, Generalized Linear Models, C5.0 con reglas, LASSO con ElasticNet, ROC curve

6. Conclusiones

La utilidad de las herramientas descritas queda reflejada en la propia simplicidad del presente trabajo. En menos de 300 líneas de código se han generado consultas, muestras, análisis plots, etc. Es importante tener en cuenta que el desarrollo descrito y su aplicación pueden no ser productivos en alumnos recién egresados, ya que muchas de las funciones, son atajos, que deben ser inicialmente dominados por el alumno, antes de poder considerar tomar estos atajos.

El desarrollo de la habilidad práctica en el campo del análisis, está al mismo nivel que la capacidad teórica del analista. La ventaja real del BigData, se encuentra en la velocidad de procesamiento, el resultado instantáneo. El valor añadido del analista está en su maletín de herramientas. Ninguna empresa utiliza un solo software, ningún desarrollador ni arquitecto de bases de datos es experto en un solo formato. La versatilidad y la capacidad de adaptación, son recursos necesarios, que deben ser practicados.

Habiendo realizado este breve trabajo, con una intención más didáctica, que de investigación, sólo queda agradecer la paciencia del lector avanzado, en las generalizaciones en descripciones de casuísticas delicadas, como la diferencia entre las bases de datos SQL y non-SQL, o la explicación somera de algunos modelos de aprendizaje automático, en las que cada una de ellas podría basarse un solo TFG. Por otro lado, queda agradecer al lector curioso que lo narrado entre líneas, haya capturado su atención y pueda provocar un inconsciente deseo de indagar en los diferentes temas descritos.

Bibliografía

1. **Press, Gil.** A Very Short History Of Big Data. *www.forbes.com.* [En línea] 9 de May de 2013. http://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/.

2. **optional, Big data for big business – analytics are no longer.** the globe and mail. *http://www.theglobeandmail.com/.* [En línea] 15 de Aug de 2015. http://www.theglobeandmail.com/report-on-business/rob-commentary/big-data-for-big-business-analytics-are-no-longer-optional/article25975741/.

3. **Barometer, 2011 Global Customer Service.** american express. *about.americanexpress.com.* [En línea] 2011. http://about.americanexpress.com/news/docs/2011x/AXP_2011_csbar_market.pdf.

4. **La tarjeta de compra de El Corte Inglés, en la cartera de 10,5 millones de ciudadanos.** abc. *www.abc.es.* [En línea] 8 de Sept de 2014. http://www.abc.es/economia/20140210/abci-tarjeta-compra-corte-ingles-201402060115.html.

5. *MapReduce and Parallel DBMSs: Friends or Foes?* **Stonebraker, Michael, y otros, y otros.** 1, 2010, Communications of the acm, Vol. 53, págs. 84-91. doi:10.1145/1629175.1629197.

6. **Dowle, Matt.** ucla. *http://user2014.stat.ucla.edu/.* [En línea] 01 de Julio de 2014. http://user2014.stat.ucla.edu/files/talk_Matt.pdf.

7. **Dowle, Matt.** github. *github.com.* [En línea] 2014. https://github.com/Rdatatable/data.table/wiki/Benchmarks-%3A-Grouping.

8. **r-forge.r-project.** datatable.r-forge.r-project. *www.datatable.r-forge.r-project.org.* [En línea] 2 de Octubre de 2014. http://datatable.r-forge.r-project.org/datatable-faq.pdf.

9. **Kuhn, Max y Mayer, Zachary.** inside-r. *www.inside-r.org.* [En línea] 12 de Jul de 2015. http://www.inside-r.org/node/86978.

10. **Feng, Lingbing, y otros, y otros.** cran.r-project. *https://cran.r-project.org.* [En línea] 14 de Mayo de 2013. https://cran.r-project.org/web/packages/imputeR/imputeR.pdf.

11. **Regression, Cubist Models For.** cran.r-project.org. *https://cran.r-project.org.* [En línea] 11 de Mayo de 2012. https://cran.r-project.org/web/packages/Cubist/vignettes/cubist.pdf.

12. **package, Generalized Boosted Models: Aguide to the gbm.** github. *https://github.com.* [En línea] 23 de Mayo de 2012. https://github.com/harrysouthworth/gbm/blob/master/inst/doc/gbm.pdf.

13. courses.cs.washington. *https://courses.cs.washington.edu.* [En línea] 2013. https://courses.cs.washington.edu/courses/cse599c1/13wi/slides/LARS-fusedlasso.pdf.

14. **Ooms, Jeroen, y otros, y otros.** cran.r-project.org. *https://cran.r-project.org.* [En línea] 25 de Agosto de 2015. https://cran.r-project.org/web/packages/RMySQL/RMySQL.pdf.

15. **Kaggle Inc 2015.** kaggle. *www.kaggle.com.* [En línea] 02 de Junio de 2015. https://www.kaggle.com/c/coupon-purchase-prediction.

16. *Next-product-to-buy models for cross-selling applications.* **Knott, A., Hayes, A. y Neslin.** 3, 2002, Journal of Interactive Marketing, Vol. 16, págs. 59-75.

17. **Jesko, Perrey y Dennis, Spillecke.** *Retail Marketing and Branding: A Definitive Guide to Maximizing ROI.* s.l. : John Wiley & Sons, 2013. pág. 400. Vol. 2.

Lista de Figuras y Tablas

Anexos

```
# Remove objects ####
{
 rm(list = ls())
 gc()
}

# Load necessary packages ####
{
 library(fasttime)
 library(data.table)
 library(caret)
 library(imputeR)
 library(lubridate)
 library(psych)
 library(corrplot)
}

# Functions and configuration ####
{
 source(file.path(Sys.getenv("HOME"),
"Dropbox/estadistica/TFG/Rscript/conf_TFG.R"))
}

users <- fread(input = file.path(conf.path.csv,"user_list.csv"),nrows = 2)
nrow(fread(input = file.path(conf.path.csv,"user_list.csv"),select = "REG_DATE"))

coupon_visit <- fread(input =
file.path(conf.path.csv,"coupon_visit_train.csv"),nrows = 2)
nrow(fread(input = file.path(conf.path.csv,"coupon_visit_train.csv"),select =
"PAGE_SERIAL"))

coupon_detail_train <- fread(input =
file.path(conf.path.csv,"coupon_detail_train.csv"),nrows = 2)
nrow(fread(input = file.path(conf.path.csv,"coupon_detail_train.csv"),select =
"ITEM_COUNT"))

coupon_list_train <- fread(input =
file.path(conf.path.csv,"coupon_list_train.csv"),nrows = 2)
nrow(fread(input = file.path(conf.path.csv,"coupon_list_train.csv"),select =
"USABLE_DATE_MON"))

coupon_list_test <- fread(input =
file.path(conf.path.csv,"coupon_list_test.csv"),nrows = 2)
nrow(fread(input = file.path(conf.path.csv,"coupon_list_test.csv"),select =
"USABLE_DATE_MON"))

coupon_list_test <- fread(input =
file.path(conf.path.csv,"coupon_list_test.csv"),nrows = 2)
nrow(fread(input = file.path(conf.path.csv,"coupon_list_test.csv"),select =
"USABLE_DATE_MON"))

nrow(fread(input = file.path(conf.path.csv,"coupon_list_train.csv"),select =
"USABLE_DATE_MON"))

### Coupons definition
```

```
#characteristics of failed coupons:
clt_b <- fread(input = file.path(conf.path.csv,"coupon_list_train.csv"),drop =
c("CAPSULE_TEXT","DISPEND",

"VALIDEND","large_area_name",

"ken_name","small_area_name") )
sum(is.na(clt_b))
sapply(clt_b,function(x)sum(is.na(x)))
clt_b[,`:=`(DISPFROM = fastPOSIXct(DISPFROM,required.components = 4),
        VALIDFROM = fastPOSIXct(DISPFROM,required.components = 3))]
setkey(clt_b,GENRE_NAME ,DISPFROM)

clt_b[is.na(VALIDFROM),head(.SD,30),by=GENRE_NAME]
clt_b[is.na(VALIDFROM), ]

clt_b[,length(unique(GENRE_NAME))]


### Visits definition
cvt_b <- fread(input = file.path(conf.path.csv,"coupon_visit_train.csv"),drop =
"REFERRER_hash")
sum(is.na(cvt_b[,!"PURCHASEID_hash",with=FALSE]))

cvt_b[,`:=`(I_DATE = fastPOSIXct(I_DATE,required.components = 5),
        month      =      format(fastPOSIXct(I_DATE,required.components      =
2),"%Y_M%m") )]
setkey(cvt_b,I_DATE,VIEW_COUPON_ID_hash)
users_month <- cvt_b[,.(Usuarios = length(unique(USER_ID_hash)),
                  Busquedas = nrow(.SD),
                  Compras = length(unique(PURCHASEID_hash))),by=month]
barplot(height = users_month$Usuarios,names.arg = users_month$month,ylim =
c(8000,12000), xpd = FALSE,
      main = "Numero de Usuarios por Mes", ylab = "N_Users")

barplot(height = users_month$Busquedas,names.arg = users_month$month,ylim
= c(150000,300000), xpd = FALSE,
      main = "Numero de Busquedas por Mes", ylab = "Busquedas")

barplot(height = users_month$Compras,names.arg = users_month$month,ylim =
c(8000,12000), xpd = FALSE,
      main = "Numero de Compras por Mes", ylab = "N_Users")


rm(cvt_b)
###

### Visits definition
cvd_t <- fread(input = file.path(conf.path.csv,"coupon_detail_train.csv"),drop =
"SMALL_AREA_NAME")
sum(is.na(cvd_t))


rm(cvd_t)
###

### users definition
```

```
ul_b <- fread(input = file.path(conf.path.csv,"user_list.csv"),drop = "PREF_NAME")
sum(is.na(ul_b[,!"WITHDRAW_DATE",with=FALSE]))
ul_b[!is.na(WITHDRAW_DATE),length(unique(USER_ID_hash))]

ul_b[,`:=`(WITHDRAW_DATE =
fastPOSIXct(WITHDRAW_DATE,required.components = 3L),
        puta_madre = fastPOSIXct(REG_DATE,required.components = 3),
         month = format(fastPOSIXct(REG_DATE,required.components =
2),"%Y_M%m"),
        month_draw = format(fastPOSIXct(WITHDRAW_DATE,required.components
= 2),"%Y_M%m"))]
setkey(ul_b,REG_DATE)
ul_b$distancia <- 0
ul_b[!is.na(WITHDRAW_DATE),`:=`(distancia =
as.numeric(difftime(WITHDRAW_DATE, REG_DATE, units = "days"))
),by=USER_ID_hash]
ul_b[!is.na(WITHDRAW_DATE) & month >
"2011_M07",describe(as.numeric(distancia) )
,by=month][,.(month,n,median,mean)]

rm(ul_b)
###

### Visits information
cvt <- fread(input = file.path(conf.path.csv,"coupon_visit_train.csv"),drop =
"REFERRER_hash")
cvt[,`:=`(I_DATE = fastPOSIXct(I_DATE,required.components = 5),
       month = format(fastPOSIXct(I_DATE,required.components =
2),"%Y_M%m") )]
clt <- fread(input = file.path(conf.path.csv,"coupon_list_train.csv"),drop =
c("CAPSULE_TEXT","DISPEND",

"VALIDEND","large_area_name",

"ken_name","small_area_name") )
# subset food, hotel and hair salon
clt[, clases :=  as.factor(GENRE_NAME) ]
clt[,clases := as.numeric(clases)]
clt[,.(unique(clases),unique(GENRE_NAME))]
clt <- clt[clases %in% c(3,7,8),];table(clt$clases)
clt[,`:=`(DISPFROM = fastPOSIXct(DISPFROM,required.components = 4),
         VALIDFROM = fastPOSIXct(DISPFROM,required.components = 3))]

#subset from cvt those coupons and join the new info
cdt <- fread(input = file.path(conf.path.csv,"coupon_detail_train.csv"),drop =
"SMALL_AREA_NAME")

setkey(cdt,COUPON_ID_hash)
setkey(clt,COUPON_ID_hash)
DT <- cdt[clt,nomatch=0]
DT[USABLE_DATE_MON == 2,USABLE_DATE_MON:=1]
DT[USABLE_DATE_TUE == 2,USABLE_DATE_TUE:=1]
DT[USABLE_DATE_WED == 2,USABLE_DATE_WED:=1]
DT[USABLE_DATE_THU == 2,USABLE_DATE_THU:=1]
DT[USABLE_DATE_FRI == 2,USABLE_DATE_FRI:=1]
DT[USABLE_DATE_SAT == 2,USABLE_DATE_SAT:=1]
DT[USABLE_DATE_SUN == 2,USABLE_DATE_SUN:=1]
```

```
DT[USABLE_DATE_BEFORE_HOLIDAY == 2,USABLE_DATE_BEFORE_HOLIDAY:=1]
DT[USABLE_DATE_HOLIDAY == 2,USABLE_DATE_HOLIDAY:=1]


DT[,USABLE_DATE_WEEK:=as.numeric((USABLE_DATE_SAT+USABLE_DATE_SUN)
==0)]

setkey(cvt,VIEW_COUPON_ID_hash)
setkey(DT,COUPON_ID_hash)

DT[clases                                                    %in%
3,table(paste0(USABLE_DATE_MON,USABLE_DATE_TUE,USABLE_DATE_WED,USAB
LE_DATE_THU,USABLE_DATE_FRI,USABLE_DATE_SAT,USABLE_DATE_SUN)) ]
DT_3_SUB          <-          DT[clases          %in%          3,
.(COUPON_ID_hash,DISPPERIOD,VALIDPERIOD,USABLE_DATE_HOLIDAY,USABLE_
DATE_BEFORE_HOLIDAY,

PRICE_RATE,CATALOG_PRICE,DISCOUNT_PRICE,USABLE_DATE_WEEK)]
DT_3_SUB <- DT_3_SUB[!duplicated(DT_3_SUB),]

DT_3 <- cvt[DT_3_SUB,]

X <- DT_3[,.(USER_ID_hash = unique(USER_ID_hash))]
# Genera la variable objetivo para elmodelo de entrenamiento
X$y <- 0
datos<-      DT_3[I_DATE      >      ("2012-06-17      00:00:00"),.(y      =
as.numeric(sum(!is.na(PURCHASEID_hash))>0)),by=USER_ID_hash]
X[USER_ID_hash %in% datos$USER_ID_hash,y:=1]
setkey(X,USER_ID_hash)

# get variables for the model

setkey(DT_3,USER_ID_hash)
datos          <-          DT_3[I_DATE          <=          ("2012-06-17
00:00:00"),lapply(.SD,mean,na.rm=T),by=USER_ID_hash,

.SDcols=c("PRICE_RATE","CATALOG_PRICE","DISCOUNT_PRICE","VALIDPERIOD")]
setnames(datos,c("USER_ID_hash",paste0(names(datos),".mean")[-1]))
setkey(datos,USER_ID_hash)

X <-X[datos,]

datos          <-          DT_3[I_DATE          <=          ("2012-06-17
00:00:00"),lapply(.SD,min,na.rm=T),by=USER_ID_hash,

.SDcols=c("PRICE_RATE","CATALOG_PRICE","DISCOUNT_PRICE","VALIDPERIOD")]
setnames(datos,c("USER_ID_hash",paste0(names(datos),"min")[-1]))
setkey(datos,USER_ID_hash)

X <-X[datos,]

datos          <-          DT_3[I_DATE          <=          ("2012-06-17
00:00:00"),lapply(.SD,max,na.rm=T),by=USER_ID_hash,

.SDcols=c("PRICE_RATE","CATALOG_PRICE","DISCOUNT_PRICE","VALIDPERIOD")]
setnames(datos,c("USER_ID_hash",paste0(names(datos),"max")[-1]))
setkey(datos,USER_ID_hash)
```

```
X <-X[datos,]

datos <- DT_3[I_DATE <= ("2012-06-17 00:00:00"),lapply(.SD,FUN=function(x){median(as.numeric(x),na.rm=T)}),by=USER_ID_hash,

.SDcols=c("USABLE_DATE_BEFORE_HOLIDAY","USABLE_DATE_HOLIDAY","USABLE_DATE_WEEK")]
setkey(datos,USER_ID_hash)

X <-X[datos,]

#datos <- DT_3[I_DATE <= ("2012-06-17 00:00:00"),.(y_pasado = as.numeric(sum(!is.na(PURCHASEID_hash))>0)),by=USER_ID_hash]
setkey(datos,USER_ID_hash)

#X <-X[datos,]

user <- fread(input = file.path(conf.path.csv,"user_list.csv"),drop = "PREF_NAME")

user <- user[is.na(WITHDRAW_DATE),]
setkey(user,USER_ID_hash)
user[,REG_DATE:=NULL]
user[,WITHDRAW_DATE:=NULL]
X <- user[X,nomatch=NA]

X[,lapply(.SD,function(x)sum(is.na(x)))]

X[,`:=`(SEX_ID=as.numeric(as.factor(SEX_ID)),
      AGE = as.numeric(AGE))]
X_USER <- X$USER_ID_hash
X[,USER_ID_hash:=NULL]
setcolorder(X,conf.col.order)
rm(cdt,clt,cvt,datos,DT,DT_3,DT_3_SUB)
write.table(X,file=paste0(conf.path.csv,"/datos.csv"),sep="\t",row.names = FALSE)

## extrae un conjunto test para evaluar los modelos
set.seed(123)
inTrain <- createDataPartition(X$y, p = .8)[[1]]
X_Train <- X[ inTrain, ]
X_Test  <- X[-inTrain, ]



xPP <- preProcess(X_Train, method = "BoxCox")

xPPTrans <- predict(xPP, X_Train)

## Resultados para AGE
xPP$bc$AGE
par(mfrow=c(1,3))
histogram(~X_Train$AGE,
        xlab = "Natural Units",
        type = "count")

histogram(~sqrt(X_Train$AGE),
```

```
        xlab = "Log Units",
        ylab = " ",
        type = "count")


histogram(~xPPTrans$AGE,
        xlab = "Transformed Data",
        ylab = " ",
        type = "count")


# impX <- impute(X_Train, lmFun = "lassoR",cFun="glmboostR")
method='medianImpute'
model_impute=preProcess(X_Train,method=method)
X_impute=predict(model_impute,X_Train)


# PCA
xPP <- preProcess(X_impute, c("BoxCox", "center", "scale"))
xPPTrans <- predict(xPP, X_impute)

xPCA <- prcomp(xPPTrans, center = TRUE, scale. = TRUE)

panelRange <- extendrange(xPCA$x[, 1:3])
splom(as.data.frame(xPCA$x[, 1:3]),
     groups = X_impute$y,
     type = c("p", "g"),
     as.table = TRUE,
     auto.key = list(columns = 2),
     prepanel.limits = function(x) panelRange)

xCorr <- cor(xPPTrans)

corrplot(xCorr, order = "hclust", tl.cex = .35)

## esta funcion identifica los regresores con altas correlaciones
highCorr <- findCorrelation(xCorr, .75)

xPPTrans$DISCOUNT_PRICEmax <- NULL
xPPTrans$DISCOUNT_PRICEmin <- NULL
xPPTrans$DISCOUNT_PRICE.mean <- NULL

ctrl <- trainControl(method = "repeatedcv",
                repeats = 5)
xPPTrans$y <- as.factor(xPPTrans$y)

xrocc <- train(y~. , data=xPPTrans,
               method = "rocc",
               trControl = ctrl,
               metric ="AUC")

xgbm <- train(y~. , data=xPPTrans,
           method = "gbm",
           trControl = ctrl)

xglm <- train(y~. , data=xPPTrans,
          method = "glm",
```

```
          trControl = ctrl)

xglmnet <- train(y~. , data=xPPTrans,
          method = "glmnet",
          trControl = ctrl)

xAdaBag <- train(y~. , data=xPPTrans,
            method = "AdaBag",
            trControl = ctrl)

xC50 <- train(y~. , data=xPPTrans,
            method = "C5.0",
            trControl = ctrl)

xC50rules <- train(y~. , data=xPPTrans,
          method = "C5.0Rules",
          trControl = ctrl)

rs <- resamples(list("ROC curve" = xrocc,
              "gbm" = xgbm,
               "Elastic Net" = xglmnet,
               "glm" = xglm,
              "C50" = xC50,
              "C50.Rules"=xC50rules))

parallelplot(rs)
```