Carrera de Especialización en Métodos Cuantitativos para la gestión y análisis de datos en Organizaciones

---

Trabajo Final de Especialización

Un modelo de predicción de próxima compra aplicado a bancos

Aplicación de técnicas de minería de datos en un entorno de Python para mejorar la estrategia de marketing directo.

Autor: Kramarz, Brian Ariel

Diciembre 2019

Introducción

Durante las últimas décadas, los bancos se han convertido en uno de los tantos tipos de organizaciones que han comenzado a recopilar datos de sus clientes de forma sistemática. Si bien existen ciertos datos que podemos esperar que toda empresa tenga respecto de sus clientes, como por pueden ser ciertos datos personales como domicilio, datos de contacto, nacionalidad, entre otros, debido a su tipo de actividad, los banco tienen también mucha otra información más sensible como podría ser el estado civil, si un cliente tiene gente a cargo, e incluso sus ingresos periódicos. Es más, con el avance de los pagos por medios electrónicos, ya sea tarjeta de crédito y/o debido, pago mis cuentas y las nuevas billeteras virtuales, los bancos puede conocer que consumimos, cuando y en donde realizamos nuestras compras, y muchos más respecto de nuestros consumos sean estos habituales o no. Cabría preguntarse cuál, si alguna de toda esa información, los bancos podrían utilizarse para sus propios intereses, lo que sin dudas nos llevaría a un debate respecto de la ética y la gestión de los datos en contextos organizacionales.

Muchas empresas, y en particular las pertenecientes a la industria de financiera, han ido modificando, y probablemente evolucionando en su estrategia de marketing. En sus inicios las empresas concentraban su estrategia de publicidad en el uso de los medios de comunicación masivos, justamente con el objetivo de alcanzar a la mayor cantidad de usuarios posible. La efectividad de este tipo de publicidad es compleja de medir, pero según Patrutiu-Baltes (2016)[i], podría concluirse que ha sido históricamente baja, debido principalmente a la gran cantidad de empresas y productos compitiendo por la atención del usuario. Muchas empresas comenzaron a utilizar nuevas estrategias de marketing basadas principalmente en datos propios de los clientes, con el objetivo de mejorar el ratio de respuesta y eficientizar las campañas de marketing. Una de las principales estrategias dentro de esta nueva rama el marketing ha sido el Telemarketing. Este consiste, básicamente, en que distintos “asesores” o “vendedores” contactan en forma directa a los potenciales clientes para ofrecer e intentar vender un determinado producto. Inicialmente ese contacto fue de forma telefónica, pero, hoy en día, gracias a la evolución de los medios de comunicación también se puede dar por otras vías o mecanismos. El telemarketing puede ser aplicado en todo tipo de industrias, pudiendo ofrecer bienes y servicios de todo tipo, pero sin duda una de las ramas que me ha aprovechado y explotado esta estrategia ha sido el sector bancario (Korkmaz, 2017)[ii].

En principio, un banco que quisiera realizar una campaña de marketing promocionando un nuevo producto, podría contactarse de forma telefónica, con cada uno de sus actuales clientes, contándole de las supuestas ventajas de este nuevo producto, de lo conveniente que fuera para el cliente contar con el mismo, e incentivándolos a que lo adquiera. Resulta evidente que esta no sería una estrategia para nada eficiente debido al alto costo que implicaría contactar a toda la cartera de clientes, cuando sería totalmente predecible que existe un subconjunto de cliente que a priori no tendría ningún interés en adquirir este nuevo producto en cuestión. Cabe destacar que el costo asociado a cada potencial cliente implica no solo el costo de la llamada telefónica sino el tiempo empleado por el recurso que la está realizando.

En este sentido, podemos entender que un objetivo de una empresa del sector financiero sea utilizar los datos personales propios de cada cliente para segmentar su cartera de clientes con algún propósito en particular. Considerando lo expuesto, uno de los requisitos para que una campaña de marketing sea no sólo exitosa, sino que además efectivamente eficiente es una buena segmentación del mercado. Podemos esperar que el banco tenga no solo los datos de contacto de cada uno de sus clientes, sino que muchos otros datos respecto de su situación socioeconómica, intereses y registros de gastos que les permitirían mejorar su clasificación entre cliente que se encontrarían interesados en adquirir el nuevo producto y quienes no, para concentrarse en el primer grupo. De la mano con lo afirmado por Fayyad, se requiere una nueva generación de técnicas y computacionales y herramientas para permitir la extracción de conocimiento útil de los grandes volúmenes de datos, que se encuentran en un proceso de crecimiento agigantado. Estas técnicas y herramientas son el sujeto central del campo emergente de descubrimiento de conocimiento en bases de datos (KDD) y minería de datos (Fayyad, 1996)[iii]

Finalmente, a fin de lograr extraer los patrones y predecir un modelo a partir de todos los datos con los que cuenta esta organización en particular, será necesario el análisis de datos en un contexto de grandes volúmenes de Información. Esto se debe no sólo a la gran cantidad de registros con los que cuenta un banco, sino que además por la gran variedad de fuentes de donde pueden provenir dichos datos. A fin de almacenarlos y procesarlos, es necesario que un banco posea un diseño eficiente de bases de datos a fin de que los mismos puedan ser procesados de forma rápido, seguro y confiable.

El objetivo del presente trabajo consistirá entonces en la selección de un modelo de minería de datos que permite la clasificación de los clientes bancarios entre aquellos que adquirirán un determinado producto, a fin de optimizar la estrategia de Marketing Directo del Banco Santander. A tales fines, será necesario estudiar en qué consiste la metodología de extracción de conocimiento a partir de los datos (KDD) y como la misma puede ser utilizada.

Considerando lo expuesto, el objetivo general del presente trabajo consiste entonces en seleccionar el mejor modelo predictivo que permita segmentar la cartera de cliente del Banco Santander entre aquellos que estarían interesados en adquirir un determinado nuevo producto bancarios y aquellos que no. A tales fines en primer lugar será necesario también entender el concepto de Extracción de Conocimiento a partir de Datos (o Knowledge Discovery in Databases - KDD) y su aplicación a los grandes volúmenes de información en general y al caso de estudio en particular. Asimismo, identificar los distintos métodos predictivos, estudiar sus particularidades y compararlos entre sí. Calibrar cada uno de los modelos en un entorno de Python a fin de seleccionar los parámetros óptimos de cada uno de los modelos. Otro de los objetivos específicos consistirá en analizar cómo se pueden comparar la performance de cada uno de los modelos y compararlos contra un bench-mark preestablecido. Finalmente, una vez seleccionado el modelo, analizar cuáles son aquellas variables que le permitirían al banco establecer las bases de segmentación de su cartera de clientes a fin de realizar recomendaciones sobre a qué clientes direccionar su campaña de marketing.

Para analizar como un banco puede clasificar a sus clientes entre aquellos que podrían adquirir un nuevo producto y quienes no estarían interesados, mediante la aplicación de métodos analíticos descriptivos, predictivos y prescriptivos, en la sección N° 2 se comenzará por plantear el marco conceptual necesario para el desarrollo de este trabajando. Se desarrollarán los conceptos que nos permitirán comprender el rol que las técnicas de minería de datos pueden tener en la estrategia de marketing directo de una empresa. En un tercer apartado, se hará una breve descripción de la base de datos, como ha sido obtenida junto con una pequeña reflexión respecto de la responsabilidad de los datos que la misma contiene. Asimismo, en esta sección se hará una descripción de las técnicas y algoritmos planteados. En la Sección Nº 4, se hará un breve análisis de los modelos propuestos y que propuestas se pueden obtener a partir de los mismos. Finalmente, en la Sección Nº 5 se exponen una serie de conclusiones y trabajos finales que podrían surgir a partir de la presente investigación.

Contenido

1. El Marketing y Minería de Datos

En el presente apartado se realizará un análisis de los principales conceptos teóricos que nos permitirán abordar la problemática planteada en la presente investigación.

1.1. Marketing Directo

El Marketing de Masas ("*mass marketing*") es una de las principales ramas del marketing, y la herramienta más tradicional empleada por las empresas para promocionarse a sí mismas o sus productos. Incluye todas las herramientas y medios mediante las cuales las empresas buscan transmitir un mensaje, siendo su principal objetivo llegar a la mayor audiencia posible, mediante el uso de distintos medios de comunicación, principalmente avisos en televisión y/o radio, publicaciones en diarios y revistas, entre otros. De acuerdo con McDaniel (2009)[iv] dentro de las ventajas identificadas del Marketing Directo se encuentran su gran masividad, y debido a esto, sus bajos costos por unidad expuesta, tantos de producción como de investigación. Sin embargo, en los últimos años, el mercado se ha puesto mucho más competitivo, y hay una variedad mucha más amplia de productos, donde todas las marcas hacen un gran esfuerzo por diferenciarse de su competencia. Si bien como se ha mencionada que este tipo de marketing llegar a audiencia altamente numerosas, dentro de sus desventajas se encuentra su baja tasa de respuesta, es decir el reducido porcentaje de personas que efectivamente compran, y más aún difícil de calcular quiénes de los clientes compraron el producto por haber visto la publicidad, volviendo así cada vez menos efectivo, y en otros términos no tan eficiente.

En contraposición al Marketing de Masas, en los últimos años se ha desarrollado una nueva rama del marketing denominada Direct Marketing (o Marketing Directo). El mismo consiste en el estudio de las características y necesidades de los potenciales clientes con el objetivo de segmentar la cartera de clientes de una empresa e identificar aquellos que potencialmente podrían y querrían adquirir un determinado producto, con el fin crear estrategias de marketing direccionadas específicamente hacia ellos con el objetivo de incentivarlos a que adquieran el producto. Algunas de estas estrategias podrían incluir el Telemarketing, mailing, couponning, televenta, email marketing, entre otras.

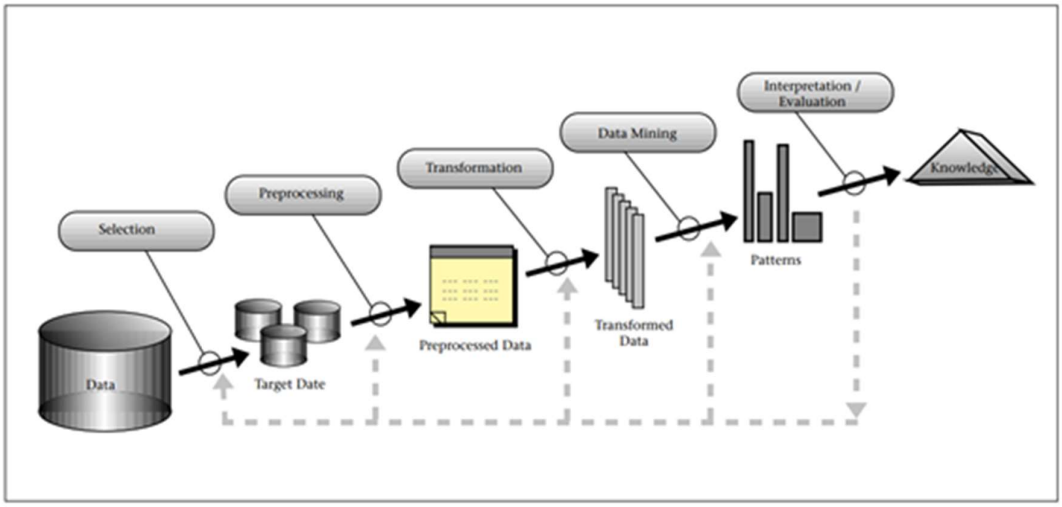
1.2. KDD

Con el transcurso del tiempo y sobre todo dependiendo del campo de estudio se le ha acuñado nombres distintos al proceso o la noción de encontrar patrones útiles a partir de los datos, incluyendo extracción de conocimiento, minería de datos, descubrimiento de información, arqueología de datos, entre tantos otros. El término Descubrimiento de conocimiento a partir de Bases de Datos (KDD) fue primeramente acuñado por Piatestsky y Schapiro[v], con el objetivo de enfatizar que el producto final del proceso radica justamente en el conocimiento con el objetivo de poder basarse en dato (data-driven). Asimismo, los mencionados autores resaltan la importancia que el proceso KDD es un concepto más amplio que las técnicas de minería de datos propiamente dichas, siendo estas tan solo un paso del proceso completo de proceso de extracción de conocimiento a partir de los datos.

De esta forma, Piatestsky y Schapiro plantean que dicho proceso podría subdividirse en 5 etapas, siendo estas no excluyentes ni exhaustivas. Según los autores, la primera etapa consiste en los relativo a la **Preparación de los datos,** la cual tuvo un impulso justamente al estudio por las bases de datos (haciendo referencia a la segunda D del proceso KDD). Dado que muchos de los algoritmos de minería de datos suponen que los datos se encuentran cargados en la memoria, las técnicas de bases, como por ejemplo el agrupamiento y ordenamiento de datos y la optimización de consultas a la base de datos, se vuelven sumamente importante con el objetivo de ganar eficiencia en el proceso. La segunda etapa consiste en la **Selección de los datos**. Parte del proceso consiste en entender la aplicación, que se desea obtener, el conocimiento relevante ya existente y el objetivo del proceso desde la perspectiva del cliente. Una vez considerado todo esto, se debe seleccionar el set de datos con el que se va a trabajar, sean esto una recopilación de distintas fuentes de datos, o un subconjunto de toda la base existente. Comúnmente, está acompañada de la tercera etapa que describen los autores que consiste en la **Limpieza de datos**. Muchas veces esta es la tarea más dificultosa y laboriosa, y sobre la cual el cientista o especialista de datos dedica una gran proporción de su tiempo, mucho más de lo deseado. Esta etapa contempla todas las tareas necesarias para poder procesar la base de datos seleccionadas, desde la extracción de "ruido", la estrategia para los valores perdidos, y hasta la operacionalización para que toda la información puede ser procesada por cada uno de los algoritmos (por ejemplo, ciertos algoritmos solo aceptan cierto tipo de formato en los atributos, ya sean estos valores numéricos o binarias, por lo que es necesario procesos de discretización de variables, creación de variables binarias, entre otros).

La cuarta etapa consiste propiamente en la **Minería de Datos**. Como se ha mencionado, la minería de datos es tan solo un paso en el proceso de KDD, y consiste en aplicar técnicas de análisis de datos y algoritmos de descubrimiento de patrones que, bajo limitaciones computacionales de eficiencia razonables, produzcan a una determinada enumeración de patrones o modelos sobre los datos. Sin dudas en los últimos años esta ha sido la etapa que más ha evolucionado a la par del avance en los procesadores y capacidad de las computadoras.

Finalmente, la última etapa del proceso de KDD consiste en la **Interpretación de resultados.** La última etapa consiste en la interpretación de los patrones obtenidos, y probablemente volver a algunas de las etapas anteriores para realizar una iteración, plantear nuevas variables, cambiar algún parámetro y hasta proponer un nuevo modelo. En esta etapa hay que entender que el usuario muchas veces no solo está interesado en las capacidades predictivas, sino que además encontrará mucho valor en entender qué es lo que dice el modelo.



El proceso de extracción de conocimiento a partir de batos puede ser clasificado en función de los objetivos, los cuales podemos agruparlos en dos principales. En primer lugar, el usuario o cliente puede tener el objetivo de ***verificar*** alguna hipótesis, sea esta ya existente o bien ya verificar que la misma se mantiene en un nuevo conjunto de datos. El segundo objetivo puede ser propiamente el de ***descubrimiento***, es decir la extracción de nuevo patrones. Asimismo, el proceso de descubrimiento se puede subdividir entre *Predicción* y *Descripción,* siendo el primer utilizado para predecir el comportamiento futuro de cierta entidad, y el segundo destinado a encontrar patrones que sean entendibles para la mente humana, y una suerte de clasificación o segmentación.

1.3. La etapa de minería de datos dentro de KDD

Habiendo definido el proceso de KDD y cada una de sus etapas, se propone ahora hacer un foco en el componente de minería de datos ya que esta suele ser la que más atención recibe. Según Fayyad la minería de datos consiste en ajustar modelos a la información observada. Dicho modelo ajustado juega el rol de conocimiento inferido. El hecho de que el modelo refleje conocimiento ya sea útil, interesante o veraz por parte de los datos ya forma parte del objetivo global del proceso de KDD.

En línea con esta visión, Aggarwal (2015)[vi] define a la minería de datos como el estudio de la recolección, limpieza, procesamiento, análisis y obtención de ideas valiosas a partir de los datos. El término de minería de datos es un amplio paraguas que es utilizado para diferentes aspectos del procesamiento de datos, y que puede ser aplicado en la resolución de una variedad de problemas sumamente amplios.

Una tercera definición de Minería de Datos puede ser la aportada por Witten, Frank y Hall (2016)[vii], según los cuales es el proceso de descubrir patrones a partir de los datos, donde este proceso debe realizarse de forma automática, o semiautomática. Los patrones encontrados deben ser significativos, en el sentido que deben permitir la obtención de cierta ventaja. Los patrones nos permiten realizar una predicción no trivial sobre una futura ocurrencia.

La mayoría de las técnicas de minería de datos se basan en técnicas probadas y testeadas del aprendizaje automático, reconocimiento de patrones y estadísticas. Asimismo, el verdadero modelo subyacente a la mayor parte de los algoritmos de minería de datos no es más que alguna de las siguientes: polinomios, splines, kernels y funciones básicas, entre otros modelos matemáticos

Asimismo, la etapa de Minería de Datos puede subdividirse en 3 componentes principales. El primero es la representación mediante un modelo, y consiste en el lenguaje o modelos seleccionado para descubrir los patrones. Es importante que el cientistia de datos tenga un amplio entendimiento de cada una de las familias de modelos, y los supuestos que puede haber subyacente a cada uno de ellos.

El segundo componente consiste en la selección de un criterio de evaluación de los modelos. Estos criterios puede ser valor cuantitativo o una función de ajuste que mida que tan bien (o mal) un determinado modelo cumple sus objetivos en el proceso de KDD. Finalmente, el tercer componente consiste en la optimización del modelo, es decir en la selección de los parámetros que maximice el criterio de evaluación seleccionado. Una vez encontrado los parámetros óptimos, se puede volver a comenzar seleccionando una nueva familia de modelos.

Como se ha mencionado anteriormente, los dos principales objetivos de la minería de datos en la práctica suelen ser ya sea la predicción o la descripción. Aunque dicha distinción suele ser más a fines de entender los objetivos del usuario final, en la práctica los límites entre descripción y predicción no son muy estrictos, y los modelos de predicción puede ser utilizados para describir y viceversa. Ya sea con el objetivo de predicción o de descripción, la mayoría de las técnicas de minería de datos puede ser agrupadas en diversas categorías, siendo la primera de ellas los algoritmos de **Clasificación.** Weiss y Kulikowski[viii] definieron a los modelos de clasificación como aquellos algoritmos cuyo objetivo sea aprender una función que mapee (clasifique) un registro entre una de muchas clases predefinidas Una segunda categoría la comprenden los modelos de **Regresión.** Los modelos de regresión son aquellos que aprenden una función que mapea un registro a un cierto valor real. Asimismo, existen otros algoritmos de **Clustering.** Jain y Dubes[ix] definieron al proceso de clusterización como una tarea descriptiva donde el objetivo es encontrar un número finito (y relativamente chico) de categorías (o clusters) que le permitan a uno lograr un mejor entendimiento del conjunto de datos. Así cada uno de los registros es asignado a un determinado cluster, o de forma más avanzada, o más de un cluster teniendo cada uno de ellos una cierta probabilidad. Finalmente, la última categoría la comprenden los algoritmos de **Sumarización.** Estos consisten en los mecanismos para lograr compactar un conjunto de registros en una única categoría. El ejemplo más simple podría ser imputar la media y el desvió de todos los registros individuales que formarían parte de un determinado subconjunto, pero existen mecanismos más sofisticados como los propuestos por Agrawal[x], a través de la derivación de reglas generales.

Habiendo realizado un desarrollo conceptual de lo que implica la extracción de conocimiento a partir de los datos, de las distintas etapas de este proceso, y hecho foco que una de las más importante es precisamente la de minería de datos, se desarrollarán a continuación algunos de los algoritmos clasificación. El espectro de posibles algoritmos es realmente amplio, y es el

investigador quien de acuerdo a su conocimiento y los datos con los que se encuentre trabajando deberá seleccionar una familia de modelos o un conjunto de familias. El o los algoritmos a utilizar depende de un conjunto de factores siendo una de las más importantes justamente el tipo de datos con los que se cuenten.

Uno de los modelos más conocidos y ampliamente utilizados son los Árboles de Decisión[xi]. Dentro de las principales razones por las cuales este algoritmo es uno de los más populares y utilizados a la hora de realizar una clasificación se debe a que el mismo tiende a mimetizar el modo de racionalizar que usan las personas. A diferencia de otros modelos que funcionan como una caja negra, este algoritmo es fácil de entender, al menos en un caso reducido. Para crear un modelo de este tipo, se debe comenzar seleccionando un atributo que se coloca en el nodo raíz y se crea una rama para cada valor posible. Esto divide el conjunto de ejemplos en subconjuntos, uno para cada valor del atributo. El proceso continúa repitiéndose de forma recursiva para cada rama, utilizando solo aquellas instancias que realmente llegan a la rama. Si en algún momento todas las instancias en un nodo tienen la misma clasificación, deja de desarrollar esa parte del árbol. El árbol generalmente se construye como una partición jerárquica de los ejemplos de capacitación, así como el algoritmo de agrupamiento de arriba hacia abajo divide los datos jerárquicamente.

La principal diferencia de la agrupación es que el criterio de partición en el árbol de decisión se supervisa con la clase etiqueta en las instancias de entrenamiento. El algoritmo posee diversos parámetros los cuales se pueden ir modificando para optimizar la precisión de la clasificación. Dentro de estos parámetros se destacan: La cantidad de ramas (o niveles), el criterio de selección de las ramas, el tamaño mínimo para crear una nueva rama, entre otros. Asimismo, existen diversos métodos de “poda” y detención de profundidad para evitar que el modelo se sobre-ajuste.

Dentro de los modelos de clasificación otros de los modelos ampliamente reconocidos consisten en la regresión logística modela directamente las probabilidades de pertenencia a clases en términos de las variables de características con una función discriminativa. De esta forma se deben estimar los parámetros del modelo probabilístico. El análisis de regresión logística se enmarca en el conjunto de Modelos Lineales Generalizados (GLM) que usa como función de enlace la función *logit*. Las probabilidades que describen el posible resultado de un único ensayo se modelan, como una función de variables explicativas, utilizando una función logística.

$$\operatorname{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}.$$

De la misma manera, otro algoritmo muy utilizado a la hora de clasificación es el de Support Vector Machine (SVM)[xii]. Este algoritmo está definido de forma natural para clasificaciones binarias, es decir cuando la variable a predecir puede tomar una de dos clases posibles, aunque también puede ser generalizado para clasificaciones de más clases. Este algoritmo se basa en un algoritmo de aprendizaje general y un núcleo de problema específico que computa el producto interno de los puntos de datos de entrada en un espacio de características. SVM realiza la clasificación al construir un hiperplano N-dimensional que separa de manera óptima los datos en dos categorías. Los modelos SVM están estrechamente relacionados con Redes neuronales. De hecho, un modelo SVM que utiliza una función de núcleo sigmoide es equivalente a una percepción de dos capas red neuronal El espacio de entrada se mapea mediante una transformación no lineal en una característica de alta dimensión espacio. El objetivo del modelado SVM es encontrar el hiperplano óptimo que separe los conjuntos de datos de tal manera que el margen entre los conjuntos de datos se maximiza. Los vectores cerca del hiperplano son los vectores de soporte. En otras palabras, el límite de decisión debe estar lo más alejado posible de los datos de ambas categorías

Las redes neuronales son un modelo de simulación del sistema nervioso humano. El sistema nervioso humano está compuesto por células, denominadas neuronas. Las neuronas biológicas están conectadas a una otro en los puntos de contacto, que se denominan sinapsis. El aprendizaje se realiza al cambiar la fuerza de las conexiones sinápticas entre las neuronas. Típicamente, la fuerza de estas conexiones cambia en respuesta a estímulos externos. Las redes neuronales pueden ser consideradas una simulación de este proceso biológico.

Como en el caso de las redes biológicas, los nodos individuales en redes neuronales artificiales se denominan neuronas. Estas neuronas son unidades de cálculo que reciben información de algunas otras neuronas, hacen cálculos en estas entradas y las alimentan en otras neuronas.
La función de cálculo en una neurona se define por los pesos en la entrada conexiones a esa neurona. Este peso puede verse como análogo a la fuerza de una conexión sináptica. Al cambiar estos pesos adecuadamente, la función de cálculo se puede aprender, lo que es análogo al aprendizaje de la fuerza sináptica en biológico Redes neuronales. El "estímulo externo" en redes neuronales artificiales para aprender estos los pesos son proporcionados por los datos de entrenamiento. La idea es modificar incrementalmente los pesos siempre que el conjunto actual de pesos haga predicciones incorrectas. Estos modelos son altamente flexibles, pero al mismo tiempo complejos, y su buen desempeño despende altamente de una correcta definición de una serie de instancias, dentro de las cuales se pueden mencionar en primer lugar la estructura de la red, es decir la cantidad de capas y nodos para cada una de ellas. En segundo lugar, en necesario definir una serie de funciones, tanto de activación como de perdidas, y la forma de calcular los ponderadores. Asimismo, se debe definir hiper parámetros para el proceso iterativo. Las redes neuronales son un algoritmo altamente flexible, pero requieren de un alto poder computacional. La clave para la efectividad de la red neuronal es la arquitectura utilizada existiendo una gran variedad de alternativas para cada una de las instancias definidas.

A la hora de utilizar modelos para la clasificación, otra de las alternativa consiste en anidar distintos modelos, sean estos del mismo o de distinto tipo todos ellos con el objetivo de mejorar la performance del mejor modelo individual que se obtuvo hasta el momento. En caso de que se aniden modelos del mismo tipo, la idea es que cada uno de ellos tenga distintos parámetros, pero también se puede utilizar modelos pertenecientes a distintas familias de modelos. La estimación final será la ponderación de las estimaciones individuales de cada uno de los modelos. Un caso particular de los modelos ensamblados se presenta cuando los modelos que se anidan son modelos de la misma familia, comprende el caso de cuando cada modelo individual es un árbol de decisión. De esta forma se proponen un conjunto de árboles de decisión, y que juntos forman un bosque, siendo cada uno de ellos entrenados con una parte de la muestra. Cada árbol es independiente de los demás. En este caso además de definir cada uno de los parámetros presente en los árboles de decisión, se deberá definir el tamaño del bosque, o cuantos árboles se desean anidar.

El concepto fundamental detrás del bosque aleatorio es simple pero poderoso: la sabiduría de las multitudes. En ciencia de datos, la razón por la que el modelo de bosque aleatorio funciona tan bien es debido a que implementa una gran cantidad de modelos (árboles) relativamente no correlacionados que operan como comité superará a cualquiera de los modelos constituyentes individuales. La baja correlación entre modelos es la clave. Al igual que las inversiones con bajas correlaciones (como acciones y bonos) se unen para formar una cartera que es mayor que la suma de sus partes, los modelos no correlacionados pueden producir predicciones de conjunto que son más precisas que cualquiera de las predicciones individuales. La razón de este maravilloso efecto es que los árboles se protegen entre sí de sus errores individuales (siempre que no se equivoquen constantemente en la misma dirección). Si bien algunos árboles pueden estar equivocados, muchos otros árboles estarán en lo correcto, por lo que, como grupo, los árboles pueden moverse en la dirección correcta.

Al igual que el Random Forest, el clasificador Ada-boost[xiii] es un algoritmo de clasificación en el cual se anida distintos modelos. El clasificador Ada-boost combina un algoritmo de clasificador débil para formar un clasificador fuerte. Este clasificador se basa en el hecho de que, si bien un solo algoritmo puede clasificar mal los objetos, si se combinan múltiples clasificadores con la selección del conjunto de entrenamiento en cada iteración y asignamos la cantidad correcta de peso en la votación final, se puede podemos tener una buena puntuación de precisión para el clasificador general. Considerando esta premisa, Ada-boost, vuelve a entrenar el algoritmo de forma iterativa eligiendo el conjunto de entrenamiento basado en la precisión del entrenamiento anterior.

El preso específico de cada clasificador entrenado en cualquier iteración depende de la precisión lograda. Como se ha mencionado, Cada clasificador débil se entrena utilizando un subconjunto aleatorio del conjunto de entrenamiento general. Después de entrenar a un clasificador en cualquier nivel, Ada-boost asigna peso a cada elemento de entrenamiento. Al elemento mal clasificado se le asigna un mayor peso para que aparezca en el subconjunto de entrenamiento del siguiente clasificador con mayor probabilidad.

Asimismo, después de entrenar a cada clasificador, el peso que se le asigna a cada uno de los clasificadores también se encuentra basado en la precisión. Al clasificador más preciso se le asigna un mayor peso para que tenga más impacto en el resultado final.

Una vez seleccionados una familia de modelos o un conjunto de familias para intentar ajustar a un determinado set de datos, es necesario definir un criterio o métrica que nos permita evaluar que tan bien, o mal, se ajusta cada uno de los modelos propuestos, poder compararlos entre sí, seleccionar los parámetros óptimos para cada familia de modelos, y de todas las familias seleccionar aquel modelo que mejor nos permita explicar el set de datos propuestos. Para ello existen diversas métricas, siendo las más utilizadas las de matriz de confusión, la precisión o accuracy y el área bajo la curva ROC.

La matriz de confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea bajo aprendizaje supervisado. 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. En caso de que se trate de una clasificación binaria, la matriz de confusión resultante será una matriz cuadrada de dos por dos.

La matriz se completa insertando en cada elemento la cantidad de observación de la clase verdadera de cada fila que son predichos bajo cada una de las clases. En el caso de las clasificaciones binarias, la matriz de confusión resultante contiene las predicciones acertadas en la diagonal principal (Verdaderos positivos y casos que no poseen el atributo y son estimados como tal). Por el otro lado, la matriz posee los casos donde el algoritmo “se equivocó”, tanto clasificando como negativamente cuando efectivamente poseía el atributo como el otro posible error de clasificar una observación positiva como negativa. Como se puede observar, las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

Si en los datos de entrada el número de muestras de clases diferentes cambia mucho la tasa de error del clasificador no es representativa de lo bien que realiza la tarea el clasificador. Si por ejemplo hay 990 muestras de la clase 1 y sólo 10 de la clase 2, el clasificador puede tener fácilmente un sesgo hacia la clase 1. Si el clasificador clasifica todas las muestras como clase 1 su precisión será del 99%. Esto no significa que sea un buen clasificador, pues tuvo un 100% de error en la clasificación de las muestras de la clase 2.

El *accuracy*, o la exactitud de un modelo, es una de las métricas que se pueden derivar de la matriz de confusión y nos da una idea respecto de que tan bien ajusta un modelo a un set de datos. En un caso de clasificación binaria, la exactitud se define como la proporción de casos falsos positivos y falsos negativas en relación al total de la muestra.

La curva ROC[xiv] es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. En otras palabras, el área bajo la curva ROC es la representación de la razón (o ratio) de verdaderos positivos (VP = Verdaderos Positivos) en relación a los falsos positivos (FP = Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo). El análisis del área bajo la curva ROC, (AUC-ROC), proporciona herramientas para seleccionar los modelos posiblemente óptimos y descartar modelos. Una de las ventajas esta métrica es que la misma es independiente de la distribución de las clases en la población. El límite inferior del área bajo la curva ROC es 0.50 (lo que sería equivalente a una elección al azar), y el límite superior es 1. Cuanto mayor sea el área bajo la curva ROC mejor es el modelo no sólo en clasificar sino en ordenar a cada uno de los registros en cuanto a la confianza de su predicción.

2. Implementación de un modelo para bancos

En la presente sección se presentarán los aspectos metodológicos del presente trabajo, comenzando con lo relativo a la base de datos utilizada y como se implementaron las técnicas de minería de datos propuestas.

2.1. Base de Datos utilizada

Como se ha planteado, el objetivo del presen trabajo es desarrollar un modelo de minería de datos capaz de clasificar a los clientes de un banco entre aquellos que adquirirán un determinado producto bancario y aquellos que no. A tales fines se utilizará una base de datos que contiene registros de cada uno de los clientes del banco. Para cada uno de los clientes se poseen datos tanto personales como de otros productos que tiene contratado cada cliente, de los cuales se hace un mayor análisis en la Sección 3.2. del presente.

El presente trabajo se realizará sobre una base de datos pública del Banco Santander compartida por la misma organización a través de la plataforma Kaggle[xv]. Kaggle Inc es una empresa perteneciente al grupo Google Inc, que se reconoce como una plataforma para el encuentro de la comunidad de cientistas de datos (data scientists) e ingenieros de machine learning. La misma comenzó en el año 2017 con el objetivo de proveer una plataforma para realizar competiciones de machine learning. Estas competencias son auspiciadas en la mayoría de los casos por las propias empresas dueñas de los datos quienes los ponen a disposición de la comunidad con diversos objetivos como puede ser encontrar el mejor algoritmo para el objetivo en cuestión, una fuente de reclutamiento de personal altamente capacitado para trabajar en la organización, crear cierta relación de la marca y los usuarios, entre muchos otros. Con el correr de los años, Kaggle incorporó nuevos servicios como por ejemplo los Kaggle Kernels, una sección donde los distintos usuarios pueden compartir códigos, análisis y hallazgos realizados en R o Python sobre los distintos data sets. Otro servicio ofrecido por Kaggle es una sección de educación online sobre diversos tópicos como Machine Learning, Data Visualization, Artificial Intelligence (AI), entre otros. Finalmente, otra de las secciones de Kaggle es un repositorio de base de datos públicas.

Como se comenta más adelante, el autor y responsable de la base de datos es el Propio Banco Santander, quien compartió la misma en el marco de una competencia llevada a cabo a través de la plataforma Kaggle. En la descripción de la competencia, el Organizador - Banco Santander, confirma que la base de datos se encuentra anonimizada, de forma tal de no contener información real que pueda ser asociada con ningún cliente real del banco. Asimismo, los términos y condiciones de la competencia permiten su uso siempre y cuando su uso se encuentre relacionado con leer y aprender de los datos, analizar, modificar y de forma más amplia preparar los modelos con propósito de la competencia.

De acuerdo a los términos y condiciones establecidos por Kaggle en la sección "Datasets", apuntan no solo a ser un repositorio, sino que cada base de datos es una comunidad en donde los usuarios pueden debatir, intercambiar análisis y publicar sus ideas y códigos con el resto de los usuarios. Kaggle se encuentra preparado para admitir que las bases se encuentren en diversos formatos (como lo son CSV, JSON, SQLite, entre otros), pero incentivan a los usuarios a compartir los sets de datos en un formato abierto y accesible, con el objetivo de facilitar el trabajo y análisis de estos por toda la comunidad.

Existen dos formas en las que se pueden compartir una base de datos a través de Kaggle: La primera alternativa, y por default en Kaggle, es subir un datasets a través de un usuario de forma particular. La segunda forma es que el responsable de la base de datos sea una organización, en cuyo caso será necesario que la propia organización cree un perfil en Kaggle bajo su nombre. Esta segunda alternativa es la más conveniente en caso la base de datos a compartir haya sido creado por otro, o sea de propiedad de un tercero.

En el caso de estudio en cuestión, la base de datos utilizada formó parte de una competencia iniciada propiamente por el Banco Santander, y que una vez finalizada la base de datos quedó disponible en la sección "Datasets".

2.2. Respecto a la privacidad de los datos

En la presente sección se reflexionará la responsabilidad del investigador en cuanto a los datos en la presente investigación bajo el marco conceptual planteado por Steinmann, Matei y Collmann[xvi].

Como se ha mencionado, el origen de la base de datos es la plataforma Kaggle. Kaggle Inc, siendo el autor y responsable de la base de datos el propio Banco Santander, quien compartió la misma en el marco de una competencia llevada a cabo a través de la mencionada plataforma Kaggle.

Dicha base de datos contiene cierta información sobre un conjunto de cliente del mencionado banco como, por ejemplo, País de residencia y ciudad, Sexo, Edad, Fecha de Alta, Ingreso anual, Forma de adquisición, y producto que dicho cliente tiene con el banco. Como se puede observar varios de esos datos pueden ser considerados pertenecientes al entorno privado de los propietarios de esos datos y sensibles en su utilización.

Desde la óptica del marco conceptual presentado por los mencionados autores, dicha investigación se encontraría enmarcada en un intermedio entre la reutilización de los datos y la asignación de un nuevo propósito planteado por los autores. La mencionada base de datos fue compartida por el propio Banco con el objetivo de alentar una competencia de minería de datos que permita la extracción de nuevo conocimiento por parte de la comunidad científica. En el presente caso de estudio se pretende utilizar dicha base de datos para extraer conocimiento respecto del poder de asociación y conexión de los distintos productos comercializados por el banco, que, si bien forma parte del alcance general, no es estrictamente el objetivo planteado en la competencia por el banco.

Como señalan los autores, en este caso se hace frente a dos posibles dilemas. El primero de ellos es compartido por el investigador secundarios y el propietario original de los datos (el banco), ya que este último ha compartido información sensible respecto de sus clientes. Las consecuencias del conocimiento que se pueda extraer sin dudas estarán repartidas entre estos dos actores, pero el investigador secundario no debiera excusarse a que la base de datos ha sido publicada por el banco.

Asimismo, recae sobre el investigador secundario la disyuntiva respecto de la legitimidad de la utilización de dicha base de datos con otros fines. Si bien el banco compartió la base de datos con el objetivo de que la comunidad científica extraiga conocimiento entre como se relacionan los clientes con los distintos productos que el banco ofrece, el objetivo del trabajo no es estrictamente el planteado en la competencia, pero cabe destacar que dentro de los términos y condiciones de la competencia: "se permite su uso siempre y cuando su uso se encuentre relacionado con leer y aprender de los datos, analizar, modificar y de forma más amplia preparar los modelos con propósito de la competencia".

Por último, también existe se deben platear los riesgos inherentes a la recombinación de los datos. Para poder compartir la base de datos de sus clientes, el banco tuvo que tomar las precauciones correspondientes para evitar que se pueda asociar e identificar a que cliente corresponde ciertos datos, ya que como se ha mencionado la base de datos publicada incluye datos sensibles. Para ello, la base de datos tuve que pasar un proceso de anonimizarían. El riesgo aquí sería que, de alguna forma, y con el apoyo de base de datos auxiliares, se logré reconstruir la base de datos original, pudiendo por ejemplo identificar a cierta persona, sabiendo su ingreso anual, si tiene un crédito hipotecario, un auto, entre otras. De nuevo resulta importante destacar los riesgos y responsabilidades de cada parte frente a esta posibilidad, y sobre todo concientizarse que los datos no nos deberían exculparnos.

2.3. Estadística Descriptiva

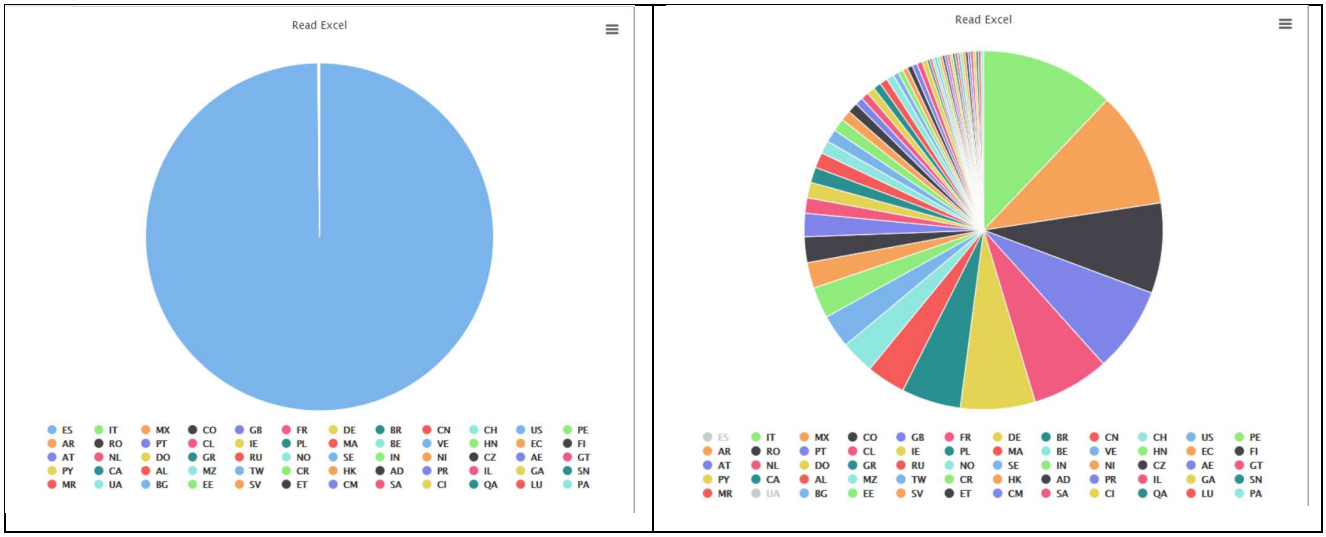
Como se ha mencionado, para el desarrollo del presente trabajo se utilizó una base de datos compartida por el propio Banco Santander. La misma cuenta con más de 500.000 registros, siendo cada uno de ellos los datos relativos a cada cliente del banco. A continuación, se hace un breve resumen de las características y campos con las que cuenta la base de datos que fue utilizada para correr cada uno de los algoritmos y modelos propios de la minería de datos, sean estos propios de la base de datos original, o creados a partir de otro campo de la base de datos:

- ✓ Código del Cliente: El mismo consiste en un código para identificar al cliente dentro de esta base de datos, pero no se corresponde con el verdadero número de cliente para el banco, ya que como se ha mencionada la base de datos ha sido anonimizada por el propio banco. Cabe

destacar que no se tiene ningún otro elemento que permita identificar al cliente (como puede ser nombre, apellido, Número de documento, etc).

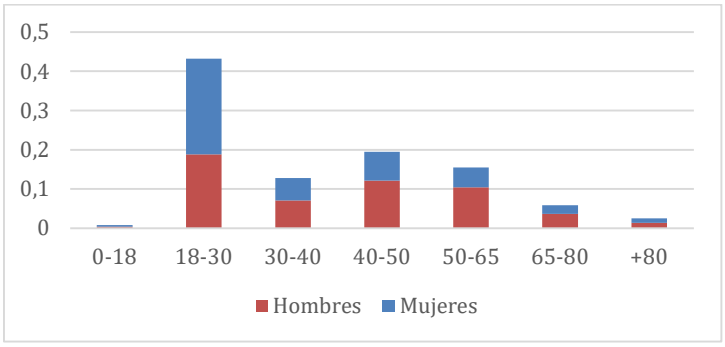
✓ Estado del cliente pudiendo el mismo ser Activo (A), exempleado (B), perteneciente a una filial (F), No empleado (N) o jubilado (P).

✓ País de residencia. Como se puede observar la gran mayoría (más del 95%) son residentes españoles, (gráfico de la izquierda) pero si excluimos España del análisis existe una gran variedad de más de 60 alternativas distintas (gráfico de la derecha), donde las que priman son Italia, México, Colombia y Gran Bretaña.

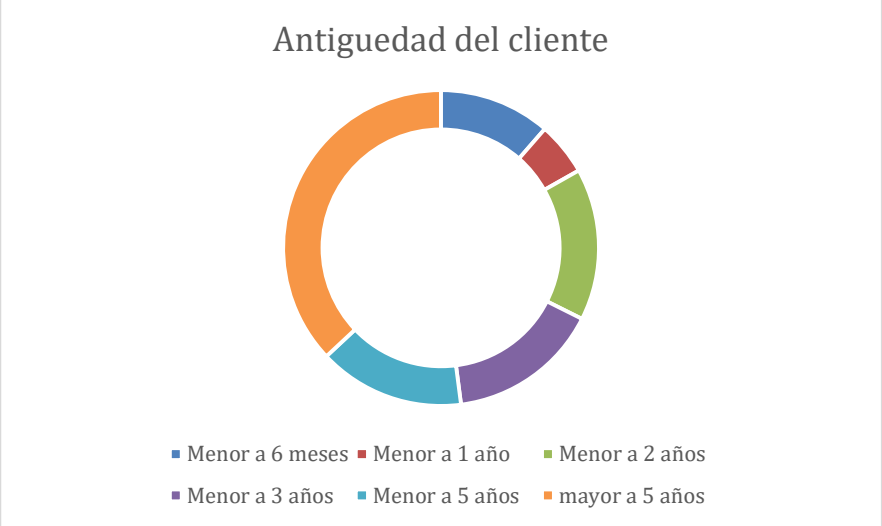


✓ Genero. Este atributo es binario y puede tomar los valores de 1 (hombre) o 0 (mujer).

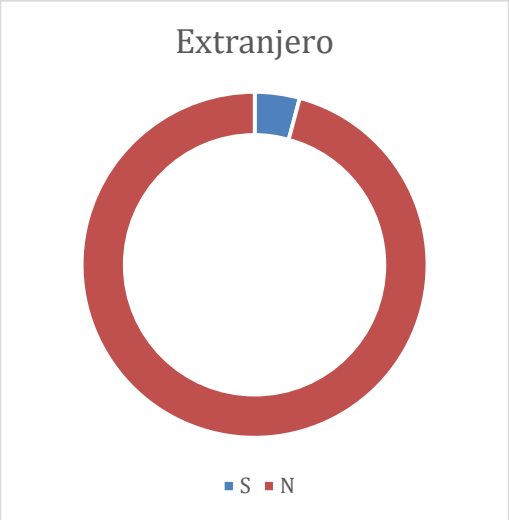
✓ Edad: Refiere a la edad del cliente. El valor mínimo era 2 años y el valor máximo es 106 años, y a partir de la misma se crearon los siguientes 7 intervalos:

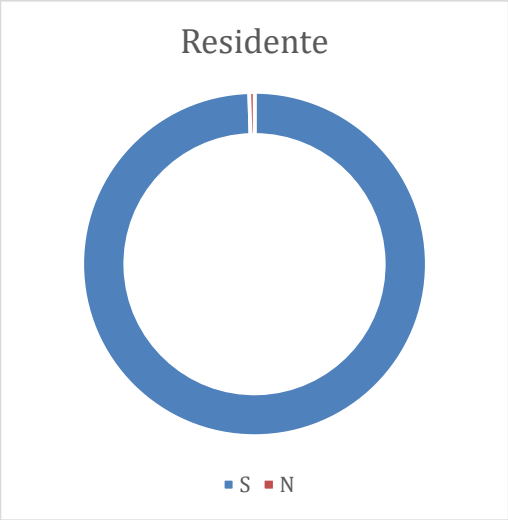


✓ Antigüedad del cliente. A partir de la fecha de alta de cada cliente, se ha calculado la antigüedad y agrupada en los siguientes intervalos



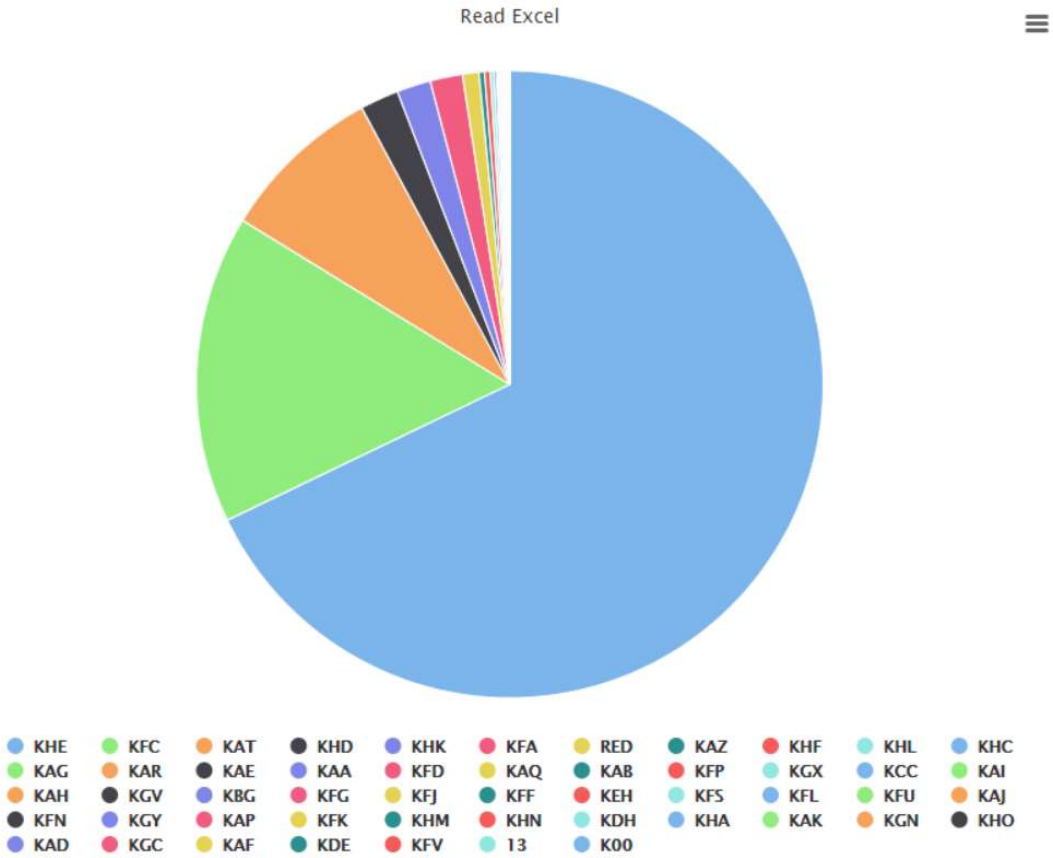
✓ Si el cliente es extranjero y si el cliente tiene residencia.





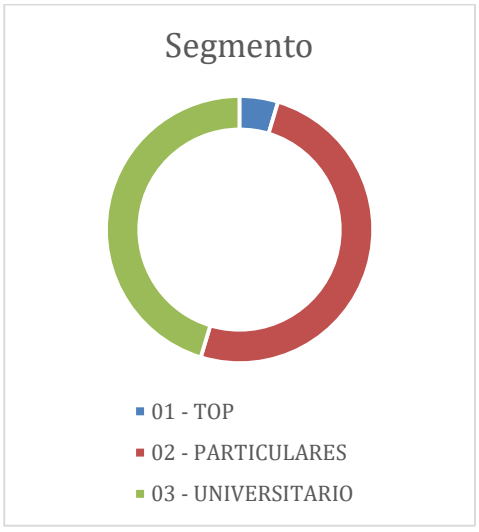
✓

✓ Canal de Venta. Indica el canal de venta por el cual el cliente ha sido incorporado. Existen 43 alternativas distintas, pero como se puede observar la gran mayoría corresponde a KHE. KFC y KAT completan el podio para llegar al 90% de la muestra

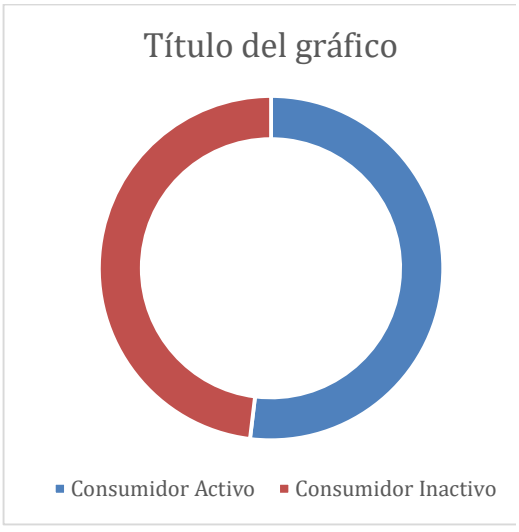


✓ Tipo de segmento al cual pertenece el cliente y tipo de consumidor.

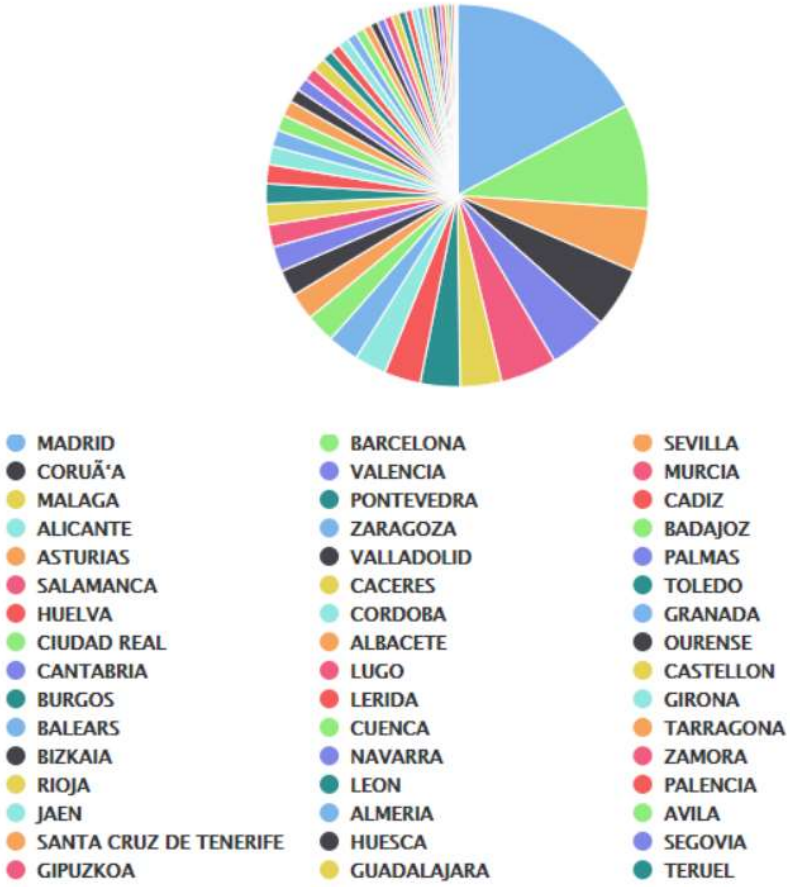
Tipo de segmento



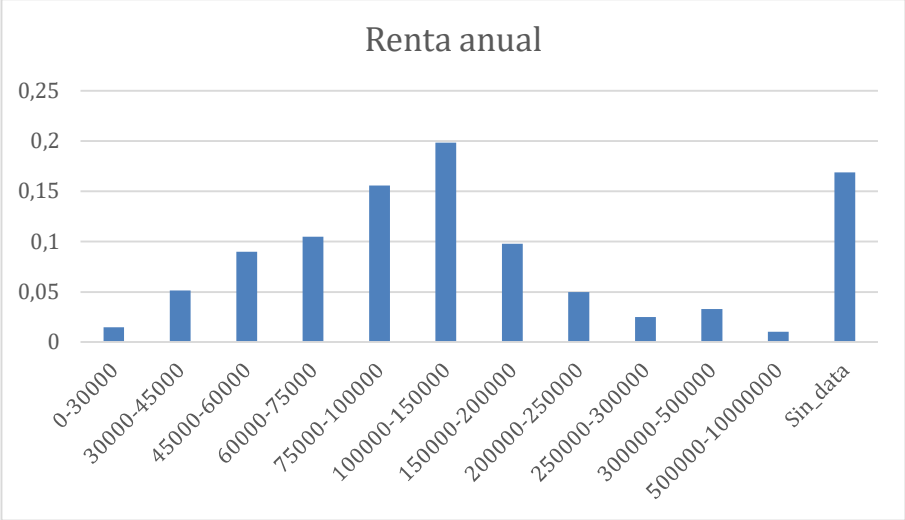
Tipo de consumidor



✓ Domicilio declarado por el cliente



✓ Renta anual del cliente, para la cual se crearon los siguientes intervalos.



✓ Productos ya adquiridos por cada cliente.

1. Caja de ahorros
2. Garantías
3. Cuenta Corriente
4. Cuenta Derivada
5. Cuenta Sueldo
6. Cuenta Juvenil
7. Cuenta Particular Mas
8. Cuenta Particular
9. Cuenta Particular Plus
10. Depósitos corto plazo
11. Depósitos mediano plazo
12. Depósitos Largo plazo
13. Hipotecas
14. Pensiones
15. Otros préstamos
16. Impuestos
17. Tarjetas de créditos
18. Securities
19. Cuentas personales
20. Cuenta Empresas
21. Débitos Automáticos
22. Otras Pensiones
23. E-Account – Billeteras virtuales. Es la variable que se intenta predecir clasificando a los clientes ente aquellos que si los compran.

2.4. Implementación de técnicas de Minería de Datos:

Como fue desarrollado en la sección 2.3. el componente de minería de datos suele sobre el cual se pone el mayor foco en el proceso de KDD. De acuerdo a lo descripto, está puede ser subdividas en 4 tareas, siendo la primera de ellas la definición de un conjunto de modelos que permitan clasificar a los clientes del Banco Santander entre aquellos que adquieren un determinado producto y aquellos que no, se propusieron diversos algoritmos. Como fue descripto en el apartado **2, se analizaron diversos algoritmos de clasificación**, cada modelo tiene diversos parámetros que deben ser especificados.

La segunda subetapa dentro del proceso de minería de datos consiste en un criterio de evaluación. El mismo será útil para poder comparar modelos dentro de la misma familia de modelos de forma tal de poder calibrar y seleccionar los parámetros óptimos, y asimismo el criterio de evaluación nos permitirá comparar los modelos de distinta clase. Para el presente caso de estudio se utilizará como criterio de evaluación el área bajo la curva ROC.

El tercer paso consiste justamente en calibrar cada uno de los modelos, justamente para seleccionar los parámetros óptimos. A tales fines se generó un algoritmo que permita "correr" cada uno de los modelos múltiples veces, cada una de ellas utilizando distintos parámetros, y para cada una de las corridas, se registró el área bajo la curva ROC para luego comparlas. Resulta importante destacar que con el objetivo de puntuar cada uno de los algoritmos, todos los métodos fueron evaluados utilizando el operador Cross-Validation. Este operador se utiliza a fin de evitar el sobre-ajuste (over fitting) del modelo, es decir que el modelo se sobre aprenda los datos de entrenamiento, pero luego no sea eficiente prediciendo el set de prueba. El operador consiste en subdividir toda la muestra en n submuestras, utilizando n-1 submuestras para el set de entrenamiento, y la submuestra restante como set de prueba, y esto lo hace de forma rotativa. Para todo este análisis se utilizó numbers of submuestras (o foldes) igual a 5. Finalmente, todos los algoritmos fueron corrido en un entorno de Python, haciéndose uso de las diferentes librerías ya desarrolladas para este fin.

El primer algoritmo en ser testeado fue el de árboles de decisión (en el Apéndice 1A se puede observar el bloque de código utilizado). Para este algoritmo se tuve que establecer en primer lugar el criterio, que consiste en la función a ser utilizada para medir la calidad de la apertura de cada una de las ramas, siendo las opciones por elegir el g*ini* y la *entropía*.

El segundo parámetro a definir es la profundidad del árbol que consiste en la cantidad de nodos con los que se quiere contar. En este caso se permitió que la misma varia entre 20 y 80, dando saltos de a 5. El tercer parámetro que se testeó fue el tamaño mínimo de muestra para que el algoritmo realice una nueva división creando un nuevo nodo. En este caso se permitió que dicho parámetro varíe entre 100 y 500, dando saltos de a 50. Como se puede observar en el apéndice 1B donde se detallan los resultados de las 208 pruebas realizadas, el mejor resultado se obtiene cuando se establece una profundidad de 20 y se divide cuando el nodo cuenta con al menos 450 observaciones, donde se obtuvo un área bajo la curva ROC de 0.856 con un error de +/- 0.053.

El segundo modelo individual en ser testeado fue el modelo de regresión logística. En este caso, el primer parámetro a definir es el algoritmo por utilizar en la optimización del problema. Asimismo, se deben definir también la función de penalización, y si se desear considerar o no un intercepto en el modelo de regresión. En el Apéndice 2A se puede observar el código utilizado, y como se puede observar en el Apéndice 2B, se obtuvieron resultados similares para todas las pruebas realizadas, siendo el mejor con AUC-ROC de 0.860 con un error de +/- 0.053.

Como se vio en el apartado de implementación de técnicas de minería de datos, una de las técnicas muy utilizadas en la minería de datos consiste en crear diferentes modelos usando muestras aleatorias con reemplazo y luego combinar o ensamblar los resultados, con el objetivo de incrementar la exactitud a partir de la combinación de las predicciones de múltiples modelos. En este sentido se utilizó el modelo de Random Forest, el cual supone la combinación de múltiples árboles de decisión individuales. De esta forma, los parámetros a definir para este modelo además de los parámetros de cada árbol, que ya han sido explicitados anteriormente, se adiciona la cantidad de árboles a combinar (n_estimator). En este caso para la cantidad de árboles para cada uno de los modelos se testearon desde 50 hasta 130, en saltos de a 10. En el Apéndice 3A se puede observar el bloque de código utilizado. De las mas de 350 pruebas realizadas, cuyo detalle puede observar en el Apéndice 3B, la prueba con el mayor AUC-ROC se obtuvo cundo se utilizó el criterio de Gini, con 60 árboles y una máxima profundidad de 10, siendo el mismo de 0.88.

Otro de los algoritmos en ser calibrados fue el Ada-boost classifier (en el Apéndice 4A se puede observar el bloque de código utilizado). En este caso los parámetros a definir eran dos. El primero de ellos el criterio de selección, pudiendo este ser SAMME o SAMME.R. El otro

parámetro a definir es la cantidad de estimadores, para lo cual se testearon desde 20 hasta 100 en intervalos de a 100.

Como se puede observar en el apéndice 4B donde se detallan los resultados de las 16 pruebas realizadas, el mejor resultado se obtiene cuando se utiliza el algoritmo SAMME.R y una cantidad de 50 estimadores, lográndose un área bajo la curva ROC, levemente superior del 0.867, pero además lográndose disminuir el error a +/- 0.049.

Uno de los algoritmos de minería de datos con crecimiento más vertiginoso en los últimos tiempos ha sido las redes neuronales, principalmente por su gran flexibilidad y versatilidad para adaptarse ante distintos conjuntos de datos. Esta versatilidad se debe principalmente a la gran cantidad de parámetros y estructuras con las que se puede definir y planear una red neuronal. En este caso, el primer parámetro que se tuve que definir es el algoritmo que el modelo va a utilizar para entrenarse, pudiéndose seleccionar entre Stochastic Gradient Descent, Adam o L-BFGS. SGD (Descenso de gradiente estocástico) actualiza los parámetros utilizando el gradiente de la función de pérdida con respecto a un parámetro que necesita adaptación, es decir donde es la tasa de aprendizaje que controla el tamaño del paso en la búsqueda de espacio de parámetros. es la función de pérdida utilizada para la red. Adam es similar al SGD en el sentido de que es un optimizador estocástico, pero puede ajustar automáticamente la cantidad para actualizar los parámetros en función de las estimaciones adaptativas de los momentos de orden inferior. L-BFGS es un solucionador que se aproxima a la matriz de Hessiana que representa la derivada parcial de segundo orden de una función. Además, se aproxima el inverso de la matriz de Hesse para realizar actualizaciones de parámetros. La implementación utiliza la versión Scipy de L-BFGS.

El siguiente parámetro a definir es la función de activación, siendo esta la función que permite que la información fluya de una serie de nodos a la siguiente. En este caso se testearon las funciones de identidad, la logística, la “tanh” y “relu”. Finalmente es necesario definir la estructura de perceptrones de cada una de las capas de la red. En este caso, se propusieron diversos esquemas con distintas cantidades de capas. En el Apéndice 5A se puede observar un detalle de todas las pruebas realizadas y los resultados obtenidos.

3. Selección de modelos y recomendaciones

Como se ha detallado en el apartado anterior, se han analizado, testeado y optimizados diversos métodos analíticos predictivos, obteniéndose para cada uno de ellos una determinada área bajo la curva ROC. A continuación, se presenta un cuadro donde se resumen todos los resultados obtenidos.

Algoritmo	#	Criterio/ Algoritmo	n_estimator (cantidad de árboles)	Maxima Profundiad	min_samples_split	AUC	Desvío_AUC
Decision Tree	8	gini		20	450	0.85634913	0.05376201
Regresión Logística	1	newton-cg	l2	FALSE		0.86065181	0.05352686
Random Forest	123	gini	60	10	200	0.88025245	0.039946
AdaBoost	4	SAMME	50			0.86704784	0.04913116
Redes Neuronales	16	lbfgs	tanh	(11,9,7,5,3)		0.88782224	0.04312936

Si bien todos los modelos han performado bastante similar con área bajo la curva ROC (AUC-ROC) relativamente altas para todas las simulaciones, se puede observar que, el modelo de Redes Neuronales con la estructura y parámetros definidos es el que mejor AUC-ROC arroja. No obstante, el AUC-ROC obtenidos con el modelo de Random Forest, si bien es levemente inferior, tiene la ventaja de haber arrojado un menor error, es decir menor variación entre cada uno de los resultados cuando se realizó el Cross Valiadation, lo que da la idea de que se trata de un modelo más estable, y que probablemente pueda tener una mejor performance ante un set de datos distintos al que fue utilizado para entrenar.

Otra de las razones que podría inclinar la balanza en favor del Modelo de Random Forest, es que el mismo se trata de un modelo más simple y fácil de explicar a un tercero, ya que el mismo se basa em reglas claras (cada una de las ramas de decisión del árbol). En contraposición, una red neuronal suele ser un modelo no solo más complejo, sino que también funciona como una caja negra. De esta forma, es posible extraer del modelo de Random Forest cuales son las variables relevantes en la clasificación (las ramas principales), lo cual es de suma importancia para la organización, ya que podrían funcionar como factores de segmentación con el objetivo de direccionar sus campañas y esfuerzos de marketing.

Con el bloque de código que se expone en el Apéndice 6, se puede extraer las variables mas relevantes del modelo seleccionado. De esta forma se puede concluir que aquellos clientes que tienden a comprar el producto objeto de estudio del presente trabajo tienden a tener las siguientes características:

- ✓ El cliente ya posee una Cuenta corrientes con la organización
- ✓ El cliente posee una antigüedad mayor a los 3 años.
- ✓ El cliente ya posee una cuenta junior (o junior plus) con la organización
- ✓ Tienen como canal de venta KHE o KHD
- ✓ Se trata de un cliente particular (no empresa)
- ✓ Clientes menores de 40 años (franjas etarias 18-30 y 30-40).
- ✓ El cliente tiene una renta menor a 75.000.

Consideraciones finales y futuros trabajos

Durante el presente trabajo se estudió como en los últimos años, todas las organizaciones, pero en particular las de la industria financiera, han comenzado a recopilar datos de sus clientes de forma sistemática. Asimismo, se analizó la importancia que pueden tener esos datos a la hora de querer clasificar la cartera de clientes de una organización, y los beneficios de una buena segmentación de mercados y potenciales usuarios con el fin de mejorar la estrategia de marketing directo de una organización.

En los últimos años se ha evidenciado un crecimiento constante y vertiginoso de las herramientas informáticas que han permitido en avanzar no solo en el almacenamiento de grandes volúmenes de información, sino que además en su procesamiento a fin de poder analizar los mismos y generar modelos que ayuden a la toma de decisiones. Como se expuso en la sección 2.1. resulta de vital importancia dirigir y segmentar el público al cual se destina toda estrategia de marketing, no solo a fines de reducir los costos que la mismas conllevan, sino que también reducir el destino de recursos (por ejemplo, de telemarketers realizando las llamadas) sino que también del costo reputacional que tiene “la molestia” en contactar a alguien que no desea adquirir el servicio.

Asimismo, durante el desarrollo del presente trabajo se describió el proceso de extracción de conocimiento a partir de los datos, realizando una descripción de cada una de las etapas. Se examinó la importancia de la recolección de los datos, y aún mas importante el preprocesamiento, la limpieza y la preparación de estos para luego poder ser procesados por los distintos métodos analíticos predictivos. En particular, se hizo foco en la etapa del proceso de KDD que más desarrollo ha tenido en los últimos años referente a las técnicas de minería de datos propiamente dichos, analizándose los algoritmos de clasificación más utilizados y realizándose una breve reseña de sus particularidades mas importantes y ventajas de cada uno.

El objetivo particular del presente trabajo fue aplicar los mencionados modelos al caso de estudio en particular, con el objetivo de lograr el mejor modelo que pudiera clasificar a los clientes del banco y predecir quienes adquirían un determinado producto. Antes de poder calibrar cada uno de los modelos, fue necesario detallar el origen de la base de datos, realizar una descripción del set de datos original y del proceso de preparación, con un detalle de las variables que fueron creadas a partir de los datos existentes.

Debido al origen de la base de datos, no menos importante en la sección 3.1.3 se realizó una reflexión de la investigación bajo el marco conceptual planteado por Steinmann, Matei y Collmann y de la corresponsabilidad del investigador debido a utilizar un set de datos público de internet.

En el apartado **2. Implementación de técnicas de Minería de Datos** se detalló como se implementaron los distintos modelos en un entorno de Pyhton, haciéndose un detalle de los diversos algoritmos utilizados, que parámetros se testearon para cada uno de los modelos y los resultados obtenidos. En el apartado Nº 3 no solo se hizo un resumen de todos los resultados, sino que se concluyó que el mejor modelo para clasificar a los clientes de esta organización en relación al nuevo producto es el modelo de Random Forest. Mas importante aún en dicho apartado, se puedo extraer cuales son las características más relevantes a la hora de clasificar a los clientes, para que la institución financiera las tenga en cuenta a la hora de segmentar su cartera de clientes y/o potenciales usuario con el fin direccionar hacia ellos los esfuerzos y recursos de las campañas de marketing directo a realizar.

Por concluir el presente trabajo de investigación, quedan pendientes algunas consideraciones para ser abordadas en próximos trabajas en un futuro cercano, que bien podrían ser desarrolladas por este investigador, o bien por cualquier otro entusiasta de la extracción del conocimiento a partir de los datos que podrían tomar este trabajo como punto de partida.

En primer lugar, se podrían evaluar implementar alguno de los modelos desarrollados en el presente trabajo en un entorno de producción. Esto implicaría tomar un modelo ya optimizado en sus parámetros y moverlo hacia una plataforma que permita la interacción directa con el usuario. De esta forma, le permitiría a la organización segmentar una determinada base de datos de clientes, ordenando los registros de acuerdo a la probabilidad que el modelo le estima a cada usuario de éxito en la adquisición del producto de interés. Las ventajas en la implementación de tal modelo para la organización son diversas, destacándose principalmente el ordenamiento de los usuarios, el descarte de cierto grupo de clientes que se espera no esté interesados en adquirir el producto, con la posibilidad de redireccionar dichos recursos y esfuerzos hacia los clientes que si se espera se encuentren interesados en el producto.

Por último, otra de los trabajos que podrían realizarse como complemento de todo lo desarrollado en el presente es un análisis de asociación entre los diversos productos. Este análisis permitiría estudiar y extraer conocimiento respecto de como los diversos usuarios del banco se comportan en relación a los mas de 25 productos que la institución financiera ofrece, con el objetivo de determinar, en caso de existir, que productos suelen adquirirse de forma simultánea, y cuales podrían llegar a ser sustitutos. Asimismo, todo este análisis permitiría potenciar los modelos desarrollados de forma de lograr un sistema de recomendación completo. Este nuevo modelo permitiría, no solo clasificar si un determinado cliente estaría interesado o no en el nuevo producto, sino que además permitiría predecir, dado el conjunto de productos que cada cliente ya tenga, cual o cuales serían los productos adicionales que el cliente podría estar interesado en adquirir.

Anexos / Apéndices

Apéndice 1A – Código de Decision Tree

```
## 0. Importar los paquetes necesarios junto con el set de datos

import pandas as pd
import sklearn as sk

from google.colab import files
uploaded = files.upload()


## 1. LEER DATOS...

data = pd.read_excel("final_v2.xlsx")

# 2. PREPARAR DATOS
#data = pd.get_dummies(data, columns=['area', 'salario'])

X = data[data.columns.drop(['ncodpers', 'ind_ecue_fin_ult1'])].values
y = data['ind_ecue_fin_ult1'].values

## DECISION TREE
# 2. VALIDACION CRUZADA EN 5 PARTES

from sklearn import model_selection
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.model_selection import cross_val_score

for criterion in ("gini", "entropy"):
  for max_depth in range(20, 85, 5):
    for min_samples_split in range(100, 500, 50):
        clf = tree.DecisionTreeClassifier(criterion=criterion, max_dept
h=max_depth, min_samples_split=min_samples_split)
        scores = model_selection.cross_val_score(clf, X, y, scoring="ro
c_auc", cv=5)
        print(criterion, max_depth, min_samples_split, "exactitud:", sc
ores.mean(), scores.std(),'|')
```

Apéndice 1B – Resultados obtenidos de Decision Tree

Algoritmo	#	Criterio	Maxima Profundiad	min_samples_split	AUC	Desvío_AUC		Algoritmo	#	Criterio	Maxima Profundiad	min_samples_split	AUC	Desvío_AUC
Decision Tree	1	gini	20	100	0.84541997	0.05539557		Decision Tree	105	entropy	20	100	0.84492109	0.0607289
Decision Tree	2	gini	20	150	0.84827934	0.05627461		Decision Tree	106	entropy	20	150	0.84931736	0.05938861
Decision Tree	3	gini	20	200	0.85165315	0.05408099		Decision Tree	107	entropy	20	200	0.85111831	0.06037991
Decision Tree	4	gini	20	250	0.85209538	0.05498287		Decision Tree	108	entropy	20	250	0.85205111	0.06019875
Decision Tree	5	gini	20	300	0.85319558	0.05554698		Decision Tree	109	entropy	20	300	0.85418945	0.05977929
Decision Tree	6	gini	20	350	0.85510611	0.05450246		Decision Tree	110	entropy	20	350	0.85503972	0.06027763
Decision Tree	7	gini	20	400	0.85590686	0.05377214		Decision Tree	111	entropy	20	400	0.8555802	0.06059681
Decision Tree	8	gini	20	450	0.85634913	0.05376201		Decision Tree	112	entropy	20	450	0.85595578	0.06115912
Decision Tree	9	gini	25	100	0.83816077	0.06052033		Decision Tree	113	entropy	25	100	0.83734121	0.06364194
Decision Tree	10	gini	25	150	0.84359916	0.06052422		Decision Tree	114	entropy	25	150	0.8435574	0.06388655
Decision Tree	11	gini	25	200	0.84765377	0.05834898		Decision Tree	115	entropy	25	200	0.8459329	0.06460686
Decision Tree	12	gini	25	250	0.84905353	0.05947605		Decision Tree	116	entropy	25	250	0.84784486	0.06467099
Decision Tree	13	gini	25	300	0.85120456	0.05932092		Decision Tree	117	entropy	25	300	0.85027236	0.06354401
Decision Tree	14	gini	25	350	0.85286598	0.05878263		Decision Tree	118	entropy	25	350	0.8515961	0.06419189
Decision Tree	15	gini	25	400	0.85364911	0.05819853		Decision Tree	119	entropy	25	400	0.85259606	0.06448391
Decision Tree	16	gini	25	450	0.8545081	0.05833948		Decision Tree	120	entropy	25	450	0.85284219	0.06532672
Decision Tree	17	gini	30	100	0.83395148	0.05948614		Decision Tree	121	entropy	30	100	0.83349903	0.06191058
Decision Tree	18	gini	30	150	0.84112973	0.06009055		Decision Tree	122	entropy	30	150	0.84084321	0.0624113
Decision Tree	19	gini	30	200	0.84570753	0.05808925		Decision Tree	123	entropy	30	200	0.84368236	0.0635035
Decision Tree	20	gini	30	250	0.84741931	0.0590517		Decision Tree	124	entropy	30	250	0.8464424	0.06347225
Decision Tree	21	gini	30	300	0.84975645	0.05914737		Decision Tree	125	entropy	30	300	0.84892564	0.06264123
Decision Tree	22	gini	30	350	0.85174858	0.05852214		Decision Tree	126	entropy	30	350	0.8504484	0.06317042
Decision Tree	23	gini	30	400	0.85277775	0.05795745		Decision Tree	127	entropy	30	400	0.85144394	0.0638651
Decision Tree	24	gini	30	450	0.85373856	0.05806859		Decision Tree	128	entropy	30	450	0.85171384	0.0647986
Decision Tree	25	gini	35	100	0.83171566	0.05948957		Decision Tree	129	entropy	35	100	0.83126926	0.06174583
Decision Tree	26	gini	35	150	0.83973395	0.05997108		Decision Tree	130	entropy	35	150	0.83917362	0.06241283
Decision Tree	27	gini	35	200	0.84476152	0.05796702		Decision Tree	131	entropy	35	200	0.84296757	0.06307811
Decision Tree	28	gini	35	250	0.84674322	0.05907097		Decision Tree	132	entropy	35	250	0.84535558	0.06291449
Decision Tree	29	gini	35	300	0.84892988	0.05952631		Decision Tree	133	entropy	35	300	0.84847314	0.06208145
Decision Tree	30	gini	35	350	0.85133969	0.05867585		Decision Tree	134	entropy	35	350	0.84983339	0.06280612
Decision Tree	31	gini	35	400	0.85264708	0.05785859		Decision Tree	135	entropy	35	400	0.85113601	0.06331388
Decision Tree	32	gini	35	450	0.85345183	0.05797771		Decision Tree	136	entropy	35	450	0.85137998	0.06452761
Decision Tree	33	gini	40	100	0.83092743	0.05917786		Decision Tree	137	entropy	40	100	0.8301915	0.06171159
Decision Tree	34	gini	40	150	0.83904971	0.0595955		Decision Tree	138	entropy	40	150	0.83836963	0.06225636
Decision Tree	35	gini	40	200	0.84439935	0.05779105		Decision Tree	139	entropy	40	200	0.84229409	0.06349689
Decision Tree	36	gini	40	250	0.84658675	0.05885436		Decision Tree	140	entropy	40	250	0.84527485	0.06314514
Decision Tree	37	gini	40	300	0.84909203	0.05890632		Decision Tree	141	entropy	40	300	0.8483453	0.06202388
Decision Tree	38	gini	40	350	0.85116185	0.05847665		Decision Tree	142	entropy	40	350	0.84982522	0.06269998
Decision Tree	39	gini	40	400	0.85249103	0.05784328		Decision Tree	143	entropy	40	400	0.85099992	0.0634754
Decision Tree	40	gini	40	450	0.853399	0.05786136		Decision Tree	144	entropy	40	450	0.85120951	0.06454456
Decision Tree	41	gini	45	100	0.83066242	0.05907057		Decision Tree	145	entropy	45	100	0.82960015	0.06168362
Decision Tree	42	gini	45	150	0.83907208	0.05953261		Decision Tree	146	entropy	45	150	0.83850453	0.06208412
Decision Tree	43	gini	45	200	0.8444495	0.05777526		Decision Tree	147	entropy	45	200	0.84196289	0.06338926
Decision Tree	44	gini	45	250	0.84649289	0.05892083		Decision Tree	148	entropy	45	250	0.84539456	0.0631916
Decision Tree	45	gini	45	300	0.84904672	0.05896623		Decision Tree	149	entropy	45	300	0.84855931	0.06210911
Decision Tree	46	gini	45	350	0.85114054	0.05846485		Decision Tree	150	entropy	45	350	0.84974137	0.06289892
Decision Tree	47	gini	45	400	0.85247743	0.05787267		Decision Tree	151	entropy	45	400	0.85100747	0.0633468
Decision Tree	48	gini	45	450	0.85344354	0.05796655		Decision Tree	152	entropy	45	450	0.85135037	0.0643848
Decision Tree	49	gini	50	100	0.83085276	0.05901489		Decision Tree	153	entropy	50	100	0.82979837	0.06166255
Decision Tree	50	gini	50	150	0.8390389	0.05960778		Decision Tree	154	entropy	50	150	0.83818361	0.06254035
Decision Tree	51	gini	50	200	0.8444007	0.05769651		Decision Tree	155	entropy	50	200	0.84249615	0.06303274
Decision Tree	52	gini	50	250	0.84652937	0.05890227		Decision Tree	156	entropy	50	250	0.84489277	0.06318304
Decision Tree	53	gini	50	300	0.84906846	0.05886234		Decision Tree	157	entropy	50	300	0.8483232	0.06205003
Decision Tree	54	gini	50	350	0.85115354	0.05845613		Decision Tree	158	entropy	50	350	0.84979176	0.06281108
Decision Tree	55	gini	50	400	0.85256438	0.05780668		Decision Tree	159	entropy	50	400	0.85089175	0.06359146
Decision Tree	56	gini	50	450	0.85344879	0.0578688		Decision Tree	160	entropy	50	450	0.85127453	0.06457207
Decision Tree	57	gini	55	100	0.83073036	0.05891246		Decision Tree	161	entropy	55	100	0.82990114	0.06151197
Decision Tree	58	gini	55	150	0.83904983	0.05955572		Decision Tree	162	entropy	55	150	0.83824467	0.06212303
Decision Tree	59	gini	55	200	0.84439218	0.05782532		Decision Tree	163	entropy	55	200	0.84244929	0.06292459
Decision Tree	60	gini	55	250	0.84656487	0.05882674		Decision Tree	164	entropy	55	250	0.84503414	0.06310641
Decision Tree	61	gini	55	300	0.84904639	0.0588553		Decision Tree	165	entropy	55	300	0.84868935	0.06194403
Decision Tree	62	gini	55	350	0.85115725	0.05854315		Decision Tree	166	entropy	55	350	0.84993204	0.06252659

Apéndice 2A – Código de Regresión Logística.

```
## 0. Importar los paquetes necesarios junto con el set de datos

import pandas as pd
import sklearn as sk

from google.colab import files
uploaded = files.upload()


## 1. LEER DATOS...

data = pd.read_excel("final_v2.xlsx")

# 2. PREPARAR DATOS
#data = pd.get_dummies(data, columns=['area', 'salario'])

X = data[data.columns.drop(['ncodpers', 'ind_ecue_fin_ult1'])].values
y = data['ind_ecue_fin_ult1'].values

from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score

for solversrt in ("newton-cg", "lbfgs", "sag", "saga"):
  for penalty in ("l1", "l2", "elasticnet"):
    for fit_intercept in ("TRUE", "FALSE"):
      clf =  LogisticRegression(penalty=penalty, fit_intercept=fit_inte
rcept)
      scores = model_selection.cross_val_score(clf, X, y, scoring="roc_
auc", cv=5)
      print(solversrt, penalty, fit_intercept, "exactitud:", scores.mea
n(), scores.std(),'|')
```

Apéndice 2B – Resultados obtenidos de Regresión Logística

Algoritmo	#	solversrt	penalty	intercecpt	AUC	Desvío_AUC
Regresión Logística	1	newton-cg	l2	FALSE	0.86065181	0.05352686
Regresión Logística	2	newton-cg	l2	TRUE	0.86065181	0.05352686
Regresión Logística	3	newton-cg	l1	FALSE	0.86011496	0.05399378
Regresión Logística	4	newton-cg	l1	TRUE	0.86012104	0.05399649

Apéndice 3A – Código de Random Forest

```
## 0. Importar los paquetes necesarios junto con el set de datos

import pandas as pd
import sklearn as sk

from google.colab import files
uploaded = files.upload()


## 1. LEER DATOS...

data = pd.read_excel("final_v2.xlsx")

# 2. PREPARAR DATOS
#data = pd.get_dummies(data, columns=['area', 'salario'])

X = data[data.columns.drop(['ncodpers', 'ind_ecue_fin_ult1'])].values
y = data['ind_ecue_fin_ult1'].values

## DECISION TREE
# 2. VALIDACION CRUZADA EN 5 PARTES

 from sklearn import model_selection
 from sklearn import ensemble
 from sklearn.model_selection import cross_val_score
 from sklearn.ensemble import RandomForestClassifier

for criterion in ("gini", "entropy"):
  for n_estimators in range(50, 130, 10):
    for max_depth in range(20, 85, 5):
      for min_samples_split in range(100, 500, 50):
        clf = RandomForestClassifier(criterion=criterion, n_estimators=
n_estimators, max_depth=max_depth, min_samples_split=min_samples_split)
        scores = model_selection.cross_val_score(clf, X, y, scoring="ro
c_auc", cv=5)
        print(criterion, n_estimators, max_depth, min_samples_split, "e
xactitud:", scores.mean(), scores.std(),'|')
```

Apéndice 3B – Resultados obtenidos de Random Forest – “Gini”

Algoritmo	#	Criterio	n_estimator (cantidad de árboles)	Maxima Profundiad	min_samples _split	AUC	Desvío_AUC	Algoritmo	#	Criterio	n_estimator (cantidad de árboles)	Maxima Profundiad	min_samples _split	AUC	Desvío_AUC
Random Forest	1	gini	50	10	100	0.87746374	0.04223849	Random Fore	106	gini	50	75	150	0.87720371	0.04026936
Random Forest	2	gini	50	10	150	0.87626051	0.040545	Random Fore	107	gini	50	75	200	0.87720612	0.04005781
Random Forest	3	gini	50	10	200	0.87857373	0.03989836	Random Fore	108	gini	50	75	250	0.87542198	0.03943209
Random Forest	4	gini	50	10	250	0.87888199	0.03937531	Random Fore	109	gini	50	75	300	0.87590203	0.03886511
Random Forest	5	gini	50	10	300	0.87979619	0.03987335	Random Fore	110	gini	50	75	350	0.87382824	0.0407977
Random Forest	6	gini	50	10	350	0.87645728	0.04082334	Random Fore	111	gini	50	75	400	0.87640587	0.04088082
Random Forest	7	gini	50	10	400	0.87567244	0.04090372	Random Fore	112	gini	50	75	450	0.87620118	0.03992687
Random Forest	8	gini	50	10	450	0.87915227	0.04189856	Random Fore	113	gini	50	80	100	0.87605126	0.03805209
Random Forest	9	gini	50	15	100	0.87600033	0.03933638	Random Fore	114	gini	50	80	150	0.87571856	0.03817753
Random Forest	10	gini	50	15	150	0.8764356	0.03962404	Random Fore	115	gini	50	80	200	0.87624504	0.03907842
Random Forest	11	gini	50	15	200	0.8765216	0.03838397	Random Fore	116	gini	50	80	250	0.87710526	0.03943678
Random Forest	12	gini	50	15	250	0.87570598	0.04040553	Random Fore	117	gini	50	80	300	0.87584828	0.0400915
Random Forest	13	gini	50	15	300	0.87546234	0.03838089	Random Fore	118	gini	50	80	350	0.87597072	0.03986271
Random Forest	14	gini	50	15	350	0.8771301	0.04053634	Random Fore	119	gini	50	80	400	0.87462809	0.03817612
Random Forest	15	gini	50	15	400	0.87649208	0.04035498	Random Fore	120	gini	50	80	450	0.87222468	0.03640273
Random Forest	16	gini	50	15	450	0.8759079	0.03872155	Random Fore	121	gini	60	10	100	0.8772614	0.04151561
Random Forest	17	gini	50	20	100	0.8765985	0.03803198	Random Fore	122	gini	60	10	150	0.87964009	0.03854613
Random Forest	18	gini	50	20	150	0.87703493	0.03883109	Random Fore	123	gini	60	10	200	0.88025245	0.039946
Random Forest	19	gini	50	20	200	0.8764824	0.03963597	Random Fore	124	gini	60	10	250	0.8774815	0.03987309
Random Forest	20	gini	50	20	250	0.87458135	0.03866563	Random Fore	125	gini	60	10	300	0.87598993	0.04065702
Random Forest	21	gini	50	20	300	0.87474883	0.03799293	Random Fore	126	gini	60	10	350	0.87426141	0.04151124
Random Forest	22	gini	50	20	350	0.87611334	0.03939321	Random Fore	127	gini	60	10	400	0.87824219	0.04139201
Random Forest	23	gini	50	20	400	0.87761384	0.03969906	Random Fore	128	gini	60	10	450	0.87606732	0.04184269
Random Forest	24	gini	50	20	450	0.87445986	0.03842239	Random Fore	129	gini	60	15	100	0.87682317	0.03943689
Random Forest	25	gini	50	25	100	0.87515372	0.03778436	Random Fore	130	gini	60	15	150	0.87552471	0.04005809
Random Forest	26	gini	50	25	150	0.87700304	0.03915774	Random Fore	131	gini	60	15	200	0.87440181	0.03884487
Random Forest	27	gini	50	25	200	0.87588374	0.03662294	Random Fore	132	gini	60	15	250	0.87670538	0.03885841
Random Forest	28	gini	50	25	250	0.87574438	0.0382527	Random Fore	133	gini	60	15	300	0.87583333	0.03900329
Random Forest	29	gini	50	25	300	0.87710006	0.03961252	Random Fore	134	gini	60	15	350	0.87531701	0.03829539
Random Forest	30	gini	50	25	350	0.87584873	0.04032761	Random Fore	135	gini	60	15	400	0.87751106	0.04009126
Random Forest	31	gini	50	25	400	0.8764876	0.03998693	Random Fore	136	gini	60	15	450	0.87712528	0.04065166
Random Forest	32	gini	50	25	450	0.87568876	0.04023531	Random Fore	137	gini	60	20	100	0.87720849	0.03919942
Random Forest	33	gini	50	30	100	0.87618643	0.03931277	Random Fore	138	gini	60	20	150	0.87621533	0.03905865
Random Forest	34	gini	50	30	150	0.87662123	0.03854999	Random Fore	139	gini	60	20	200	0.87586834	0.03741146
Random Forest	35	gini	50	30	200	0.87608578	0.04012899	Random Fore	140	gini	60	20	250	0.87671877	0.03956071
Random Forest	36	gini	50	30	250	0.87661468	0.04021243	Random Fore	141	gini	60	20	300	0.87535661	0.03811249
Random Forest	37	gini	50	30	300	0.87722448	0.03984745	Random Fore	142	gini	60	20	350	0.87573914	0.03877764
Random Forest	38	gini	50	30	350	0.87598475	0.03953557	Random Fore	143	gini	60	20	400	0.87565321	0.03988186
Random Forest	39	gini	50	30	400	0.87582618	0.0403764	Random Fore	144	gini	60	20	450	0.87645634	0.04008562
Random Forest	40	gini	50	30	450	0.87602545	0.03976248	Random Fore	145	gini	60	25	100	0.87679421	0.03959329
Random Forest	41	gini	50	35	100	0.87466517	0.03741314	Random Fore	146	gini	60	25	150	0.87697983	0.04002182
Random Forest	42	gini	50	35	150	0.87557415	0.03927937	Random Fore	147	gini	60	25	200	0.87792404	0.03967157
Random Forest	43	gini	50	35	200	0.87603114	0.03906793	Random Fore	148	gini	60	25	250	0.87530914	0.03800025
Random Forest	44	gini	50	35	250	0.87702572	0.03906947	Random Fore	149	gini	60	25	300	0.8770617	0.04056018
Random Forest	45	gini	50	35	300	0.87550938	0.04057923	Random Fore	150	gini	60	25	350	0.87640101	0.0402803
Random Forest	46	gini	50	35	350	0.87667745	0.04098106	Random Fore	151	gini	60	25	400	0.87648249	0.03946455
Random Forest	47	gini	50	35	400	0.87722015	0.04073328	Random Fore	152	gini	60	25	450	0.87592488	0.03938825
Random Forest	48	gini	50	35	450	0.87651405	0.03963382	Random Fore	153	gini	60	30	100	0.8760767	0.03843818
Random Forest	49	gini	50	40	100	0.87584862	0.04022317	Random Fore	154	gini	60	30	150	0.87617984	0.03952391
Random Forest	50	gini	50	40	150	0.87542157	0.03837815	Random Fore	155	gini	60	30	200	0.87550196	0.03981001
Random Forest	51	gini	50	40	200	0.87557257	0.03800611	Random Fore	156	gini	60	30	250	0.87623463	0.04051485
Random Forest	52	gini	50	40	250	0.87608897	0.03880761	Random Fore	157	gini	60	30	300	0.8749988	0.03865396
Random Forest	53	gini	50	40	300	0.87690913	0.04039996	Random Fore	158	gini	60	30	350	0.87655958	0.04091968
Random Forest	54	gini	50	40	350	0.87623246	0.03893351	Random Fore	159	gini	60	30	400	0.87679471	0.04009575
Random Forest	55	gini	50	40	400	0.8757528	0.03963644	Random Fore	160	gini	60	30	450	0.87671618	0.03995711
Random Forest	56	gini	50	40	450	0.87580908	0.03957799	Random Fore	161	gini	60	35	100	0.87679349	0.04050977
Random Forest	57	gini	50	45	100	0.87555236	0.03834675	Random Fore	162	gini	60	35	150	0.87631514	0.03984363
Random Forest	58	gini	50	45	150	0.87542722	0.03772816	Random Fore	163	gini	60	35	200	0.87513036	0.03903606
Random Forest	59	gini	50	45	200	0.87632195	0.04068049	Random Fore	164	gini	60	35	250	0.87568253	0.03859873
Random Forest	60	gini	50	45	250	0.87576071	0.03927258	Random Fore	165	gini	60	35	300	0.87602036	0.0390001
Random Forest	61	gini	50	45	300	0.87526568	0.03977634	Random Fore	166	gini	60	35	350	0.87538845	0.03867114
Random Forest	62	gini	50	45	350	0.87670315	0.03812022	Random Fore	167	gini	60	35	400	0.8765227	0.03991046
Random Forest	63	gini	50	45	400	0.87535585	0.04006663	Random Fore	168	gini	60	35	450	0.87549733	0.04030202
Random Forest	64	gini	50	45	450	0.87627782	0.03971618	Random Fore	169	gini	60	40	100	0.87512743	0.03724705
Random Forest	65	gini	50	50	100	0.87585986	0.03989153	Random Fore	170	gini	60	40	150	0.87623211	0.03989111
Random Forest	66	gini	50	50	150	0.87693778	0.03866296	Random Fore	171	gini	60	40	200	0.87575731	0.03879573
Random Forest	67	gini	50	50	200	0.87503282	0.03823747	Random Fore	172	gini	60	40	250	0.87636631	0.03885081
Random Forest	68	gini	50	50	250	0.87537346	0.03921199	Random Fore	173	gini	60	40	300	0.87746646	0.04071575
Random Forest	69	gini	50	50	300	0.87578281	0.03851766	Random Fore	174	gini	60	40	350	0.87616138	0.03897234
Random Forest	70	gini	50	50	350	0.87424993	0.03730484	Random Fore	175	gini	60	40	400	0.87488653	0.03763654
Random Forest	71	gini	50	50	400	0.87448626	0.03763657	Random Fore	176	gini	60	40	450	0.87570978	0.03911968
Random Forest	72	gini	50	50	450	0.8749788	0.03990947	Random Fore	177	gini	60	45	100	0.8759447	0.03821722
Random Forest	73	gini	50	55	100	0.87575715	0.03909773	Random Fore	178	gini	60	45	150	0.87592667	0.03865313
Random Forest	74	gini	50	55	150	0.87524141	0.03845449	Random Fore	179	gini	60	45	200	0.87734041	0.03936452
Random Forest	75	gini	50	55	200	0.87611184	0.0383573	Random Fore	180	gini	60	45	250	0.87573766	0.03841445
Random Forest	76	gini	50	55	250	0.87521113	0.03860927	Random Fore	181	gini	60	45	300	0.87555536	0.04034289
Random Forest	77	gini	50	55	300	0.87610276	0.03995542	Random Fore	182	gini	60	45	350	0.87621936	0.04036936
Random Forest	78	gini	50	55	350	0.87489485	0.04129555	Random Fore	183	gini	60	45	400	0.87594218	0.03966452
Random Forest	79	gini	50	55	400	0.87630479	0.03965082	Random Fore	184	gini	60	45	450	0.8757215	0.04028466
Random Forest	80	gini	50	55	450	0.87549473	0.03905905	Random Fore	185	gini	60	50	100	0.87491109	0.03841767
Random Forest	81	gini	50	60	100	0.87533563	0.03907686	Random Fore	186	gini	60	50	150	0.87703079	0.03869115
Random Forest	82	gini	50	60	150	0.87537041	0.0390653	Random Fore	187	gini	60	50	200	0.87605245	0.03889173
Random Forest	83	gini	50	60	200	0.8760617	0.03895207	Random Fore	188	gini	60	50	250	0.87536809	0.03912632
Random Forest	84	gini	50	60	250	0.87593555	0.03871134	Random Fore	189	gini	60	50	300	0.87609727	0.03986855
Random Forest	85	gini	50	60	300	0.87438395	0.03918064	Random Fore	190	gini	60	50	350	0.87567879	0.03909679
Random Forest	86	gini	50	60	350	0.87574602	0.03912087	Random Fore	191	gini	60	50	400	0.87648642	0.03918677
Random Forest	87	gini	50	60	400	0.87550807	0.03986851	Random Fore	192	gini	60	50	450	0.87657182	0.04016584
Random Forest	88	gini	50	60	450	0.87435185	0.0379296	Random Fore	193	gini	60	55	100	0.87556508	0.03833786
Random Forest	89	gini	50	65	100	0.87688812	0.03985956	Random Fore	194	gini	60	55	150	0.87649708	0.03804172
Random Forest	90	gini	50	65	150	0.87660239	0.04074325	Random Fore	195	gini	60	55	200	0.8772995	0.03944368
Random Forest	91	gini	50	65	200	0.87602761	0.03929437	Random Fore	196	gini	60	55	250	0.87550089	0.03906878
Random Forest	92	gini	50	65	250	0.87669952	0.03969331	Random Fore	197	gini	60	55	300	0.8752424	0.03807431
Random Forest	93	gini	50	65	300	0.87554456	0.04096229	Random Fore	198	gini	60	55	350	0.87588356	0.03745933
Random Forest	94	gini	50	65	350	0.87645303	0.03984918	Random Fore	199	gini	60	55	400	0.8765111	0.04083177
Random Forest	95	gini	50	65	400	0.87542945	0.03988984	Random Fore	200	gini	60	55	450	0.87580747	0.03970175
Random Forest	96	gini	50	65	450	0.87475846	0.03923706	Random Fore	201	gini	60	60	100	0.87611908	0.03925324
Random Forest	97	gini	50	70	100	0.87542615	0.03810409	Random Fore	202	gini	60	60	150	0.87628351	0.04020036
Random Forest	98	gini	50	70	150	0.8757078	0.03907994	Random Fore	203	gini	60	60	200	0.87565729	0.03937326
Random Forest	99	gini	50	70	200	0.87622691	0.03821637	Random Fore	204	gini	60	60	250	0.8766774	0.04011889
Random Forest	100	gini	50	70	250	0.87516697	0.03951193	Random Fore	205	gini	60	60	300	0.87652278	0.03925977
Random Forest	101	gini	50	70	300	0.8758879	0.03821671	Random Fore	206	gini	60	60	350	0.87693307	0.0412875
Random Forest	102	gini	50	70	350	0.8764366	0.03947602	Random Fore	207	gini	60	60	400	0.876414	0.04080703
Random Forest	103	gini	50	70	400	0.87529596	0.03877926	Random Fore	208	gini	60	60	450	0.87618714	0.04157754
Random Forest	104	gini	50	70	450	0.87496127	0.041052	Random Fore	209	gini	60	65	100	0.87448951	0.03703272
Random Forest	105	gini	50	75	100	0.875729	0.03921604	Random Fore	210	gini	60	65	150	0.87693066	0.04031691

Apéndice 3B – Resultados obtenidos de Random Forest – “Entropy”

Algoritmo	#	Criterio	n_estimator (cantidad de árboles)	Maxima Profundiad	min_samples _split	AUC	Desvío_AUC	Algoritmo	#	Criterio	n_estimator (cantidad de árboles)	Maxima Profundiad	min_samples _split	AUC	Desvío_AUC
Random Forest	212	entropy	50	10	100	0.87695236	0.04071092	Random Fores	290	entropy	90	40	300	0.87700649	0.040029
Random Forest	213	entropy	50	10	200	0.87776518	0.04185887	Random Fores	291	entropy	90	40	400	0.87726539	0.04114127
Random Forest	214	entropy	50	10	300	0.87747874	0.04123856	Random Fores	292	entropy	90	50	100	0.87712524	0.03995196
Random Forest	215	entropy	50	10	400	0.87791745	0.04209183	Random Fores	293	entropy	90	50	200	0.87858773	0.04075315
Random Forest	216	entropy	50	20	100	0.87728628	0.03926935	Random Fores	294	entropy	90	50	300	0.87692201	0.04020797
Random Forest	217	entropy	50	20	200	0.87726687	0.04000972	Random Fores	295	entropy	90	50	400	0.87671675	0.0400305
Random Forest	218	entropy	50	20	300	0.87775401	0.04081746	Random Fores	296	entropy	90	60	100	0.87645314	0.03929383
Random Forest	219	entropy	50	20	400	0.87613872	0.03912178	Random Fores	297	entropy	90	60	200	0.87736001	0.03977879
Random Forest	220	entropy	50	30	100	0.87561028	0.03943363	Random Fores	298	entropy	90	60	300	0.87838463	0.0413
Random Forest	221	entropy	50	30	200	0.87832409	0.04184378	Random Fores	299	entropy	90	60	400	0.87715739	0.03997068
Random Forest	222	entropy	50	30	300	0.87809137	0.04022017	Random Fores	300	entropy	90	70	100	0.87724331	0.03910386
Random Forest	223	entropy	50	30	400	0.87636499	0.03992304	Random Fores	301	entropy	90	70	200	0.87755782	0.03971853
Random Forest	224	entropy	50	40	100	0.87627174	0.03957144	Random Fores	302	entropy	90	70	300	0.8782439	0.0410873
Random Forest	225	entropy	50	40	200	0.87752975	0.04157866	Random Fores	303	entropy	90	70	400	0.8783525	0.04172032
Random Forest	226	entropy	50	40	300	0.87731007	0.04032841	Random Fores	304	entropy	90	80	100	0.87740477	0.04001626
Random Forest	227	entropy	50	40	400	0.87698128	0.04132301	Random Fores	305	entropy	90	80	200	0.87789336	0.04095327
Random Forest	228	entropy	50	50	100	0.87519769	0.03792583	Random Fores	306	entropy	90	80	300	0.87749774	0.04065174
Random Forest	229	entropy	50	50	200	0.87764387	0.04053064	Random Fores	307	entropy	90	80	400	0.8785765	0.04053705
Random Forest	230	entropy	50	50	300	0.87581433	0.03890036	Random Fores	308	entropy	110	10	100	0.87853955	0.04078369
Random Forest	231	entropy	50	50	400	0.87669507	0.04122627	Random Fores	309	entropy	110	10	200	0.87920341	0.04028569
Random Forest	232	entropy	50	60	100	0.87719715	0.04001492	Random Fores	310	entropy	110	10	300	0.87914166	0.04112556
Random Forest	233	entropy	50	60	200	0.87541224	0.03940801	Random Fores	311	entropy	110	10	400	0.87802933	0.04105146
Random Forest	234	entropy	50	60	300	0.87489715	0.03980396	Random Fores	312	entropy	110	20	100	0.87814032	0.03925461
Random Forest	235	entropy	50	60	400	0.87840066	0.04068123	Random Fores	313	entropy	110	20	200	0.87784236	0.04027767
Random Forest	236	entropy	50	70	100	0.87641012	0.03952666	Random Fores	314	entropy	110	20	300	0.87860694	0.04033725
Random Forest	237	entropy	50	70	200	0.87610635	0.03922412	Random Fores	315	entropy	110	20	400	0.87788936	0.0400899
Random Forest	238	entropy	50	70	300	0.87788523	0.04217239	Random Fores	316	entropy	110	30	100	0.87743318	0.04056294
Random Forest	239	entropy	50	70	400	0.87696414	0.04003709	Random Fores	317	entropy	110	30	200	0.87774034	0.04002521
Random Forest	240	entropy	50	80	100	0.8759751	0.04107328	Random Fores	318	entropy	110	30	300	0.8778511	0.04087196
Random Forest	241	entropy	50	80	200	0.87784419	0.04091481	Random Fores	319	entropy	110	30	400	0.87765756	0.04095993
Random Forest	242	entropy	50	80	300	0.87813929	0.04185551	Random Fores	320	entropy	110	40	100	0.8770362	0.04037516
Random Forest	243	entropy	50	80	400	0.87675155	0.03943353	Random Fores	321	entropy	110	40	200	0.8776912	0.04005952
Random Forest	244	entropy	70	10	100	0.87905841	0.0401361	Random Fores	322	entropy	110	40	300	0.87746609	0.04078223
Random Forest	245	entropy	70	10	200	0.87833076	0.0409749	Random Fores	323	entropy	110	40	400	0.8771732	0.04042734
Random Forest	246	entropy	70	10	300	0.87873035	0.04067567	Random Fores	324	entropy	110	50	100	0.87736722	0.04013045
Random Forest	247	entropy	70	10	400	0.87740326	0.04114707	Random Fores	325	entropy	110	50	200	0.87732134	0.03969772
Random Forest	248	entropy	70	20	100	0.87756564	0.0397363	Random Fores	326	entropy	110	50	300	0.87662604	0.03941838
Random Forest	249	entropy	70	20	200	0.87843179	0.04207074	Random Fores	327	entropy	110	50	400	0.87760021	0.04081741
Random Forest	250	entropy	70	20	300	0.87782518	0.03970253	Random Fores	328	entropy	110	60	100	0.87781045	0.04031269
Random Forest	251	entropy	70	20	400	0.87673427	0.03959083	Random Fores	329	entropy	110	60	200	0.87747399	0.03970736
Random Forest	252	entropy	70	30	100	0.87746485	0.04051422	Random Fores	330	entropy	110	60	300	0.87777364	0.03984647
Random Forest	253	entropy	70	30	200	0.87826195	0.04063035	Random Fores	331	entropy	110	60	400	0.87787764	0.04103836
Random Forest	254	entropy	70	30	300	0.87794713	0.04113535	Random Fores	332	entropy	110	70	100	0.87701756	0.03992904
Random Forest	255	entropy	70	30	400	0.8774586	0.04099985	Random Fores	333	entropy	110	70	200	0.87769366	0.04029407
Random Forest	256	entropy	70	40	100	0.87646081	0.03885353	Random Fores	334	entropy	110	70	300	0.87716265	0.04005185
Random Forest	257	entropy	70	40	200	0.87819597	0.0413504	Random Fores	335	entropy	110	70	400	0.87771939	0.04045585
Random Forest	258	entropy	70	40	300	0.87717188	0.04003331	Random Fores	336	entropy	110	80	100	0.87770188	0.03955687
Random Forest	259	entropy	70	40	400	0.87794287	0.04143015	Random Fores	337	entropy	110	80	200	0.87712953	0.04064568
Random Forest	260	entropy	70	50	100	0.87679619	0.03939914	Random Fores	338	entropy	110	80	300	0.87807878	0.04046088
Random Forest	261	entropy	70	50	200	0.87760908	0.03977979	Random Fores	339	entropy	110	80	400	0.87731236	0.0400601
Random Forest	262	entropy	70	50	300	0.87555269	0.04026279	Random Fores	340	entropy	130	10	100	0.87770978	0.04261046
Random Forest	263	entropy	70	50	400	0.87788292	0.04028997	Random Fores	341	entropy	130	10	200	0.87826719	0.04209186
Random Forest	264	entropy	70	60	100	0.87734603	0.0395135	Random Fores	342	entropy	130	10	300	0.87940038	0.04068261
Random Forest	265	entropy	70	60	200	0.87694583	0.0404459	Random Fores	343	entropy	130	10	400	0.87781711	0.04097515
Random Forest	266	entropy	70	60	300	0.87744781	0.03991617	Random Fores	344	entropy	130	20	100	0.87833651	0.03859638
Random Forest	267	entropy	70	60	400	0.87779068	0.04026261	Random Fores	345	entropy	130	20	200	0.87831733	0.04088795
Random Forest	268	entropy	70	70	100	0.87696923	0.03924221	Random Fores	346	entropy	130	20	300	0.87736863	0.0396921
Random Forest	269	entropy	70	70	200	0.87773342	0.04028269	Random Fores	347	entropy	130	20	400	0.87796622	0.03998957
Random Forest	270	entropy	70	70	300	0.8777826	0.04194794	Random Fores	348	entropy	130	30	100	0.87771442	0.03957575
Random Forest	271	entropy	70	70	400	0.87735319	0.0411951	Random Fores	349	entropy	130	30	200	0.87839413	0.04069855
Random Forest	272	entropy	70	80	100	0.87707209	0.04007201	Random Fores	350	entropy	130	30	300	0.87796007	0.03995931
Random Forest	273	entropy	70	80	200	0.87716638	0.04016537	Random Fores	351	entropy	130	30	400	0.87743083	0.04084078
Random Forest	274	entropy	70	80	300	0.87725002	0.04052527	Random Fores	352	entropy	130	40	100	0.87695781	0.03866437
Random Forest	275	entropy	70	80	400	0.87737334	0.04085657	Random Fores	353	entropy	130	40	200	0.87751348	0.04036304
Random Forest	276	entropy	90	10	100	0.8793092	0.04091418	Random Fores	354	entropy	130	40	300	0.87693539	0.04097662
Random Forest	277	entropy	90	10	200	0.87759379	0.04167646	Random Fores	355	entropy	130	40	400	0.87845543	0.04131107
Random Forest	278	entropy	90	10	300	0.8769071	0.04117228	Random Fores	356	entropy	130	50	100	0.87725959	0.03906449
Random Forest	279	entropy	90	10	400	0.87894419	0.04078248	Random Fores	357	entropy	130	50	200	0.87715267	0.03914582
Random Forest	280	entropy	90	20	100	0.87801965	0.04003091	Random Fores	358	entropy	130	50	300	0.87704918	0.0407225
Random Forest	281	entropy	90	20	200	0.87802648	0.03946966	Random Fores	359	entropy	130	50	400	0.87751118	0.04039321
Random Forest	282	entropy	90	20	300	0.8780122	0.04025027	Random Fores	360	entropy	130	60	100	0.87762418	0.04027672
Random Forest	283	entropy	90	20	400	0.87757519	0.04213493	Random Fores	361	entropy	130	60	200	0.87839426	0.04034439
Random Forest	284	entropy	90	30	100	0.87747333	0.04021228	Random Fores	362	entropy	130	60	300	0.87806171	0.04064273
Random Forest	285	entropy	90	30	200	0.87800956	0.04057576	Random Fores	363	entropy	130	60	400	0.87801055	0.04028522
Random Forest	286	entropy	90	30	300	0.87825219	0.04127677	Random Fores	364	entropy	130	70	100	0.87707226	0.03895302
Random Forest	287	entropy	90	30	400	0.8772633	0.04164158	Random Fores	365	entropy	130	70	200	0.87827058	0.04040807
Random Forest	288	entropy	90	40	100	0.87781724	0.04101787	Random Fores	366	entropy	130	70	300	0.87668477	0.03952977
Random Forest	289	entropy	90	40	200	0.87756427	0.04005837	Random Fores	367	entropy	130	70	400	0.87797279	0.04139937

Apéndidce 4A – Código de AdaBoost

```
## 0. Importar los paquetes necesarios junto con el set de datos

import pandas as pd
import sklearn as sk

from google.colab import files
uploaded = files.upload()


## 1. LEER DATOS...

data = pd.read_excel("final_v2.xlsx")

# 2. PREPARAR DATOS
#data = pd.get_dummies(data, columns=['area', 'salario'])

X = data[data.columns.drop(['ncodpers', 'ind_ecue_fin_ult1'])].values
y = data['ind_ecue_fin_ult1'].values

from sklearn import model_selection
from sklearn.ensemble import AdaBoostClassifier

for algorithm in ("SAMME", "SAMME.R"):
  for n_estimators in range (20,100,10):
    clf = sk.ensemble.AdaBoostClassifier(n_estimators=n_estimators)
    scores = sk.model_selection.cross_val_score(clf, X, y, scoring="roc
_auc", cv=5)
    print(algorithm, n_estimators, "exactitud:", scores.mean(), scores.
std(),'|')
```

Apéndice 4A – Resultados obtenidos de AdaBoost

Algoritmo	#	Algorithm	n parameters	AUC	Desvío_AUC
AdaBoost	1	SAMME	20	0.86266812	0.05091928
AdaBoost	2	SAMME	30	0.86333534	0.04994779
AdaBoost	3	SAMME	40	0.8641961	0.05058345
AdaBoost	4	SAMME	50	0.86704784	0.04913116
AdaBoost	5	SAMME	60	0.86681234	0.04948466
AdaBoost	6	SAMME	70	0.86579564	0.04934866
AdaBoost	7	SAMME	80	0.86578128	0.04915402
AdaBoost	8	SAMME	90	0.86584354	0.04915836
AdaBoost	9	SAMME.R	20	0.86266812	0.05091928
AdaBoost	10	SAMME.R	30	0.86333534	0.04994779
AdaBoost	11	SAMME.R	40	0.8641961	0.05058345
AdaBoost	12	SAMME.R	50	0.86704784	0.04913116
AdaBoost	13	SAMME.R	60	0.86681234	0.04948466
AdaBoost	14	SAMME.R	70	0.86579564	0.04934866
AdaBoost	15	SAMME.R	80	0.86578128	0.04915402
AdaBoost	16	SAMME.R	90	0.86584354	0.04915836

Apéndice 5A – Código de Redes Neuronales

```
from sklearn import model_selection
from sklearn import ensemble
from sklearn.model_selection import cross_val_score
from sklearn.neural_network import MLPClassifier

for solver in ("lbfgs", "sgd", "adam"):
  for activation in ("identity", "logistic", "tanh", "relu"):
    for hidden_layer_sizes in ((5,3), (7,5,3), (9,7,5,3), (11,9,7,5,3),
 (13,11,9,7,5,3), (21,15,9,3)):
      clf = MLPClassifier(solver=solver, activation=activation, alpha=1
e-5, hidden_layer_sizes=hidden_layer_sizes, random_state=1)
      scores = sk.model_selection.cross_val_score(clf, X, y, scoring="r
oc_auc", cv=5)
      print(solver, activation, hidden_layer_sizes, scores.mean(), scor
es.std(),'|')
```

Apéndice 5B – Resultados obtenidos de Redes Neuronales

Algoritmo	#	Algoritmo	Funcion de activacion	Estructura	AUC	Desvío_AUC		Algoritmo	#	Algoritmo	Funcion de activacion	Estructura	AUC	Desvío_AUC
Redes Neuronales	1	lbfgs	identity	(5,3)	0.85811557	0.0578678		Redes Neurona	37	sgd	tanh	(5,3)	0.88156738	0.04614304
Redes Neuronales	2	lbfgs	identity	(7,5,3)	0.85928944	0.05501596		Redes Neurona	38	sgd	tanh	(7,5,3)	0.87466926	0.05049405
Redes Neuronales	3	lbfgs	identity	(9,7,5,3)	0.85701674	0.05775205		Redes Neurona	39	sgd	tanh	(9,7,5,3)	0.8831407	0.04431997
Redes Neuronales	4	lbfgs	identity	(11,9,7,5,3)	0.85960432	0.05523463		Redes Neurona	40	sgd	tanh	(11,9,7,5,3)	0.88636741	0.03915643
Redes Neuronales	5	lbfgs	identity	(13,11,9,7,5,3)	0.8595505	0.05570637		Redes Neurona	41	sgd	tanh	(13,11,9,7,5,3)	0.88414322	0.04041246
Redes Neuronales	6	lbfgs	identity	(21,15,9,3)	0.85963468	0.05621495		Redes Neurona	42	sgd	tanh	(21,15,9,3)	0.88427476	0.0442243
Redes Neuronales	7	lbfgs	logistic	(5,3)	0.87682343	0.0470888		Redes Neurona	43	sgd	relu	(5,3)	0.86674537	0.04934428
Redes Neuronales	8	lbfgs	logistic	(7,5,3)	0.87278381	0.04775394		Redes Neurona	44	sgd	relu	(7,5,3)	0.87939539	0.04651211
Redes Neuronales	9	lbfgs	logistic	(9,7,5,3)	0.69669365	0.08153834		Redes Neurona	45	sgd	relu	(9,7,5,3)	0.87996558	0.04120093
Redes Neuronales	10	lbfgs	logistic	(11,9,7,5,3)	0.44824952	0.06788125		Redes Neurona	46	sgd	relu	(11,9,7,5,3)	0.87501784	0.04277814
Redes Neuronales	11	lbfgs	logistic	(13,11,9,7,5,3)	0.3564934	0.14688834		Redes Neurona	47	sgd	relu	(13,11,9,7,5,3)	0.877061	0.04895362
Redes Neuronales	12	lbfgs	logistic	(21,15,9,3)	0.27859535	0.10235737		Redes Neurona	48	sgd	relu	(21,15,9,3)	0.88413117	0.04128863
Redes Neuronales	13	lbfgs	tanh	(5,3)	0.87609877	0.04437908		Redes Neurona	49	adam	identity	(5,3)	0.85760957	0.05769905
Redes Neuronales	14	lbfgs	tanh	(7,5,3)	0.88414944	0.04536765		Redes Neurona	50	adam	identity	(7,5,3)	0.86202298	0.05307184
Redes Neuronales	15	lbfgs	tanh	(9,7,5,3)	0.88055858	0.04949618		Redes Neurona	51	adam	identity	(9,7,5,3)	0.86025403	0.05271358
Redes Neuronales	16	lbfgs	tanh	(11,9,7,5,3)	0.88782224	0.04312936		Redes Neurona	52	adam	identity	(11,9,7,5,3)	0.86119867	0.05247567
Redes Neuronales	17	lbfgs	tanh	(13,11,9,7,5,3)	0.88298376	0.04840562		Redes Neurona	53	adam	identity	(13,11,9,7,5,3)	0.85955517	0.0556278
Redes Neuronales	18	lbfgs	tanh	(21,15,9,3)	0.88432028	0.04271349		Redes Neurona	54	adam	identity	(21,15,9,3)	0.8604346	0.05262875
Redes Neuronales	19	lbfgs	relu	(5,3)	0.76988594	0.16781753		Redes Neurona	55	adam	logistic	(5,3)	0.88346259	0.04356749
Redes Neuronales	20	lbfgs	relu	(7,5,3)	0.83562671	0.09348695		Redes Neurona	56	adam	logistic	(7,5,3)	0.88342494	0.03952955
Redes Neuronales	21	lbfgs	relu	(9,7,5,3)	0.87978031	0.04448801		Redes Neurona	57	adam	logistic	(9,7,5,3)	0.88190435	0.04279432
Redes Neuronales	22	lbfgs	relu	(11,9,7,5,3)	0.8752064	0.0446602		Redes Neurona	58	adam	logistic	(11,9,7,5,3)	0.88614594	0.04052862
Redes Neuronales	23	lbfgs	relu	(13,11,9,7,5,3)	0.52003181	0.04006363		Redes Neurona	59	adam	logistic	(13,11,9,7,5,3)	0.88243527	0.04055279
Redes Neuronales	24	lbfgs	relu	(21,15,9,3)	0.88662863	0.04011936		Redes Neurona	60	adam	logistic	(21,15,9,3)	0.88252373	0.03983591
Redes Neuronales	25	sgd	identity	(5,3)	0.86256137	0.05282942		Redes Neurona	61	adam	tanh	(5,3)	0.88135704	0.03772752
Redes Neuronales	26	sgd	identity	(7,5,3)	0.85989317	0.05408672		Redes Neurona	62	adam	tanh	(7,5,3)	0.88024338	0.04232707
Redes Neuronales	27	sgd	identity	(9,7,5,3)	0.85973702	0.05387656		Redes Neurona	63	adam	tanh	(9,7,5,3)	0.88303656	0.04330461
Redes Neuronales	28	sgd	identity	(11,9,7,5,3)	0.86030911	0.05257009		Redes Neurona	64	adam	tanh	(11,9,7,5,3)	0.88316276	0.04036142
Redes Neuronales	29	sgd	identity	(13,11,9,7,5,3)	0.86190868	0.05384885		Redes Neurona	65	adam	tanh	(13,11,9,7,5,3)	0.87531416	0.05213801
Redes Neuronales	30	sgd	identity	(21,15,9,3)	0.86048838	0.05306772		Redes Neurona	66	adam	tanh	(21,15,9,3)	0.87867699	0.03932291
Redes Neuronales	31	sgd	logistic	(5,3)	0.86124723	0.05138885		Redes Neurona	67	adam	relu	(5,3)	0.88166351	0.04571206
Redes Neuronales	32	sgd	logistic	(7,5,3)	0.7698591	0.08254627		Redes Neurona	68	adam	relu	(7,5,3)	0.87380403	0.04242091
Redes Neuronales	33	sgd	logistic	(9,7,5,3)	0.72989652	0.07037349		Redes Neurona	69	adam	relu	(9,7,5,3)	0.88311551	0.03740613
Redes Neuronales	34	sgd	logistic	(11,9,7,5,3)	0.45320431	0.0683945		Redes Neurona	70	adam	relu	(11,9,7,5,3)	0.87960204	0.0369097
Redes Neuronales	35	sgd	logistic	(13,11,9,7,5,3)	0.3591786	0.14671174		Redes Neurona	71	adam	relu	(13,11,9,7,5,3)	0.87237822	0.05291438
Redes Neuronales	36	sgd	logistic	(21,15,9,3)	0.30764215	0.10930984		Redes Neurona	72	adam	relu	(21,15,9,3)	0.88415551	0.03962361

Apéndice 5B – Extracción de Variables relevantes

```
## RANDOM FOREST
# 2.MEJOR MODELO

from sklearn import model_selection
from sklearn.ensemble import RandomForestClassifier

criterion="gini"
n_estimator=60
max_depth=10
min_samples_split=200
clf = sk.ensemble.RandomForestClassifier(criterion=criterion, n_estimat
ors=n_estimators, max_depth=max_depth, min_samples_split=min_samples_sp
lit)
scores = sk.model_selection.cross_val_score(clf, X, y, scoring="roc_auc
", cv=5)
print(scores.mean(), scores.std(),'\n')


import numpy as np
clf.fit(X, y)
importances=clf.feature_importances_
std = np.std([tree.feature_importances_ for tree in clf.estimators_],
             axis=0)

indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")

for f in range(X.shape[1]):
    print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indic
es[f]]))
```

Referencias

[i] Patrutiu-Baltes, L. (2016). Inbound Marketing-the most important digital marketing strategy. *Bulletin of the Transilvania University of Brasov. Economic Sciences. Series V*, *9*(2), 61.

[ii] *Mitik, M., Korkmaz, O., Karagoz, P., Toroslu, I. H., & Yucel, F. (2017). Data Mining Approach for Direct Marketing of Banking Products with Profit/Cost Analysis. The Review of Socionetwork Strategies, 11(1), 17–31.doi:10.1007/s12626-017-0002-5*

[iii] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27-34.

[iv] LAMB, Charles W.; HAIR, Joe F.; MCDANIEL, Carl. *Essentials of marketing*. Cengage Learning, 2011.

[v] Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases*, 229-238.

[vi] Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer

[vii] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[viii] Weiss, S. I., and Kulikowski, C. 1991. Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning, and Expert Systems. San Francisco, Calif.: Morgan Kaufmann.

[ix] Jain, A. K., and Dubes, R. C. 1988. Algorithms for Clustering Data. Englewood Cliffs, N.J.: PrenticeHall.

[x] Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; and Verkamo, I. 1996. Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining, eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 307–328. Menlo Park, Calif.: AAAI Press.

[xi] Rodrigo C. Barros, André C.P.L.F de Carvalho, Alex A. Freita, “Automatic Design of Decision-Tree Induction Algorithms”, 2015 - ISBN 978-3-319-14230-2, 978-3-319-14231-9

[xii] Sahu, S. K., Prasad, M. B. N. V., & Tripathy, B. K. A Support Vector Machine Binary Classification and Image Segmentation of Remote Sensing Data of Chilika Lagloon.

[xiii] Reyzin, L., & Schapire, R. E. (2006, June). How boosting the margin can also boost classifier complexity. In *Proceedings of the 23rd international conference on Machine learning* (pp. 753-760). ACM.

[xiv] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.

[xv] https://www.kaggle.com/contact

[xvi] STEINMANN, Michael; MATEI, Sorin Adam; COLLMANN, Jeff. A theoretical framework for ethical reflection in big data research. En *Ethical Reasoning in Big Data*. Springer, Cham, 2016. p. 11-27.