

Trabajo Fin de Máster

Máster Técnicas Estadísticas

Fecha: 31-08-22

Desarrollo de un Priorizador para Campañas Comerciales Empleando Modelos Predictivos

José Damián García Noya

Este documento resume el proyecto del Trabajo Fin de Máster *Desarrollo de un Priorizador para Campañas Comerciales Empleando Modelos Predictivos* dirigido por don Rubén Fernández Casal, Profesor Contratado Doctor de la Universidade da Coruña, don Guillermo López Taboada, Profesor Titular de la Universidade da Coruña y don Juan Manuel Mazaira Gómez, Mánager del equipo de *Modelos Predictivos y Prospección* de ABANCA. Por motivos de confidencialidad no es posible la publicación de la memoria completa.

Resumen

Este proyecto se realizó en el área de *Inteligencia de Clientes* de la entidad ABANCA *Corporación Bancaria S.A* que se dedica a la inteligencia de negocio o *Business Intelligence (BI)*. Su principal objetivo por lo tanto es el uso de distintas estrategias y herramientas a fin de transformar los datos en conocimiento que pueda aplicarse en el proceso de toma de decisiones de la compañía. En particular, el área de *Inteligencia de Clientes* se especializa en el proceso de recopilación y análisis de los datos disponibles en la entidad con el objetivo de adelantarse a las necesidades de sus clientes mediante acciones comerciales afines a cada perfil. Así pues, el área de *Inteligencia de Clientes* es la encargada, entre otras cosas, del desarrollo de modelos analíticos como pueden ser los modelos de propensión a la contratación de productos financieros¹.

Los **modelos de propensión**² son modelos predictivos de clasificación que ordenan la cartera de clientes por su probabilidad estimada de contratar un determinado producto financiero. En particular, este tipo de modelos obtienen estimaciones de la probabilidad de que un determinado cliente vaya a contratar el producto en los meses posteriores (“éxito”), lo que resulta de utilidad a la hora de determinar cuál es el público objetivo de las diferentes acciones comerciales.

Una de las líneas actuales de trabajo del área de *Inteligencia de Clientes* es el desarrollo de un **priorizador para campañas comerciales** que permita mejorar el plan de comunicación de campañas existente utilizando la información de los modelos de propensión junto con otros indicios del interés mostrado por el cliente en dichos productos. A efectos prácticos, sería deseable que el plan de comunicación de campañas comerciales

¹A partir de aquí se hará utilizará simplemente el término producto para hacer referencia a productos financieros

²En lo que sigue, por simplicidad, se hará referencia a los modelos de propensión para la contratación de productos financieros simplemente como modelos de propensión.

priorice el público objetivo de los productos ofertados y seleccione los canales de marketing³ óptimos para el contacto con los distintos clientes según sus necesidades. En otras palabras, lo que se busca es saber **a qué clientes se deben dirigir las campañas de cada producto y a través de qué canales de marketing se deberían comunicar la ofertas** de los mismos, todo ello teniendo en cuenta y respetando las distintas restricciones de negocio.

El enfoque más sencillo para tratar el problema es una metodología conocida en la literatura como *Next-Best Offer (NBO)* o también *Next-Product-to-Buy* (Blattberg et al. 2008). El primer paso del procedimiento nombrado es el desarrollo de un modelo de propensión para cada uno de los productos financieros que se pretenden ofertar. La segunda fase consiste en obtener, a partir de los modelos de propensión anteriores y para cada uno de los clientes, la estimación de la probabilidad de éxito $P(Y = 1|X = x)$ asociada a cada producto. Por último, se asocia a cada cliente la oferta con la mayor probabilidad de éxito, es decir, se le asigna a cada cliente la oferta del producto que es más probable que contrate en los próximos meses.

A pesar de que la metodología anterior es una buena estrategia comercial al estar centrada en el cliente, la progresiva transformación digital de la empresa, acelerada por eventos recientes como la pandemia del *COVID-19*, ha propiciado la creciente necesidad de abordar ciertas **problemáticas**. En primer lugar, la metodología *NBO* no tiene en cuenta en ningún momento la **elección del canal de marketing** por el que se transmite la oferta al cliente. Como es razonable, los clientes son más o menos susceptibles de recibir favorablemente una proposición si se hace por un medio con el que se sienten cómodos de modo que surge la necesidad de escoger el canal de forma adecuada. Tradicionalmente, la elección de canal se lleva a cabo en base al segmento al que pertenece el cliente o por criterios expertos aunque sería deseable basar la decisión en algún tipo de modelo analítico que quizás podría ser más eficiente. Por otro lado, en la metodología *NBO* no existe ninguna preocupación por **restringirse a las necesidades de negocio**. Algunos ejemplos notables de este tipo de requerimientos podrían ser respetar la cantidad máxima de contactos que puede recibir un cliente cada cierto tiempo, la necesidad de realizar un número mínimo de ofertas relativas a ciertos productos, la necesidad de restringirse a la capacidad máxima de cada uno de los canales, etc. Lo que se hace en ocasiones es aplicar las restricciones con posterioridad a la asignación de ofertas a los clientes, lo que se traduce en una disminución de la calidad de la decisión y un aumento considerable de la complejidad asociada a la gestión de campañas. Finalmente, la metodología *NBO* no tiene en cuenta que los **beneficios** asociados a cada uno de los productos ofertados son en principio distintos. Consecuentemente, una buena metodología prioriza los productos más rentables.

Así pues, con el objetivo de resolver las problemáticas anteriores, se ha llevado a cabo una revisión exhaustiva de la literatura existente en el tema derivándose las siguientes conclusiones:

- La manera habitual de tratar el problema es a través de un **modelo de optimización** en el que se incorporan las métricas proporcionadas por los modelos de propensión en la función objetivo. Así pues, se trata de maximizar algún tipo de

³En este proyecto el término canal de marketing hace referencia a los canales utilizados para la comunicación de acciones comerciales.

medida como por ejemplo los **beneficios esperados** para una determinada asignación de ofertas. Algunos artículos en los que se presenta esta metodología son Cohen (2004), Nobibon et al. (2011) y Delanote et al. (2013).

- En lo que respecta a las necesidades de negocio, estas se incorporan fácilmente en el problema de optimización en forma de **restricciones** que la solución debe cumplir. Algunas restricciones propuestas en Cohen (2004) son la mínima tasa de retorno de la inversión, el número máximo y mínimo de productos ofertados de cada tipo, el presupuesto de campaña y la capacidad de los canales de marketing, entre otras.
- Finalmente, en lo que respecta a los canales de marketing, algunos autores como Nobibon et al. (2011) no los tienen en cuenta en ningún momento mientras que otros como Cohen (2004) y Delanote et al. (2013) utilizan modelos de propensión que incorporan por defecto la información del canal y el producto simultáneamente.

En este caso particular, la propuesta de autores como Cohen (2004) o Delanote et al. (2013) no puede aplicarse directamente porque los modelos de propensión disponibles en *ABANCA* no tienen en cuenta el efecto de los canales de marketing. El motivo es que tener en cuenta los canales en los modelos tendría muchos inconvenientes ya que hay productos que no se pueden contratar *online* y, dependiendo del momento del ciclo de vida del cliente y el producto considerado, puede ser preferible recurrir a un gestor o no. Adicionalmente, considerar los canales resultaría en una reducción de la importancia de las variables que se introducen actualmente en los modelos de propensión, lo cual tampoco es deseable. Ante esta situación se propuso incorporar la información del canal de marketing de forma externa considerando que las ofertas que generan una respuesta en el cliente son aquellas tales que el cliente es propenso a contratar el producto ofertado y para las que además el canal utilizado es el **canal preferente para la comunicación de acciones comerciales** dirigidas al cliente. Así pues, suponiendo la independencia entre los sucesos anteriores, se estima la probabilidad de que una oferta genere una respuesta en el cliente como el producto de la probabilidad de que el cliente contrate el producto (obtenida con los modelos de propensión disponibles en *ABANCA*) y la probabilidad de que el canal de marketing utilizado sea el canal preferente. De esta forma surge la necesidad de construir un modelo predictivo que trate de estimar esta última probabilidad.

El primer paso para el desarrollo del **modelo predictivo para el canal preferente** fue la obtención de los datos. Para ello se hizo uso de la información recopilada y trabajada previamente por el equipo de *Modelos Predictivos y Prospección* a la que se accedió por medio del lenguaje *SQL*.

A continuación se proporcionó una definición del concepto de canal preferente consensuada con el equipo de *Modelos Predictivos y Prospección* y basada en la estrategia de negocio en *ABANCA*. Posteriormente, se construyó la variable objetivo en base a la definición anterior y a los datos relativos a la utilización de los diferentes canales de marketing por parte de los clientes para hacer contrataciones. Así pues, la variable objetivo tomó la forma de una variable categórica nominal con tantas categorías como canales de marketing disponibles para realizar las distintas ofertas.

El siguiente paso en la modelización consistió en realizar un análisis exhaustivo de todas y cada una de las variables explicativas disponibles (un total de 227) con el objetivo de entender su relación con la variable objetivo y de seleccionar aquellas que podrían ser de mayor utilidad para la construcción del modelo predictivo. En particular, el proceso

constó de dos etapas. En primer lugar se obtuvieron análisis gráficos (estimaciones no paramétricas de la función de densidad, gráficos de cajas, ...) y medidas de carácter exploratorio de las variables (cuantiles de la distribución, *V de Cramér* para variables categóricas, ...) en relación con la variable objetivo. En un segundo paso se ajustaron iterativamente modelos con una cierta capacidad predictiva como los árboles de decisión para eliminar de forma progresiva las variables con menores medidas de importancia que no producían disminuciones apreciables en las métricas de evaluación, más concretamente, el *AUC* correspondiente a cada una de las categorías de la variable objetivo. El primer paso sirvió para descartar un primer conjunto de variables explicativas para las que no se observaron indicios de que pudieran ser relevantes en la predicción y para reducir la carga computacional del siguiente paso. A su vez, el segundo paso permitió reducir aún más la complejidad asociada al proceso de construcción del modelo eliminando variables no útiles en la predicción manteniendo un número razonable de predictores. Al finalizar el proceso se seleccionaron un total de 42 variables explicativas que terminarían formando parte del modelo final. En particular, la memoria original recoge el análisis exploratorio de las 6 variables que resultaron más influyentes en la predicción mostrando las medidas exploratorias de interés, los gráficos violín asociados y las tablas de contingencia para variables de tipo categórico.

Una vez hecho esto, se prosiguió con la modelización haciendo uso de diferentes algoritmos de aprendizaje automático para resolver el problema de clasificación, en particular, *Random Forest*, *Multi-Layer Perceptron*, *XGBoost*, *LightGBM* y *Super Learner*. Para ajustar cada uno de estos modelos se consideró una partición de los datos disponibles en conjuntos de datos de entrenamiento, validación y test con porcentajes del 60%, 20% y 20%, respectivamente. El proceso más costoso del procedimiento fue la selección de los hiperparámetros de cada uno de los modelos. En particular, se realizaron diversas simulaciones para cada uno de los métodos haciendo uso de varias combinaciones de candidatos para los hiperparámetros de tal forma que para cada uno de los algoritmos se ajustaron, sobre los datos de entrenamiento, varios modelos distintos, cada uno con una combinación concreta de hiperparámetros. Posteriormente, se utilizó el conjunto de datos de validación para seleccionar, de entre las distintas configuraciones de hiperparámetros, la que presentaba los mejores resultados de acuerdo al *AUC* calculado por categorías. Una vez seleccionada la configuración “óptima” de hiperparámetros, la evaluación del modelo escogido se realizó en base al conjunto de datos de test mediante el análisis de la matriz de confusión asociada y todas las medidas que se pueden derivar de la misma (sensibilidad, especificidad, precisión balanceada, puntuación F_1 , valor predictivo positivo *PPV*, valor predictivo negativo *NPV*, ...) así como el análisis de las curvas *ROC* y los *AUC* correspondientes a cada una de las categorías individuales. Finalmente y a modo informativo, se proporcionaron también los macro-promedios, macro-promedios ponderados y micro-promedios asociados a cada una de las cantidades anteriores.

En cuanto al *software* empleado para la construcción de los modelos, se utilizó fundamentalmente el lenguaje de programación *R* para servir de interfaz con los distintos entornos de trabajo. En particular, para construir los modelos *Random Forest* y los *MLP*'s se empleó la librería *h2o* que sirve de interfaz con *H2O*, que es una plataforma de código abierto que permite construir modelos de aprendizaje automático con grandes cantidades de datos (*Big Data*) de forma distribuida y escalable mediante el paradigma *MapReduce*. Por otro lado, para construir los modelos desarrollados por la librería de código abierto *XGBoost* se utilizó la librería *xgboost* de *R* y para construir los modelos desarrollados por la librería de código abierto *LightGBM* se utilizó la librería *lightgbm* de *R*. Por últi-

mo y de forma auxiliar se empleó el lenguaje *Python* por mayor comodidad a la hora de calcular ciertas métricas de evaluación y de generar ciertas representaciones gráficas de los resultados. La ejecución de los distintos programas se realizó en el servidor analítico de la entidad lo que permitió una mayor eficiencia computacional al posibilitar la realización de algunas tareas de forma paralela. En particular, los entornos de desarrollo integrados (*IDE*) utilizados fueron *RStudio Server* y *Jupyter Lab* para *R* y *Python* respectivamente.

Luego de analizar los resultados se observó que el modelo con los mejores *AUC* por categoría era el que proporcionaba el algoritmo *Super Learner* pero, debido a que la complejidad del modelo no justificaba el pequeño incremento de los *AUC* con respecto a los de otras metodologías, se optó por seleccionar el siguiente mejor modelo. Así pues, dado que los siguientes mejores modelos eran los proporcionados por *XGBoost* y *LightGBM* siendo sus *AUC* muy similares, se optó por realizar un pequeño estudio del tiempo computacional y seleccionar el más eficiente. Más concretamente, el estudio consistió en ajustar los modelos sobre los mismos datos de entrenamiento unas 30 veces obteniendo en cada caso el tiempo requerido para realizar el ajuste. Los resultados mostraron que el algoritmo implementado por *LightGBM* era en promedio unas 17 veces más rápido que el implementado por *XGBoost* de modo que se seleccionó como modelo final el proporcionado por *LightGBM*. Finalmente se analizaron las variables más influyentes en la predicción de acuerdo a la reducción de la impureza en los nodos de los árboles de decisión que formaban parte del ensamble *LightGBM* y se estudiaron los efectos principales de acuerdo al modelo escogido mediante la utilización de los gráficos *PDP* (*Partial Dependence Plot*) asociados. En particular, la memoria original recoge el estudio de las 6 variables que resultaron tener las mayores medidas de importancia.

En cuanto al problema de optimización, este se modelizó a través de un modelo de **programación lineal y entera (ILP)**. Las variables de decisión utilizadas fueron variables binarias de tal forma que permitieran determinar para cada una de las ofertas posibles si debería realizarse o no. A su vez, la función objetivo se definió como los beneficios netos esperados al realizar un conjunto particular de ofertas. Estos beneficios netos esperados se estimaron a través de la probabilidad de que la oferta generara una respuesta en el cliente y las estimaciones existentes del margen ordinario anual generado por los diferentes productos y del coste marginal asociado a la realización de ofertas por ciertos canales. Por último, se incorporaron ciertas necesidades de negocio apropiadas para el problema concreto en forma de restricciones.

Finalmente, para el **método de resolución** del problema se optó por utilizar un *software* de optimización, también denominado *solver*, que incorpora fundamentalmente métodos deterministas como el algoritmo *Branch & Cut* y algún tipo de algoritmo heurístico como método auxiliar. Particularmente, dado que en la actualidad *ABANCA* no disponía de licencia para *solvers* comerciales como *CPLEX* o *Gurobi*, que son los más eficientes en la resolución de los problemas, se usaron para este fin los *solvers* de código abierto incluidos en el proyecto *COIN-OR* (*Computational Infrastructure for Operations Research*). Por último, para servir de interfaz entre los datos del problema y el *solver* se utilizó el paquete *Pyomo* (*Python Optimization Modeling Objects*) de *Python* como herramienta de **modelado algebraico** del problema.

Referencias

- [1] Alonzo T.A, Pepe M.S (2002). *Distribution-free ROC analysis using binary regression techniques*. 3(3):421-32.
- [2] Balas E, Ceria S, Cornuejols G, Natraj N (1996). *Gomory cuts revisited*. Operations Research Letters. 19(1):1-9
- [3] Bazaraa M, Jarvis J, Sherali H (2010). *Linear programming and networks flows*. John Wiley and Sons.
- [4] Beaulieu A (2009). *Learning SQL* (Second Edition). O'Reilly Media.
- [5] Bhatti M.A (2000). *Practical optimization methods*. Springer-Verlag.
- [6] Blattberg R.C, Kim B, Neslin S.A (2008). *Database Marketing Analyzing and Managing Customers*. Springer.
- [7] Bradley S.P, Hax A.C, Magnanti T.L (1977). *Applied Mathematical Programming*. Addison Wesley.
- [8] Bradley A.P (1997). *The use of the area under the ROC curve in the evaluation of machine learning algorithms*. Pattern Recognition 30(7):1145-1159.
- [9] Breiman L, Friedman J, Olshen R, Stone C (1984). *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA.
- [10] Breiman L (1996). *Bagging predictors*. Machine Learning, 24(2), 123-140.
- [11] Breiman L (2001). *Statistical modeling: The two cultures (with comments and a rejoinder by the author)*. Statistical Science. 16(3):199-231.
- [12] Chen T, Guestrin C (2016). *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 16.
- [13] Cohen M.D (2004). *Exploiting response models-optimizing cross-sell and up-sell opportunities in banking*. Information Systems. 29:327-341.
- [14] Goodfellow I, Bengio Y, Courville A (1943). *Deep Learning*. MIT Press. <https://www.deeplearningbook.org>
- [15] Cybenko G (1989). *Approximation by superpositions of a sigmoidal function*. Mathematics of Control, Signals, and Systems. 2:303-314.
- [16] Delanote S, Leus R, Nobibon F.T (2013). *Optimization of the annual planning of targeted offers in direct marketing*. Journal of the Operational Research Society. 64:1770-1779.
- [17] Dietterich T.G (2000). *An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization*. Machine Learning. 40(2):139-157.

- [18] Efron B (1979). *Bootstrap methods: Another look at the jackknife*. The Annals of Statistics. 7(1):1-26.
- [19] Fawcett, Tom (2006). *An introduction to ROC analysis*. Pattern recognition letters 27(8):861-874.
- [20] Fernández-Casal R, Costa-Bouzas J, Oviedo M (2022). *Aprendizaje Estadístico*. https://rubenfcasal.github.io/aprendizaje_estadistico/.
- [21] Freund Y, Schapire R.E (1996). *Experiments with a new boosting algorithm*. Thirteenth International Conference on ML. 41(1):67-95.
- [22] Friedman J.H (2001). *Greedy function approximation: a gradient boosting machine*. The Annals of Statistics. 29(5):1189-1232.
- [23] Friedman J.H (2002). *Stochastic gradient boosting*. Computational Statistics & data analysis. 38(4):367-378.
- [24] Friedman J.H, Popescu B.E (2008). *Predictive learning via rule ensembles*. The Annals of Applied Statistics. 2(3):916-954.
- [25] Gareth J, Hastie T, Tibshirani R, Witten D (2013). *An Introduction to Statistical Learning with Applications in R*. Springer Texts in Statistics.
- [26] Gedeon T (1983). *Data mining of inputs: Analysing magnitude and functional measures*. International Journal of Neural Systems. 24(2):123-140.
- [27] Géron A (2019). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (Second Edition). O'Reilly Media.
- [28] Goldstein A, Kapelner A, Bleich J, Pitkin E (2015). *Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation*. Journal of Computational and Graphical Statistics. 24(1):44-65.
- [29] Gomory R (1958). *Outline of an algorithm for integer solutions to linear programs*. Bulletin of American Mathematical Society. 64:275-278.
- [30] Greenwell B.M (2017). *pdp: An R Package for Constructing Partial Dependence Plots*. The R Journal. 9(1):421-436.
- [31] Hand D.J, Till R.J (2017). *A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems*. Machine Learning. 45:171-186.
- [32] Hanley J.A, McNeil B.J (1982). *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. Radiology 143:29-36.
- [33] Hart W, Watson J, Woodruff D (2001). *Pyomo: modeling and solving mathematical programs in Python*. Mathematical Programming Computation. 3(3):219-260.
- [34] Hart W, Laird C, Watson J, Woodruff D, Hackebeil G, Bethany L, Siirola J (1992). *Pyomo – Optimization Modeling in Python*. Springer.

- [35] Hastie T, Tibshirani R, Friedman J (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Second Edition). Springer Series in Statistics.
- [36] Hornik K, Stinchcombe M, White H (1989). *Multilayer feedforward networks are universal approximators*. Neural Networks. 2:359-366.
- [37] Jurafsky D, Martin J.H (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Third Edition). Prentice Hall.
- [38] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T (2016). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. Advances in Neural Information Processing Systems. 30.
- [39] Kearns M, Valiant L (1994). *Cryptographic limitations on learning Boolean formulae and finite automata*. Journal of the ACM. 41(1):67-95.
- [40] Koopialipour M, Noorbakhsh A (2020). *Applications of Artificial Intelligence Techniques in Optimizing Drilling* in A. Azizi (ed.) Emerging Trends in Mechatronics. Intech Open.
- [41] Lougee-Heimer R, (2003). *The Common Optimization Interface for Operations Research*. IBM Journal of Research and Development. 47(1):57-66.
- [42] McCulloch W, Pitts W (1943). *A Logical Calculus of Ideas Immanent in Nervous Activity*. Bulletin of Mathematical Biophysics. 5(4):115-133.
- [43] Molnar C (2020). *Interpretable Machine Learning*. <https://christophm.github.io/interpretable-ml-book>.
- [44] Nemhauser G.L, Wolsey L.A (1988). *Integer and Combinatorial Optimization*. John Wiley and Sons.
- [45] Nesterov Y (1983). *A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$* . Doklady ANSSSR (translated as Soviet.Math.Docl.). 269:543-547.
- [46] Nobibon F.T, Leus R, Spieksma F.C.R (2011). *Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms*. European Journal of Operational Research. 210:670-683.
- [47] Padberg M, Rinaldi G (1991). *A branch-and-cut algorithm for the resolution of large-scale symmetric travelling salesman problems*. SIAM Review. 33:60-100.
- [48] Ripley B.D (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.
- [49] Rumelhart D, Hinton G, Williams R (1986). *Learning representations by back-propagating errors*. Nature. 323:533-536.
- [50] Valiant L.G (1984). *A Theory of the Learnable*. Communications of the ACM. 27(11):1134-1142.
- [51] Walpert D (1992). *Stacked generalization*. Neural Networks. 5(2):241-259.

[52] Wolsey L.A (1998). *Integer Programming*. John Wiley and Sons.