Máster en Ingeniería Computacional y Matemática

Trabajo Final de Máster

Desarrollo de un recomendador de productos basado en Extreme Gradient Boosting

Estudiante: Pablo López Serrano
Máster Ingeniería Computacional y Matemática
Inteligencia Artificial

Nombre Consultor/a: Samir Kanaan Izquierdo
Nombre Profesor/a responsable de la asignatura: Carles Ventura Royo

31 de Mayo 2017

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Desarrollo de un recomendador de productos basado en Extreme Gradient Boosting*
Nombre del autor:	*Pablo López Serrano*
Nombre del consultor/a:	*Samir Kanaan Izquierdo*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega (mm/aaaa):	05/2017
Titulación::	*Master de Ingeniería Computacional y Matemática*
Área del Trabajo Final:	*Inteligencia Artificial*
Idioma del trabajo:	*Castellano*
Palabras clave	*Recomendador, Predicción, Extreme Gradient Boosting*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El trabajo consiste en el desarrollo de un recomendador de productos aplicando el algoritmo Extreme Gradient Boositng. Se resuelve el problema de la plataforma Kaggle “Santander Product Recomendation” que trata de predecir los siete productos del banco con más probabilidad de ser contratados por los clientes en el futuro.

La finalidad del trabajo consiste en resolver un problema de predicción real con un gran volumen de datos y aplicar una de las técnicas predictivas de más éxitos en las competiciones de Machine Learning. Además se pretende comprar los resultados alcanzados con los obtenidos por los mejores usuarios de la plataforma Kaggle.

En total se realizan 15 experimentos distintos con una fase de Ingeniería de Características muy exhaustiva, donde cada uno de los cambios aplicados se aprueban o se descartan en función de las puntuaciones obtenidas en la plataforma.

Los resultados finales hubieran alcanzado la posición 49 de la clasificación final de la competición.

Abstract (in English, 250 words or less):

In this final project has been developed a product classifier algorithm using Extreme Gradient Boosting technique. “Santander Product Recommendation” is the problem to be solved selected from Kaggle platform. In this problem the competitor has to predict the seven products with higher probability to be purchased in the future.

The purpose of this project is to resolve a real predictive problem with higher data and apply one of the most success predictive method in the competitions of Machine Learning. Further, at the end of the resolution, the author will compare the results with the best users of the competition.

Fifteen different experiments are carried out with a very exhaustive Feature Engineering phase, where each of the applied changes are approved or discarded according to the scores obtained in the platform.

The final position in the competition had been 49th.

Índice

Índice de tablas

Índice de figuras

1 Introducción del TFM

1.1 Contexto y justificación del Trabajo

En los últimos años, la evolución de las nuevas tecnologías y la industria del Big Data están teniendo un gran impacto en la sociedad. Tanto es así que se han generado nuevas profesiones que requieren de profesionales con competencias en diversos campos como las matemáticas, la estadística, la programación, el análisis de datos, la economía, etc.

Una de las profesiones más atractivas de este siglo XXI es el Científico de Datos o Data Scientist, que consiste en un perfil profesional que traduce los grandes volúmenes de datos, que provienen de todo tipo de fuentes de información masivas y las convierten en respuestas.

El objetivo de un científico de datos es obtener respuestas fiables a cualquier tipo de problemas, como por ejemplo: saber cuál es el mejor momento para comprar un determinado producto, predecir los gustos de ciertos consumidores y recomendar las mejores opciones, o descubrir si una persona está en riesgo de padecer una enfermedad.

Dentro del contexto de Big Data, a lo largo de los últimos años, han surgido en internet, comunidades, foros, conjuntos de usuarios y plataformas con el fin de explotar y poner en común los conocimientos de sus integrantes.

Una de las plataformas más famosas dentro del mundo del Data Science es Kaggle que nació como una idea de convertir el análisis de datos estadísticos en un juego. Los creadores formaron un punto de encuentro de analistas y científicos de datos para resolver problemas reales en sus momentos de ocio.
Esta idea tan simple, ha llegado a juntar a cerca de 200.000 usuarios y se ha convertido en la comunidad de ciencia de datos más grande del mundo.

Lo que empezó como una simple competición por entrenamiento se ha convertido en un laboratorio que se ofrece a empresas e instituciones para resolver problemas que tengan con los datos.
Las empresas pueden evaluar múltiples soluciones y elegir un ganador en cada competición. Este mérito, proporciona al usuario un cierto prestigio y además una recompensa económica.

En el actual trabajo, se ha escogido una de las competiciones propuestas en la plataforma kaggle. El título de la competición es “Santander Product Recomendation” y consiste en predecir los productos que los clientes del Banco Santander van a contratar en el futuro.

El problema escogido es de gran interés para el autor del presente trabajo porque tiene dos características muy contrastadas. Por una parte tiene la particularidad de resolver un problema clásico en el mundo comercial:

¿Qué productos son los que se van a vender más en el futuro? Tener la respuesta a esta pregunta es, prácticamente, sinónimo de éxito.
Y por otro lado, el problema pertenece al mundo de la Banca que es uno de los sectores principales del análisis de datos.

La aportación del presente autor para afrontar el problema "Santander Product Recomendation" será aplicar algoritmos predictivos basados en árboles de decisión realizando una exhaustiva fase de Ingeniería de Caracterísitcas (Feature Engineering).

Una vez se hayan obtenido los resultados se compararán la predicciones con los resultados finales de la competición aportados por los usuarios de Kaggle.

1.2 Objetivos del Trabajo

Los objetivos del trabajo se dividen en objetivos generales y objetivos del problema:

1.2.1 Objetivos Generales

- Resolver un problema real de predicción y recomendación
- Trabajar con un gran volumen de datos (Big Data).
- Estudio de las diferentes técnicas predictivas más utilizadas.
- Aplicar algoritmos predictivos (Extreme Gradient Boosting)
- Ingeniería de características (Feature Engineering). Escoger las mejores características para diseñar los modelos predictivos.
- Ensamblaje de los diferentes modelos predictivos.
- Aplicar los conocimientos adquiridos en el Máster de Ingeniería Computacional y Matemática.

1.2.2 Objetivos del Problema

- Definición y aplicación de métricas.
- Obtención de un recomendador fiable y de precisión
- Comparativa de los resultados con otros usuarios de la plataforma Kaggle.
- Obtener la máxima puntuación en la plataforma Kaggle durante el periodo de tiempo del proyecto.

1.3 Enfoque y método seguido

Los problemas de predicción no tienen una estrategia única de resolución. Es decir, no hay una metodología ni un procedimiento recomendado para poder solucionar el problema.
Esto es así, ya que, todos los problemas de predicción, dependen de los datos. En función de éstos, los problemas se tienen que encarar de una manera u otra.

Lo más importante y decisivo en este tipo de problemas es escoger las características que se emplearán en los modelos. Escoger las

características adecuadas para la fase de entrenamiento será la tarea más ardua y compleja del trabajo.

1.4 Planificación del Trabajo

El periodo de trabajo para la ejecución del proyecto es de Diciembre de 2016 a Mayo de 2017 y se planifica de la siguiente manera:

- Tarea 1: Revisión de la literatura para encontrar artículos relacionados con la temática del trabajo.
- Tarea 2: Revisión de publicaciones realizadas en la plataforma Kaggle relacionadas con el Banco Santander.
- Tarea 3: Aprendizaje de técnicas de predicción y recomendación.
- Tarea 4: Definición completa del problema.
- Tarea 5: Obtención y descripción de los datos del problema.
- Tarea 6: Minería de datos.
- Tarea 7: Limpieza de datos.
- Tarea 8: Realización de los modelos predictivos.
- Tarea 9: Comprobación de resultados y ensamblaje de modelos.
- Tarea 10: Creación de modelo final y testeo.
- Tarea 11: Comparación y entrega de los resultados en la plataforma Kaggle.
- Tarea 12: Redacción de memoria.

A continuación se muestra el diagrama de Gantt [1] para la planificación inicial:

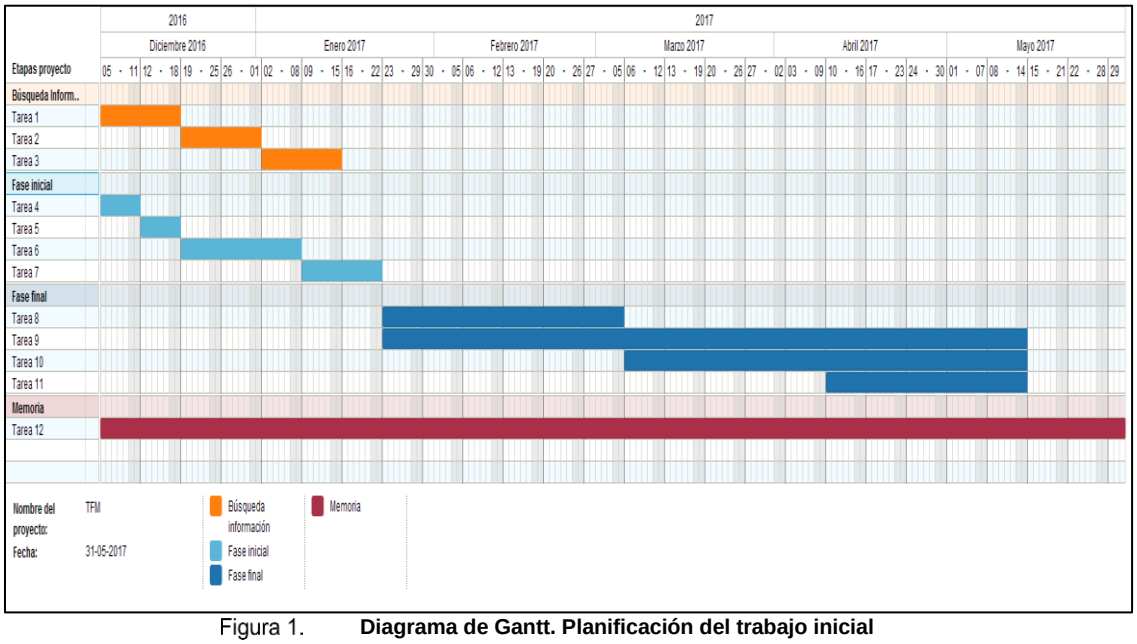


Figura 1. **Diagrama de Gantt. Planificación del trabajo inicial**

Debido al carácter iterativo del problema, durante el desarrollo del proyecto, se ha tenido que reajustar la planificación.

Por un lado, las tareas 8, 9 y 11 están estrechamente ligadas y son consecutivas e iterativas, mientras que la tarea 10 es la última en aplicarse.

Finalmente las tareas se redefinen de la siguiente manera:

- Tareas 1-7: Igual que en la planificación inicial
- Tarea 8: Realización de los modelos predictivos. (Iterativa)
- Tarea 9: Comprobación de resultados y ensamblaje de modelos. (Iterativa)
- Tarea 10: Comparación y entrega de los resultados en la plataforma Kaggle. (Iterativa)
- Tarea 11: Creación de modelo final y testeo.
- Tarea 12: Redacción de memoria.

Y el diagrama de Gantt queda de la siguiente manera:

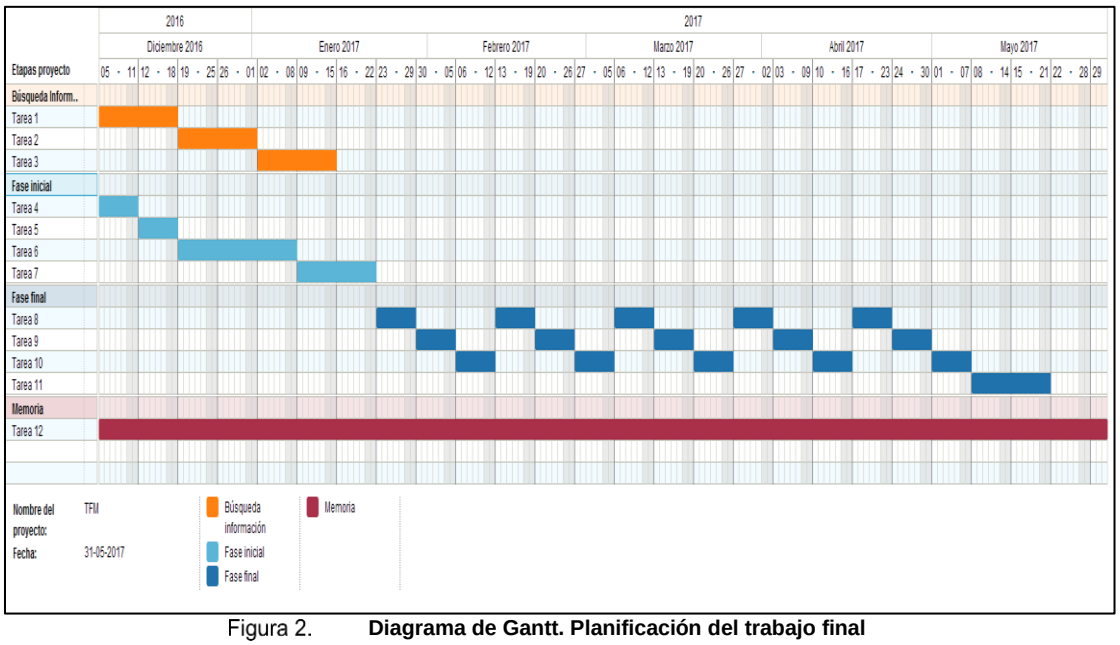


Figura 2. **Diagrama de Gantt. Planificación del trabajo final**

1.5 Breve descripción de los otros capítulos de la memoria

Capítulo 2-Descripción del problema: Se define completamente el problema
Capítulo 3-Elección de las herramientas y del lenguaje de programación: Se escogen las herramientas y el lenguaje de programación a emplear
Capítulo 4-Primeros pasos. Inspección de datos: Toma de contacto con los datos, limpieza y minería de datos
Capítulo 5-Experimentos Computacionales: Desarrollo de todos los experimentos
Capítulo 6-Conclusiones: Comentarios finales y posibles mejoras
Capítulo 7-Glosario
Capítulo 8-Bibliografía
Capítulo 9-Anexos

2 Descripción del problema

El Banco Santander [2] quiso emprender una nueva estrategia comercial para poder recomendar a sus clientes los productos más adecuados según su naturaleza y sus necesidades. Se dieron cuenta que hasta ese momento, tan solo un número reducido de clientes eran reactivos a las nuevas ofertas y productos que el banco ofrecía, mientras que sobre el resto de clientes no tenían ningún tipo de respuesta o "feedback" positivo. Para poder mejorar en este aspecto, propusieron el problema a la plataforma kaggle. El objetivo es predecir los productos que los clientes contratarán en el futuro, basándose en el comportamiento pasado de dichos clientes y en características similares entre clientes.
De esta forma, el banco tendrá una recomendación de productos más efectiva y podrán satisfacer las necesidades individuales de cada cliente.

El punto de partida del problema consiste en un juego de datos en el que se registran mensualmente los comportamientos de la cartera de clientes del banco. Los datos propuestos son de un año y medio (empiezan en Enero de 2015 y acaban en Mayo de 2016).
En concreto, lo que se pide es predecir, para cada cliente, los 7 productos con más probabilidad de ser contratados en el mes de Junio de 2016 adicionalmente de los que tienen contratados en Mayo de 2016.

Los archivos[3] aportados por la plataforma son:

- **train.csv**: es el conjunto de entrenamiento del problema. Consiste en todos los datos de los clientes durante el periodo de tiempo de un año y medio. Cada registro del conjunto de datos es una "fotografía" del cliente a final de mes. Los datos de este registro son las características físicas o personales de cada cliente (fecha, edad, país de residencia, sexo, antigüedad en el banco, etc…) y los productos que tiene contratados en ese mes (cuentas de ahorro, tarjeta de crédito, pensiones, etc.)
- **test.csv**: es el conjunto de prueba del problema. Consiste en los datos personales o físicos de cada cliente en el mes de Junio de 2016.
- **sample_submission.csv:** es el archivo de muestra para hacer la entrega de la solución. Tan solo se tienen los códigos identificativos de los clientes.

A continuación se describen todos los campos del conjunto de datos. Son las características tanto físicas o personales como los productos que tienen contratados.

2.1 Datos personales o físicos

Numero	Nombre de la columna	Descripción	Ejemplo de cliente
1	Fecha_dato	Fecha del registro	2015-01-28
2	Ncodpers	Código identificativo del cliente	1375586
3	Ind_empleado	Índice del empleado: A (activo), B (exempleado), F (filial), N (no empleado), P (pasivo)	N
4	País_residencia	Residencia del cliente	ES
5	Sexo	Hombre/Mujer	H
6	Age	Edad	35
7	Fecha_alta	La fecha en la que el cliente firmó el primer contrato con el banco	2015-01-12
8	Ind_nuevo	Índice 1 si el cliente fue registrado en el banco en los últimos 6 meses.	0
9	Antigüedad	Numero de meses del cliente en el banco.	6
10	Indrel	Indice de relación del cliente con el banco. 1(Cliente principal al principio y al final del mes), 99(Cliente principal al principio del mes, pero no al final)	1
11	Ult_fec_cli_1t	Última fecha en la que el cliente es principal	-
12	Indre_1mes	Tipo de cliente al principio del mes. 1(Cliente principal), 2(Cliente secundario), P(Potencial), (Ex cliente principal), 4(Ex cliente secundario)	1

13	Tiprel_1mes	Tipo de relación del cliente con el banco a principio de mes. A(activo), I(inactivo), P(ex cliente), R(Potencial)	A
14	Indresi	Índice de residencia (S o N si el país de residencia del cliente es el mismo que el del banco)	S
15	Indtext	Índice de nacionalidad. (S o N si el país de nacimiento del cliente es el mismo que el del banco)	N
16	Conyuemp	Índice de matrimonio. 1 (si el cliente es marido o mujer de algún empleado del banco)	-
17	Canal_entrada	Método de captación del cliente.	KHL
18	Indfall	Índice de fallecimiento. Si el cliente ha fallecido o no	N
19	Tipodom	Dirección principal del cliente	1
20	Cod_prov	Código de la provincia	29
21	Nomprov	Nombre de la provincia	MALAGA
22	Ind_actividad_cliente	Indice de actividad (1 cliente activo, 0 cliente inactivo)	1
23	Renta	Ingresos brutos del cliente	87218.10
24	Segmento	Segmentación del banco. 01(VIP), 02(Individual), 03(graduado)	02-PARTICULARES

Tabla 1. Datos del problema “Santander Product Recommendation”. Datos personales.

2.2 Datos de productos

25	Ind_ahor_fin_ult1	Cuenta de ahorro	0
26	Ind_aval_fin_ult1	Avales	0
27	Ind_cco_fin_ult1	Cuenta corriente	1
28	Ind_cder_fin_ult1	Cuenta derivada	0
29	Ind_cno_fin_ult1	Cuenta de nomina	0
30	Ind_ctju_fin_ult1	Cuenta junior	0
31	Ind_ctma_fin_ult1	Cuenta particular adicional	0
32	Ind_ctop_fin_ult1	Cuenta particular	0
33	Ind_ctpp_fin_ult1	Cuenta particular Plus	0
34	Ind_deco_fin_ult1	Depósito a corto plazo	0
35	Ind_deme_fin_ult1	Depósito a medio plazo	0
36	Ind_dela_fin_ult1	Depósito a largo plazo	0
37	Ind_ecue_fin_ult1	Cuenta e	0
38	Ind_fond_fin_ult1	Fondos	0
39	Ind_hip_fin_ult1	Hipoteca	0
40	Ind_plan_fin_ult1	Plan de pensiones	0
41	Ind_pres_fin_ult1	Prestamos	0
42	Ind_reca_fin_ult1	Tasas	0
43	Ind_tjcr_fin_ult1	Tarjeta de crédito	0
44	Ind_valo_fin_ult1	Seguro	0
45	Ind_viv_fin_ult1	Cuenta de hogar	0
46	Ind_nomina_ult1	Nómina	0
47	Ind_nom_pens_ult1	Cuenta de pensiones	0
48	Ind_recibo_ult1	Debito directo	0

Tabla 2. Datos del problema “Santander Product Recommendation”. Datos de productos

En el ejemplo anterior, se han mostrado los datos de un registro del archivo “train.csv”. Se comprueba que se tienen un total de 24 características personales y 24 de productos.

Además, el archivo contiene un total de 13647309 registros, por lo que la tabla de datos es una matriz de 13647309 filas x 48 columnas.

2.3 Competición de Kaggle: “Santander Product Recomendation”

La competición “Santander Product Recomendation” tuvo una duración de dos meses. Empezó el 26 de Octubre de 2016 y finalizó el 21 de Diciembre de 2016.
En total participaron 1787 equipos (Los equipos pueden estar formados por uno o más usuarios) y el premio de la competición fue de 60.000 dólares repartidos entre los tres mejores participantes. (30.000 para el primero, 20.000 para el segundo y 10.000 para el tercero) [4]

2.4 Métricas de evaluación de Kaggle [5]

Las resoluciones entregadas por los competidores son evaluadas por la plataforma Kaggle mediante la Media de Precisión Promedio (Mean Avarage Precision – MAP). En concreto, se deben predecir los siete productos con más probabilidad, por lo tanto se requiere calcular MAP@7.

Esta métrica se calcula de la siguiente manera:

$$MAP@7 = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{\sum_{k=1}^{\min(n,7)} P(k)}{\min(m,7)}$$

Donde $|U|$ es el número de clientes, $P(k)$ es la precisión para en el punto k , n es el numero de productos predichos y m es el número de productos adicionales contratados por el usuario u.

O lo que es lo mismo, es la media de los Promedios de Precisión (Avarage Precision)

$$MAP@7 = \frac{1}{|U|}\sum_{u=1}^{|U|} AP@7$$

En el caso que un cliente no contrate ningún producto ($m = 0$), se considera que la AP@7 de ese usuario es 0.
Para entenderlo mejor, se procede a poner un ejemplo y a calcular el MAP@7 para 3 clientes.

Cliente 1
Productos contratados: [A,B,C,D,E,F,G]
Productos predichos: [V,B,L,X,A,T,F]

[A,B,C,D,E,F,G]			
Posicion	Prediccion	Correcto	Precision
1	V	NO	0
2	B	SI	1/2
3	L	NO	0
4	X	NO	0
5	A	SI	2/5
6	T	NO	0
7	F	NO	3/7

Tabla 3. Ejemplo Cliente 1

$$AP@7 = \frac{\frac{1}{2}+\frac{2}{5}+\frac{3}{7}}{7} = 0.189796$$

Cliente 2
Productos contratados: [H,I,J,K,L,M,N]
Productos predichos: [C,B,R,H,A,P,J]

[H,I,J,K,L,M,N]			
Posicion	Prediccion	Correcto	Precision
1	C	NO	0
2	B	NO	0
3	R	NO	0
4	H	SI	1/4
5	A	NO	0
6	P	NO	0
7	J	SI	2/7

Tabla 4. Ejemplo Cliente 2

$$AP@7 = \frac{\frac{1}{4} + \frac{2}{7}}{7} = 0.076531$$

Cliente 3
Productos contratados: [O,P,Q,R,S,T,U]
Productos predichos: [A,B,C,D,W,U,S]

[O,P,Q,R,S,T,U]			
Posicion	Prediccion	Correcto	Precision
1	A	NO	0
2	B	NO	0
3	C	NO	0
4	D	NO	0
5	W	NO	0
6	U	SI	1/6
7	S	SI	2/7

Tabla 5. Ejemplo Cliente 3

$$AP@7 = \frac{\frac{1}{6} + \frac{2}{7}}{7} = 0.064626$$

Por lo tanto el Mean Avarage Precision 7 (MAP@7) es:

Cliente	Productos contratados	Productos predichos	AP@7
1	[A,B,C,D,E,F,G]	[V,B,L,X,A,T,F]	0.189796
2	[H,I,J,K,L,M,N]	[C,B,R,H,A,P,J]	0.076531
3	[O,P,Q,R,S,T,U]	[A,B,C,D,W,U,S]	0.064626

Tabla 6. MAP@7

$$MAP@7 = \frac{0.189796 + 0.076531 + 0.064626}{3} = 0.110317$$

3 Elección de las herramientas y del lenguaje de programación

Antes de empezar de lleno en la resolución del problema se ha considerado oportuno, describir la máquina con la que se realizarán las pruebas computacionales.
El ordenador es de la marca LENOVO con las siguientes características:

- Procesador Intel® Core™ i7-3632QM CPU @2.20GHz
- Memoria RAM: 8GB

Además, se cree conveniente realizar un estudio comparativo para escoger las herramientas y el lenguaje de programación más adecuado teniendo en cuenta la naturaleza y las características del problema.

En el mundo de la ciencia de datos, los dos lenguajes de alto nivel más empleados son Python y R. [6]
A continuación se hace un breve resumen de las características y requisitos mínimos de cada uno de estos lenguajes.

3.1 Lenguajes de programación

1. Conocimientos previos de programación: Python es un lenguaje más generalista y el aprendizaje más duro pasa por aplicar librerías como (Pandas, Numpy, Scipy…etc). Por lo contrario R es un lenguaje diferente ya que tiene su origen es académico-estadístico.
2. Herramientas User Friendly y GUI: R dispone de herramientas como (Rattle, Rstudio o Rcommander) que tienen buenas interfaces visuales que pueden resolver problemas analíticos sin tener la necesidad de programar. Python, por el contrario tiene menos herramientas, aunque dispone de los notebooks que permiten programar mediante el navegador.
3. Coste de las herramientas: Tanto R como Python son gratuitos.
4. Capacidad de innovación: Ambos lenguajes tienen detrás una comunidad de científicos de datos muy amplia que constantemente están creando aplicaciones, métodos y librerías de ayuda para los programadores.
5. Primeros pasos del autor: Tanto en R como en Python el autor del proyecto tiene conocimientos básicos. Se instalan los dos lenguajes y se hacen distintas pruebas para cargar, leer y modificar los datos.
6. Prueba en Python y R: Para Python se emplea la librería pandas y para R se utiliza la librería data.table. Se carga el archivo train.csv, se visualiza el dataframe y se modifican algunos datos en ambas pruebas. El resultado de ambas pruebas dan como resultado que la librería data.table ha resultado más rápida y cómoda para el tratamiento de datos.

3.2 Algoritmos predictivos

Para llevar a cabo la resolución del problema será necesario emplear alguna técnica predictiva.
A día de hoy existen una gran cantidad de algoritmos predictivos [7]. Entre ellos destacan los siguientes:

- Algoritmos de Regresión línea (Linear Regression): Es un modelo matemático que consiste en aproximar un conjunto de puntos a una línea recta. Dicho de otra manera, tiene el objetivo de establecer una relación lineal entre las variables independientes de un problema con la variable de respuesta o variable dependiente.

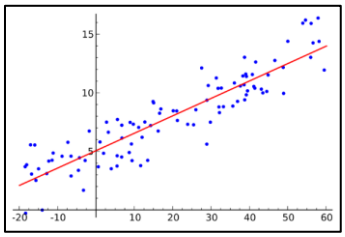


Figura 3. **Regresión Lineal**

- Algoritmos de Regresión logística (Logistic Regression): Es una técnica en la que la variable dependiente es categórica. Es decir, solo puede adoptar un número limitado de categorías, en función de las variables independientes o predictoras.

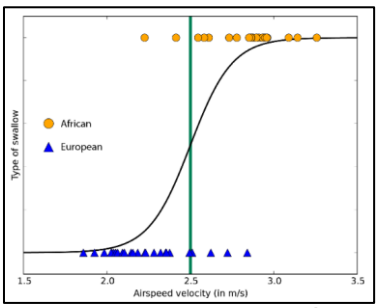


Figura 4. **Regresión Logística**

- Árboles de decisión (Decision trees): Es una forma gráfica y analítica de representar todos los eventos que pueden surgir a partir de una decisión asumida en un cierto momento. Sirven para tomar la decisión "más acertada" desde un punto de vista probabilístico, ante un abanico de posibles decisiones.

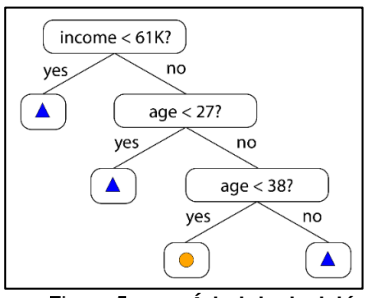


Figura 5. **Árbol de decisión**

- Máquinas de vectores de soporte (Support vector machine): Como en todos los algoritmos predictivos, el SVM, a partir de un conjunto de puntos (en el que cada punto pertenece a una de dos posibles categorías) construye un modelo capaz de predecir si un nuevo punto pertenece a una categoría u otra. Los SVM buscan un hiperplano que tenga la máxima distancia con los puntos que estén más cercanos a él mismo.
  De esta forma, los puntos a los que se les asignará una categoría estarán a un lado del hiperplano y los que se etiqueten en la otra categoría estarán al otro lado.

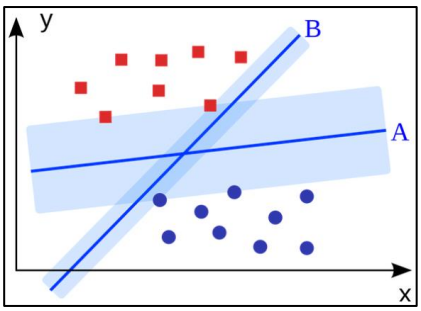


Figura 6. **SVM**

- Redes neuronales (Neural networks): Este tipo de algoritmos simulan el comportamiento del sistema neuronal humano. Una red neuronal suele implicar un número de procesadores dispuestos en niveles que trabajan de forma paralela. El primer nivel recibe la información de entrada, la procesan y devuelven una respuesta que será la entrada del siguiente nivel de la red. El último nivel de la red produce la salida del sistema.

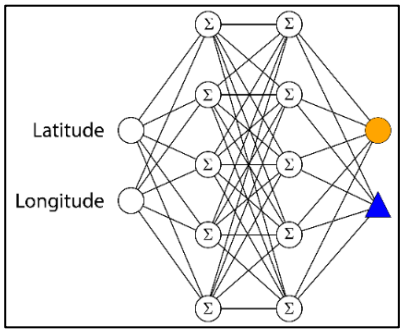


Figura 7. **Red neuronal**

3.3 Gradient Boosting

Uno de los algoritmos más importantes y potentes en las competiciones de Machine Learning es el Gradient Boosting. [8] [9] [10] [11]

Para poder definir esta técnica, previamente vale la pena describir los conceptos de “Boosting” y “Gradient”.

- Boosting (Aumentar): Es una técnica secuencial que se basa en el ensamblaje de modelos. Tiene como objetivo combinar un conjunto de modelos simples (aprendices) para obtener un modelo final con

mejores prestaciones. De esta manera, se aumenta la precisión sin la necesidad de crear un modelo complejo.

A partir de un determinado momento t, se asignan pesos a los resultados obtenidos en el momento anterior t-1. Los resultados predichos correctamente se les asigna un peso bajo, mientras que los predichos erróneamente se les asigna un peso alto.

Para entenderlo mejor, se procede poner un ejemplo:

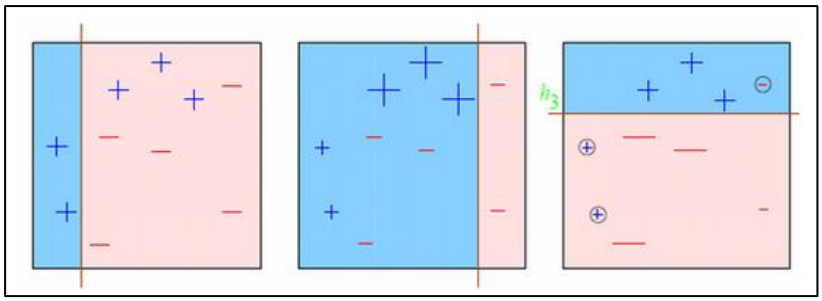


Figura 8. **Ejemplo visual de "Boosting"**

En el primer modelo simple (aprendiz 1, gráfico izquierdo), se puede observar como todos los puntos tienen el mismo peso. El modelo predice correctamente 2 puntos + y 5 puntos -, mientras que falla en las predicciones de 3 puntos +.

En el segundo modelo simple (aprendiz 2, gráfico central), se observa que los puntos correctamente predichos en el modelo anterior, tienen un peso inferior a los puntos fallados. Este nuevo modelo, se centra más en los puntos con mayor peso y predice correctamente 5 puntos + y 2 puntos -, mientras que falla en las predicciones de 3 puntos -.

En el tercer modelo simple (aprendiz 3, gráfico derecho), se observa como se ha reconfigurado los pesos de los puntos en función de si han sido predichos correctamente en el modelo anterior. En este nuevo caso, el modelo predice correctamente 3 puntos + y 4 puntos -, mientras que falla en 2 puntos + y 1 punto -.

Finalmente se combinan los aprendices para obtener el siguiente modelo final:

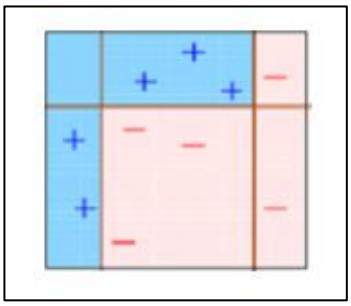


Figura 9. **Modelo final**

- Gradient (Gradiente): Emplea el procedimiento de Gradiente Descendiente para optimizar una función f(x).
  Para encontrar el mínimo de una función, iterativamente, se procede de la siguiente manera:

$$x_i = x_{i-1} - \mu \frac{df(x_{i-1})}{dx}$$

  Donde μ es una constante.

  En efecto, el valor de x encontrado en la actual iteración es igual al valor de x en la iteración anterior más una fracción del gradiente evaluado en el punto anterior.
  El proceso iterativo finaliza cuando $x_i = x_{i-1}$.

  Se empieza el procedimiento asignando el valor inicial x_0.

  Cuando el proceso x_{i-1} corresponde al punto A, el gradiente negativo es alto y consecuentemente x_i (en el punto A') se desplaza considerablemente. En el punto B, el gradiente es considerablemente menor y por lo tanto B' queda cerca de B.
  Por último, en el punto C=C' se cumple la condición de finalización del proceso iterativo y el gradiente es cero.

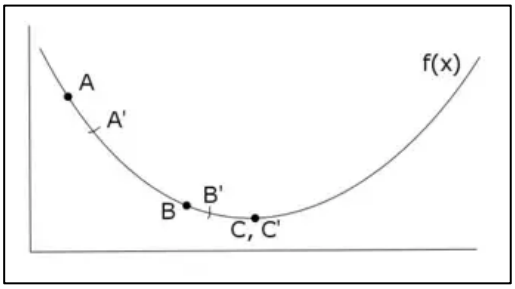


Figura 10. **Gradiente Descendiente**

Por lo tanto, una vez explicados los conceptos clave, se define el Gradient Boosting, como una técnica que emplea el "boosting" para minimizar los errores de los aprendices simples empleando el procedimiento de Gradiente Descendiente.

Los elementos principales del Gradient Boosting son:

- Loss function (función a minimizar): Esta función depende del problema que se pretenda solucionar. Normalmente para problemas de regresión se emplean los errores cuadrados y para problemas de clasificación se usa la "logistic loss"
- Weak learners (Aprendices simples): Se emplean arboles de decisión como modelos simples. Normalmente se restringen las condiciones de los árboles, por ejemplo el número de nodos, niveles o ramas del árbol, para asegurar la simplicidad del modelo
- Additive Model (Modelos añadidos): Son los modelos a ensamblar. Se añade un árbol para cada iteración. Los árboles existentes no se alteran.

3.4 Extreme Gradient Boosting

Después de hacer una búsqueda intensiva de los modelos y algoritmos predictivos que se usan en la actualidad, se encuentra que la técnica más empleada y que está cosechando más éxitos es el Extreme Gradient Boosting. Además, este algoritmo está dominando las resoluciones de Machine Learning de la plataforma Kaggle. [12] [13]
Tal y como su nombre indica, este método se basa en el Gradient Boosting. El término "Extreme" se refiere al objetivo de llevar al límite los recursos computacionales de la técnica de Gradient Boosting para obtener mejores resultados. [14] [15]

Este algoritmo creado por Tianqi Chen [16] (estudiante de la Universidad de Washington) nació con la idea de crear un sistema escalable del algoritmo Gradient Boosting. Lo que finalmente obtuvo, fue un algoritmo con las siguientes mejoras:

- Regularización: Sirve para evitar el over-fitting (sobre entrenamiento). Otras implementaciones del Gradient Boosting no tienen regularización.
- Procesamiento en paralelo: Permite procesamiento en paralelo y es mucho más rápido que otras implementaciones de GB.
- Mayor flexibilidad: Permite a los usuarios definir sus propias funciones de optimización y criterios de evaluación
- Built-in Cross-Validation: Permite al usuario ejecutar cross-validation (técnica para evaluar los resultados y garantizar que son independientes entre los datos de entrenamiento y de prueba) en cada una de las iteraciones.
- Continuar en el modelo existente: Permite a los usuarios retomar un proceso de entrenamiento que se había dejado a medias. (Puede ser una ventaja significativa en ciertas aplicaciones).

Finalmente se escoge el Extreme Gradient Boosting para realizar las predicciones de este trabajo.

4 Primeros pasos. Inspección de datos

Antes de entrar de lleno en el problema, se realiza una primera toma de contacto con los datos del problema. Se pretende realizar una introspección de los datos con la finalidad de encontrar posibles errores, faltas, datos en blanco, valores null, incoherencias, formatos erróneos, etc…

4.1 Carga de ficheros

A continuación se detallan los pasos seguidos:

El primer paso es cargar la librería data.table y definir el directorio de trabajo:

```
# Libraries and workind directory
library(data.table)
library(dplyr)
library(xgboost)
setwd("C:/Users/Pablo/Desktop/Santander/Santander1")
```

Seguidamente se lee el archivo train.csv y se comprueba que la carga de los datos se haya efectuado satisfactoriamente. Se ejecuta el siguiente comando:

```
# Read data
data <- fread("train_ver2.csv")
```

La librería data.table muestra un mensaje de alerta en el que avisa que la columna 12 (indrel_1mes) contiene datos de diferente tipología.

Para comprobar los tipos de datos se ejecuta los siguientes comandos:

```
# Check warnings and look diferent values of "indrel_1mes
levels <- levels(as.factor(data[["indrel_1mes"]]))
levels
```

El resultado es el siguiente:

```
[1] ""    "1"   "1.0" "2"   "2.0" "3"   "3.0" "4"   "4.0" "P"
```

Se observa que hay datos numéricos, datos con decimales y datos que son caracteres. Por lo tanto, y debido a la tabla inicial con los posibles valores del campo indrel_1mes se decide convertir todos los datos a strings.

```
# Convert indrel_1mes to categorical
numIdsTrain <- !data[["indrel_1mes"]] %in% c("", "P")
data[["indrel_1mes"]][numIdsTrain] <-
as.numeric(data[["indrel_1mes"]][numIdsTrain])
levels <- levels(as.factor(data[["indrel_1mes"]]))
```

El resultado es el siguiente:

```
[1] ""  "1" "2" "3" "4" "P"
```

Se observa, pues que lo datos se han corregido correctamente.
Finalmente, se lee el archivo test.csv de la misma forma que el archivo anterior y se comprueban los datos.

```
# Read data
data2 <- fread("test_ver2.csv")
```

En este caso, no ha saltado ningún mensaje de aviso, pero igualmente se comprueban los datos de la columna indrel_1mes.

```
# Check warinings and look diferent valnues of "indrel_1mes
levels <- levels(as.factor(data2[["indrel_1mes"]]))
levels
```

El resultado es el siguiente:

```
[1] "1"  "3"
```

Se comprueba que realmente los datos son correctos.

4.2 Comprobación del número filas y del número de columnas

Se procede a contabilizar el número de filas y columnas de los archivos aportados, para tener una idea de la dimensiones del problema.

Para ello se ejecutan las siguientes instrucciones:

```
# Check the number of rows and columns for train.csv
nrow(data)
ncol(data)
```

El resultado es:

```
[1] 13647309
[1] 48
```

Por lo tanto el archivo train.csv contiene un total de 13647309 filas (registros de clientes por meses) por 48 columnas (datos personales y de productos).

A priori, el archivo train.csv debería tener todos los datos históricos de los clientes. Es decir, cada cliente debería tener un total de 17 registros ya que el periodo de tiempo es de 17 meses y cada registro corresponde a 1 mes.
Para comprobar esta suposición, se procede a comprobar cuántos clientes distintos existen en el archivo train.csv. Se ejecuta la siguiente instrucción:

```
# Check the number of diferent customers in train.csv
customer.list <-  unique(data$ncodpers)
length(customer.list)
```

El resultado es:

```
[1] 956645
```

Se obtienen 956645 clientes distintos.
Seguidamente se procede a comprobar si todos los clientes tienen los 17 registros.

```
# Number diferent customers in train.csv
total <- length(customer.list)*17
```

El resultado obtenido es:

```
[1] 16262965
```

Por lo que, claramente, se concluye que no todos los clientes están registrados en los 17 meses. Esto es debido a que puede haber clientes que hayan sido captados después del inicio del periodo de estudio o que haya clientes que hayan roto las relaciones con el banco durante el periodo de estudio.

Seguidamente, se comprueban las filas y columnas del archivo test.csv.

```
# Check the number of rows and columns for test.csv
nrow(data2)
ncol(data2)
```

El resultado es:

```
[1] 929615
[1] 24
```

El test.csv contiene un total de 929615 filas (registros de clientes en el mes de Junio de 2016) por 24 columnas (datos personales).

Por lo tanto, del total de 956645 clientes que contiene el archivo train.csv, tan solo se deben predecir los productos contratados de 929615 clientes.

4.3 Histórico de contrataciones de cada producto

Para este problema, uno de los puntos más importantes, antes de realizar cálculos predictivos, es conocer aquellos productos que se han contratado más durante el periodo de tiempo estudiado.
Se cree conveniente, pues, realizar un histórico de las contrataciones de los productos.

Para ello, se procede a contabilizar las contrataciones de los productos mes a mes. Es decir, se hace un recuento de las veces que un determinado producto pasa de tener valor 0 (mes anterior) a valer 1 (mes actual). Se realiza para todos los productos en todos los meses.

El resultado se muestra en las siguientes tablas:

month	ind_ahor_fin_ult1	ind_aval_fin_ult1	ind_cco_fin_ult1	ind_cder_fin_ult1	ind_cno_fin_ult1
28/02/2015	1	1	2132	12	1884
28/03/2015	0	1	2293	15	1786
28/04/2015	0	1	2366	12	1984
28/05/2015	0	0	2264	7	1900
28/06/2015	0	0	6588	9	1857
28/07/2015	0	0	3056	10	2771
28/08/2015	0	0	5730	10	2412
28/09/2015	0	0	4389	7	1775
28/10/2015	0	0	5682	3	2409
28/11/2015	0	0	4670	6	2930
28/12/2015	0	0	9077	9	3869
28/01/2016	0	1	4436	12	2155
28/02/2016	0	0	3988	9	2211
28/03/2016	0	0	4551	5	2428
28/04/2016	0	0	4027	5	2381
28/05/2016	1	0	3854	5	2346

Tabla 7. Histórico de contrataciones. Primera parte

month	ind_ctju_fin_ult1	ind_ctma_fin_ult1	ind_ctop_fin_ult1	ind_ctpp_fin_ult1	ind_deco_fin_ult1
28/02/2015	7	291	249	171	348
28/03/2015	18	272	284	165	481
28/04/2015	10	217	278	167	263
28/05/2015	10	205	289	172	238
28/06/2015	7	210	213	151	289
28/07/2015	18	191	252	164	265
28/08/2015	37	240	231	145	322
28/09/2015	22	471	191	122	387
28/10/2015	43	468	223	153	349
28/11/2015	31	633	209	140	110
28/12/2015	69	592	283	155	8
28/01/2016	37	545	259	149	0
28/02/2016	48	770	217	141	1
28/03/2016	45	643	225	131	0
28/04/2016	40	596	240	161	0
28/05/2016	40	512	226	131	0

Tabla 8. Histórico de contrataciones. Segunda parte

month	ind_deme_fin_ult1	ind_dela_fin_ult1	ind_ecue_fin_ult1	ind_fond_fin_ult1	ind_hip_fin_ult1
28/02/2015	33	724	1472	451	5
28/03/2015	23	814	1594	570	7
28/04/2015	26	775	1459	492	4
28/05/2015	25	885	1461	396	9
28/06/2015	33	939	1204	245	4
28/07/2015	24	1138	1085	204	6
28/08/2015	33	1024	924	223	1
28/09/2015	30	1348	1072	130	8
28/10/2015	22	1390	1205	146	7
28/11/2015	1	908	1588	169	3
28/12/2015	0	1199	1816	205	3
28/01/2016	0	447	2272	145	2
28/02/2016	0	684	2342	82	6
28/03/2016	0	239	1909	92	2
28/04/2016	0	99	2249	83	5
28/05/2016	0	46	2709	60	3

Tabla 9. Histórico de contrataciones. Tercera parte

month	ind_plan_fin_ult1	ind_pres_fin_ult1	ind_reca_fin_ult1	ind_tjcr_fin_ult1	ind_valo_fin_ult1
28/02/2015	27	9	217	3471	259
28/03/2015	24	12	217	5076	205
28/04/2015	22	17	871	4627	194
28/05/2015	21	11	343	4071	201
28/06/2015	19	8	2931	4655	152
28/07/2015	27	5	589	4886	232

28/08/2015	22	13	264	4379	445
28/09/2015	21	13	271	4463	328
28/10/2015	19	9	593	4461	911
28/11/2015	37	8	416	4312	133
28/12/2015	157	11	235	4295	255
28/01/2016	104	7	445	3792	739
28/02/2016	45	6	557	3881	320
28/03/2016	30	7	199	4401	163
28/04/2016	21	4	807	4100	128
28/05/2016	22	7	279	4248	183

Tabla 10. Histórico de contrataciones. Cuarta parte

month	ind_viv_fin_ult1	ind_nomina_ult1	ind_nom_pens_ult1	ind_recibo_ult1
28/02/2015	8	5659	5656	8285
28/03/2015	4	4479	4473	9135
28/04/2015	5	2663	5628	8346
28/05/2015	4	3369	3345	7085
28/06/2015	3	5070	8135	9023
28/07/2015	9	4408	4850	9172
28/08/2015	2	2950	2968	7115
28/09/2015	2	4683	4753	11210
28/10/2015	2	4818	4790	12100
28/11/2015	5	4395	4409	10102
28/12/2015	8	5071	5148	9888
28/01/2016	2	2437	2892	10078
28/02/2016	2	9499	12687	11528
28/03/2016	4	4974	5038	10117
28/04/2016	3	3790	4424	9799
28/05/2016	7	5488	5513	10163

Tabla 11. Histórico de contrataciones. Quinta parte

A simple vista se destacan las siguientes observaciones:

- Los productos ind_ahor_fin_ult1 e ind_aval_fin_ult1 no tienen prácticamente contrataciones.

- Los productos ind_cder_fin_ult1, ind_ctju_fin_ult1, in_deme_fin_ult1, ind_deme_fin_ult1, ind_hip_fin_ult1, ind_plan_fin_ult1, ind_pres_fin_ult1 e ind_viv_fin_ult1 el máximo de contrataciones en un mes no alcanza las 200 unidades.

- Los productos ind_ctma_fin_ult1, ind_ctop_fin_ult1, ind_deco_fin_ult1, ind_fond_fin_ult1 e ind_valo_fin_ult1 el máximo de contrataciones en un mes no alcanza las 800 unidades

- Los productos ind_cco_fin_ult1, ind_cno_fin_ult1, ind_dela_fin_ult1, ind_ecue_fin_ult1, ind_reca_fin_ult1,

ind_tjcr_fin_ult1, ind_nomina_ult1, ind_nom_pens_ult1, ind_recibo_ult1 tienen más de 800 contrataciones.

Se puede decir que el grupo de estos 9 últimos productos son los que tiene más éxito entre los clientes.

A continuación se procede a mostrar gráficamente la evolución de las contrataciones de los productos más importantes

4.3.1 Histórico de contrataciones de los productos con más de 800 contrataciones.

A continuación se muestra el gráfico de contrataciones para estos determinados productos.

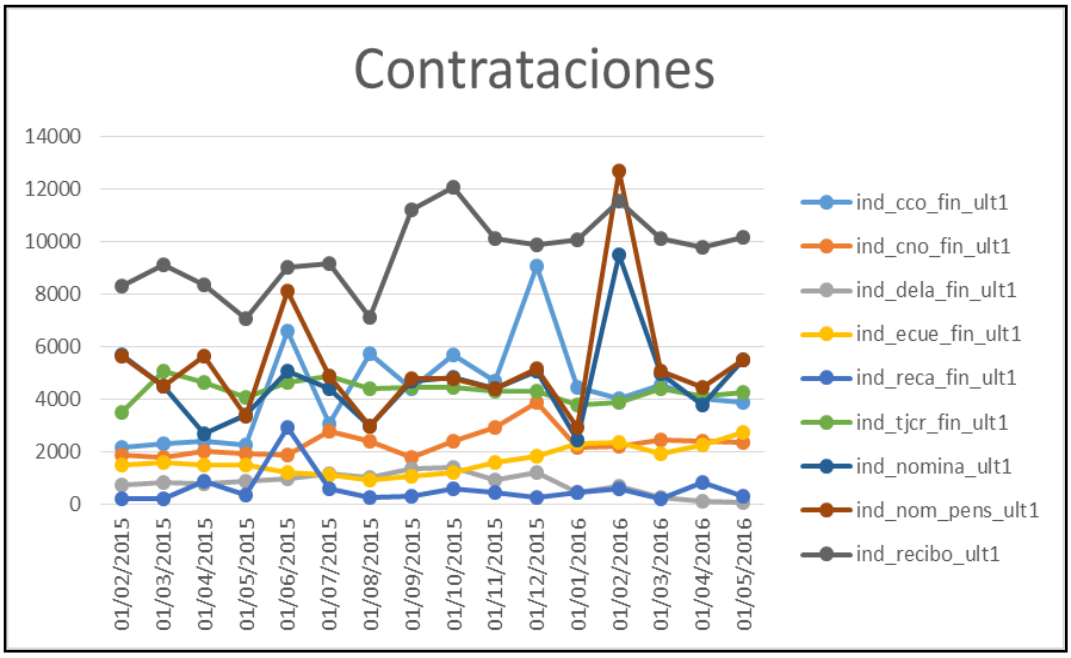


Figura 11. **Histórico de los productos con más de 800 contrataciones**

Se puede observar como hay productos que tienen ciertos picos de contrataciones en determinados meses. Esta información puede ser muy valiosa en el planteamiento del problema ya que se deberá prestar especial atención en estas anomalías.

- Los productos ind_cco_fin_ult1, ind_reca_fin_ult1 e ind_nom_pens_ult1 tienen picos de contrataciones en Junio de 2015.

- El producto ind_cco_fin_ult1 también tiene un pico de contrataciones en Diciembre de 2015

- Los productos ind_nom_pens_ult1 e ind_nomina_ult1 tienen un pico de contrataciones en Febrero de 2016.

- Para el resto de productos, sí que hay un altibajo en las contrataciones, pero se puede decir que la tendencia es relativamente estable

4.3.2 Histórico de contrataciones de los productos con más de 200 contrataciones.

Se muestra el gráfico para los productos del segundo nivel de contrataciones

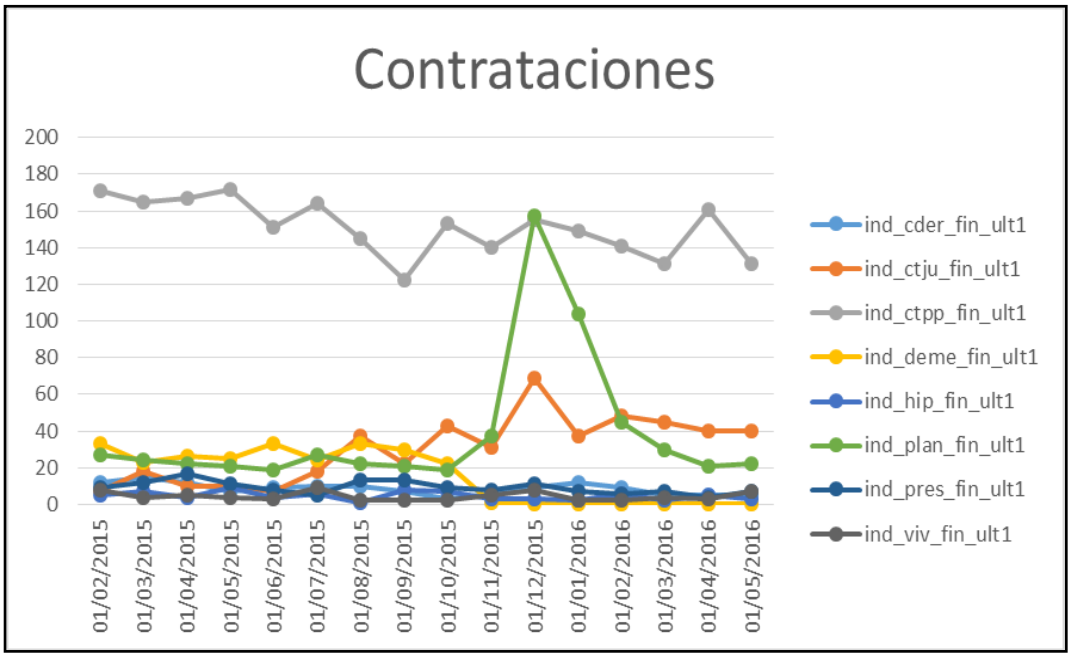


Figura 12. **Histórico de los productos con más de 200 contrataciones**

En este gráfico se observan claramente como los productos con menos contrataciones la evolución de las curvas es relativamente estable.

Sin embargo, los productos ind_plan_fin_ult1 e ind_ctju_fin_ult1 sufren un incremento notable de las contrataciones en el mes de Diciembre de 2015.

También vale la pena destacar que el producto ind_deme_fin_ult1 deja de ser contratado a partir de Diciembre de 2015.

Por último comentar que el producto ind_deco_fin_ult1, también deja de ser contratado en estos últimos meses.

4.4 Máxima puntuación de MAP@7

Tal y como se ha comentado en el apartado del MAP@7, cuando un cliente no contrata ningún producto, la métrica AP@7 es cero.
La máxima puntuación del MAP es 1, siempre y cuando todos los clientes contraten algún producto en Junio de 2016 y se predigan todos los productos correctamente.

Como es lógico, puede haber clientes que no contraten productos en el mes de Junio de 2016. Esto provoca que la métrica MAP@7 sea inferior a 1. Al no saber los clientes que no contratarán ningún producto en Junio, no se puede saber la máxima puntuación del MAP@7.
Sin embargo, se puede hacer una estimación de la máxima puntuación para Junio de 2016, comparándola con los MAP@7 de otros meses.
Para ello, se procede a contabilizar los clientes que contratan algún producto en los meses de Mayo, Abril y Marzo ya que son los meses más cercanos al mes de predicción.

En Mayo de 2016, del total de 931453 clientes, 29717 contrataron al menos uno de los productos del banco.
Esto implica que:

- 901736 clientes no contrataron ningún producto, por lo que su AP@7 es cero.
- Si se predicen correctamente los 7 productos con mayor probabilidad de ser escogidos, para los 29717 clientes el AP@7 es igual a 1.
- Finalmente el MAP@7 será:

$$MAP@7(Mayo) = \frac{29717}{931453} = 0.03190392$$

Se procede de la misma manera para Abril y Marzo y los resultados son:

$$MAP@7(Abril) = \frac{28707}{928274} = 0.03092514$$

$$MAP@7(Marzo) = \frac{30544}{925076} = 0.03301783$$

Mes	MAP@7
Mayo	0.03190392
Abril	0.03092514
Marzo	0.03301783

Tabla 12. Mean Avarage Precision Maxima para los meses de Mayo, Abril y Marzo

Es razonable pensar, que para Junio de 2016, la puntuación máxima del MAP@7 será un valor parecido a los anteriores.

4.4.1 Puntuaciones de Kaggle de los dos mejores usuarios en la competición

A continuación se muestran las puntuaciones del MAP@7 de los dos mejores clasificados en la competición de Kaggle. [17]

1. Idle_speculation. Puntuación: 0.031409
2. Tom Van de Wiele. Puntuación: 0.0313171

Como se observa, la estimación calculada anteriormente se aproxima bastante a las mejores notas de los usuarios de la competición. El objetivo del trabajo, será intentar acercarse lo máximo a estas puntuaciones

4.5 Información complementaria de los dos mejores usuarios en la competición

En este apartado, se pretende realizar una recopilación del "state of art". Es decir, se obtendrá información de los trabajos de los mejores competidores del problema.
Al haber finalizado la competición, gran parte de los competidores publicaron en el foro de la plataforma las soluciones de sus trabajos.
Con esta información, se comprobarán aproximaciones, se identificarán modelos y metodologías y se tendrá una idea global del ejercicio.

4.5.1 Primer clasificado: Idle_speculation [18]

Este usuario fue el primer clasificado de la competición.

Su propuesta fue ensamblar doce modelos empleando redes neuronales con ocho modelos de Gradient Boosting.

A continuación se explican los detalles de su publicación:

- Características
  Emplea las características aportadas por el juego de datos y añade las siguientes:
  - Tiempos en los que un producto permanece o no permanece contratado
  - Tiempo desde la última contratación de un producto
  - Medias de productos contratados
  - Tiempos desde que se detecta un cambio en características que no son productos (por ejemplo: canal_entrada, ind_actividad_cliente, etc…)

- Modelos Gradient Boosting
  Emplea modelos predictivos multiclase. Es decir, tiene como objetivo obtener la probabilidad de todas las posibles clases.
  Divide el dominio del problema en 17 clases. Las 16 primeras son los 16 productos más contratados mientras que la 17 indica cero contrataciones o alguna contratación de los productos menos contratados.

- Modelos de Redes neuronales
  A diferencia de los modelos anteriores, los modelos de redes neuronales se basan en la presencia de un producto en mes determinado. No le da importancia a si el producto ha sido contratado en ese mes o si lleva contratado meses atrás.
  La estructura de la red es de una capa de entrada, dos capas intermedias con 512 nodos y una capa de salida.

- Post-proceso
  Para cada uno de los submodelos obtiene el resultado para Junio 2016, Diciembre 2015 y Junio 2015.

Los resultados finales son los obtenidos en Junio de 2016 para todos los productos exceptuando el producto “cco” que añaden los resultados de Diciembre de 2015 y el producto “reca” que añade los resultados de Junio de 2015.

4.5.2 Segundo clasificado: Tom Van de Wiele [19]

Este usuario fue el segundo clasificado de la competición.

A continuación se comentan los detalles de su publicación:

- Características
  Le da máxima importancia a las diferencias entre las variables del problema entre el mes actual y el mes anterior. Es decir, se centra en los cambios de los valores de las variables entre dos meses consecutivos.
  Añade las siguientes características:
  - Para cada producto comprueba las diferencias entre el mes actual y el anterior.
  - Recuento de meses desde que un producto es contratado hasta que deja de ser contratado.
  - Recuento de meses desde que un producto no ha sido contratado hasta que es contratado.
  - Número de contrataciones y descontrataciones de cada producto.

- Modelos
  Construye los modelos empleando el algoritmo Extreme Gradient Boosting.
  Realiza las predicciones de los 24 productos, basándose, por separado, en los datos de los meses desde Febero de 2015 hasta Mayo de 2016.

- Ensamblaje de modelos
  Una vez obtenidos los resultados de cada uno de los modelos, realiza un ensamblaje para obtener las predicciones finales.
  El ensamblaje lo realiza, aplicando pesos a los modelos en función de los resultados obtenidos en las entregas en la plataforma Kaggle.
  Por ejemplo, para el producto ind_cco_fin_ult1 aplica un peso de 0.3 al modelo de Febrero de 2015, un 0.3 al modelo de Marzo de 2015, un peso de 13 para el modelo de Junio de 2015...etc.

5 Experimentos Computacionales

En este capítulo se explicarán en detalle todos los experimentos computacionales realizados para la resolución del problema.
En cada uno de los experimentos ser procederá de la siguiente manera:

- Se añadirá un cambio en los modelos planteados.
- Se ejecutará el código computacional y se obtendrán las predicciones.
- Se entregarán las predicciones a la plataforma Kaggle y se obtendrá la puntuación del MAP@7.
- Se aceptarán los cambios si la puntuación obtenida es superior a la del experimento anterior.
- Se declinarán los cambios si los resultados no mejoran la puntuación del experimento anterior.

Los cambios que se realizarán en cada uno de los experimentos serán, básicamente, añadir o quitar características de los modelos.
Esta tarea es la más importante del proyecto, ya que, escoger las características o variables más decisivas para realizar las predicciones, es la clave del éxito del problema. En total se han realizado 15 experimentos que se resumen en la siguiente tabla:

Experimento	Modelos
A	Se usan como características los productos del mes anterior. Se empieza por Mayo de 2016
B	Se eliminan los modelos de los productos que tienen muy pocas contrataciones en los últimos meses y se fija su probabilidad en 10^{-10}
C	Se eliminan de los modelos los clientes que en el mes anterior tenían contratado el producto a modelizar
D	Se contabilizan el comportamiento de compra de los clientes a lo largo de todo el periodo de estudio. Se añaden count_00, count_01, count_10 y count_11
E	Se añaden las características físicas o personales de los clientes. Se convierten los datos categóricos a numéricos
F	Se repite el mismo experimento para Abril de 2016
G	Se repite el mismo experimento para Marzo de 2016
H	Se repite el mismo experimento para Febrero de 2016
I	Se repite el mismo experimento para Enero de 2016
J	Se repite el mismo experimento para Diciembre de 2015
K	Para el modelo del producto ind_cco_fin_ult1 se emplean los datos de Diciembre de 2015. Para el resto de productos, los de Abril de 2016
L	Para el modelo del producto ind_nom_pens_ult1 se emplean los datos de Febrero de 2016. Para el resto de productos, los de Abril de 2016
M	Se repite el mismo el experimento F pero para Junio de 2015.
N	Para el modelo del producto ind_reca_fin_ult1 se emplean los datos de Junio de 2015. Para el resto de productos, los de Abril de 2016
FINAL	Se repite el experimento N cuatro veces, cambiando los datos de Abril por los de Mayo, Marzo, Febrero y Enero. Se ensamblan los modelos calculando la media de las probabilidades de los 5 experimentos.

Tabla 13. Tabla resumen de los experimentos computacionales

Vale la pena comentar, que los cambios realizados se irán acumulando experimento tras experimento si las puntuaciones obtenidas son mejores que las anteriores. A continuación se explican en detalle cada uno de los experimentos realizados.

5.1 Experimento A

5.1.1 Introducción y aspectos básicos

El planteamiento inicial de este ejercicio consistirá en predecir las contrataciones del mes de Junio de 2016, teniendo en cuenta tan solo los datos del mes anterior. Es decir, a partir de los datos del mes de Mayo de 2016, se predecirán las compras de Junio de 2016.
Se ha elegido esta opción como una primera toma de contacto y para familiarizarse con la metodología de trabajo.
Además, es de sentido común empezar por este mes ya que a nivel económico, social o de oportunidades, la situación del cliente en Junio de 2016 será más parecida a la de Mayo de 2016.

Vale la pena comentar, que el problema no es un ejercicio de predicción único. Es decir, no se debe predecir tan solo un producto, sino que se deben predecir las contrataciones de los 24 productos disponibles en el banco.
Por lo tanto, se deberán plantear 24 modelos y se deberán solucionar, por separado, las 24 resoluciones.

5.1.2 Matriz inicial de datos

El punto de partida del ejercicio es la obtención de la matriz inicial de datos. Después de la limpieza y modificación de datos del archivo train.csv, la tabla resultante tiene la siguiente forma:

Nº	ID	fecha registro	Datos personales (22 columnas)			
	ncodpers	fecha_dato	ind_empleado	pais_residencia	...	segmento
1	15889	28/01/2015	F	ES	...	01 - TOP
2	15890	28/01/2015	A	ES	...	02 - PARTICULARES
…	…	…	…	…	…	…
…	…	…	…	…	…	…
13647309	1550586	28/05/2016	N	ES	…	03 - UNIVERSITARIO

Tabla 14. Tabla de datos inicial. Datos personales

Nº	Datos de productos (24 columnas)			
	ind_ahor_fin_ult1	ind_aval_fin_ult1	...	ind_recibo_ult1
1	0	0	...	0
2	0	0	...	1
…	…	…	…	…
…	…	…	…	…
13647309	0	0	...	0

Tabla 15. Tabla de datos inicial. Datos de productos

A partir de esta tabla inicial de datos, se montarán las tablas necesarias para crear los modelos predictivos.

5.1.3 Modelos

La creación de los modelos consiste en decidir qué datos se van a emplear para la fase de entrenamiento.
Debido a que el planteamiento inicial es predecir los datos del mes de Junio teniendo en cuenta tan solo los datos de Mayo, se necesitarán también los datos del mes de Abril para poder entrenar los modelos.
En este caso, pues, se desprecian todos los datos del periodo de tiempo entre Enero de 2015 y Marzo de 2016.
Además, de forma intuitiva, se cree que los datos más relevantes para realizar la predicción son los datos de los productos, por lo que también se desprecian los datos personales o físicos de los clientes.

Como ya se ha comentado anteriormente, las predicciones se realizarán individualmente. Es decir, se montará un modelo para cada producto del banco y se realizará la predicción de Junio de 2016.
A modo de ejemplo, se detallan los pasos realizados para el modelo del producto ind_cco_fin_ult1.

5.1.4 Fase de entrenamiento

Se monta el modelo mediante el método xgboost (Extreme Gradient Boosting).

- Se construye la matriz X de entrenamiento: Son los datos de los productos del mes de Abril de 2016.

Nº	ID	Datos de productos del mes **anterior** (24 columnas)			
	ncodpers	ind_ahor_fin_ult1_last	ind_aval_fin_ult1_last	...	ind_recibo_ult1_last
1	15889	0	0	...	0
2	15890	0	0	...	1
…	…	…	…	…	…
…	…	…	…	…	…
926663	1548207	0	0	…	0

Tabla 16. Tabla de datos de entrenamiento. Matriz X. Mes de Abril de 2016

- Se pasa la respuesta o Y: Son los datos en el mes de Mayo de 2016 del producto que se va a predecir.

Datos del producto en el mes **actual**		
Nº	ncodpers	ind_cco_fin_ult1
1	15889	1
2	15890	0
…	…	…
…	…	…
926663	1548207	0

Tabla 17. Tabla de datos de entrenamiento. Matriz Y. Mes de Mayo de 2016

5.1.5 Fase de prueba

Una vez entrenado el modelo, se procede a predecir las respuestas del mes de Junio empleando la matriz de Mayo.

- Se construye la matriz X de prueba: Son los datos de los productos del mes de Mayo de 2016.

Nº	ID	Datos de productos del mes **actual** (24 columnas)			
	ncodpers	ind_ahor_fin_ult1_last	ind_aval_fin_ult1_last	...	ind_recibo_ult1_last
1	15889	0	0	...	0
2	15890	0	0	...	1
...	...	...	...	...	...
...	...	...	...	...	...
929615	1553689	0	0	...	0

Tabla 18. Tabla de datos de prueba. Matriz X. Mes de Mayo de 2016

5.1.6 Resultados de la predicción

La solución se muestra en la siguiente tabla:

Nº	ncodpers	ind_cco_fin_ult1
1	15889	0.990870893
2	15890	0.003800573
...	...	...
929615	1553689	0.008232009

Tabla 19. Resultados de la predicción del producto ind_cco_fin_ult1 para el mes de Junio de 2016

5.1.7 Comentarios de los resultados

Se obtienen las predicciones de todos los productos para todos los clientes del mes de Junio de 2016. Por lo tanto, cada cliente tendrá asociadas 24 probabilidades distintas (1 por cada producto del banco).
Seguidamente, se ordenan las probabilidades para cada cliente y se escogen los 7 productos que han obtenido una mayor probabilidad.
Estos 7 productos serán los que se recomendarán en la entrega final del ejercicio.

Finalmente la matriz de recomendación queda de la siguiente manera:

Lista de los 7 productos recomendados			
15889	15890		1553689
ind_valo_fin_ult1	ind_plan_fin_ult1		ind_cco_fin_ult1
ind_ctpp_fin_ult1	ind_ecue_fin_ult1		ind_recibo_ult1
ind_cco_fin_ult1	ind_ctpp_fin_ult1		ind_deme_fin_ult1
ind_tjcr_fin_ult1	ind_cno_fin_ult1	...	ind_ctma_fin_ult1
ind_recibo_ult1	ind_recibo_ult1		ind_nom_pens_ult1
ind_nom_pens_ult1	ind_nom_pens_ult1		ind_ecue_fin_ult1
ind_nomina_ult1	ind_nomina_ult1		ind_nomina_ult1

Tabla 20. Lista de los 7 productos recomendados para cada cliente

5.1.8 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 13. **Evaluación de kaggle. MAP@7 del experimento A**

5.2 Experimento B

Haciendo un repaso de los gráficos históricos comentados en el apartado 4.3, da la casualidad que hay dos productos que prácticamente no tienen contrataciones y hay dos productos que se dejan de contratar en los últimos meses.

Estos productos son: ind_ahor_fin_ult1, ind_aval_fin_ult1, ind_deco_fin_ult1 e ind_deme_fin_ult1

month	ind_ahor_fin_ult1	ind_aval_fin_ult1	ind_deco_fin_ult1	ind_deme_fin_ult1
28/02/2015	1	1	348	33
28/03/2015	0	1	481	23
28/04/2015	0	1	263	26
28/05/2015	0	0	238	25
28/06/2015	0	0	289	33
28/07/2015	0	0	265	24
28/08/2015	0	0	322	33
28/09/2015	0	0	387	30
28/10/2015	0	0	349	22
28/11/2015	0	0	110	1
28/12/2015	0	0	8	0
28/01/2016	0	1	0	0
28/02/2016	0	0	1	0
28/03/2016	0	0	0	0
28/04/2016	0	0	0	0
28/05/2016	1	0	0	0

Tabla 21. Histórico de productos con pocas contrataciones

Viendo estos datos, sería lógico pensar que en Junio de 2016 no va a haber contrataciones de estos productos.

5.2.1 Modelos

Por lo tanto, el primer cambio que se va a plantear es eliminar estos productos de los modelos y fijar la probabilidad de ser contratados a un valor muy bajo (10^{-10}),

A modo de ejemplo, se detallan los pasos realizados para el modelo del producto ind_ctop_fin_ult1.

5.2.2 Fase de entrenamiento

Se monta el modelo mediante el método xgboost (Extreme Gradient Boosting).

- Se construye la matriz X de entrenamiento: Son los datos de los productos del mes de Abril de 2016 sin los productos eliminados

Nº	ID	Datos de productos del mes **anterior** (20 columnas)			
	ncodpers	ind_cco_fin_ult1_last	ind_cder_fin_ult1_last	...	ind_recibo_ult1_last
1	15889	1	0	...	0
2	15890	0	0	...	1
…	…	…	…	…	…
…	…	…	…	…	…
926663	1548207	0	0	…	0

Tabla 22. Tabla de datos de entrenamiento. Matriz X. Mes de Abril de 2016 sin productos eliminados

- Se pasa la respuesta o Y: Son los datos en el mes de Mayo de 2016 del producto que se va a predecir.

Datos del producto en el mes **actual**		
Nº	ncodpers	ind_ctop_ult1
1	15889	0
2	15890	0
…	…	…
…	…	…
926663	1548207	0

Tabla 23. Tabla de datos de entrenamiento. Matriz Y. Mes de Mayo de 2016 sin productos eliminados

5.2.3 Fase de prueba

Una vez entrenado el modelo, se procede a predecir las respuestas del mes de Junio empleando matriz de Mayo.

- Se construye la matriz X de prueba: Son los datos de los productos del mes de Mayo de 2016 sin los productos eliminados

Nº	ID	Datos de productos del mes **actual** (20 columnas)			
	ncodpers	ind_cco_fin_ult1_last	ind_cder_fin_ult1_last	...	ind_recibo_ult1_last
1	15889	1	0	...	0
2	15890	0	0	...	1
…	…	…	…	…	…
…	…	…	…	…	…
929615	1553689	0	0	…	0

Tabla 24. Tabla de datos de prueba. Matriz X. Mes de Mayo de 2016 sin productos eliminados

5.2.4 Resultados de la predicción

La solución se muestra en la siguiente tabla:

Nº	ncodpers	ind_cco_fin_ult1
1	15889	0.0011633243
2	15890	0.0007303815
…	…	…
929615	1553689	0.0001107773

Tabla 25. Resultados de la predicción del producto ind_ctop_fin_ult1 para el mes de Junio de 2016

5.2.5 Comentarios de los resultados

Se obtienen las predicciones de todos los productos para todos los clientes del mes de Junio de 2016 y se ordenan en orden descendente. Se escogen los 7 primeros productos.

Finalmente la matriz de recomendación queda de la siguiente manera:

Lista de los 7 productos recomendados			
15889	15890		1553689
ind_valo_fin_ult1	ind_plan_fin_ult1		ind_cco_fin_ult1
ind_ctpp_fin_ult1	ind_ecue_fin_ult1		ind_recibo_ult1
ind_cco_fin_ult1	ind_cno_fin_ult1		ind_ctma_fin_ult1
ind_tjcr_fin_ult1	ind_ctpp_fin_ult1	…	ind_nom_pens_ult1
ind_recibo_ult1	ind_recibo_ult1		ind_ecue_fin_ult1
ind_nom_pens_ult1	ind_nomina_ult1		ind_tjcr_fin_ult1
ind_nomina_ult1	ind_nom_pens_ult1		ind_cno_fin_ult1

Tabla 26. Lista de los 7 productos recomendados para cada cliente

5.2.6 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 14. **Evaluación de kaggle. MAP@7 del experimento B**

Se observa una ligera mejoría con respecto al Experimento A. Por lo tanto, se cree conveniente seguir a partir de este cambio en los siguientes experimentos.

5.3 Experimento C

Para intentar mejorar las puntuaciones obtenidas en el experimento B, se cree conveniente aplicar algunos cambios en los datos seleccionados.

Tal y como se ha comentado, en el experimento anterior se han empleado:

- Todos los clientes del mes de Abril para la matriz X de la fase de entrenamiento.
- La respuesta de Mayo para la columna Y de la fase de entrenamiento.

- Todos los clientes del mes de Mayo para la matriz X de la fase de prueba.

5.3.1 Modelos

El cambio propuesto para este nuevo experimento será emplear:

- Todos **los clientes del mes de Abril en los que el producto modelizado tenga valor igual a 0** para la matriz X de la fase de entrenamiento.
- La respuesta de Mayo para la columna Y de la fase de entrenamiento, **de los clientes filtrados en la selección anterior.**
- Todos **los clientes del mes de Mayo en los que el producto modelizado tenga valor igual a 0** para la matriz X de la fase de prueba.
- Se predicen las respuestas de Junio.

A modo de ejemplo, se detallan los pasos realizados para el modelo del producto ind_fond_fin_ult1.

5.3.2 Fase de entrenamiento

Se monta el modelo mediante el método xgboost (Extreme Gradient Boosting).

- Se construye la matriz X de entrenamiento: Son los datos de los productos del mes de Abril de 2016 con la restricción de los clientes que tienen el producto modelizado igual a 0.

Nº	ID	Datos de productos del mes anterior (20 columnas)				
	ncodpers	ind_cco_fin_ult1_last	...	**ind_fond_ult1_last**	...	ind_recibo_ult1_last
1	15893	0	...	**0**	...	0
2	15898	0	...	**0**	...	0
...	...	...	...	**0**	...	...
...	...	...	...	**0**	...	...
300248	1548207	0	...	**0**	...	0

Tabla 27. Tabla de datos de entrenamiento. Matriz X. Mes de Abril de 2016 con restricción

- Se pasa la respuesta o Y: Son los datos en el mes de Mayo de 2016 del producto que se va a predecir con la restricción de los clientes anteriormente seleccionados

Datos del producto en el mes actual		
Nº	ncodpers	ind_fond_fin_ult1
1	15893	0
2	15898	0
...	...	...
...	...	...
300248	1548207	0

Tabla 28. Tabla de datos de entrenamiento. Matriz Y. Mes de Mayo de 2016 con restricción

5.3.3 Fase de prueba

Una vez entrenado el modelo, se procede a predecir las respuestas del mes de Junio empleando matriz de Mayo.

- Se construye la matriz X de prueba: Son los datos de los productos del mes de Mayo de 2016 con la restricción de los clientes que tienen el producto modelizado igual a 0.

Nº	ID	Datos de productos del mes anterior (20 columnas)				
	ncodpers	ind_cco_fin_ult1_last	...	**ind_aval_fond_ult1_last**	...	ind_recibo_ult1_last
1	15889	1	...	**0**	...	0
2	15890	0	...	**0**	...	1
...	...	...	...	**0**	...	...
...	...	...	...	**0**	...	...
914928	1553689	0	...	**0**	...	0

Tabla 29. Tabla de datos de prueba. Matriz X. Mes de Mayo de 2016 con restricción

5.3.4 Resultados de la predicción

La solución se muestra en la siguiente tabla:

Nº	ncodpers	ind_fond_fin_ult1
1	15889	4.112769e-04
2	15890	2.639858e-04
...	...	...
914928	1553689	4.782745e-06

Tabla 30. Resultados de la predicción del producto ind_fond_fin_ult1 para el mes de Junio de 2016

5.3.5 Comentarios de los resultados

Se obtienen las predicciones de todos los productos para todos los clientes del mes de Junio de 2016 y se ordenan en orden descendente. Se escogen los 7 primeros productos.

Finalmente la matriz de recomendación queda de la siguiente manera:

Lista de los 7 productos recomendados			
15889	15890		1553689
ind_recibo_ult1	ind_cco_fin_ult1		ind_cco_fin_ult1
ind_nom_pens_ult1	ind_reca_fin_ult1		ind_recibo_ult1
ind_nomina_ult1	ind_valo_fin_ult1		ind_ctma_fin_ult1
ind_ecue_fin_ult1	ind_fond_fin_ult1	...	ind_ecue_fin_ult1
ind_cno_fin_ult1	ind_ctop_fin_ult1		ind_nom_pens_ult1
ind_ctop_fin_ult1	ind_dela_fin_ult1		ind_nomina_ult1
ind_reca_fin_ult1	ind_ctma_fin_ult1		ind_tjcr_fin_ult1

Tabla 31. Lista de los 7 productos recomendados para cada cliente

5.3.6 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 15. **Evaluación de kaggle. MAP@7 del experimento C**

Se observa una mejoría muy significativa con respecto al Experimento B. Escoger aquellos clientes que no tienen contratado el producto modelizado aporta una gran ventaja al problema.

Por lo tanto, los clientes que tienen contratado el producto a modelizar distorsiona la predicción.

Esto es así, ya que a la hora de montar el árbol de decisión, el modelo se condiciona en gran manera por aquellos clientes que tienen contratado el producto modelizado.

A partir de los siguientes experimentos, se aplicará este cambio.

5.4 Experimento D

Una vez obtenido, en el experimento C, una clara mejoría en la puntuación de kaggle, se propone como novedad añadir un recuento del histórico de compra de cada cliente para cada producto.

5.4.1 Modelos

En concreto, para cada cliente y para cada producto del banco, el recuento se va a realizar de la siguiente manera:

- Numero de ocurrencias que un cliente NO tiene contratado un producto en el mes anterior y NO tiene contratado el producto en el mes actual. Pasa de 0 a 0 (count_00)
- Numero de ocurrencias que un cliente NO tiene contratado un producto en el mes anterior y SI tiene contratado el producto en el mes actual. Pasa de 0 a 1 (count_01)
- Numero de ocurrencias que un cliente SI tiene contratado un producto en el mes anterior y NO tiene contratado el producto en el mes actual. Pasa de 1 a 0 (count_10)
- Numero de ocurrencias que un cliente SI tiene contratado un producto en el mes anterior y SI tiene contratado el producto en el mes actual. Pasa de 1 a 1 (count_11)

Por ejemplo, el cliente 15892, para el producto ind_cco_fin_ult, hasta el mes de Abril de 2016 ha tenido la siguiente evolución:

ncodpers	fecha_dato	ind_cco_fin_ult1
15892	28/01/2015	0
15892	28/02/2015	0
15892	28/03/2015	0
15892	28/04/2015	0
15892	28/05/2015	0
15892	28/06/2015	1
15892	28/07/2015	1
15892	28/08/2015	1
15892	28/09/2015	1
15892	28/10/2015	1
15892	28/11/2015	1
15892	28/12/2015	1
15892	28/01/2016	1
15892	28/02/2016	1
15892	28/03/2016	1
15892	28/04/2016	1
15892	28/05/2016	1

Tabla 32. Tabla de evolución de compras del cliente 15892 para el producto ind_cco_fin_ult1

Como se puede comprobar, el cliente 15892 hasta el mes de Junio de 2015 no contrató el producto. En ese mes, lo contrató y hasta el final del periodo lo ha seguido teniendo contratado.

Por lo tanto el recuento realizado hasta el mes de Abril de 2016 es el siguiente:

Hasta el mes de Abril de 2016					
ncodpers	producto	count_00	count_01	count_10	count_11
15892	ind_cco_fin_ult1	4	1	0	10

Tabla 33. Tabla de recuento de la evolución del cliente 15892 para el producto ind_cco_fin_ult1 hasta Abril de 2016

Y el recuento realizado hasta el mes de Mayo de 2016 es el siguiente:

Hasta el mes de Mayo de 2016					
ncodpers	producto	count_00	count_01	count_10	count_11
15892	ind_cco_fin_ult1	4	1	0	11

Tabla 34. Tabla de recuento de la evolución del cliente 15892 para el producto ind_cco_fin_ult1 hasta Mayo de 2016

A modo de ejemplo, se detallan los pasos realizados para el modelo del producto ind_hip_fin_ult1.

5.4.2 Fase de entrenamiento

Se monta el modelo mediante el método xgboost (Extreme Gradient Boosting).

- Se construye la matriz X de entrenamiento: Se siguen manteniendo los cambios realizados en el experimento C y se añade el recuento de la evolución de cada cliente con cada producto. El recuento es hasta Abril de 2016.

Nº	ID	Datos de productos del mes anterior (20 columnas)				
	ncodpers	ind_cco_fin_ult1_last	...	**ind_hip_ult1_last**	...	ind_recibo_ult1_last
1	15889	1	...	**0**	...	0
2	15890	0	...	**0**	...	1
3	15892	1	...	**0**	...	1
...	...	...	...	**0**	...	...
922116	1548203	0	...	**0**	...	0

Tabla 35. Tabla de datos de entrenamiento. Matriz X. Mes de Abril de 2016 con recuentos. Primera parte

Nº	ID	Datos del recuento (80 columnas)					
	ncodpers	...	count_00_cco	count_01_cco	count_10_cco	count_11_cco	...
1	15889	...	0	0	0	15	...
2	15890	...	15	0	0	0	...
3	15892	...	4	1	0	10	...
...	...	...	...	0	...	...	...
922116	1548203	...	0	0	0	0	...

Tabla 36. Tabla de datos de entrenamiento. Matriz X. Mes de Abril de 2016 con recuentos. Segunda parte.

- Se pasa la respuesta o Y: Son los datos en el mes de Mayo de 2016 del producto que se va a predecir con los cambios del experimento C.

Datos del producto en el mes actual		
Nº	ncodpers	ind_hip_fin_ult1
1	15889	0
2	15890	0
...	...	...
...	...	...
922116	1548203	0

Tabla 37. Tabla de datos de entrenamiento. Matriz Y. Mes de Mayo de 2016

5.4.3 Fase de prueba

- Se construye la matriz X de prueba: Se mantienen los cambios realizados en el experimento C y se añade el recuento de la evolución de cada cliente con cada producto. El recuento es hasta Mayo de 2016.

Nº	ID	Datos de productos del mes anterior (20 columnas)				
	ncodpers	ind_cco_fin_ult1_last	...	**ind_hip_ult1_last**	...	ind_recibo_ult1_last
1	15889	1	...	**0**	...	0
2	15890	0	...	**0**	...	1
3	15892	1	...	**0**	...	1
...	...	...	...	**0**	...	...
...	1548203	1	...	**0**	...	0
...	...	...	...	0	...	...
925085	1553687	0	...	**0**	...	0

Tabla 38. Tabla de datos de prueba. Matriz X. Mes de Mayo de 2016 con recuentos. Primera parte

Nº	ID	Datos del recuento (80 columnas)					
	ncodpers	...	count_00_cco	count_01_cco	count_10_cco	count_11_cco	...
1	15889	...	0	0	0	16	...
2	15890	...	16	0	0	0	...
3	15892	...	4	1	0	11	...
...	...	...	...	0	...	...	...
...	1548203	...	0	1	0	0	...
...	...	...	...	0	...	...	...
922116	1548203	...	0	0	0	0	...

Tabla 39. Tabla de datos de prueba. Matriz X. Mes de Mayo de 2016 con recuentos. Segunda parte

5.4.4 Resultados de la predicción

La solución se muestra en la siguiente tabla:

Nº	ncodpers	ind_hip_fin_ult1
1	15889	3.686434e-07
2	15890	3.896465e-05
...	...	...
925087	1553689	3.661010e-06

Tabla 40. Resultados de la predicción del producto ind_hip_fin_ult1 para el mes de Junio de 2016

5.4.5 Comentarios de los resultados

Se obtienen las predicciones de todos los productos para todos los clientes del mes de Junio de 2016 y se ordenan en orden descendente. Se escogen los 7 primeros productos.

Finalmente la matriz de recomendación queda de la siguiente manera:

Lista de los 7 productos recomendados			
15889 ind_recibo_ult1 ind_ecue_fin_ult1 ind_cno_fin_ult1 ind_reca_fin_ult1 ind_nomina_ult1	15890 ind_valo_fin_ult1 ind_cco_fin_ult1 ind_reca_fin_ult1 ind_pres_fin_ult1 ind_ctop_fin_ult1	...	1553689 ind_cco_fin_ult1 ind_ctma_fin_ult1 ind_recibo_ult1 ind_nom_pens_ult1 ind_nomina_ult1

ind_nom_pens_ult1 ind_ctop_fin_ult1	ind_fond_fin_ult1 ind_viv_fin_ult1		ind_cno_fin_ult1 ind_ecue_fin_ult1

Tabla 41. Lista de los 7 productos recomendados para cada cliente

5.4.6 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 16. **Evaluación de kaggle. MAP@7 del experimento D**

De nuevo se observa una mejoría en la puntuación con respecto al anterior experimento.
El recuento de la evolución de los clientes es una característica que aporta decisión en los modelos.
Vale la pena comentar, que añadir los recuentos en el experimento implica un aumento de 80 características a cada uno de los modelos. Son 4 recuentos para los 20 productos escogidos (Se omiten los 4 productos anteriormente eliminados de los modelos). Esto implica complejidad computacional y tiempo de resolución.

A pesar de esta nueva complejidad computacional, en los siguientes experimentos se aplicará este nuevo cambio.

5.5 Experimento E

Después de los resultados obtenidos en el experimento anterior, se propone añadir las características físicas o personales de los clientes en los modelos.
A continuación, como recordatorio, se visualizan los datos personales de uno de los clientes.

ncodpers	fecha_dato	ind_empleado	pais_residencia	sexo	age
1375586	28/01/2015	N	ES	H	35

fecha_alta	ind_nuevo	antigüedad	indrel	ult_fec_cli_1t	indrel_1mes
12/01/2015	0	6	1		1

tiprel_1mes	indresi	indtext	conyuemp	canal_entrada	indfall
A	S	N		KHL	N

tipodom	cod_prov	nomprov	ind_actividad_cliente	renta	segmento
1	29	MALAGA	1	87218.10	02 - PARTICULARES

Tabla 42. Datos personales de un cliente (Registro de Enero de 2015)

Muchos de los campos de los datos personales no son numéricos, sino que son datos categóricos.
El recomendador empleado no permite características categóricas, por lo tanto, se deben transformar los datos a numéricos para que el recomendador pueda trabajar correctamente.

5.5.1 Modelos

Para transformar los datos, en cada modelo, se reemplazará el valor categórico con la media del producto que se esté modelizando en función del valor categórico.
A continuación se muestra un ejemplo para la característica "sexo" en el modelo del producto ind_recibo_ult1. Se muestra una tabla con los datos de la variable y del producto modelizado

Nº	ncodpers	sexo	ind_hip_ult1_last
1	15889	V	0
2	15890	V	1
3	15892	H	1
4	15893	V	0
5	15894	V	1
…	…	…	…
814940	1548207	V	0

Tabla 43. Datos de la variable categórica "sexo" con el producto modelizado

Para la variable "sexo" hay 3 tipos de valores posibles:

- V: Varón
- H: Hembra
- "": Valor en blanco

Seguidamente, se contabiliza:

- El número total de varones.
- El número de varones que tienen contratado el producto ind_recibo_ult1.
- El número total de hembras.
- El número de hembras que tienen contratado el producto ind_recibo_ult1.
- El número de "blancos".
- El número de "blancos" que tienen contratado el producto.
- Y se calculan las medias.

Se muestran en la siguiente tabla:

Sexo	total	ind_recibo_ult1 = 1	media
Varones	502857	67201	0.1336384
Hembras	423801	44521	0.1050517
Blancos	5	1	0.2

Tabla 44. Cálculo de medias de la variable sexo en función del producto ind_recibo_ult1

A modo de ejemplo, se detallan los pasos realizados para el modelo del producto ind_recibo_fin_ult1.

5.5.2 Fase de entrenamiento

Se monta el modelo mediante el método xgboost (Extreme Gradient Boosting).

- Se construye la matriz X de entrenamiento: Se añaden las variables personales o físicas de los clientes con la transformación de variables categóricas a numéricas.

Nº	ID	Datos de productos del mes anterior (20 columnas)				
	ncodpers	ind_cco_fin_ult1_last	...	ind_hip_ult1_last	...	ind_recibo_ult1_last
1	15889	1	...	0	...	0
2	15893	0	...	0	...	0
3	15896	1	...	0	...	0
4	15898	0	...	0	...	0
...	...	...	...	...	...	...

Nº	ID	Datos del recuento (80 columnas)					
	ncodpers	...	count_00_cco	count_01_cco	count_10_cco	count_11_cco	...
1	15889	...	0	0	0	15	...
2	15893	...	15	0	0	0	...
3	15896	...	3	1	1	5	...
4	15898	...	9	0	0	0	...
...	...	...	...	...	...	...	...

Nº	ID	Datos personales del cliente (22 columnas)								
	ncodpers	...	sexo	age	...	antiguedad	...	canal_entrada	...	segmento
1	15889	...	0.1336384	56	...	255	...	0.1951157	...	0.3805167
2	15893	...	0.1336384	63	...	256	...	0.1951157	...	0.1498972
3	15896	...	0.1336384	50	...	256	...	0.1951157	...	0.3805167
4	15898	...	0.1050517	61	...	256	...	0.1951157	...	0.1498972
...	...	...	...	...	...	...	...	...	...	...

Tabla 45. Tabla de datos de entrenamiento. Matriz X. Datos de productos, recuento y datos personales de Abril de 2016.

- Se pasa la respuesta o Y: Son los datos en el mes de Mayo de 2016 del producto que se va a predecir.

Datos del producto en el mes actual		
Nº	ncodpers	ind_ recibo _fin_ult1
1	15889	0
2	15893	0
3	15896	0
...	...	...

Tabla 46. Tabla de datos de entrenamiento. Matriz Y. Mes de Mayo de 2016

5.5.3 Fase de prueba

- Se construye la matriz X de prueba: Se aplican los cambios de este experimento:

Nº	ID	Datos de productos del mes anterior (20 columnas)				
	ncodpers	ind_cco_fin_ult1_last	...	ind_hip_ult1_last	...	ind_recibo_ult1_last
1	15889	1	...	0	...	0
2	15893	0	...	0	...	0
3	15896	1	...	0	...	0
4	15898	0	...	0	...	0
...	...	...	...	...	...	...

Nº	ID	Datos del recuento (80 columnas)					
	ncodpers	...	count_00_cco	count_01_cco	count_10_cco	count_11_cco	...
1	15889	...	0	0	0	16	...
2	15893	...	16	0	0	0	...
3	15896	...	3	1	1	6	...
4	15898	...	10	0	0	0	...
...	...	...	...	...	...	...	...

Nº	ID	Datos personales del cliente (22 columnas)								
	ncodpers	...	sexo	age	...	antiguedad	...	canal_entrada	...	segmento
1	15889	...	0.1346640	56	...	256	...	01965866	...	0.3846945
2	15893	...	0.1346640	63	...	257	...	0.1965866	...	0.1513024
3	15896	...	0.1346640	50	...	257	...	0.1965866	...	0.3846945
4	15898	...	0.1060328	61	...	257	...	0.1965866	...	0.1513024
...	...	...	...	...	...	...	...	...	...	...

Tabla 47. Tabla de datos de entrenamiento. Matriz X. Datos de productos, recuento y datos personales de Mayo de 2016.

5.5.4 Resultados de la predicción

La solución se muestra en la siguiente tabla:

Nº	ncodpers	ind_recibo_ult1
1	15889	8.509911e -03
2	15893	5.379012e-04
...	...	...
816600	1553689	5.593390e-02

Tabla 48. Resultados de la predicción del producto ind_recibo_ult1 para el mes de Junio de 2016

5.5.5 Comentarios de los resultados

Se obtienen las predicciones de todos los productos para todos los clientes del mes de Junio de 2016 y se ordenan en orden descendente. Se escogen los 7 primeros productos.

Finalmente la matriz de recomendación queda de la siguiente manera:

Lista de los 7 productos recomendados			
15889	15890		1553689
ind_recibo_ult1	ind_valo_fin_ult1		ind_cco_fin_ult1
ind_ecue_fin_ult1	ind_reca_fin_ult1		ind_recibo_ult1
ind_nomina_ult1	ind_pres_fin_ult1		ind_ctma_fin_ult1
ind_ctop_fin_ult1	ind_fond_fin_ult1	...	ind_nomina_ult1
ind_nom_pens_ult1	ind_cco_fin_ult1		ind_nom_pens_ult1
ind_cno_fin_ult1	ind_ctop_fin_ult1		ind_cno_fin_ult1
ind_reca_fin_ult1	ind_dela_fin_ult1		ind_ecue_fin_ult1

Tabla 49. Lista de los 7 productos recomendados para cada cliente

5.5.6 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 17. **Evaluación de kaggle. MAP@7 del experimento E**

Se aumenta ligeramente la nota del MAP@7 incluyendo las características personales de los clientes, por lo tanto se mantendrán estos cambios en los experimentos futuros.

Cada nuevo cambio introducido en los experimentos provoca un aumento de la nota, sin embargo también provoca un aumento de la complejidad computacional debido al incremento de las matrices de los modelos.

5.6 Experimento F

Hasta este momento, el planteamiento del problema ha sido emplear los datos del mes anterior para predecir las contrataciones del mes actual. En todos los experimentos se han utilizado los datos del mes de Mayo de 2016 para predecir las contrataciones del mes de Junio de 2016.

En este nuevo experimento, se pretende predecir las contrataciones del mes de Junio de 2016 empleando los datos del mes de Abril de 2016.

En el modelo, la fase de entrenamiento y la fase de prueba son exactamente iguales que en el experimento anterior.

5.6.1 Resultados de la predicción y comentarios

Se obtienen las predicciones de todos los productos para todos los clientes del mes de Junio de 2016 y se ordenan en orden descendente. Se escogen los 7 primeros productos.

Finalmente la matriz de recomendación queda de la siguiente manera:

Lista de los 7 productos recomendados			
15889	15890		1553689
ind_recibo_ult1	ind_cco_fin_ult1		ind_cco_fin_ult1
ind_ecue_fin_ult1	ind_valo_fin_ult1		ind_recibo_ult1
ind_nomina_ult1	ind_reca_fin_ult1		ind_ctma_fin_ult1
ind_reca_fin_ult1	ind_ctop_fin_ult1	...	ind_nom_pens_ult1
ind_ctop_fin_ult1	ind_cder_fin_ult1		ind_nomina_ult1
ind_nom_pens_ult1	ind_fond_fin_ult1		ind_cno_fin_ult1
ind_cno_fin_ult1	ind_hip_fin_ult1		ind_ecue_fin_ult1

Tabla 50. Lista de los 7 productos recomendados para cada cliente.

5.6.2 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

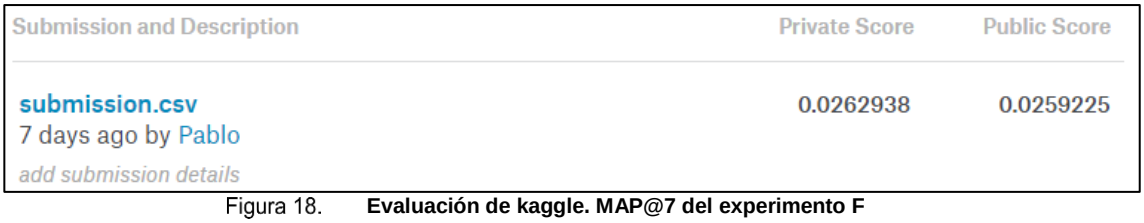


Figura 18. **Evaluación de kaggle. MAP@7 del experimento F**

Se observa que la puntuación obtenida en este nuevo experimento, utilizando los datos del mes de Abril de 2016 es ligeramente superior a la puntuación obtenida en el experimento E.

Este nuevo resultado, aporta una información valiosa. El mes de Abril de 2016 es mejor predictor que el mes de Mayo de 2016 para realizar las predicciones de Junio de 2016.

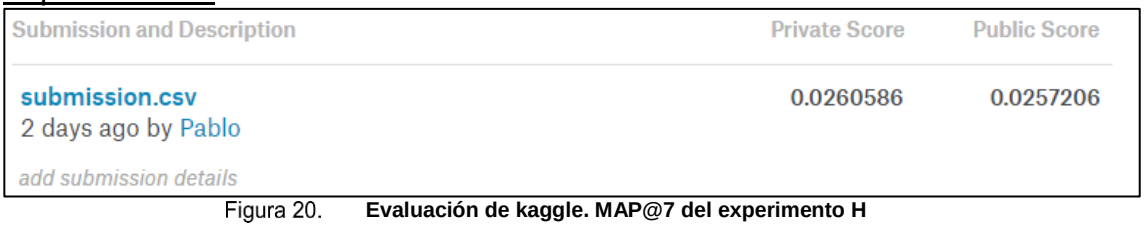
5.7 Experimentos G, H, I, J

A continuación se realiza la misma práctica que en el experimento anterior para los meses de Marzo (Experimento G), Febrero (Experimento H), Enero (Experimento I) y Diciembre (Experimento J).
Los resultados obtenidos son los siguientes:

5.7.1 Experimento G

Figura 19. **Evaluación de kaggle. MAP@7 del experimento G**

5.7.2 Experimento H



Submission and Description	Private Score	Public Score
submission.csv 2 days ago by Pablo *add submission details*	0.0260586	0.0257206

Figura 20. **Evaluación de kaggle. MAP@7 del experimento H**

5.7.3 Experimento I

Submission and Description	Private Score	Public Score
submission.csv 14 hours ago by Pablo *add submission details*	0.0257223	0.0254735

Figura 21. **Evaluación de kaggle. MAP@7 del experimento I**

5.7.4 Experimento J

Submission and Description	Private Score	Public Score
submission.csv a day ago by Pablo *add submission details*	0.0287602	0.0285665

Figura 22. **Evaluación de kaggle. MAP@7 del experimento J**

Se observa que los experimentos G, H, I y J no superan la puntuación obtenida en el experimento F.
Sin embargo, el experimento J (Diciembre de 2015) es notablemente superior.

5.7.5 Análisis y estudio de las tendencias de compra

Es sorprendente que el mes de Diciembre de 2015 sea el mes que mejores puntuaciones esté obteniendo para predecir las contrataciones de Junio de 2016.
A priori, parece raro, que los datos de 6 meses atrás sean los que aporten mayor precisión, ya que lo lógico sería que fueran los meses más cercanos a Junio de 2016.
¿Por qué está sucediendo esto?
Para poder dar una explicación, vale la pena centrarse en el gráfico de contrataciones comentado en la introducción del trabajo.

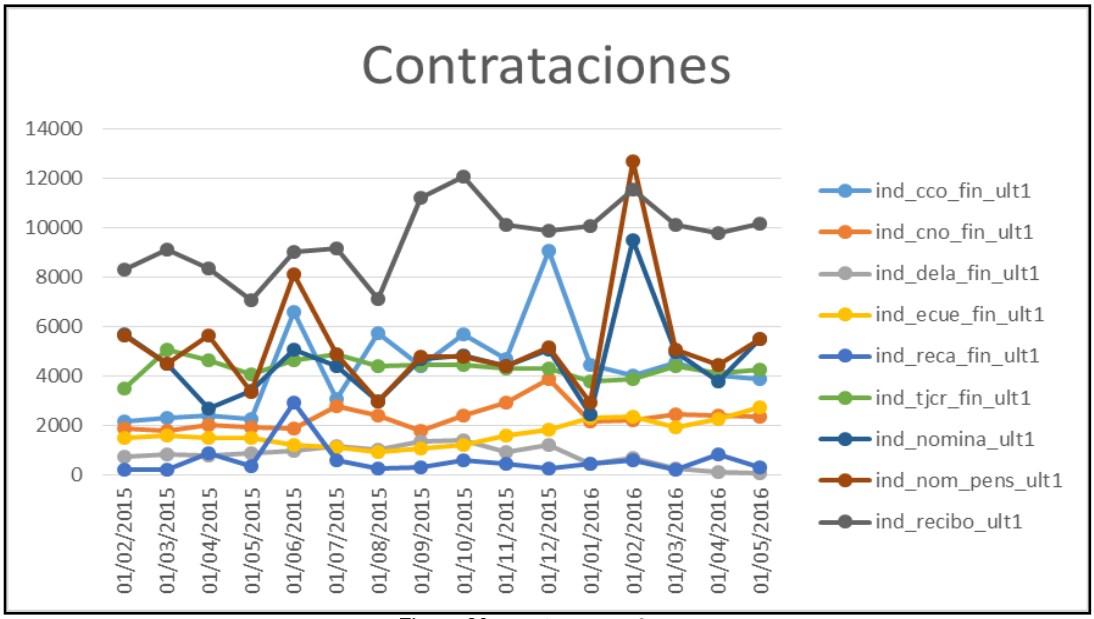


Figura 23. **Contrataciones**

¿Qué relación pueden tener las contrataciones del mes de Diciembre con las de Junio?
Se puede observar que, el único comportamiento anómalo en ambos meses es el del producto ind_cco_fin_ult. Para los dos meses, hay un pico contrastado de las contrataciones. Esto hace pensar que la tendencia de compra de los clientes pueda ser periódica en los meses de Junio y Diciembre, y por lo tanto, este suceso podría repetirse en Junio de 2016.

A continuación se detallan las contrataciones de este producto:

- Desde Febrero de 2015 hasta Mayo de 2015, las contrataciones se mantienen constantes en un valor de 2000 aproximadamente.
- En Junio de 2015, hay un cambio brusco en la curva que supera las 6000 contrataciones.
- En Julio de 2015, las contrataciones bajan a un valor aproximadamente de 3000.
- Desde Agosto hasta Noviembre de 2015 la curva oscila alrededor de las 5000 contrataciones.
- En Diciembre de 2015 vuelve a haber un incremento de las contrataciones. En concreto se llega hasta las 9000.
- Desde Enero de 2016 hasta Mayo de 2016 la tendencia vuelve a estabilizarse a valores de 4000 contrataciones.

Seguidamente, se comparan las contrataciones del mes de Diciembre de 2015 con las contrataciones del mes de Abril de 2016 (Mes que hasta el momento ha obtenido mejores puntuaciones. Experimento F). Se omiten los productos con pocas contrataciones.

Producto	28/12/2015	28/04/2016
ind_cco_fin_ult1	9077	4027
ind_cno_fin_ult1	3869	2381
ind_dela_fin_ult1	1199	99

ind_ecue_fin_ult1	1816	2249
ind_reca_fin_ult1	235	807
ind_tjcr_fin_ult1	4295	4100
ind_nomina_ult1	5071	3790
ind_nom_pens_ult1	5148	4424
ind_recibo_ult1	9888	9799

Tabla 51. Contrataciones

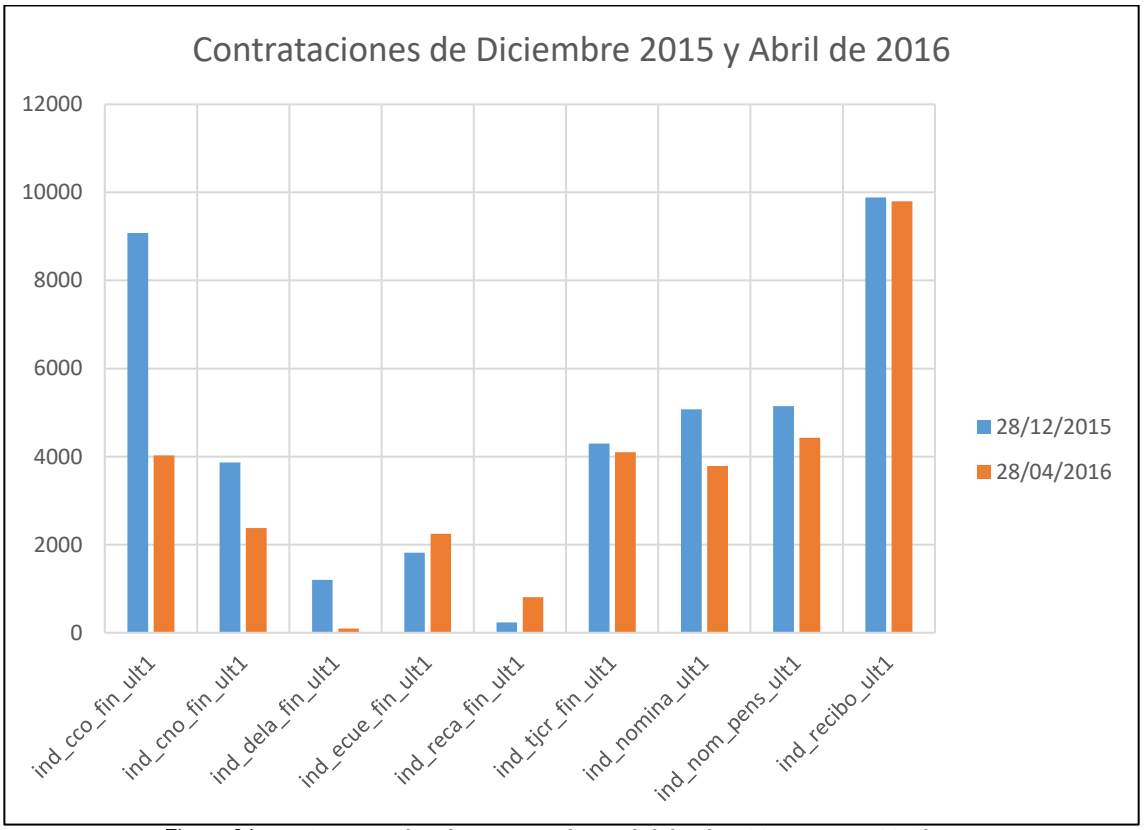


Figura 24. **Comparativa de contrataciones (Diciembre 2015 y Mayo 2016)**

Se observa que las contrataciones en Diciembre de 2015 y en Abril de 2016 son bastante similares para todos los productos excepto para el producto ind_cco_fin_ult1.
Por lo tanto, tiene lógica, que la puntuación del MAP@7 empleando los datos de Diciembre de 2015, sea superior a la de Abril de 2016.

Adicionalmente, se observa que en el mes de Junio de 2015, también hay picos de contrataciones para los productos ind_nom_pens_ult1 e ind_reca_fin_ult1.

A diferencia del producto ind_cco_fin_ult1, el producto ind_nom_pens_ult1 tiene otro pico de contrataciones en el mes de Febrero de 2016, pero este comportamiento no sería periódico ya que en Febrero de 2015 no se observa ningún pico.

Por último, el producto ind_reca_fin_ult1, no tiene otro mes en el que se disparan considerablemente las contrataciones por lo que, en este caso la tendencia de compra podría ser periódica solo en los meses de Junio.

En los siguientes experimentos, se procederá a incluir los comportamientos periódicos de estos productos para comprobar que influencia tienen en las predicciones de Junio de 2016.

Se realizarán:

- Experimento K: El modelo para predecir las contrataciones del producto **ind_cco_fin_ult1** se realizará empleando los datos de **Diciembre de 2015.**
- Experimento L: El modelo para predecir las contrataciones del producto **ind_nom_pens_ult1** se realizará empleando los datos de **Febrero de 2016.**
- Experimento M: Se realizará el mismo experimento que los realizados para los meses anteriores (Experimentos E-J) pero para el mes de **Junio de 2015**.
- Experimento N: El modelo para predecir las contrataciones del producto **ind_reca_fin_ult1** se realizará empleando los datos de **Junio de 2015.**

5.8 Experimento K

Hasta el momento, las mejores predicciones realizadas han sido empleando los meses de Abril de 2016 y de Diciembre de 2015.
El comportamiento de compra de los clientes, parece ser un factor muy importante tal y como se ha observado en el capítulo anterior.

5.8.1 Modelos

Por lo tanto, en este nuevo experimento, se pretende compaginar ambas predicciones. Se tendrán en cuenta las compras de los clientes para el producto ind_cco_fin_ult en el mes de Diciembre de 2015.

Como ya se sabe, cada producto debe ser modelizado por separado. En este ejercicio, el modelo del producto ind_cco_fin_ult1 se utilizarán los datos de Diciembre de 2015 y para el resto de productos se emplearán los datos de Abril de 2016. (Iguales que los del Experimento F)

5.8.2 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 25. **Evaluación de kaggle. MAP@7 del experimento K**

El resultado obtenido es satisfactorio. Se ha aumentado la puntuación obtenida en el experimento J. Por lo tanto, se confirma que los comportamientos irregulares como los del producto ind_cco_fin_ult1 se deben tener en cuenta en las predicciones.

5.9 Experimento L

En este nuevo experimento, se pretende realizar la misma prueba que el experimento anterior para el producto ind_nom_pens_ult1.

Este producto tiene el siguiente comportamiento:

- Desde Febrero de 2015 hasta Mayo de 2015, las contrataciones se tienen una tendencia oscilante entre el rango de 6000 a 3000 compras.
- En Junio de 2015, hay un cambio brusco en la curva que supera las 8000 contrataciones.
- En Julio de 2015, las contrataciones bajan a 5000. (Rango del primer tramo)
- En Agosto las contrataciones bajan en picado hasta las 3000.
- Desde Septiembre hasta Diciembre de 2015 la curva se mantiene en valores constantes, alrededor de las 4000 contrataciones.
- En Enero de 2016, vuelve a haber un descenso notable. Las contrataciones bajan hasta las 3000 unidades.
- En Febrero de 2016, las contrataciones tienen un aumento muy significativo. Se alcanza el máximo valor. El número de contrataciones supera las 12000 unidades.
- Desde Marzo hasta Mayo de 2016 las contrataciones se mantienen constantes en un rango de 4000-5000 unidades.

Como se puede observar, el comportamiento de este producto es similar al de ind_cco_fin_ult1. Tiene dos picos muy contrastados. Uno en Junio de 2015 y otro en Febrero de 2016.
La principal diferencia es que los picos de este producto no parecen ser periódicos, ya que en Febrero de 2015 no hubo un incremento en las contrataciones.
Por lo tanto, es probable que en este caso, predecir las contrataciones de Junio de 2016 empleando los datos de Febrero de 2016 no dé resultados tan contrastados como los anteriores.

5.9.1 Modelos

De la misma manera que en el Experimento K. Para el modelo del producto ind_cco_fin_ult1 se utilizarán los datos de Diciembre de 2015, para el producto ind_nom_pens_ult1 se utilizarán los datos de Febrero de 2016 y para el resto de productos se emplearán los datos de Abril de 2016.

5.9.2 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 26. **Evaluación de kaggle. MAP@7 del experimento L**

En este caso la puntuación obtenida no supera la del Experimento K. Esto implica que emplear el mes de Febrero de 2016 para el modelo del producto ind_nom_pens_ult1 no es un indicativo notable en las contrataciones. El hecho de que la curva de contrataciones no tenga ese carácter periódico, resulta ser un factor destacado.

5.10 Experimento M

Tal y como se ha observado en estos últimos experimentos, el mes de Junio de 2015 parece ser un mes clave ya que las contrataciones de ciertos productos aumentan considerablemente.
En este nuevo experimento se procede a predecir las contrataciones de Junio de 2016 teniendo en cuenta tan solo los datos del mes de Junio de 2015.

5.10.1 Modelos

Los modelos son exactamente iguales que los realizados en los experimentos E-J utilizando los datos de Junio de 2015.

5.10.2 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 27. **Evaluación de kaggle. MAP@7 del experimento M**

De la misma manera que sucedió con las predicciones de Diciembre de 2015 (Experimento J), la nota obtenida en este experimento es superior a la obtenida en el experimento F (Experimento que daba mejor puntuación hasta el momento).
Este resultado hace pensar que la hipótesis, de que el comportamiento de compra del producto ind_reca_fin_ult1 pueda ser periódico en los meses de Junio, gana enteros.

5.11 Experimento N

De la misma manera que en el experimento K, en esta nueva prueba se pretende añadir el comportamiento del producto ind_reca_fin_ult1 en Junio de 2015 en las predicciones.

5.11.1 Modelos

En este ejercicio:

- El modelo del producto ind_cco_fin_ult1 se realiza empleando los datos de Diciembre de 2015.
- El modelo del producto ind_reca_fin_ult1 se realiza utilizando los datos de Junio de 2016.
- El resto de modelos se utilizarán los datos de Abril de 2016.

5.11.2 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Submission and Description	Private Score	Public Score
submission.csv a day ago by Pablo *add submission details*	0.0306401	0.0303799

Figura 28. **Evaluación de kaggle. MAP@7 del experimento N**

Se obtiene la mejor puntuación hasta el momento. Utilizar los datos de Diciembre de 2015 para el modelo del producto ind_cco_fin_ult1 y los datos de Junio de 2015 para el modelo del producto ind_reca_fin_ult1 ha resultado ser exitoso.
Las tendencias de compra de los clientes son importantes. Es decir, hay productos que suelen tener más aceptación en ciertos meses. Es un factor importante que no se puede pasar por alto.

Por lo tanto, en este último experimento se predicho las contrataciones del mes de Junio de 2016 con:

- Modelo del producto ind_cco_fin_ult1: Datos de Diciembre de 2015
- Modelo del producto ind_reca_fin_ult1: Datos de Junio de 2015
- Resto de productos: Datos de Abril de 2016

Como ya se ha visto, los datos de Abril de 2016, son los más idóneos para predecir las contrataciones de Junio de 2016. Sin embargo, toda la información que se tiene de los meses anteriores no se está utilizando.

En la siguiente prueba se pretenden usar estos meses para realizar las predicciones. La idea será hacer una media de las probabilidades de las contrataciones de cada uno de los clientes en los meses de Mayo de 2016 hasta Enero de 2016.

5.12 Experimento final

Se realiza la misma prueba que el experimento N, para los meses de Mayo de 2016 hasta Enero de 2016. En todos ellos, para las predicciones del producto ind_cco_fin_ult1 se emplearán los datos de Diciembre de 2015 y para las predicciones del producto ind_reca_fin_ult1 se usarán los datos de Junio de 2015. Para el resto de productos:

- Experimento O: Datos de Mayo de 2016
- Experimento P: Datos de Marzo de 2016
- Experimento Q: Datos de Febrero de 2016
- Experimento R: Datos de Enero de 2016

Como ya se ha visto en las pruebas anteriores (Experimentos E-J), la mejor puntuación es la que emplea los datos de Abril de 2016. Por lo tanto, los resultados de los experimentos O-R no se entregan en la plataforma Kaggle, ya que la puntuación más alta es la del Experimento N.

Sin embargo, se calculará la media de las probabilidades de contratación de cada cliente para cada producto de los meses de Enero hasta Mayo.

Por ejemplo, el cliente 15889:

ncopders	Producto	Mayo 2016	Abril 2016	Marzo 2016	Febrero 2016	Enero 2016
15889	ind_ahor_fin_ult1	1,00E-10	1,00E-10	1,00E-10	1,00E-10	1,00E-10
15889	ind_aval_fin_ult1	1,00E-10	1,00E-10	1,00E-10	1,00E-10	1,00E-10
15889	ind_cder_fin_ult1	9,49E-08	1,27E-04	3,01E-06	4,86E-06	3,65E-06
15889	ind_cno_fin_ult1	1,05E-03	3,04E-04	1,45E-03	6,22E-04	4,71E-04
15889	ind_ctju_fin_ult1	6,32E-06	1,54E-06	1,25E-06	1,25E-06	9,39E-07
15889	ind_ctma_fin_ult1	2,45E-05	2,60E-06	2,87E-06	2,65E-06	3,97E-06
15889	ind_ctop_fin_ult1	1,34E-03	3,70E-04	3,90E-04	6,57E-05	5,47E-04
15889	ind_deco_fin_ult1	1,00E-10	1,00E-10	1,00E-10	1,00E-10	1,00E-10
15889	ind_deme_fin_ult1	1,00E-10	1,00E-10	1,00E-10	1,00E-10	1,00E-10
15889	ind_dela_fin_ult1	3,50E-05	1,81E-04	1,41E-04	1,21E-04	8,11E-05
15889	ind_ecue_fin_ult1	7,19E-03	6,95E-03	6,17E-03	5,50E-03	2,94E-03
15889	ind_fond_fin_ult1	5,80E-05	1,58E-04	3,88E-05	5,18E-04	2,05E-05
15889	ind_hip_fin_ult1	2,23E-06	6,96E-06	2,19E-06	6,88E-07	2,21E-06
15889	ind_plan_fin_ult1	1,40E-05	4,89E-08	3,24E-06	9,32E-06	1,95E-04
15889	ind_pres_fin_ult1	1,21E-04	6,34E-06	2,14E-05	6,60E-07	4,40E-05
15889	ind_reca_fin_ult1	6,45E-03	6,45E-03	6,45E-03	6,45E-03	6,45E-03
15889	ind_viv_fin_ult1	5,28E-08	2,96E-05	1,51E-05	2,19E-06	2,21E-06
15889	ind_nomina_ult1	1,51E-03	4,55E-04	1,52E-03	7,61E-04	2,48E-04
15889	ind_nom_pens_ult1	1,29E-03	3,59E-04	1,03E-03	1,20E-03	2,34E-04
15889	ind_recibo_ult1	8,51E-03	1,09E-02	2,81E-03	7,16E-03	1,67E-02

Tabla 52. Probabilidades de contratación para el cliente 15889 para los meses de Mayo hasta Enero de 2016

Y se calcula la media de las probabilidades de cada producto para cada cliente. Es decir, $\frac{1}{5}\sum_{i=1}^{5} pr_i$

En el ejemplo del cliente 15889 sería:

ncopders	Producto	promedio
15889	ind_ahor_fin_ult1	1,00E-10
15889	ind_aval_fin_ult1	1,00E-10
15889	ind_cder_fin_ult1	2,77E-05
15889	ind_cno_fin_ult1	7,81E-04
15889	ind_ctju_fin_ult1	2,26E-06
15889	ind_ctma_fin_ult1	7,32E-06
15889	ind_ctop_fin_ult1	5,42E-04
15889	ind_deco_fin_ult1	1,00E-10
15889	ind_deme_fin_ult1	1,00E-10
15889	ind_dela_fin_ult1	1,12E-04
15889	ind_ecue_fin_ult1	5,75E-03

15889	ind_fond_fin_ult1	1,59E-04
15889	ind_hip_fin_ult1	2,86E-06
15889	ind_plan_fin_ult1	4,43E-05
15889	ind_pres_fin_ult1	3,86E-05
15889	ind_reca_fin_ult1	6,45E-03
15889	ind_viv_fin_ult1	9,83E-06
15889	ind_nomina_ult1	9,00E-04
15889	ind_nom_pens_ult1	8,22E-04
15889	ind_recibo_ult1	9,21E-03

Tabla 53. Probabilidades medias calculadas para el cliente 15889

Finalmente, una vez obtenidas las probabilidades medias de cada uno de los clientes para todos los productos, se procede a escoger los 7 productos con más probabilidad para cada uno de los clientes y se forma la lista que se entrega en kaggle.

5.12.1 Entrega de los resultados en kaggle y obtención de métrica

Se hace entrega de la predicción y el resultado calculado por kaggle es:

Figura 29. **Evaluación de kaggle. MAP@7 del experimento final**

La puntuación obtenida es mejor que la obtenida en el experimento N. Además, es una puntuación que se acerca bastante a los resultados de los primeros clasificados en la plataforma kaggle.
Con esta puntuación se hubiera alcanzado la posición 49 en la clasificación final de la competición.

Llegados a este punto, se decide no se seguir haciendo más experimentos ya que las puntuaciones obtenidas son satisfactorias.

5.13 Evolución de los experimentos

Por último, se detallará gráficamente un esquema de la evolución del proyecto, con todos los pasos y las puntuaciones obtenidas en los experimentos.

De esta manera se podrán observar los cambios y sus respectivas mejoras con respecto a los experimentos anteriores y se podrá hacer un seguimiento más exhaustivo del problema

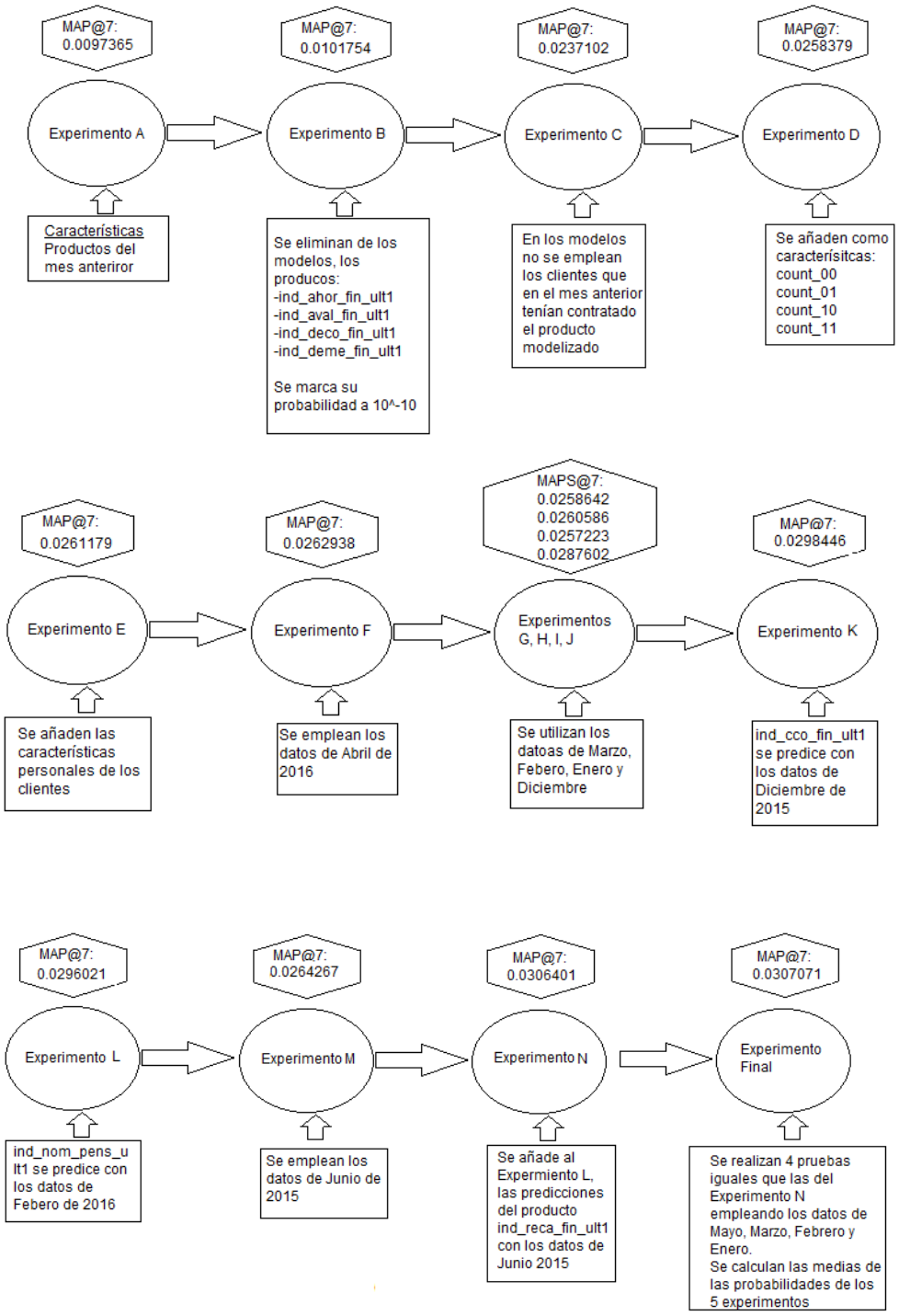


Figura 30. **Gráfico evolutivo del problema**

Vale la pena comentar, que los cambios propuestos han aumentado considerablemente la puntuación MAP@7. Sin embargo, también ha repercutido en un aumento de los costes computacionales debido al incremento de características.

6 Conclusiones

6.1 Resumen

En este proyecto se ha realizado el problema “Santander Product Recomendation” de la plataforma Kaggle.
El trabajo ha consistido en predecir los siete productos del Banco de Santander, con más probabilidad de ser contratados, para cada uno de sus clientes.

Como tareas iniciales del proyecto, se ha realizado una revisión de la literatura relacionada con el tema, se han consultado publicaciones en el foro de la plataforma kaggle, se ha hecho un estudio de las diferentes técnicas predictivas más importantes y se han descrito las herramientas y el lenguaje de programación usado.

Una vez realizadas estas tareas, se ha obtenido toda la información necesaria para definir y entender el problema completo. Además se han inspeccionado los datos (minería de datos) y se han realizado tareas de limpieza de los mismos.

A partir de este punto, se han realizado un total de quince experimentos.
Para cada uno de los experimentos se obtienen unos resultados que son entregados en la plataforma Kaggle. La puntuación registrada por Kaggle ha servido como indicativo para evaluar los experimentos.
Cada experimento introduce un cambio significativo con respecto al anterior, de tal forma que si la puntuación obtenida es superior a la obtenida anteriormente el cambio se mantendrá en los siguientes experimentos.

Durante la fase experimental se han observado los siguientes puntos relevantes:

- Se realizan un total de 24 modelos (1 por cada producto del banco).
- Se empieza el problema empleando únicamente los datos de Mayo de 2016.
- Los productos ind_ahor_fin_ult1, ind_aval_fin_ult, ind_deco_fin_ult e ind_deme_fin_ult no son productos de éxito en los clientes y su probabilidad de ser contratados es muy baja.
- El hecho de tener contratado un producto en el mes anterior afecta considerablemente de forma negativa en las predicciones. Por lo tanto, tan solo se han usado los clientes que en el mes anterior no tenían contratado el producto modelizado.
- El mes de Abril de 2016 es mejor predictor que el de Mayo de 2016.
- Los recuentos históricos de las contrataciones y descontrataciones (count_00, count_01, count_10 y count_11) son datos importantes.
- Las contrataciones de ciertos productos en determinados meses es un factor decisivo con respecto a las predicciones.
- El producto ind_cco_fin_ult1 y el producto ind_reca_fin_ult1 tienen comportamientos de compra periódicos.

- El ensamblaje de los modelos empleando la media de los meses de Enero-Mayo de 2016 ayuda a incrementar la nota final.
- La nota final obtenida es MAP@7=0.0307071. Es una puntuación que se acerca bastante a la obtenida por los mejores clasificados de la competición. En concreto, se hubiera alcanzado la posición 49 de la clasificación final.

6.2 Seguimiento de la planificación

Durante la ejecución del proyecto se ha tenido que reajustar la planificación. Tal y como se ha descrito en la introducción del documento, varias de las tareas que inicialmente estaban organizadas durante una franja de tiempo concreta han resultado ser de carácter iterativo.

También vale la pena destacar que este tipo de proyectos se basan en gran manera en la metodología "prueba-error". Es decir, a medida que se iban realizando los experimentos se han ido descubriendo tendencias y características que han servido para tomar las decisiones adecuadas.

6.3 Trabajos futuros

Como posibles líneas de trabajo futuras para mejorar los resultados del proyecto se debería:

- Aplicar otros algoritmos predictivos como pueden ser las redes neuronales (método usado por el primer clasificado de la competición)
- Mejores técnicas de ensamblaje: Aplicar distintos modelos predictivos con diferentes técnicas y añadir pesos en función de los resultados obtenidos. Por ejemplo: usar redes neuronales, support vector machines, extreme gradient boosting y aplicar los pesos correspondientes.
- Ingeniería de características: Aunque la mayor parte del proyecto ha consistido en una exhaustiva fase de ingeniería de características se debería profundizar algo más. Por ejemplo, los datos personales o físicos de los clientes que tan solo se han transformado de categóricos a numéricos, se podrían emplear para obtener más características que ayuden a mejorar las predicciones.

6.4 Comentarios finales

Vale la pena destacar que aunque el problema resuelto se ha definido como "real" no es del todo cierto.
Las predicciones obtenidas se han ido mejorando en función de las puntuaciones obtenidas por la plataforma Kaggle.
En un problema 100% real estas puntuaciones no se tendrán, por lo que a la hora de la verdad, la complejidad es aún mayor.

7 Glosario

Big Data: Concepto que hace referencia a conjunto de datos muy grandes. A menudo se emplea también como sinónimo de datos masivos.

Data Scientist: Nuevo perfil profesional que traduce grandes volúmenes de datos en respuestas fiables.

Machine Learning: Aprendizaje automático. Es un campo de las ciencias de la computación cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

Kaggle: Plataforma web de competiciones de Machine Learning

Feature Engineering: Es lo que se le llama “Ingeniería de Caracterísitcas”. Trata de escoger las variables más adecuadas para entrenar los modelos.

MAP@7 (Mean Avarage Precision): Métrica empleada para puntuar las predicciones del ejercicio.

Python y R: Lenguajes de programación de alto nivel.

Linear Regression, Logistic Regression, Decision Trees, Support Vector Machines, Neural Networks: Modelos predictivos más importantes

Gradient Boosting: Técnica que emplea la optimización del Gradiente Descendiente con el “Boosting” (Aumentar la precisión)

Extreme Gradient Boosting: Técnica mejorada del Gradient Boosting. Es uno de los métodos que más éxitos está cosechando en las competiciones de Kaggle.

Loss function: Función que se desea minimizar

Weak learners: Modelos simples. A menudo se les llama “aprendices”

Additive Model: Modelos a ensamblar en la técnica de Gradient Boosting.

Matriz de datos inicial: Es el conjunto de datos de partida del problema.

Fase de entrenamiento: Es la fase en la que se “aporta conocimiento” a los modelos. A partir de los datos, se entrenan los modelos para posteriormente realizar las predicciones.

Fase de prueba: Es la fase predictiva. Se emplean los modelos entrenados en la fase anterior para predecir las respuestas del problema.

8 Bibliografía

[1] (Diciembre 2017)
https://www.tomsplanner.es

[2] (Diciembre 2017)
https://www.kaggle.com/c/santander-product-recommendation

[3] (Diciembre 2017)
https://www.kaggle.com/c/santander-product-recommendation/data

[4] (Diciembre 2017)
https://www.kaggle.com/c/santander-product-recommendation#evaluation

[5] (Diciembre 2017)
https://www.kaggle.com/c/santander-product-recommendation#prizes

[6] (Enero 2017)
https://blogs.deusto.es/bigdata/eligiendo-una-herramienta-de-analitica-sas-r-o-python/

[7] (Enero 2017)
https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/

[8] (Febrero 2017)
https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/

[9] (Febrero 2017)
https://www.quora.com/What-is-an-intuitive-explanation-of-Gradient-Boosting

[10] (Febero2017)
https://www.analyticsvidhya.com/blog/2015/09/complete-guide-boosting-methods/

[11](Febrero 2017)
http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

[12] (Marzo 2017)
http://cran.fhcrc.org/web/packages/xgboost/vignettes/xgboost.pdf

[13] (Marzo 2017)
https://arxiv.org/pdf/1603.02754v2.pdf

[14] (Marzo 2017)
http://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

[15] (Marzo 2017)
http://xgboost.readthedocs.io/en/latest/model.html

[16] (Marzo 2017)
http://homes.cs.washington.edu/~tqchen/2016/03/10/story-and-lessons-behind-the-evolution-of-xgboost.html

[17] (Abril 2017)
https://www.kaggle.com/c/santander-product-recommendation/leaderboard

[18] (Abril 2017)
https://www.kaggle.com/c/santander-product-recommendation/discussion/26835

[19] (Mayo 2017)
https://www.kaggle.com/c/santander-product-recommendation/discussion/26824

9 Anexos

En este apartado se adjunta el código final empleado para la resolución del problema (Experimentos E,F,G,H,I,J), el código de mezcla empleando datos de distintos meses (Experimentos N,O,P,Q,R) y el código de ensamblaje final de los modelos con la entrega final.

9.1 Código final (Experimentos E,F,G,H,I,J)

```
library(data.table)
library(dplyr)
library(xgboost)

setwd("C:/Users/Pablo/Desktop/ProyectoSantander/CodigoFinal")

# Lectura del archivo de entrenamiento
dataTrain <- fread("train_ver2.csv")

# Lectura del archivo de prueba
dataTest <- fread("test_ver2.csv")

# Comprobacion de las alertas de los valores de las variables "indrel_1mes y "conyuemp" en el archivo de entrenamiento
levels <- levels(as.factor(dataTrain[["indrel_1mes"]]))
levels
levels <- levels(as.factor(dataTrain[["conyuemp"]]))
levels

# Conversion de indrel_1mes a variable categorica.
numIdsTrain <- !dataTrain[[12]] %in% c("", "P")
dataTrain[[12]][numIdsTrain] <- as.numeric(dataTrain[[12]][numIdsTrain])
levels <- levels(as.factor(dataTrain[["indrel_1mes"]]))
levels

# Comprobacion de las alertas de los valores de las variables "indrel_1mes y "conyuemp" en el archivo de entrenamiento
levels <- levels(as.factor(dataTest[["indrel_1mes"]]))
levels
levels <- levels(as.factor(dataTest[["conyuemp"]]))
levels

# Creacion de listas de productos, fechas y clientes
# Se eliminan los productos con pocas contrataciones ("ind_ahor_fin_ult1", "ind_aval_fin_ult1","ind_deco_fin_ult1", "ind_deme_fin_ult1")
product.list <-
colnames(dataTrain)[25:ncol(dataTrain)][!colnames(dataTrain)[25:ncol(dataTrain)] %in% c("ind_ahor_fin_ult1", "ind_aval_fin_ult1",

"ind_deco_fin_ult1", "ind_deme_fin_ult1",
```

```
"ind_ahor_fin_ult1_last", "ind_aval_fin_ult1_last",

"ind_deco_fin_ult1_last", "ind_deme_fin_ult1_last")]
date.list <- c(unique(dataTrain$fecha_dato), "2016-06-28")
customer.list <- unique(dataTrain$ncodpers)

# Se guardan los datos de Junio 2015, Diciembre 2015, Enero 2016,
# Febrero 2016, Marzo 2016, Abril 2106 y Mayo 2016 en archivos separados
for(i in c(6,12:length(date.list))) {
  print(date.list[i])
  if(date.list[i] != "2016-06-28") {
    out <- dataTrain[fecha_dato==date.list[i]]
    out <- merge(out, dataTrain[fecha_dato==date.list[i-1],
c("ncodpers",product.list), with=FALSE], by="ncodpers", suffixes=c("","_last"))
    write.csv(out, paste0("train_", date.list[i], ".csv"), row.names=FALSE)
  } else {
    out <- dataTest
    out <- merge(out, dataTrain[fecha_dato==date.list[i-1],
c("ncodpers",product.list), with=FALSE], by="ncodpers", suffixes=c("","_last"))
    colnames(out)[(ncol(out)-19):ncol(out)] <- paste0(colnames(out)[(ncol(out)-
19):ncol(out)], "_last")
    write.csv(out, paste0("test_", date.list[i], ".csv"), row.names=FALSE)
  }
}

# Se guardan los recuentos de contrataciones para los meses de Junio 2015,
Diciembre 2015, Enero 2016,
# Febrero 2016, Marzo 2016, Abril 2106 y Mayo 2016 en archivos separados.
(count_00, count_01, count_10 y count_11)
for(i in c(6,12:length(date.list))) {
  print(date.list[i])
  if(date.list[i] != "2016-06-28") {
    out <- merge(dataTrain[fecha_dato==date.list[i], .(ncodpers)],
dataTrain[fecha_dato==date.list[i-1], .(ncodpers)], by="ncodpers")
  } else {
    out <- fread("test_ver2.csv", select=2)
  }
  for(product in product.list) {
    print(product)
    temp <- dataTrain[fecha_dato %in% date.list[1:(i-1)],
c("fecha_dato","ncodpers",product), with=FALSE]
    temp <- temp[order(ncodpers, fecha_dato)]
    temp$n00 <- temp$ncodpers==lag(temp$ncodpers) & lag(temp[[product]])==0
& temp[[product]]==0
    temp$n01 <- temp$ncodpers==lag(temp$ncodpers) & lag(temp[[product]])==0
& temp[[product]]==1
    temp$n10 <- temp$ncodpers==lag(temp$ncodpers) & lag(temp[[product]])==1
& temp[[product]]==0
```

```
    temp$n11 <- temp$ncodpers==lag(temp$ncodpers) & lag(temp[[product]])==1
& temp[[product]]==1
    temp[is.na(temp)] <- 0
    count <- temp[, .(sum(n00, na.rm=TRUE), sum(n01, na.rm=TRUE), sum(n10,
na.rm=TRUE), sum(n11, na.rm=TRUE)), by=ncodpers]
    colnames(count)[2:5] <- paste0(product, c("_00","_01","_10","_11"))
    out <- merge(out, count, by="ncodpers")
  }
  write.csv(out[, -1, with=FALSE], paste0("count_", date.list[i], ".csv"),
quote=FALSE, row.names=FALSE)
}

# Inicio de bucle. Experimentos de Junio 2015, Diciembre 2015, Enero 2016,
# Febrero 2016, Marzo 2016, Abril 2106 y Mayo 2016.
for(i in c(6,12:(length(date.list)-1))) {
  # Bucle de los modelos para cada uno de los productos
  for(product in product.list) {

    #Lectura de los archivos train_ y count_
    data.1 <- fread(paste0("train_", date.list[i], ".csv"))
    data.2 <- fread(paste0("count_", date.list[i], ".csv"))
    data.train <- cbind(data.1, data.2)
    data.train <- data.train[order(ncodpers)]

    #Lectura de los archivos test_ y count_
    data.1 <- fread("test_2016-06-28.csv")
    data.2 <- fread("count_2016-06-28.csv")
    data.test <- cbind(data.1, data.2)
    data.test <- data.test[order(ncodpers)]

    rm(data.1)
    rm(data.2)

    # Bucle para convertir las variables categoricas a numéricas.
    exp.var <- colnames(data.test)[-(1)]
    for(var in exp.var) {
      if(class(data.train[[var]])=="character") {
        data.temp <- data.train[, c(var, paste0(product, "_last")), with=FALSE]
        colnames(data.temp)[ncol(data.temp)] <- "target"
        target.mean <- data.temp[, .(target=mean(target)), by=var]
        data.train[[var]]          <-          target.mean$target[match(data.train[[var]],
target.mean[[var]])]

        data.temp <- data.test[, c(var, paste0(product, "_last")), with=FALSE]
        colnames(data.temp)[ncol(data.temp)] <- "target"
        target.mean <- data.temp[, .(target=mean(target)), by=var]
        data.test[[var]]          <-          target.mean$target[match(data.test[[var]],
target.mean[[var]])]
      }
    }
```

```
    rm(data.temp)
       # Restriccion para filtrar los clientes que no tienen contratado el producto a
modelizar
    data.train <- data.train[data.train[[paste0(product,"_last")]] == 0]
    data.test <- data.test[data.test[[paste0(product,"_last")]] == 0]

    # Creación de las matrices de entrenamiento y de prueba para la ejecucion de
xgb
    x.train <- data.train[!is.na(data.train[[product]]), exp.var, with=FALSE]
    x.train[] <- lapply(x.train, as.numeric)
    y.train <- data.train[!is.na(data.train[[product]]), product, with=FALSE]
    y.train <- y.train[[product]]
    dtrain <- xgb.DMatrix(data=as.matrix(x.train), label=y.train)
    rm(data.train)
    rm(x.train)
    x.test <- data.test[, exp.var, with=FALSE]
    x.test[] <- lapply(x.test, as.numeric)
    dtest <- xgb.DMatrix(data=as.matrix(x.test), label=rep(NA, nrow(data.test)))
    data.test <- data.test[, .(ncodpers)]
    rm(x.test)

    # Parametros de xgb
    nrounds <- 500
    early_stopping_round <- 50
    params <- list("eta"=0.05,
                   "max_depth"=4,
                   "min_child_weight"=1,
                   "objective"="binary:logistic",
                   "eval_metric"="auc")

    # Ejecución de entrenamiento de xgb
    set.seed(0)
    model.xgb <- xgb.train(params=params,
                           data=dtrain,
                           nrounds=nrounds,
                           watchlist=list(train=dtrain),
                           early_stopping_round=early_stopping_round,
                           print_every_n=10,
                           base_score=mean(y.train, na.rm=TRUE))

    # Ejecución de las predicciones de xgb
    result <- data.table(data.test$ncodpers, predict(model.xgb, dtest))
    colnames(result) <- c("ncodpers",product)
    result <- result[order(ncodpers)]
    write.csv(result, paste0("submission_", product, "_", date.list[i], ".csv"),
quote=FALSE, row.names=FALSE)

    rm(dtrain)
    rm(dtest)
    rm(result)
```

```
  }

  # Montaje de todas las proabilidades en un data.table
  product.list.final <- colnames(fread("train_ver2.csv", select=25:48, nrows=0))
  submission <- fread("sample_submission.csv")[, .(ncodpers)]

  result <- data.table()
  result.temp <- data.table()

  for (j in c(1:length(product.list.final))){
    if(j %in% c(1,2,10,11)) {
      temp <- submission
      temp$pr  <- 1e-10
    }
    else{
      temp  <-  fread(paste0("submission_",  product.list.final[j],  "_",  date.list[i],
".csv"))
      colnames(temp)[2] <- "pr"
      temp$product <- product.list.final[j]
      result.temp <- rbind(result.temp, temp)
    }
  }

  # data.table final y ordenación en función de la probabilidad
  result <- result.temp
  result <- result[order(ncodpers, -pr)]

  # Elección de los 7 productos con mayor probabilidad de ser contratados
  for(k in 1:7) {
    temp <- result[!duplicated(result, by="ncodpers"), .(ncodpers, product)]
    submission <- merge(submission, temp, by="ncodpers", all.x=TRUE)
    result <- result[duplicated(result, by="ncodpers"), .(ncodpers, product)]
    colnames(submission)[ncol(submission)] <- paste0("p",k)

    if(nrow(result) == 0) {
      break
    }
  }

  # Creación del archivo de entrega
  submission[is.na(submission)] <- ""
  submission$added_products <- submission[[paste0("p",1)]]
  for(z in 2:7) {
    submission$added_products       <-       paste(submission$added_products,
submission[[paste0("p",z)]])
  }
  mode <- paste0("submission_", date.list[i])
  submission <- submission[order(ncodpers)]
  file.name <- paste0(mode, ".csv")
```

```
  write.csv(submission[, .(ncodpers, added_products)], file.name, quote=FALSE,
row.names=FALSE)
}
```

9.2 Código de mezcla empleando datos de distintos meses (Experimentos N,O,P,Q,R)

```
library(data.table)

setwd("C:/Users/Pablo/Desktop/ProyectoSantander/CodigoFinal")

product.list <- colnames(fread("train_ver2.csv", select=25:48, nrows=0))
date.list <- c("2016-05-28", "2016-04-28", "2016-03-28", "2016-02-28", "2016-01-
28")

#Bucle para crear los modelos con datos de distintos meses
for(i in 1:length(date.list)) {
  submission <- fread("sample_submission.csv")[, .(ncodpers)]
  result.temp <- data.table()
  result <- data.table()
  for(j in 1:length(product.list)) {
    product <- product.list[j]
    print(product.list[j])
    if(j %in% c(1,2,10,11)) {
      temp <- submission
      temp$pr  <- 1e-10
    } else {
      if(product == "ind_cco_fin_ult1") {
        train.date <- "2015-12-28"
        print(train.date)
      } else if(product == "ind_reca_fin_ult1") {
        train.date <- "2015-06-28"
        print(train.date)
      } else {
        train.date <- date.list[i]
        print(train.date)
      }
      temp <- fread(paste0("submission_", product, "_", train.date, ".csv"))
      colnames(temp)[2] <- "pr"
      temp$product <- product
    }
    temp$product <- product
    result.temp <- rbind(result.temp, temp)
    result <- result.temp
  }
  result <- result[order(ncodpers,-pr)]

  # Elección de los 7 productos con mayor probabilidad de ser contratados
  for(k in 1:7) {
    print(k)
```

```
    temp <- result[!duplicated(result, by="ncodpers"), .(ncodpers, product)]
    submission <- merge(submission, temp, by="ncodpers", all.x=TRUE)
    result <- result[duplicated(result, by="ncodpers"), .(ncodpers, product)]
    colnames(submission)[ncol(submission)] <- paste0("p",k)
    if(nrow(result) == 0) {
      break
    }
  }
  # Creación del archivo de entrega
  submission[is.na(submission)] <- ""
  submission$added_products <- submission[[paste0("p",1)]]
  for(z in 2:7) {
    submission$added_products <- paste(submission$added_products,
submission[[paste0("p",z)]])
  }
  submission <- submission[order(ncodpers)]
  file.name <- paste0("submission_", date.list[i],"_mix.csv")
  write.csv(submission[, .(ncodpers, added_products)], file.name, quote=FALSE,
row.names=FALSE)
}}
```

9.3 Código de ensamblaje de modelos (Experimento final)

```
library(data.table)

setwd("C:/Users/Pablo/Desktop/ProyectoSantander/CodigoFinal")

product.list <- colnames(fread("train_ver2.csv", select=25:48, nrows=0))
submission <- fread("sample_submission.csv")[, .(ncodpers)]
date.list <- c("2016-05-28", "2016-04-28", "2016-03-28", "2016-02-28", "2016-01-
28")

#Bucle para crear los modelos final
result.temp <- data.table()
result <- data.table()
for(i in 1:length(date.list)) {
  print(date.list[i])
  result.temp <- data.table()
  for(j in 1:length(product.list)) {
    product <- product.list[j]
    print(product.list[j])
    if(j %in% c(1,2,10,11)) {
      temp <- submission
      temp$pr  <- 1e-10
    } else {
      if(product == "ind_cco_fin_ult1") {
        train.date <- "2015-12-28"
        print(train.date)
      } else if(product == "ind_reca_fin_ult1") {
        train.date <- "2015-06-28"
```

```
      print(train.date)
    } else {
      train.date <- date.list[i]
      print(train.date)
    }
    temp <- fread(paste0("submission_", product, "_", train.date, ".csv"))
    colnames(temp)[2] <- "pr"
    temp$product <- product
  }
  temp$product <- product
  result.temp <- rbind(result.temp, temp)
 }
 result <- rbind(result, result.temp)
}

result <- result[order(ncodpers,product)]
pred.sum <- result[, .(med=sum(pr)/5), by=.(ncodpers, product)]
result <- pred.sum[order(ncodpers, -med)]

# Elección de los 7 productos con mayor probabilidad de ser contratados
for(i in 1:7) {
 print(i)
 temp <- result[!duplicated(result, by="ncodpers"), .(ncodpers, product)]
 submission <- merge(submission, temp, by="ncodpers", all.x=TRUE)
 result <- result[duplicated(result, by="ncodpers"), .(ncodpers, product)]
 colnames(submission)[ncol(submission)] <- paste0("p",i)
 if(nrow(result) == 0) {
  break
 }
}
# Creación del archivo final de entrega
submission[is.na(submission)] <- ""
submission$added_products <- submission[[paste0("p",1)]]
for(i in 2:7) {
 submission$added_products <- paste(submission$added_products,
submission[[paste0("p",i)]])
}

submission <- submission[order(ncodpers)]
file.name <- paste0("submission_final.csv")
write.csv(submission[, .(ncodpers, added_products)], file.name, quote=FALSE,
row.names=FALSE)
```