CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Matemáticas e Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Elaboración y evaluación de modelos predictivos de negocio

Autor: Pablo Gaspar Peral

Director: Nelson Medinilla Martínez

MADRID, JUNIO 2015

Resumen

Este trabajo de fin de grado es un estudio de un caso real de predicción sobre grandes cantidades de datos. Se ha llevado a cabo a partir de los datos recogidos en un sitio web de competiciones de data mining y se han elaborado dos modelos predictivos mediante diferentes técnicas de modelización para tratar de predecir si un hecho ocurre o no ocurre.

Este trabajo, sin dejar de lado el desarrollo del problema, se centra en cómo se evalúa un modelo predictivo. Por ello, se ha construido una escala propia de clasificación de modelos en función a su poder de predicción y se ha realizado un esfuerzo por definir formalmente las medidas de validación de modelos que se iban a utilizar.

Cada problema concreto de datos supone un contexto diferente. Por tanto, todo el trabajo aquí desarrollado debe interpretarse en el contexto planteado y ante otros datos diferentes, no se debe tomar como verdad absoluta lo que en este documento se plantea.

Este trabajo nace en un entorno empresarial de alto nivel y trata de ser un apoyo en el campo de data analytics, que hoy en día se encuentra en plena expansión.

Palabras clave: Modelos predictivos, conjuntos de datos, XGBoost, regresión logística, validación, curva ROC, Lift

Abstract

This final bachelor work is a study of a real case of prediction over huge amounts of data. This work has been done using datasets that were downloaded from a website about data mining competitions and two predictive models have been developed using different modeling techniques to try to predict if something occurs or not.

This work, without leaving apart the develop of the problem, has an important component related to the evaluation of predictive models. Thus, an own scale has been built in order to classify predictive models regarding to its predictive power and it has made an effort to formaly define the validation measures that were going to be used.

Each data science problem entail a different context. Therefore, this work must be interpreted in the context raised and with different datasets, the things written in this document, should not be taken as an absolute truth.

This work grows in a high-level business environment and tries to be helpful in the data analytics field, that is rising nowadays.

Palabras clave: Predictive models, datasets, XGBoost, logistic regression, validation, ROC curve, Lift

Índice

Índice de figuras

Índice de cuadros

1. Introducción

La minería de datos ha dado lugar a una paulatina sustitución del análisis de datos dirigido a la verificación por un enfoque de análisis de datos dirigido al descubrimiento del conocimiento. La principal diferencia entre ambos se encuentra en que en el último se descubre información sin necesidad de formular previamente una hipótesis. La aplicación automatizada de algoritmos de minería de datos permite detectar diversos patrones no obvios en los datos, razón por la cual esta técnica es mucho más eficiente que el análisis dirigido a la verificación cuando se intenta explorar datos procedentes de repositorios de gran tamaño y complejidad elevada.

Dichas técnicas emergentes se encuentran en continua evolución como resultado de la colaboración entre campos de investigación tales como bases de datos, reconocimiento de patrones, inteligencia artificial, sistemas expertos, estadística, visualización, recuperación de información, y computación de altas prestaciones.

1.1. Problema general y solución planteada

Un modelo predictivo mide la relación entre la respuesta específica de un sujeto ante un hecho a predecir y una o más características del mismo sujeto. El objetivo de un modelo predictivo es evaluar la probabilidad de que un sujeto similar tenga el mismo rendimiento en una muestra diferente. Un ejemplo podría ser un modelo que midiera si un paciente es propenso a tener cierta enfermedad en función a ciertos parámetros médicos.

En general, la elaboración de modelos predictivos no es una tarea sencilla. Aunque cada vez existen más herramientas y algoritmos que permiten su desarrollo, es importante tener los recursos suficientes para tratar grandes cantidades de datos, es necesario comprender las particularidades del conjunto de datos que se posee y también hay que comprender en profundidad qué medidas se utilizan para evaluar modelos predictivos.

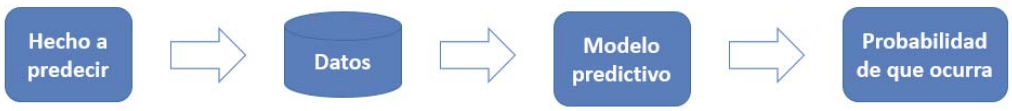

Figura 1: Realización de un modelo

En la figura 1 puede observarse el esquema general del problema planteado en este trabajo. En primer lugar, se desea predecir la probabilidad de que ocurra un hecho concreto y se suele disponer de una gran cantidad de datos. Sobre esos datos se desarrolla un modelo predictivo. Más tarde, sobre nuevos datos se aplica el modelo y se obtiene una probabilidad de que ocurra el hecho a predecir para cada nuevo registro.

La solución que se va a dar en este trabajo al problema planteado, consiste en la elaboración de dos modelos predictivos sobre ciertos datos mediante dos técnicas diferentes de modelización para después evaluar y comparar los resultados de ambas técnicas para poder determinar cual es la solución más apropiada.

1.2. Objetivos y organización del trabajo

Los objetivos de este trabajo han sido los siguientes:

- Desarrollar dos modelos predictivos de negocio aplicando dos técnicas diferentes para realizar predicciones sobre una variable binaria.
- Estudiar y definir formalmente las diferentes medidas de validación existentes, puesto que en la bibliografía consultada no se encuentran definiciones formales.
- Analizar los resultados de los modelos predictivos al aplicarlos a una parte de la población y comparar las medidas de validación para el problema en concreto.

Estos objetivos convierten el trabajo en un ejercicio bastante completo e interesante, donde va a resultar necesario informarse adecuadamente y desarrollar soluciones apropiadas para el problema planteado trabajando con grandes cantidades de datos reales en un entorno con recursos limitados.

Para el desarrollo de este trabajo, el esquema general desarrollado es el siguiente:

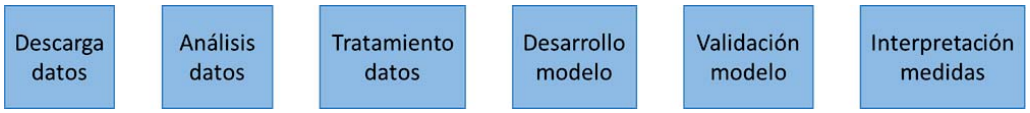

Figura 2: Organización del trabajo

Tal y como se observa en la figura 2, en primer lugar se han descargado diversos conjuntos de datos sobre los que se iban a desarrollar los modelos predictivos, después se ha realizado un análisis exhaustivo de los mismos y se han pretratado en función a los resultados del análisis obtenido. Después, se ha elaborado el modelo y se han obtenido las diferentes medidas de validación para poder interpretarlas y compararlas. Cabe destacar que todo este proceso ha sido altamente iterativo, sobre todo, a la hora de probar diferentes versiones de los modelos predictivos.

Para desarrollar el trabajo, se ha contado con un equipo portátil con un sistema operativo Windows 7, un procesador Intel Core i7 de 2.2GHz y con una memoria RAM de 6GB.

1.3. Estructura de esta memoria

La memoria se estructura como sigue, en la sección 2 se definen formalmente y en profundidad diversos conceptos que se utilizan para la evaluación de modelos predictivos, en la sección 3 se comenta una primera aproximación al problema planteado, en la sección 4 se plantea el enfoque definitivo del trabajo y se explica en detalle el desarrollo de los dos modelos predictivos mediante técnicas diferentes, en la sección 5 se muestran y comentan los resultados obtenidos en los diferentes modelos predictivos y en la finalmente, en la sección 6, se concluye la memoria y se muestran algunas líneas futuras de trabajo que no se han desarrollado.

2. Dimensiones de la validación

A continuación se introducen diferentes conceptos sobre la validación de modelos predictivos que resultan necesarios para comprensión del trabajo desarrollado.

2.1. Validación

En general, siempre que se desarrolla un modelo predictivo es recomendable evaluarlo. A esta fase de la modelización se le denomina **validación** y no existe una única característica que pueda definir si un modelo es satisfactorio. Cabe destacar que cuando se trata de predecir cualquier suceso, existe una gran diferencia en cuanto a métodos y complejidad en función a si el hecho a predecir es una variable binaria (un hecho se produce o no se produce) o si lo que se pretende predecir son variables de tres o más categorías. La complejidad aumenta considerablemente si el hecho a predecir no se corresponde con una variable binaria y la forma de evaluar los modelos es diferente. A continuación, se va a definir el concepto de umbral y después, para poder evaluar si los modelos desarrollados en este trabajo son satisfactorios, se define formalmente el porcentaje de acierto, el Lift y el área bajo la curva ROC.

- Umbral

Tal y como se ha comentado previamente, cuando se tiene un modelo predictivo desarrollado, lo que se obtiene del mismo es, para cada registro, la probabilidad de que ocurra el hecho que se desea predecir. En este contexto, resulta necesario definir un umbral en el intervalo $[0,1]$ de forma que si la probabilidad es mayor a ese umbral, entonces se considera que sí ocurre el hecho y si la probabilidad es menor, entonces se predice que no va a ocurrir el hecho en cuestión.

La definición de este umbral puede depender de criterios de exigencia empresariales, del científico de datos o simplemente, pueden utilizarse diferentes umbrales para saber cuál da resultados más satisfactorios para ciertos modelos predictivos. Para el desarrollo de este trabajo, por razones de simplicidad, se ha fijado el umbral en 0,5.

- Porcentaje de acierto

El porcentaje de acierto puede ser una propiedad útil, pero es necesario interpretarla en función al contexto para determinar si un modelo predictivo es satisfactorio o no lo es. Para su formalización, se definen las siguientes variables:

$N = \{N^{o} \textit{ de registros a predecir}\} \in \mathbb{N}$
$P_i = \{\textit{Predicción realizada sobre la variable objetivo del registro } i\} \in \mathbb{Z}_2$

$R_i = \{Suceso\ ocurrido\ realmente\ para\ el\ registro\ i\} \in \mathbb{Z}_2$

Se define la función *error* como:

$$error : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$$

$$error(x,y) = \begin{cases} 0 & si \quad x = y \\ 1 & si \quad x \neq y \end{cases}$$

$F = \{N^o\ de\ fallos\ al\ predecir\} = \sum_{i=1}^{N} error(P_i, R_i)$

Se define el porcentaje de acierto como:

$$Acierto = 1 - \frac{F}{N} \in [0,1]$$

De esta propiedad, es necesario tener en cuenta que cuando lo que se quiere predecir está balanceado, es decir, el número de veces que ocurre un suceso es similar al número de veces que no ocurre, el porcentaje de acierto suele aportar información útil. Sin embargo, la interpretación del porcentaje de acierto puede ser totalmente errónea si el hecho que quiere predecirse ocurre con una tasa muy baja. Supongamos el siguiente ejemplo: se desea predecir si un cliente de un servicio online va a cancelar su suscripción. Estamos ante un problema de clasificación binaria: el usuario cancela la suscripción al servicio (target=1) o la mantiene (target=0). Definimos los siguientes parámetros:

Tenemos un servicio con 100 clientes suscritos $\implies N = 100$

Sin ningún estudio previo ni aplicando ninguna técnica de modelización, se desarrolla un modelo que predice que ningún usuario va a cancelar su suscripción $\implies P_i = 0\ \forall\ i \in [0,100]$

En realidad, 2 de los 100 usuarios suscritos la cancelan $\implies R_i = \begin{cases} 0 & \forall\ i \in \{3,4,...,100\} \\ 1 & \forall\ i \in \{1,2\} \end{cases}$

Por tanto, realizando los cálculos definidos anteriormente:

$$F = \sum_{i=1}^{100} error(P_i, R_i) = 2$$

El porcentaje de acierto resultante sería el siguiente:

$$Acierto = 1 - \frac{2}{100} = 0{,}98$$

Obtenemos que este modelo que tan solo consiste en afirmar que ningún cliente va a cancelar su suscripción tiene un porcentaje de acierto del 98 %. A priori, sería un porcentaje satisfactorio, pero en realidad, no se está prediciendo nada ni tendría utilidad este modelo por lo que siempre es necesario tener en cuenta el contexto en el que se evalúa.

- Lift

El lift es una medida de orden para poder controlar cómo funciona un modelo predictivo respecto a una predicción aleatoria [1]. Algunos parámetros necesarios para definir formalmente el lift son los siguientes:

$N = \{N^{o}\ de\ registros\ a\ predecir\} \in \mathbb{N}$

$P_i(target = 1) = \{Probabilidad\ de\ que\ ocurra\ el\ hecho\ a\ predecir\ para\ el\ registro\ i\} \in [0,1]$

$R_i = \{Suceso\ ocurrido\ realmente\ para\ el\ registro\ i\} \in \mathbb{Z}_2$

$Ratio = \{Ratio\ medio\ de\ ocurrencia\ de\ la\ variable\ objetivo\} = \frac{\Sigma_1^N R_i}{N} \in [0,1]$

Una vez desarrollado un modelo predictivo y habiendo obtenido la probabilidad P_i de la variable objetivo para cada registro, se aplica el siguiente algoritmo:

- Se ordenan todos los registros sobre los que se ha realizado una predicción de menor a mayor probabilidad predicha.
- Una vez ordenados, se agrupan los usuarios por deciles de forma que se forman 10 grupos con el mismo número de registros. Se define $N_d = \{N^{o}\ de\ registros\ por\ decil\}$
- Se calcula el ratio real de ocurrencia de la variable objetivo para cada decil. Es decir:

$$Ratio\ real = \frac{\Sigma_1^{N_d} R_i}{N_d} \in [0,1]\ N_d \in \{1,2,...,9,10\}$$

Para cada decil, se calcula el lift que se define de la siguiente forma:

$$Lift = \frac{Ratio\ real}{Ratio}$$

Para ilustrar esta definición del lift, vamos a considerar el mismo ejemplo que se ha utilizado en el apartado anterior. El lift se calcularía de la siguiente forma:

- Se ordenan los usuarios de menor a mayor probabilidad predicha de cancelar su suscripción.
- Se agrupan los usuarios por la probabilidad predicha de cancelar la suscripción por deciles.
- Se calcula el porcentaje de cancelación de suscripción para cada grupo. Es decir, se suma el número de veces que se cancela una suscripción y se divide entre el número de clientes en cada grupo.

El propósito de nuestro modelo de ejemplo es estimar si es probable que un usuario cancele su suscripción o no. Esto quiere decir, que si el modelo funciona de forma satisfactoria, el aumento de la probabilidad predicha debe ser directamente proporcional al aumento del ratio de cancelaciones. Es decir, cuanto más elevada sea la probabilidad de que un cliente cancele, más elevado debe ser el ratio de cancelaciones reales.

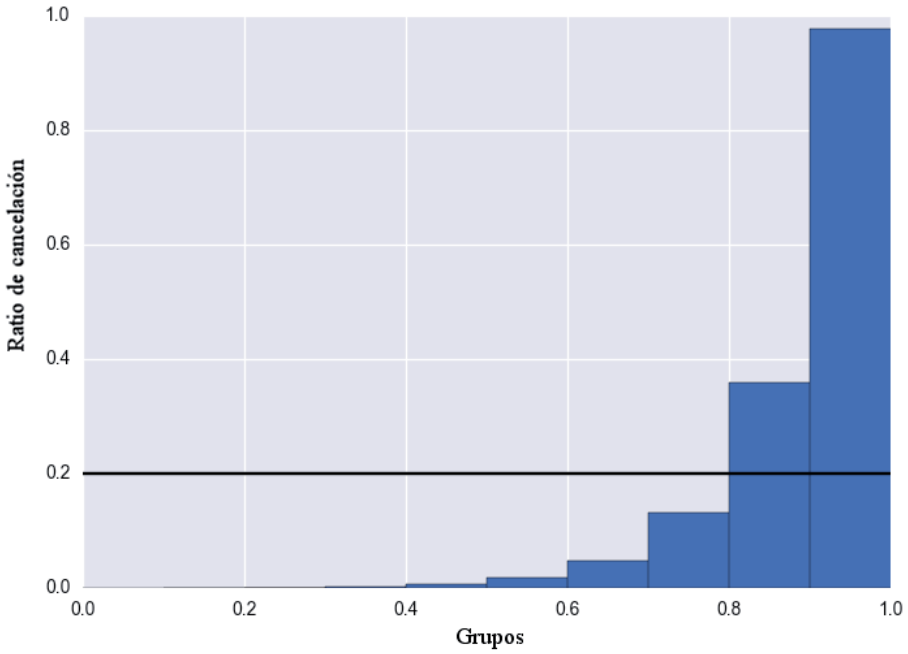

Figura 3: Ratios de cancelación por decil

En la figura 3 puede observarse que el ratio de cancelaciones en la parte de la derecha es más elevado, tal y como era de esperar. Para probabilidades inferiores al 50 % el ratio de los grupos es aproximadamente 0. Este gráfico puede utilizarse para comprobar el que el modelo está funcionando correctamente. También cabe destacar, que esta medida es perfectamente comparable entre cada variación o mejora del modelo.

El grupo con mayor probabilidad de cancelar tiene un lift de $0{,}97/0{,}2 = 4{,}85$ y el segundo grupo con probabilidad más alta tiene un lift del 1.8. Esto se interpreta de la siguiente forma, si seleccionamos al grupo de clientes con una probabilidad de cancelación mayor, podemos esperar capturar un número de usuarios que van a cancelar la suscripción cinco veces mayor que si seleccionásemos al mismo grupo de gente de forma aleatoria.

Matriz de confusión

En el campo de la minería de datos, una matriz de confusión es una herramienta que permite la visualización del desempeño de un modelo predictivo. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real [2].

Para comprender esta característica es necesario definir el concepto de falso positivo y falso negativo. Se produce un falso positivo cuando se ha predicho que un hecho va a ocurrir pero este hecho finalmente no ocurre. Por otro lado, un falso negativo se produce cuando se predice que un hecho no va a ocurrir y finalmente ocurre.

Uno de los beneficios de las matrices de confusión es que facilitan ver la distribución de los ”fallos”del modelo predictivo. Se puede ver si el modelo tiene un mayor número de fallos en falsos positivos o en falsos negativos y en función de lo que se esté prediciendo, dos modelos con el mismo porcentaje de acierto, pueden considerarse de forma diferente en función a su distribución de errores observada en la matriz de confusión.

		Predicción	
		Cancela	No cancela
Valor Real	Cancela	1000	150
	No cancela	500	380

Cuadro 1: Ejemplo de matriz de confusión

Continuando con el mismo ejemplo mencionado anteriormente (expuesto ahora para otra población total), en la matriz de confusión del cuadro 1 se observa que existen 500 personas sobre las que se predice que van a cancelar y en realidad no lo hacen (falsos positivos) mientras que hay 150 personas que se predice que no van a cancelar y en realidad sí lo hacen (falsos negativos). En este caso, si la empresa que plantea este problema quiere evitar cancelaciones inesperadas, estaríamos ante un modelo que probablemente no estaría cumpliendo su cometido ya que tiene un elevado porcentaje de falsos negativos. Cabe destacar que, en general, siempre es necesario plantearse en función al hecho predicho, cómo afectaría la distribución de falsos positivos y falsos negativos del modelo.

El espacio ROC

Para poder definir la curva ROC, primero es necesario tener claro en qué espacio se dibuja. Para dibujar una curva ROC sólo son necesarias las siguientes razones que se obtienen de la matriz de confusión:

- Ratio de verdaderos positivos $\implies VPR = \frac{|VP|}{|Realidad\ positivos|}$

- Ratio de falsos positivos $\Longrightarrow FPR = \frac{|FP|}{|Realidad\ negativos|}$

El ratio VPR mide hasta qué punto un modelo es capaz de detectar o clasificar los casos positivos correctamente, de entre todos los casos positivos disponibles. El ratio FPR define cuántos resultados positivos son incorrectos de entre todos los casos negativos disponibles durante la prueba [3].

El espacio ROC se define por FPR y VPR como ejes x e y respectivamente, y representa los intercambios entre verdaderos positivos y falsos positivos. En diferente documentación consultada, VPR se considera equivalente a sensibilidad y FPR se considera equivalente a $1 - especificidad$. Por tanto, el gráfico ROC también es conocido como la representación de sensibilidad frente a $(1 - especificidad)$. De cada resultado de predicción se puede obtener una instancia de la matriz de confusión y así representar un punto en el espacio ROC.

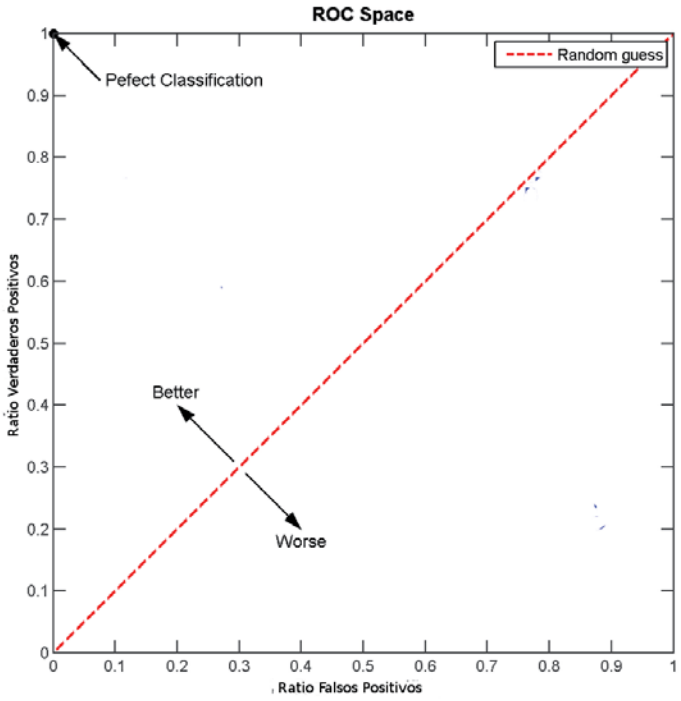

Figura 4: Espacio ROC

Analizando la figura 4, el mejor método posible de predicción se situaría en un punto en la esquina superior izquierda, o coordenada (0,1) del espacio ROC, representando un 100 % de sensibilidad (ningún falso negativo) y un 100 % también de especificidad (ningún falso positivo). A este punto (0,1) también se le llama clasificación perfecta. Por el contrario, una clasificación totalmente aleatoria daría un punto a lo largo de la línea diagonal, que se llama también

línea de no-discriminación, y se dibuja desde el extremo inferior izquierdo hasta la esquina superior derecha.

La diagonal divide el espacio ROC. Los puntos por encima de la diagonal representan resultados del modelo predictivo que pueden ser satisfactorios (mejor que el azar), mientras que los puntos por debajo de la línea implican que los resultados del modelo son insatisfactorios (peor que al azar). Nótese que la salida de un modelo predictivo que sea insatisfactorio podría ser invertida para obtener un modelo que predice con mayor acierto.

Área bajo la curva ROC

Una vez definido el espacio ROC, dado un modelo, vamos a querer dibujar su curva ROC asociada y vamos a calcular el área bajo esa curva puesto que nos dará información sobre el funcionamiento del modelo. Para hacerlo se aplica el siguiente algoritmo:

1. Mediante el modelo, se obtiene la probabilidad de que ocurra el hecho a predecir para cada registro.

2. Se seleccionan N valores en el intervalo $[0,1]$. Los valores seleccionados representan distintos umbrales y son, en principio, equidistantes aunque pueden no serlo. Se fija V al primer valor de N. $V = N_1$

3. Se aplica la siguiente función a cada registro para decidir si ocurre o no ocurre el hecho a predecir:

$$ocurre(x) = \begin{cases} 0 & si \quad P(target = 1) \leq N_i \\ \\ 1 & si \quad P(target = 1) > N_i \end{cases}$$

4. Tras aplicar la función definida, se calcula la matriz de confusión y se obtienen los ratios VPR y FPR que representan un punto que se dibuja en el espacio ROC.

5. Si no se han cubierto todos los valores de N, se fija V al siguiente valor de N, $V = N_{i+1}$ y se vuelve al paso (3). En caso contrario, se tienen todos los puntos dibujados en el espacio ROC.

6. Sobre los puntos dibujados se puede interpolar linealmente para aproximar la curva ROC y así poder calcular el área que queda encerrada bajo la misma o pueden realizarse otro tipo de ajustes diferentes a la interpolación lineal.

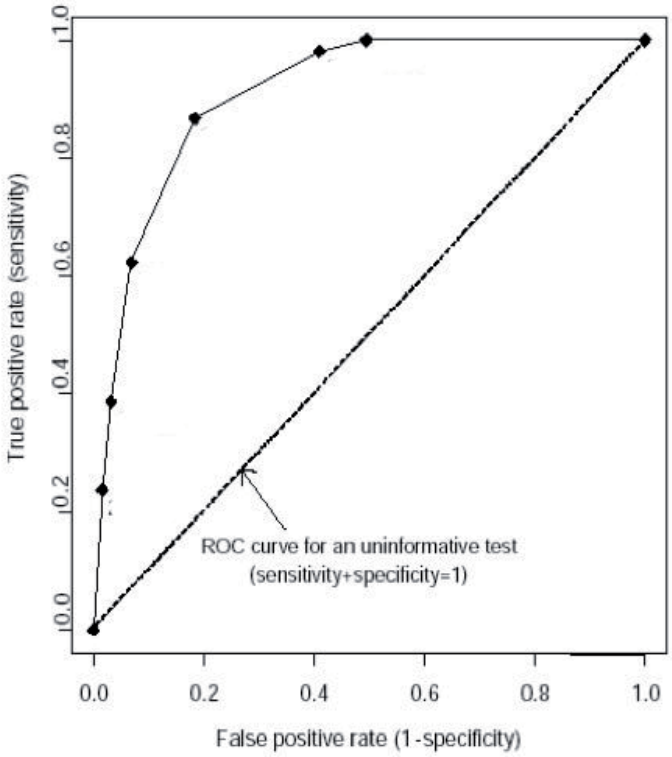

Figura 5: Ejemplo de curva ROC

Tal y como se ha comentado previamente, los puntos representados en el espacio ROC pueden servir para evaluar el funcionamiento del modelo puesto que se representa el punto compuesto por el ratio de falsos positivos y el de verdaderos positivos. Por tanto, el algoritmo detallado anteriormente puede utilizarse también para saber donde fijar el corte en la probabilidad para considerar que el hecho modelizado ocurre.

Para comparar dos pruebas, no podemos centrarnos exclusivamente en el dato objetivo de que el área bajo la curva ROC de un modelo sea mayor que el área utilizando otro modelo. Pueden existir dos pruebas con sendas curvas ROC muy distintas de forma, hecho que puede tener importantes implicaciones prácticas, y que, sin embargo, sean prácticamente iguales. En general, nunca debería prescindirse de un examen visual detenido de un gráfico que muestre simultáneamente diferentes curvas ROC.

2.2. Análisis comparativo

En primer lugar, para poder realizar un análisis comparativo de estas medidas, cabe destacar que en el ámbito de los modelos predictivos **no existe el óptimo**. Es posible desarrollar un

modelo que tenga un lift, un porcentaje de acierto y un área bajo la curva ROC que resulten satisfactorios alcanzando una solución de compromiso. Sin embargo, no podrá afirmarse que es la solución óptima puesto que no es seguro que el modelo desarrollado no vaya a ser superado por modelos con mejor funcionamiento en el futuro. Además, los modelos predictivos se desarrollan bajo la suposición de que una población que actualmente se comporta de una forma, va a seguir haciéndolo de forma parecida en el futuro y estos comportamientos también pueden cambiar (normalmente, las empresas recomiendan actualizar los modelos anualmente).

Entonces, ¿cómo puede decidirse si un modelo es mejor que otro? Es una pregunta que no tiene una respuesta trivial. Generalmente, es necesario plantearse una solución de compromiso entre lo que se desea predecir, el coste en tiempo de desarrollar una solución, el coste en recursos de ejecutar el modelo predictivo y las medidas de validación tomadas en cuenta .

Teniendo en cuenta únicamente las medidas de validación, cabe destacar que estas medidas **no son ordenables**. Un modelo con mejor porcentaje de acierto no tiene por qué tener mejor lift o una mayor área bajo la curva ROC ni ello debe significar que sea un modelo que proporcione mejores resultados.

2.3. Función de evaluación

Dado que para evaluar un modelo, es necesario tener en cuenta todas las medidas definidas anteriormente, para este trabajo se ha desarrollado una función de evaluación propia que es una simplificación que tiene en cuenta las medidas anteriores además de una penalización por falsos positivos y falsos negativos en función al criterio que quiera darle la persona que evalúa el modelo. Para definir la función, contamos con los siguientes parámetros:

$P = \{Porcentaje\ de\ acierto\} \in [0,1]$
$R = \{Area\ bajo\ la\ curva\ ROC\} \in [0,1]$
$L = \{Lift\ del\ decil\ con\ mayor\ probabilidad\} \in [0,\infty)$
$FP = \{Ratio\ de\ falsos\ positivos\} \in [0,1]$
$FN = \{Ratio\ de\ falsos\ negativos\} \in [0,1]$
$I_{FP} = \{Importancia\ de\ los\ falsos\ positivos\} \in [0,10]$
$I_{FN} = \{Importancia\ de\ los\ falsos\ negativos\} \in [0,10]$

Y la función de evaluación desarrollada se define de la siguiente forma:

$$fun = \text{mín}(0, 40R + 30P + 3L - (5FPI_{FP} + 5FNI_{FN}))$$

Esta función se ha desarrollado de forma empírica para ser utilizada en la evaluación de modelos predictivos como una primera aproximación para determinar si su funcionamiento es satisfactorio o no lo es. La función se divide en dos partes. La primera parte es la correspondiente al cálculo de una puntuación base teniendo en cuenta el área bajo la curva ROC, el porcentaje

de acierto y el lift. La segunda parte es una penalización en función al ratio de falsos positivos y al de falsos negativos, combinado con la importancia que se le haya dado.

La función es positiva y cada sumando está acotado salvo el correspondiente al lift. No obstante, el lift en la gran mayoría de los casos se encuentra en el intervalo $[0, 10]$ y en extrañas ocasiones supera ese valor por lo que se considera que es un factor acotado también. La figura 6, que pueda observarse a continuación, representa la interpretación de la capacidad predictiva del modelo en función a los resultados de la función:

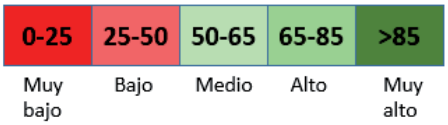

Figura 6: Escala correspondiente a la función de evaluación

Supongamos que tenemos dos modelos predictivos sobre los mismos datos, el modelo A y el modelo B. Entre el experto y el cliente, se han puesto de acuerdo para determinar la importancia de los falsos negativos y los falsos positivos, siendo esta $I_{FN} = 10$ y $I_{FP} = 7$ respectivamente. Al validar ambos modelos, se obtienen las siguientes medidas que se pueden analizar en el cuadro 2:

Modelo A	Modelo B
R=0.7	R=0.77
P=0.8	P=0.85
L=3	L=3.4
FP=0.15	FP=0.05
FN=0.05	FN=0.1

Cuadro 2: Datos sobre los modelos A y B

Y al aplicar la función se obtienen los siguientes resultados:

Modelo A	Modelo B
Base=61	Base=66.5
Penalización=7.75	Penalización=6.75
Total=53.25	Total=59,75

Cuadro 3: Función de evaluación aplicada al modelo A y B

Tal y como puede observarse, según la escala, el modelo B tiene mayor capacidad predictiva que el modelo A. No obstante, la penalización es aproximadamente la misma porque el modelo

B tiene un mayor porcentaje de falsos negativos que el modelo A aunque el porcentaje de acierto del modelo B sea más elevado. Cuando las diferencias al aplicar la función desarrollada son tan pequeñas como en este caso, lo aconsejable sería analizar detenidamente las diferentes medidas de validación para finalmente decidir si un modelo es más satisfactorio que otro.

3. Primera aproximación

Una vez introducidos todos los conceptos relacionados con la validación de modelos en este trabajo, en los siguientes apartados va a contarse cómo se han obtenido los datos sobre los que desarrollar los modelos predictivos, cómo se han desarrollado y se van a interpretar los resultados y sus comparaciones.

Como primera aproximación, se encontró una competición en el sitio web Kaggle [1] que se llamaba *"Springleaf Marketing Response"* y el objetivo de la competición era, dados unos datos anonimizados de los clientes de una empresa de marketing, determinar qué clientes iban a responder de forma positiva a una oferta que se le enviase por correo electrónico.

Los datos descargados se reflejan en el siguiente cuadro:

Nombre	Descripción	Número de variables	Número de registros	Comentarios
train.csv	Conjunto de datos de entrenamiento	1.935	145.231	Datos liberados al comienzo de la competición
test.csv	Conjunto de datos de validación	1.934 (igual que train pero sin la variable objetivo)	145.232	

Cuadro 4: Datos obtenidos

El problema planteado, en primer lugar, parecía apropiado para este trabajo puesto que se trataba de realizar una predicción sobre una variable binaria (respuesta positiva o negativa al correo electrónico). No obstante, una vez se comenzaron a estudiar los datos, se tomó la decisión de **descartarlos** y buscar unos nuevos datos ya que, el volumen del conjunto de datos descargado era muy elevado y el equipo con el que se ha trabajado no tenía suficiente memoria RAM como para tratarlos e iba a ser muy complicado llegar a analizar concretamente cualquier variable dado que se desconocía por completo su significado (las variables del conjunto de datos estaban numeradas, pero no tenían ningún nombre significativo).

[1] Consultar el apartado 7.1 en los anexos

4. Cambio de enfoque

Tras descartar los datos anteriores, se encontró una nueva competición en el sitio web Kaggle denominada *"Santander Customer Satisfaction"*. Esta competición tiene como objetivo identificar clientes insatisfechos de forma prematura. De esta forma, el banco podría realizar diferentes acciones para mejorar la satisfacción de ciertos clientes antes de que estos abandonen en el banco.

Siguiendo el esquema de trabajo definido en la introducción, se han descargado los datos de la página web y se definen a continuación en el cuadro 5:

Nombre	Descripción	Número de variables	Número de registros	Comentarios
train	Conjunto de datos de entrenamiento	371	76.020	Datos liberados al comienzo de la competición
test	Conjunto de datos de validación	370 (igual que train pero sin la variable objetivo)	75.818	

Cuadro 5: Datos obtenidos de la competición *"Santander Customer Satisfaction"*

Tal y como puede consultarse en la tabla, se descargaron dos conjuntos de datos, un conjunto denominado train que se iba a utilizar para entrenar el modelo predictivo. Por otro lado, el conjunto de datos test se iba a utilizar para dar una probabilidad de estar insatisfecho a cada cliente de ese conjunto, utilizando el modelo entrenado. Después, se podía subir al sitio web un fichero con las probabilidades dadas para cada cliente. Dadas esas probabilidades, los servidores de la página web calculaban el área bajo la curva ROC (solo se basaban en esa medida) y te decían en que posición estabas respecto al resto de participantes.

Por las necesidades de evaluación de este trabajo, tan sólo se ha trabajado con el conjunto de datos de entrenamiento, dividiéndolo en dos tablas de la siguiente forma:

Nombre	Descripción	Número de variables	Número de registros
entrenamiento	Conjunto de datos de entrenamiento	371	50.000
validación	Conjunto de datos de validación	371	16.020

Cuadro 6: Tablas generadas a partir del conjunto de datos train

El objetivo de dividir la tabla de entrenamiento en dos tablas era obtener una tabla para entrenar el modelo, y otra tabla que serviría para validarlo, ya que en ambas la variable sobre la

que se realiza la predicción está informada.

El siguiente paso, fue analizar los conjuntos de datos para poder tener información sobre la población que íbamos a utilizar para predecir qué clientes están insatisfechos (variable objetivo con valor 1) o qué clientes no lo están (variable objetivo con valor 0).

Para poder realizar este análisis se llevó a cabo un pequeño desarrollo de un código [2] que servía para obtener datos sobre cada variable de la población, agrupándolos todos en en un fichero de tipo pdf para facilitar su análisis. Para cada variable se obtenía su distribución en un histograma, y también se obtenían la media, la mediana, la desviación típica, el mínimo, el máximo y su nivel de información.

Tras realizar un análisis exhaustivo de la población, destacan las siguientes características principales:

- Todos los datos están informados al 100 %, no obstante, existen ciertas variables que tienen el valor -999999 lo que sugiere que se le ha asignado ese valor a casos en los que el mismo era desconocido.
- Existen 34 variables que son constantes, por tanto no aportan ninguna información de cara a caracterizar a diversos miembros de la población.
- Existen 60 variables que tienen exactamente los mismos datos que otras 60 variables, por tanto, se está duplicando la información.
- Se observa que una parte de la población que está satisfecha, tiene aproximadamente las mismas características que parte de la población insatisfecha. Esto sugiere que no va a ser sencillo predecir la insatisfacción del cliente.
- Se observa que el conjunto de datos está desbalanceado respecto a la variable objetivo, es decir, hay muy pocos clientes que se identifican como insatisfechos (aproximadamente un 4 %) mientras que el resto están satisfechos.
- Se infieren los significados de varias variables en base a los histogramas, a destacar:
  - Se deduce que var3 es la nacionalidad del cliente.

Valor	2	8	-999999	9	3	1	13	7	4	12
Frecuencia	74.165	138	116	110	108	105	98	97	86	85

Cuadro 7: Tabla de frecuencias de la variable var3

[2] Consultar el código tratamiento_datos.R anexado en el apartado 7.3

- Se deduce que *num_var_4* es el número de productos que tiene contratado cada cliente.

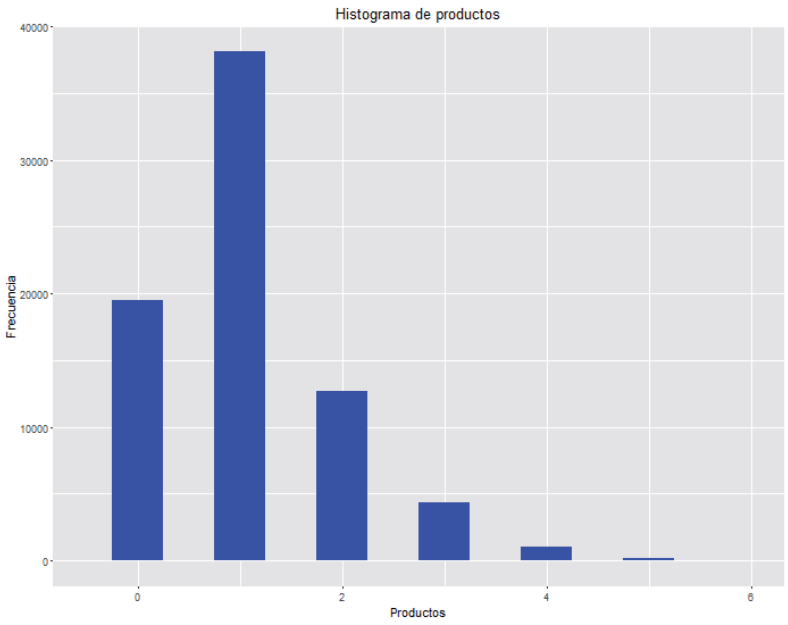

Figura 7: Distribución de los productos por cliente

- Al estudiar esta variable, se observa que la mayoría de clientes insatisfechos no tienen ningún producto.

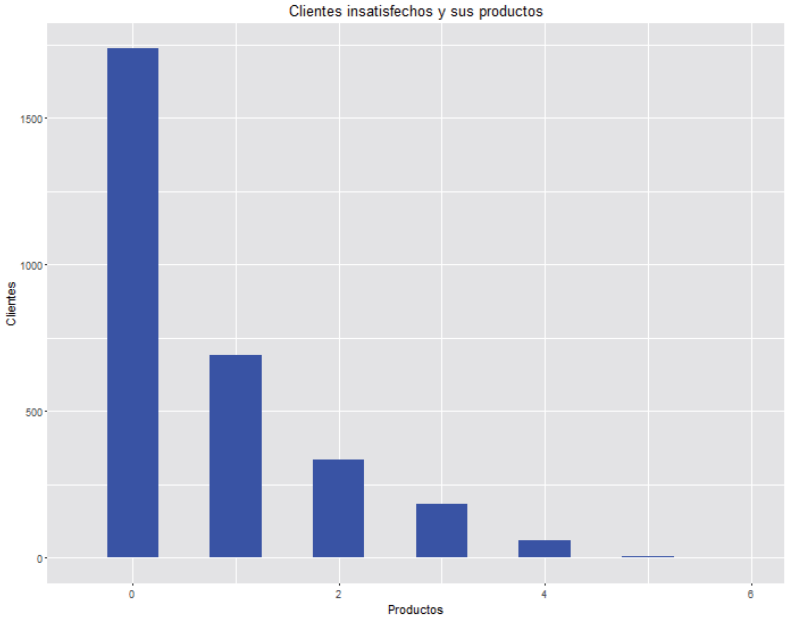

Figura 8: Distribución de los productos entre los clientes insatisfechos

- Se deduce que var15 es la edad del cliente y su distribución queda reflejada en la figura 9 a continuación:

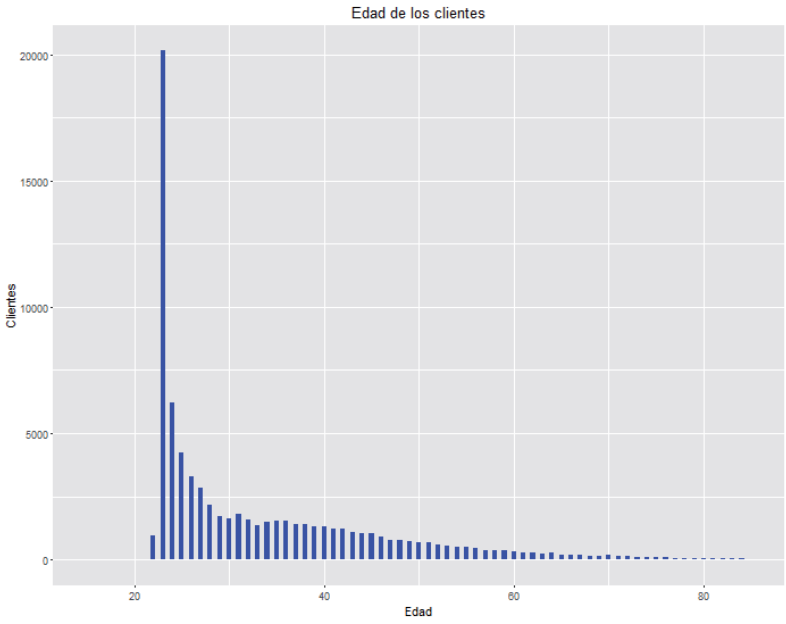

Figura 9: Edad de los clientes estudiados

Tras estudiar y comprender las características del conjunto de datos con el que se estaba trabajando, se llevó a cabo un pretratamiento de los datos de la siguiente forma:

- Se elimina del conjunto de datos las 34 variables que son constantes puesto que no aportan ninguna información del cliente dado que todos los clientes tienen el mismo valor.
- Se elimina del conjunto de datos las 60 variables duplicadas, dado que al haber, para cada variable, otra con los mismos datos, se está duplicando información de forma innecesaria.
- Se realiza un análisis de correlación de Pearson [3] y se eliminan las variables cuyo coeficiente de correlación es mayor que 0.85.
- Se calcula la varianza de cada variable y se eliminan las variables con varianza menor que 0.001, es decir, variables que prácticamente son constantes.
- Se modifica el valor de las variables que tenían asignado -999999 y se les asigna la media de la población para no desviar bruscamente la distribución de los datos.

En los dos próximos apartados, se explica cómo se han desarrollado dos modelos predictivos mediante dos técnicas diferentes partiendo del conjunto de datos resultante tras el pretratamiento.

4.1. Modelo Gradient Boosting (XGBoost)

Bajo la metodología del Gradient Boosting estudiada en la bibliografía [4], se ha utilizado la librería XGBoost para programar en R un modelo que sirviese para predecir la insatisfacción del cliente.

Una vez tratados los datos, se ha ido desarrollando de forma iterativa diferentes versiones del modelo, de forma que se iba observando una mejora en su funcionamiento a través de las diferentes medidas de validación planteadas. La estrategia utilizada ha sido la siguiente:

- Se implementa el algoritmo utilizando la librería XGBoost.
- Se prueba el algoritmo para unos parámetros determinados.
- Una vez se ha comprobado el correcto funcionamiento del algoritmo, se implementa un código iterativo que recorre los diferentes parámetros del algoritmo (submuestreo, rondas y profundidad de los árboles).
- Para cada modelo obtenido, se puntúa el conjunto de datos de validación y se calculan las diferentes medidas de validación.

[3] Consultar el apartado 7.2 de los anexos

- Teniendo en cuenta los resultados obtenidos, se selecciona el mejor modelo.
- Una vez extraído el modelo, se procede a puntuar el conjunto de datos de validación definitivamente.

Tras llevar a cabo la selección de un modelo mediante la estrategia definida anteriormente, investigando la bibliografía, se descubre que existe una forma de mejorar la precisión de un modelo más sencilla y poderosa que la selección efectuada: se pueden aunar modelos y ensamblarlos. Crear un ensamblado consiste en dos pasos: el primero, construir diversos modelos y el segundo, combinar sus pesos. Cualquier método de ensamblado compite de forma satisfactoria contra el mejor de los algoritmos individuales que lo componen.

Este fenómeno fue descubierto por un grupo de investigadores, separada y simultáneamente, para mejorar la clasificación, usando tanto árboles de decisión (Ho,Hull and Srihari, 1990), como redes neuronales (Hansen and Salamon, 1990). Los avances tempranos más importantes fueron desarrollados por Breiman, L. (1996) con la técnica de Bagging, y Freund and Shapire (1996) con el algoritmo AdaBoost.

Figura 10: Esquema de ensamblado de modelos

Por tanto, una vez obtenidos los mejores parámetros para el modelo, se definieron 20[4] semillas aleatorias fijas, y a partir de esas semillas, se desarrolló un modelo para cada una de ellas. Finalmente, combinando los 20 modelos con idéntico peso para cada uno de ellos, se obtuvieron los mejores resultados, que se analizan en el siguiente capítulo.

4.2. Modelo mediante una regresión logística

Tras desarrollar el modelo predictivo mediante la técnica denominada Gradient Boosting, se planteó el desarrollo de otro modelo basado en una regresión logística[5] puesto que era más sencillo y se desarrolló en un tiempo sustancialmente menor, para después poder comparar las diferencias existentes en el funcionamiento de ambos modelos sobre el conjunto de validación.

[4]De forma empírica se han ido seleccionando una distinta cantidad de semillas aleatorias y el mejor resultado se ha obtenido con 20 semillas.

En primer lugar, trabajando sobre el conjunto de datos resultante tras el pretratamiento, se llevó a cabo un análisis de varianza sobre todas la variables en función de la variable objetivo. Tras estudiar en la bibliografía cómo se interpreta un análisis de varianza (ANOVA [6]), se seleccionan las variables que producen diferencias significativas en el comportamiento de la variable dependiente, es decir, que influyen a la hora de determinar si el cliente está insatisfecho o no lo está. Las variables seleccionadas para el desarrollo del modelo, aunque no pueden interpretarse puesto que no tienen nombres significativos, se pueden consultar en el cuadro 8 a continuación:

Variable	Tipo
var15	Numérica
imp_op_var40_efect_ult1	Numérica
imp_op_var41_efect_ult1	Numérica
ind_var1_0	Binaria
ind_var5	Binaria
ind_var12_0	Binaria
ind_var13_0	Binaria
ind_var14_0	Binaria
ind_var30	Binaria
ind_var39_0	Binaria
num_var14_0	Numérica
saldo_var5	Numérica
saldo_var8	Numérica
num_ent_var16_ult1	Numérica
num_var22_hace2	Numérica
num_var22_ult1	Numérica

Cuadro 8: Variables seleccionadas para desarrollar el modelo

Por último, se desarrolla un modelo predictivo mediante una regresión logística utilizando las variables seleccionadas y se puntúa el conjunto de validación. Los resultados se muestran en el siguiente apartado.

5. Resultados

A continuación se muestran y se analizan los resultados de validación obtenidos por cada modelo desarrollado en este trabajo. Más tarde, se comparan los resultados y se concluye el trabajo seleccionando la técnica que se considera más apropiada para la resolución del problema.

5.1. Resultados modelo XGBoost

Validando el modelo desarrollado mediante la técnica de Extreme Gradient Boosting (XGBoost) y para un umbral situado en 0,5 se han obtenido los siguientes resultados sobre el conjunto de validación:

- El **porcentaje de acierto** es del $97,41\,\%$, lo que supone un alto porcentaje de acierto y se debe al hecho de que la gran mayoría de la población no se identifica como insatisfecha, es decir, el conjunto de datos no está balanceado.
- La **matriz de confusión** obtenida se puede observar en el cuadro 9:

		Valor real	
		Satisfecho	Insatisfecho
Predicción	Satisfecho	15375	411
	Insatisfecho	3	231

Cuadro 9: Matriz de confusión

  Tal y como puede observarse en los datos, la gran mayoría de los fallos de predicción que se producen en el conjunto de validación, se dan con clientes que están insatisfechos pero el modelo no es capaz de detectarlos como clientes insatisfechos (la probabilidad obtenida es menor que el umbral).

- Al calcular el lift se obtiene el cuadro 10 con 1602 usuarios en cada decil:

Decil	Media Decil	Media Conjunto	Lift
1	0.06	4.09	0.01
2	0.25	4.09	0.06
3	0.69	4.09	0.17
4	1.00	4.09	0.24
5	1.44	4.09	0.35
6	1.63	4.09	0.40
7	2.75	4.09	0.67
8	4.44	4.09	1.09
9	9.44	4.09	2.31
10	18.44	4.09	4.51

Cuadro 10: Tabla resumen del lift

A partir de los datos calculados, se representa de forma inmediata en la figura 11 el gráfico del lift:

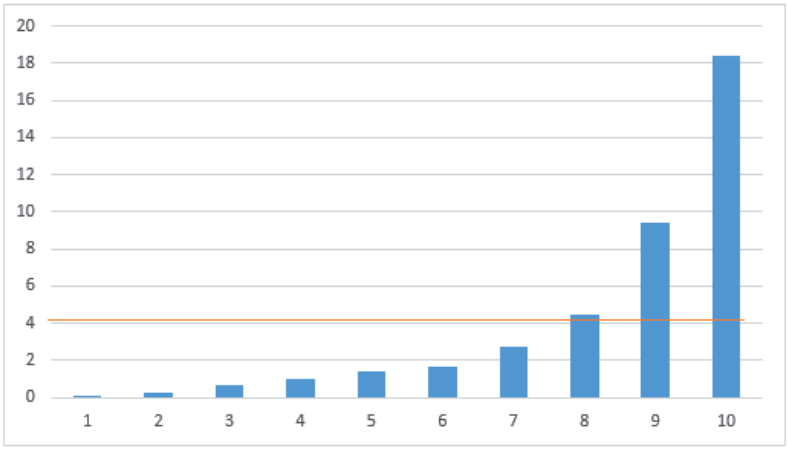

Figura 11: Gráfico del Lift

En la figura 11, se observa un crecimiento continuo de la población insatisfecha conforme se alcanzan los deciles con mayor probabilidad predicha. Este hecho se puede interpretar como un correcto funcionamiento del modelo, ya que cuanto mayor es la probabilidad de estar insatisfecho, mayor es la tasa real de insatisfacción.

- La **curva ROC** correspondiente es la siguiente y encierra un área de 0.837:

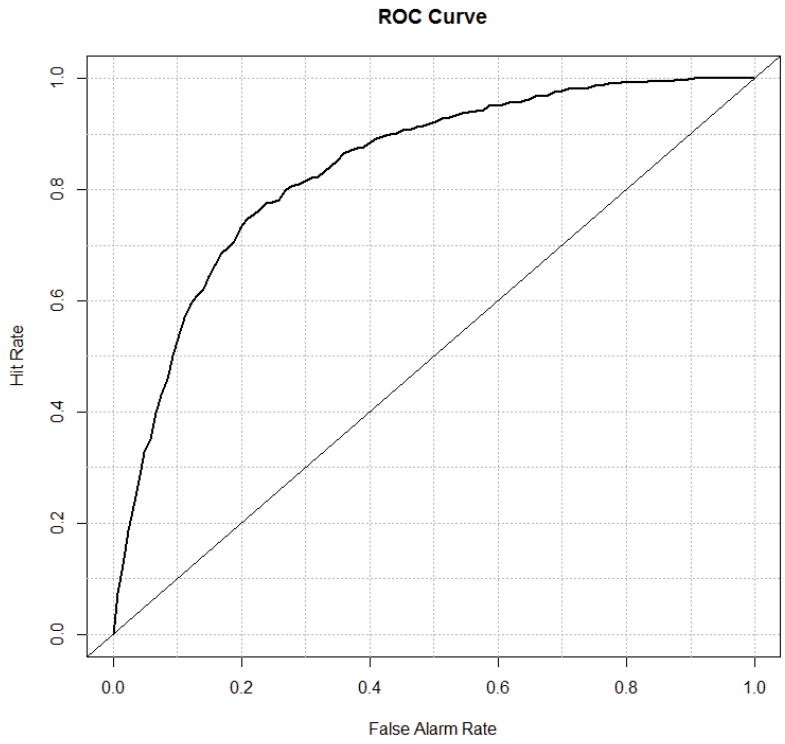

Figura 12: Gráfico de la curva ROC

Se observa que la función resultante es convexa y no tiene cambios de tendencia.

5.2. Resultados modelo regresión logística

Evaluando las mismas características sobre el modelo desarrollado mediante una regresión logística simple, se han obtenido los siguientes resultados sobre el mismo conjunto de validación:

- El porcentaje de acierto es del 96,59 % lo que supone un alto porcentaje de acierto y se debe al hecho de que la gran mayoría de la población no se identifica como insatisfecha, es decir, el conjunto de datos no está balanceado.
- La matriz de confusión obtenida es la siguiente:

		Valor real	
		Satisfecho	Insatisfecho
Predicción	Satisfecho	15373	541
	Insatisfecho	5	101

Cuadro 11: Matriz de confusión

Tal y como puede observarse en la matriz, se obtienen unos datos muy parecidos al anterior modelo, sólo que se detectan menos clientes insatisfechos. Por tanto, la gran mayoría de los fallos de predicción que se producen en el conjunto de validación, se dan también con clientes que están insatisfechos pero el modelo no es capaz de detectarlos como clientes insatisfechos (la probabilidad obtenida es menor que el umbral).

- Al calcular el lift se obtiene de la siguiente tabla con 1602 usuarios en cada decil:

Decil	Media Decil	Media Conjunto	Lift
1	0.94	4.09	0.23
2	0.69	4.09	0.17
3	0.94	4.09	0.23
4	1.31	4.09	0.32
5	1.56	4.09	0.38
6	2.88	4.09	0.70
7	4.81	4.09	1.18
8	3.44	4.09	0.84
9	6.88	4.09	1.68
10	16.69	4.09	4.08

Cuadro 12: Tabla resumen del lift

A partir de los datos calculados, se representa de forma inmediata en la figura 13 el gráfico del lift:

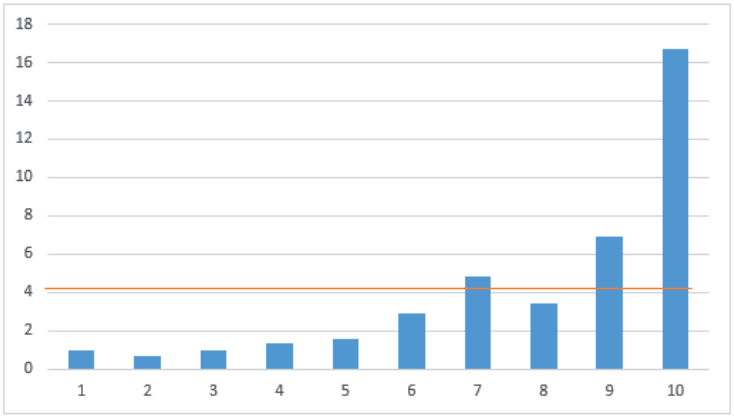

Figura 13: Gráfico del Lift

En el gráfico puede observarse que, tan sólo en los últimos deciles con mayor probabilidad de insatisfacción, parece que se detecta de forma relativamente satisfactoria a clientes insatisfechos. Este hecho se puede interpretar como un correcto funcionamiento del modelo tan solo en el 20 % de la población con mayor probabilidad de insatisfacción, ya que en el resto de la población, se dan comportamientos extraños y parece que el modelo no es capaz de detectar clientes insatisfechos.

- La curva ROC correspondiente es la siguiente y encierra un área de 0.77:

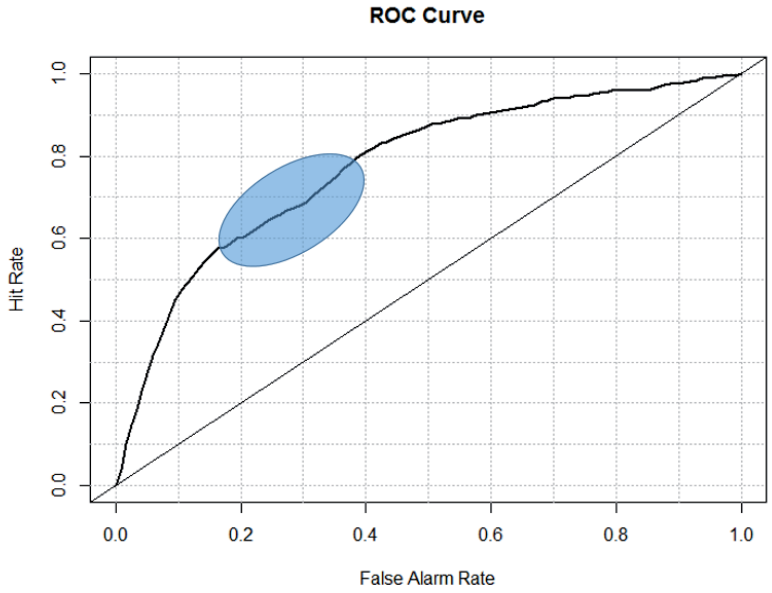

Figura 14: Gráfico de la curva ROC

En la zona remarcada en azul de esta curva ROC, se observa un cambio de tendencia, lo que implica un cambio en la segunda derivada de la misma puesto que la función pasa de se convexa a ser cóncava. Esto ocurre puesto que en el conjunto de validación, existe un grupo de clientes a los que se les ha dado una probabilidad de insatisfacción muy cercana y se les está dando una probabilidad parecida tanto a clientes satisfechos como a insatisfechos. Por tanto, crece el ratio de falsos positivos y no el de verdaderos positivos provocando el cambio de tendencia. No se ha llegado a realizar el estudio, pero se sospecha que estos problemas ocurren por la gran similitud en los datos en un grupo de clientes compuesto por clientes satisfechos e insatisfechos.

5.3. Gráficos comparativos

Tras evaluar las medidas individualmente, se han dibujado tanto un gráfico comparativo del Lift como un gráfico comparativo de las curvas ROC para poder analizar y comparar mejor ambos modelos. Además, se ha utilizado la escala desarrollada en este trabajo para estudiar si servía para discriminar alguno de los dos modelos desarrolados:

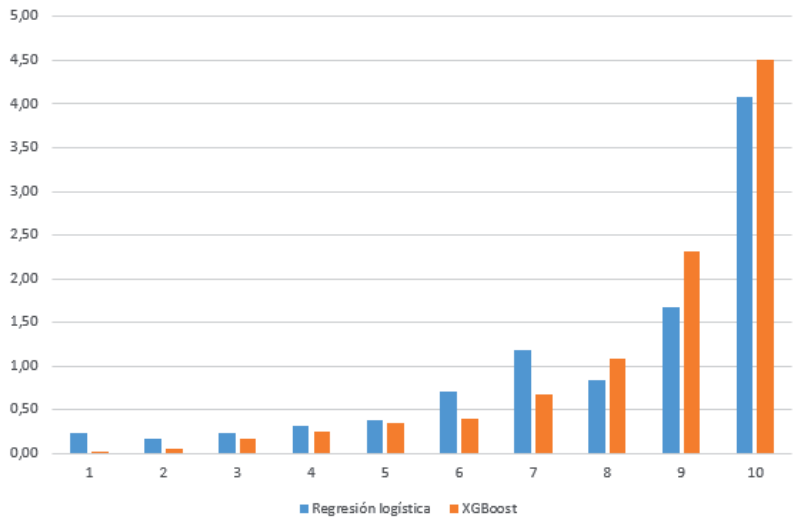

Figura 15: Comparación gráfico Lift

Al comparar el lift de los dos modelos representados en la figura 15, se observa que, tan sólo en los últimos deciles con mayor probabilidad de insatisfacción, la detección de clientes insatisfechos funciona de manera satisfactoria en ambos modelos. Sin embargo, en el resto de deciles, parece que el modelo XGBoost actúa de la forma esperada mientras que el modelo basado en una regresión logística no parece captar satisfactoriamente a los clientes insatisfechos.

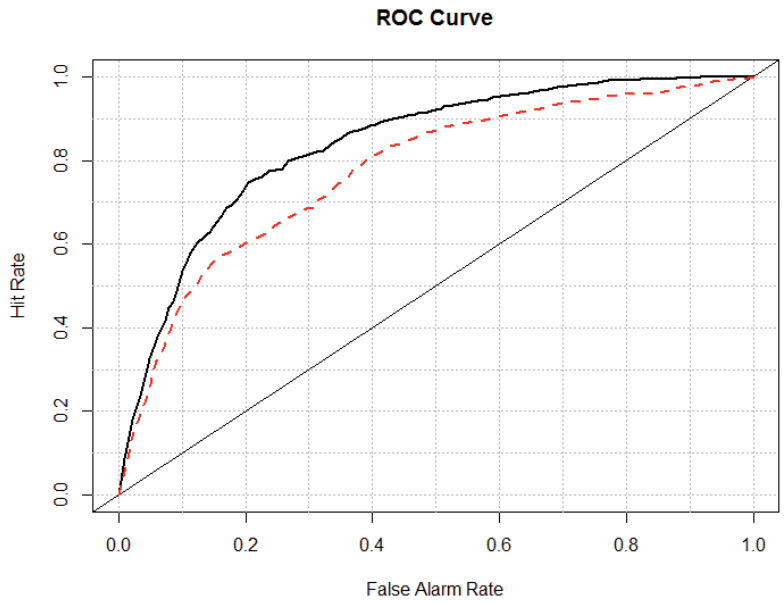

Figura 16: Comparación curvas ROC

En la figura 16 se observa como la curva ROC del modelo XGBoost está siempre por encima de la curva ROC del modelo basado en una regresión logística. Esto puede interpretarse de la siguiente forma, para cualquier valor del umbral, el modelo XGBoost asegura un mejor funcionamiento respecto al modelo basado en una regresión logística.

En el cuadro 13, se reflejan todas la medidas de validación calculadas, así como el porcentaje de falsos negativos y el porcentaje de falsos positivos que se han generado al puntuar el conjunto de validación para los dos modelos desarrollados.

Modelo XGBoost	**Modelo regresión logística**
R=0.837	R=0.775
P=0.974	P=0.966
L=4.51	L=4.08
FP=0	FP=0
FN=0.026	FN=0.034

Cuadro 13: Medidas de evaluación sobre los modelos desarrollados

Para terminar de tener las medidas que nos ayuden a evaluar qué modelo sería el elegido para resolver el problema planteado de predicción de insatisfacción de cliente, se calcula, mediante la escala desarrollada en este trabajo, la capacidad predictiva de cada modelo. Cabe destacar que para este problema concreto, se ha determinado la importancia de los falsos negativos como muy

alta ($I_{FN} = 10$) puesto que se quiere penalizar que existan clientes insatisfechos que se estén identificando como satisfechos y la importancia de los los falsos positivos se ha considerado como importancia media, siendo esta $I_{FP} = 5$.

Modelo XGBoost	**Modelo regresión logística**
Base=76.23	Base=72.22
Penalización=1.31	Penalización=1.70
Total=74.92	Total=70.52

Cuadro 14: Función de evaluación aplicada a los dos modelos desarrollados

Al comparar los diferentes modelos y medidas de validación, se obtiene que, al aplicar la función de evaluación definida para el modelo XGBoost este obtiene una puntuación de 74.92, es decir, tiene una capacidad predictiva alta y el segundo modelo obtiene una puntuación de 70.52, lo que quiere decir que ambos modelos tienen una capacidad predictiva relativamente satisfactoria y la escala, en este caso, no ayuda a discriminar un modelo frente a otro. Por tanto, es necesario analizar las medidas de forma individual.

Tras analizar individualmente las medidas, se observa un ligero peor funcionamiento del modelo desarrollado mediante una regresión logística mientras que el modelo XGBoost, según se evalúa en este trabajo, es capaz de detectar de mejor forma la insatisfacción de un cliente. Dado que el modelo de regresión logística obtiene unos buenos resultados de validación para el 20 % de la población con mayor probabilidad dada, la elección del modelo a utilizar sería el mismo si se desea alcanzar por parte del Santander a un 20 % de la cartera. Además, es un modelo más simple, rápido de desarrollar, consume menos recursos, es explicable y, para esa parte de la población, no funciona especialmente peor que el modelo XGBoost. Sin embargo, si el banco Santander quisiera alcanzar a toda su cartera, el modelo seleccionado para esta solución sería el modelo XGBoost, aunque es un modelo que no es sencillo de explicar a la empresa y podría ser satisfactorio, pero no sería comprensible.

6. Conclusiones y líneas futuras de trabajo

Como conclusiones de este trabajo destacan que se ha podido estudiar y trabajar con conjuntos de datos reales con un volumen relativamente grande. Se ha participado activamente en una competición de Kaggle quedando en el Top 20 % (posición 928 de 5123 participantes) y se ha mostrado un funcionamiento más satisfactorio en el modelo elaborado mediante la técnica de Gradient Boosting frente a un modelo desarrollado con una regresión logística.

Por otro lado, el trabajo se ha centrado en la evaluación de modelos predictivos. De esta forma, se ha construido una escala propia de clasificación de modelos en función a su poder de predicción y se han definido formalmente las medidas de validación de modelos que se iban a utilizar para poder trabajar con ellas.

Como líneas futuras de trabajo, se podrían elaborar modelos predictivos mediante redes neuronales, que es un campo que se encuentra en crecimiento debido a la capacidad de cómputo existente hoy en día y también se evaluarían con la escala y las medidas tratadas en este trabajo. Además, se podría estudiar más a fondo la función de evaluación de modelos que se ha construido, puesto que incluir otras medidas, añadir más criterios de penalización o dar otros pesos a los sumandos quizás puedan dar un mejor comportamiento a la misma.

7. Anexos

7.1. ¿Qué es Kaggle?

Figura 17: Logotipo de la empresa

La empresa Kaggle, fundada en abril del año 2010[7], se define en su página web como "la comunidad más grande del mundo de los científicos de datos. Éstos, compiten entre sí para resolver problemas complejos con una gran cantidad de datos, y se tratan problemas de negocio interesantes y sensibles de algunas de las mayores compañías del mundo a través de grandes competiciones."

Su labor, desarrollada por completo a través de su página web, se resume en poner en contacto a empresas que tienen necesidades de soluciones a problemas de negocio (que implican el manejo, análisis y explotación de grandes cantidades de datos), con los denominados "data scientists", es decir, con personas capaces de analizar, tratar y explotar los datos para obtener información y conclusiones de éstos.

Figura 18: Esquema de funcionamiento de Kaggle

La empresa es una plataforma online que sirve para hospedar competiciones públicas de datos, en las que empresas o instituciones, públicas o privadas, envían su problema a Kaggle y éste se encarga de presentar ese problema al público, que compite entre sí para crear la mejor solución. Así pues, Kaggle organiza las competiciones y prepara los conjuntos de datos, a cambio de un coste que le cobra a la empresa cliente. Por su parte, los participantes reciben algún tipo de recompensa, normalmente de tipo económica, si logran posicionar su solución entre las ganadoras (normalmente primer, segundo y tercer puesto).

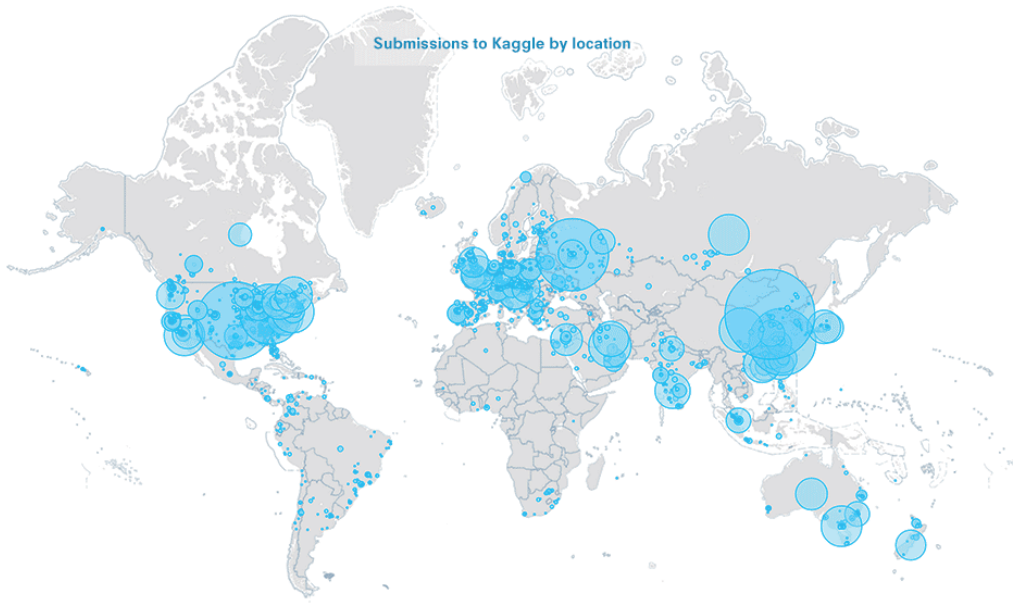

Figura 19: Envíos por localización

Kaggle proporciona resultados científicos con datos completamente actuales a empresas de todos los tamaños. Tiene un historial comprobado de resolución de problemas del mundo real a través de una amplia gama de industrias, incluyendo ciencias de la vida, servicios financieros, energía, tecnología de la información y el comercio minorista.

El sitio web ofrece varias competiciones al mismo tiempo. Actualmente aloja una cantidad de 214, de las cuales 12 están activas y el resto ya han concluido. Estas competiciones varían tanto en duración, como en temática, estructura o premio. Así pues, es posible encontrar competiciones que duran más de un año y otras que solo lo hacen unos pocos meses; unas cuyos datos son de tipo financiero, y otras en las que son textos o imágenes; y algunas en las que no se permiten equipos, y en otras, por ejemplo, en las que no hay solo una meta sino varios hitos marcados en diferentes fechas.

En cuanto al premio, éste siempre suele resultar atractivo al participante. Básicamente se dan tres tipos de premio:

- Económico. Se entrega una cantidad de dinero al ganador/es.
- Empleo. Se le asegura un puesto de trabajo al ganador. Es inusual.
- Didáctico. Se le da la oportunidad al público de manejar datos reales o de características especiales para practicar o investigar.

7.2. Análisis de correlación de Pearson

En ocasiones nos puede interesar estudiar si existe o no algún tipo de relación entre dos variables aleatorias. Así, por ejemplo, podemos preguntarnos si hay alguna relación entre las notas de la asignatura Estadística I y las de Álgebra I. Una primera aproximación al problema consistiría en dibujar en el plano R^2 un punto por cada alumno: la primera coordenada de cada punto sería su nota en estadística, mientras que la segunda sería su nota en matemáticas. Así, obtendríamos una nube de puntos la cual podría indicarnos visualmente la existencia o no de algún tipo de relación (lineal, parabólica, exponencial, etc.) entre ambas notas[8].

En este caso, nos interesa cuantificar la intensidad de la relación lineal entre dos variables. El parámetro que nos da tal cuantificación es el coeficiente de correlación lineal de Pearson r , cuyo valor oscila entre –1 y 1.

$$-1 \leq r = \frac{Cov(X,Y)}{S_x S_y} = \frac{\sum X_i Y_i - n\vec{X}\vec{Y}}{(n-1)s_x s_y} \leq 1$$

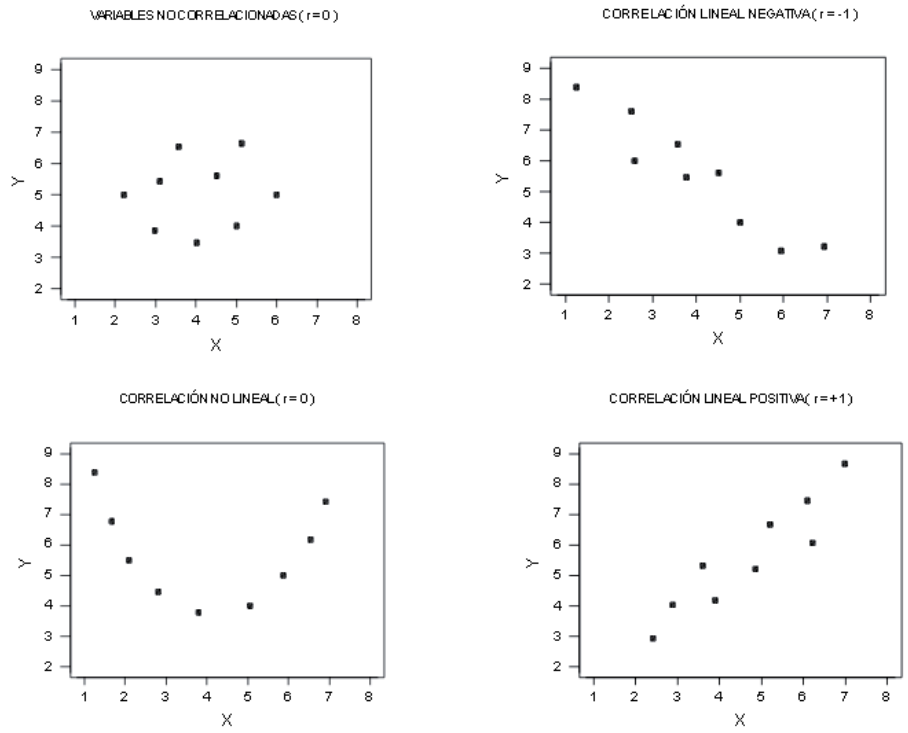

Figura 20: Diferentes tipos de correlación

7.3. Códigos

```r
# Se leen los datos
dat_train <- read.csv("C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/train.csv",
      stringsAsFactors = F)
dat_test <- read.csv("C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/test.csv",
     stringsAsFactors = F)

# Se juntan test y train
dat_test$TARGET <- NA
all_dat <- rbind(dat_train, dat_test)

# Se eliminan las variables constantes
train_names <- names(dat_train)[-1]
for (i in train_names)
{
  if (class(all_dat[[i]]) == "integer")
  {
    u <- unique(all_dat[[i]])
    if (length(u) == 1)
    {
      all_dat[[i]] <- NULL
    }
  }
}

#Se eliminan las columnas duplicadas
train_names <- names(all_dat)[-1]
fac <- data.frame(fac = integer())

for(i in 1:length(train_names))
{
  if(i != length(train_names))
  {
    for (k in (i+1):length(train_names))
    {
      if(identical(all_dat[,i], all_dat[,k]) == TRUE)
      {
        fac <- rbind(fac, data.frame(fac = k))
      }
    }
  }
}
same <- unique(fac$fac)
all_dat <- all_dat[,-same]

#Se eliminan las variables altamente correlacionadas
cor_v<-abs(cor(all_dat))
diag(cor_v)<-0
cor_v[upper.tri(cor_v)] <- 0
cor_f <- as.data.frame(which(cor_v > 0.85, arr.ind = T))
row.names(cor_f)<-NULL
all_dat <- all_dat[,-unique(cor_f$row)]

#Calculamos la varianza para cada variable
variable<-apply(all_dat,2,FUN=var)
#Seleccionamos las que tienen varianza pequena (casi constantes)
varianzaConstante<-variable[variable<0.001]
varsConstantes<-names(varianzaConstante[!is.na(varianzaConstante)])
all_dat<-all_dat[,!colnames(all_dat) %in% varsConstantes]

```

```
58
59
60 # Se vuelve a dividir los datos en test, train y validacion
61 train <- all_dat[(1:nrow(dat_train)), ]
62 validation<-train[sample(nrow(train), 16020), ]
63 idsValidation<-validation$ID
64 train <- train[!train$ID %in% idsValidation,]
65 test <- all_dat[-(1:nrow(dat_train)), ]
66 save(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/train.RData", train)
67 save(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/test.RData", test)
68 save(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/validation.RData",
       validation)
```

Listing 1: Código tratamiento_datos.R

```
1  # #install.packages("xgboost")
2  # #install.packages("ggplot2")
3  library(xgboost)
4  library(ggplot2)
5
6  #Se cargan los datos
7  load(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/train.RData")
8  load(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/test.RData")
9  load(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/validation.RData")
10
11 #Se construye el modelo
12 results_test<-NULL
13 #semillas<-round(runif(20, 0, 100))
14 semillas<-c(67, 31, 57, 47, 46, 21, 54, 64, 86,  3, 37, 29, 20, 56, 41, 40, 69, 83, 16,
       38)
15
16 for (i in semillas){
17   set.seed(i)
18   param <- list("objective" = "binary:logistic",booster = "gbtree",
19                 "eval_metric" = "auc",colsample_bytree = 0.85, subsample = 0.95)
20
21   y <- as.numeric(train$TARGET)
22   print(i)
23   #De las pruebas se obtienen los siguientes parametros
24   xgbmodel <- xgboost(data = as.matrix(train[,colnames(train) != "TARGET"]), params =
         param,
25                       nrounds = 380, max.depth = 5, eta = 0.03,
26                       label = y, maximize = T)
27
28   #Se predice sobre el conjunto de validacion
29   res <- predict(xgbmodel, newdata = data.matrix(validation[,colnames(validation) != "
         TARGET"]))
30   results_test<-cbind(results_test,res)
31 }
32
33 #Se calcula la media de la probabilidad de los 20 modelos
34 res<-apply(results_test, 1, FUN=mean)
35 res <- data.frame(ID = validation$ID, prob = res)
36
37
38 tablaValidacion<-merge(validation, res, by="ID")
39 tablaValidacion<-tablaValidacion[,c("ID","TARGET", "prob")]
40
41 write.csv(res, "C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/submission.csv",
       row.names = FALSE)
```

Listing 2: Código modelo_XGB.R

```r
#install.packages("pscl")
#install.packages("e1071")

library(ggplot2)
library(pscl)
library(e1071)

load(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/train.RData")
load(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/test.RData")
load(file="C:/Users/pablo/Google Drive/Carrera/TFG/Datos/Santander/validation.RData")

# Se seleccionan las variables que parecen ser significativas
myaov<-aov(TARGET~.,data=train)
bestfeatures<-trimws(rownames(summary(myaov)[[1]][which(summary(myaov)[[1]][,5]<0.001),]))

#Se elabora un modelo mediante la regresion logistica respecto a esas variables
formula<-as.formula(paste("TARGET~",paste(bestfeatures,collapse="+")))
mymodel<-glm(formula,data=train,family=binomial(link='logit'))

#Se predice sobre el conjunto de validacion
fitres<-as.numeric(predict(mymodel,validation,type='response'))
tablalogistica<-data.frame(ID=validation$ID, TARGET=validation$TARGET ,prob=fitres)
```

Listing 3: Código regresion_logistica.R

```r
#install.packages("sqldf")
#install.packages("SDMTools")
#install.packages("verification")

library(sqldf)
library(SDMTools)
library(verification)

cont<-0
corte<-0.5

tablaValidacion<-tablalogistica

#Porcentaje de acierto
for(i in 1:nrow(tablaValidacion)){
  if(tablaValidacion[i,"prob"]>corte & tablaValidacion[i,"TARGET"]==1){
    cont<-cont+1
  }
  else if(tablaValidacion[i,"prob"]<corte & tablaValidacion[i,"TARGET"]==0){
    cont<-cont+1
    }
}

porc<-cont/nrow(validation)

print(paste0("Porc acierto es: ",porc))

#Se calcula el lift
tablaValidacionSorted<-tablaValidacion[order(tablaValidacion$prob),]
#Ordenar por deciles
```

```
31 tablaValidacionSorted$decil<-as.numeric(cut(tablaValidacionSorted$prob,quantile(
       tablaValidacionSorted$prob,(0:10)/10),include.lowest=TRUE,lable=TRUE))
32 sqldf("select decil, count(*) as users, sum(target) as positivos, 4.09 as media, round((
       cast(sum(target) as float)/1600*100),2) as mediadecil, round(round((cast(sum(target)
       as float)/1600*100),2)/4.09, 2) as lift  from tablaValidacionSorted group by decil")
33
34 #Matriz de confusion
35 target = tablaValidacionSorted$TARGET
36 pred = tablaValidacionSorted$prob
37
38 #Se calcula la matriz de confusion
39 confusion.matrix(target,pred,threshold=corte)
40
41 #tablaLogisticaSorted<-tablalogistica[order(tablalogistica$prob),]
42
43 # tablaMezclada<-merge(tablaValidacion,tablalogistica, by="ID")
44 # #Area bajo la ROC
45 # datos<-data.frame(prob1=tablaMezclada$prob.x, prob2=tablaMezclada$prob.y)
46 #
47 # roc<-roc.plot(tablaMezclada$TARGET.x, datos, thresholds = NULL, show.thres = F)
48
49 #Se calcula la curva roc
50 roc<-roc.plot(target, pred, show.thres = F)
```

Listing 4: Código validacion_TFG.R

8. Referencias bibliográficas

Referencias

[1] DANIEL T. LAROSE AND CHANTAL D. LAROSE, *Data mining and predictive Analytics*, 2nd ed, 2015

[2] S. CHRISTIAN ALBRIGHT AND WAYNE L. WINSTON, *Business Analytics, Data Analysis and Decision Making*, 6th ed, 2016

[3] WOJTEK J. KRZANOWSKI AND DAVID J. HAND, *ROC Curves for Continuous Data*, 1st ed, 2009

[4] (2015) *Introduction to boosted trees* [Online]. http://xgboost.readthedocs.io/en/latest/model.html

[5] TREVOR HASTIE, ROBERT TIBSHIRANI AND JEROME FRIEDMAN, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 1st ed, 2001

[6] KEITH E. MULLER AND BETHEL A. FETTERMAN, *Regression and ANOVA*, 1st ed, 2002

[7] *Information about Kaggle* [Online]. http://kaggle.com

[8] J. K. SHARMA, *Business Statistics*, 2nd ed, 2007

[9] *Generating tables for LaTeX* [Online]. http://tablesgenerator.com

[10] MARK GARDENER, *Beginning R: The Statistical Programming Language*, 3rd ed, 2012

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Thu Jun 09 20:29:25 CEST 2016
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)