UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

TESIS

“Optimización de las campañas de marketing en la Industria Financiera: Un enfoque basado en Machine Learning”

PARA OBTENER EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS EN CIENCIA DE LA COMPUTACIÓN CON MENCIÓN EN LA ESPECIALIDAD: COMPUTACIÓN CIENTÍFICA

ELABORADA POR:

EDMUNDO DE ELVIRA MORI ORRILLO

ASESOR:

DR. JOSÉ MANUEL CASTILLO CARA

LIMA – PERÚ

2022

El esfuerzo de mis padres por darme una educación de calidad ha hecho posible los logros académicos y profesionales que he conseguido; a ellos va dedicado este trabajo.

Agradecimientos

Mis padres han dedicado los mejores años de su vida a proveerme una sólida escala de valores y darme la mejor formación académica desde mi infancia hasta estos días. Esto ha posibilitado la consecución de los logros académicos y profesionales que estoy consiguiendo. Mi eterno agradecimiento a ellos.

Mi Alma Mater, la Universidad Nacional de Ingeniería, personificada en la Facultad de Ingeniería Industrial y de Sistemas y en la Facultad de Ciencias, la cual se ha encargado de proveerme la sólida formación académica que poseo. A mis profesores de estas dos facultades mi sincero agradecimiento.

Al doctor Manuel Castillo-Cara, mi profesor en el curso de Minería de Datos en la Maestría de Ciencias de la Computación de la Facultad de Ciencias, quien me ha provisto sólidos conocimientos referidos a la construcción de modelos analíticos. Además, su asesoramiento y guía en el desarrollo de esta tesis ha hecho posible su exitosa finalización. A él mi infinito agradecimiento.

Finalmente, agradezco a mis compañeros de trabajo de la institución donde laboro, Ellos con sus consejos y enseñanzas posibilitan día a día mi crecimiento profesional.

Resumen

Las campañas de marketing que implementan las organizaciones financieras están centradas en el producto. Las empresas del sistema ofertan en promedio 17 productos entre activos, pasivos y de servicios; los valores de los indicadores de efectividad y de rentabilidad de estas campañas no son óptimos. Por tanto, para incrementar el valor de estos indicadores es necesario construir un modelo analítico denominado "Próxima Mejor Oferta" (NBO, por sus siglas del inglés) que permita definir el conjunto de productos a ofertar, priorizados según su propensión y rentabilidad.

Para desarrollar este modelo, en primer lugar, se calcula la rentabilidad esperada utilizando modelos predictivos que utilizan los siguientes algoritmos: Regresión Lineal, Bosques Aleatorios, Regresión Lineal Regularizada, Regresión con Vectores de Apoyo, etc; en segundo lugar, calculamos las probabilidades de adquisición de cada producto utilizando modelos de clasificación; los algoritmos utilizados son: Regresión Logística, Arboles de Decisión, Máquinas de Soporte Vectorial, entre otros. Para calcular las probabilidades de adquisición se utilizan diferentes modelos de clasificación; debido a esto, para compararlas entre sí, es necesario estandarizarlas utilizando un único modelo. La estandarización se realiza mediante la utilización de modelos ensamblados que combinan diferentes algoritmos de clasificación o empleando un modelo basado en inteligencia artificial denominado Redes Neuronales. La comparación de la capacidad de generalización de cada método nos indica que el modelo ensamblado obtiene los mejores resultados de estandarización, esta capacidad se calculó utilizando el indicador Área Bajo la Curva (AUC, por sus siglas del inglés). La probabilidad estandarizada y la rentabilidad serán las variables usadas para construir la función de priorización, output del modelo NBO, la cual permite ordenar los productos financieros de acuerdo a su grado de aceptabilidad.

El modelo NBO permite al área de Inteligencia Comercial construir estrategias de marketing visión cliente con una efectividad y rentabilidad mayor. Esta afirmación se demuestra comparando los resultados de las campañas comerciales de los 6 primeros meses del año 2021, donde se utilizó reglas de experto, con los resultados de las campañas comerciales de los últimos 6 meses del mismo año, donde se diseñaron estrategias basadas en el modelo NBO. La comparación obtuvo como resultado que la efectividad y la rentabilidad de las campañas comerciales se incrementó en los 6 últimos meses del año.

Abstract

Marketing campaigns implemented by financial organizations are focused on the product. The companies in the system offer an average of 17 different products between assets, liabilities and services; obtaining that the effectiveness and profitability of these campaigns is not optimal. Therefore, to increase these indicators, it is necessary to build an analytical model called "Next Best Offer" (NBO) that allows defining the set of products to be offered, prioritized according to their propensity and profitability.

To develop this model, first of all, the expected profitability is calculated using predictive models that use the following algorithms: Linear Regression, Random Forests, Regularized Linear Regression, Regression with Support Vectors, etc; secondly, we calculate the acquisition probabilities of each product using classification models that use the following algorithms: Logistic Regression, Decision Trees, Support Vector Machines, among others. Different analytical models are used to calculate acquisition probabilities, due to this to compare them with each other, it is necessary to standardize them using a single model. This standardization can be done using assembled models that combine different classification algorithms or using a model based on artificial intelligence called Neural Networks. The comparison of the prediction capacity of each method indicates that the assembled model obtains the best standardization results, this capacity was calculated using the indicator Area Under the Curve (AUC, acronym in the English language). The standardized probability and profitability will be the variables used to build the prioritization function, output of the NBO model, which allows ordering financial products according to their degree of acceptability.

The NBO model allows the Business Intelligence area to build marketing strategies customer vision with greater effectiveness and profitability; This affirmation is demonstrated by analyzing the results of the commercial campaigns of the first 6 months of the year 2021. These campaigns were carried out using marketing strategies product vision, in their construction, empirical rules called expert rules were used, and comparing them with the results of the commercial campaigns of the last 6 months of the same year (strategies built using the NBO model). The comparison obtained as a result that the effectiveness and profitability of each campaign increased substantially in the last 6 months.

Índice General

Índice de Figuras

Índice de Tablas

Índice de Acrónimos

NBO Próxima Mejor Oferta (Next Best Offer)

CRM Gestión de Relaciones con Clientes (Customer RelationShip Management)

BA Analítica de Negocios (Business Analytics)

IT Tecnologías de Información (Information Technologies)

ANN Red Neuronal Artificial (Artificial Neural Network)

ESS Suma de los Cuadrados de los Errores (Error Sum of Square)

NL Logaritmo Natural (Natural Logarithm)

KNN K-Vecinos mas Cercanos (K-Nearest Neighbors)

SVM Máquina de Soporte Vectorial (Support Vector Machine)

PCA Análisis de Componentes Principales (Principal Component Analysis)

PSI Índice de Estabilidad de la Población (Population Stability Index)

SGD Descenso de Gradiente Estocástico (Stochastic Gradient Descent)

AUC Área Bajo la Curva (Area Under Curve)

ROC Característica Operativa del Receptor (Receiver Operating Characteristic)

MSE Error Cuadrático Medio (Mean Square Error)

XGB Aumento de Gradiente Extremo (Xtreme Gradient Boost)

LGBM Maquina Leve de Gradiente Ascendente (Light Gradient Boosting Machine)

GBC Clasificador de Gradiente Ascendente (Gradient Boosting Classifier)

SVC Clasificador de Soporte Vectorial (Support Vector Classifier)

Capítulo 1

Planteamiento del Problema

La existencia de un mercado cada vez más competitivo y la disminución de la capacidad adquisitiva de la población conduce a las empresas del sector financiero a rediseñar sus estrategias comerciales. Ante ello, nace la necesitada de diseñar estrategias de marketing visión cliente utilizando un enfoque de *Machine Learning*.

1.1. Descripción de la Problemática

En la actualidad, en las empresas modernas, se reconoce ampliamente que el análisis de la información es la piedra angular del desarrollo de todos los campos de la actividad económica. Por ello, es obligatorio que la infraestructura tecnológica dentro de las empresas esté fuertemente adaptada para respaldar un crecimiento continuo a lo largo de la línea de tiempo empresarial. Además, la utilización del aprendizaje automático, entendido como una ciencia de los algoritmos, permite a las organizaciones extraer patrones (minería de datos) que optimizan la toma de decisiones. En esta perspectiva, las empresas deben diseñar sus estrategias comerciales para ser aplicadas en una campaña de marketing basada en el cliente. Esta visión obliga a construir un modelo analítico visión cliente denominado Próxima Mejor Oferta (NBO, por sus siglas del inglés), que se construye a partir de modelos predictivos que calculan las rentabilidades esperadas y las probabilidades de adquisición por producto financiero.

Debido a la digitalización, la transformación del marketing actual está impulsada por el uso de la tecnología y por una visión de marketing centrada en el cliente. Esto permite a la organización ofertar al mercado productos financieros que el cliente necesita para satisfacer sus necesidades. En este contexto, el crecimiento exponencial de los datos y su correcta utilización ofrecen valor comercial y una ventaja competitiva.

Las probabilidades estandarizadas y la rentabilidad esperada (en caso de que la persona tome el producto) intervienen, a través de un producto aritmético, en la construcción de una función de priorización (característica fundamental del modelo NBO) que proporciona un conjunto de productos priorizados que se le debe ofertar a los clientes y no clientes de la organización.

1.2. Formulación del Problema

¿En cuánto se incrementará la eficiencia de las campañas comerciales al utilizar el modelo NBO? y ¿Cuál es el método óptimo para estandarizar las probabilidades obtenidas de los modelos estadísticos?

1.3. Hipótesis

En la medida que las hipótesis orientan el proceso de investigación y condicionan su diseño, es necesario que estén bien definidas para que puedan validarse estadísticamente y permitan llegar a conclusiones concretas.

1.3.1. Hipótesis General

El uso de un modelo analítico denominado NBO incrementa la efectividad y la rentabilidad de las campañas comerciales.

1.3.2. Hipótesis Específicas

- El modelo NBO permite convertir las campañas de marketing centradas en el producto en campañas comerciales centradas en el cliente.
- Las estrategias comerciales, construidas con la información que brinda el modelo NBO, permiten a la organización satisfacer la demanda de productos financieros.
- Las capacidades de generalización de los modelos de rentabilidad están por encima del 70%.
- Los modelos ensamblados constituyen el mejor método para estandarizar las probabilidades de adquisición de los productos financieros.

1.4. Justificación

Todo proyecto de investigación tiene un costo, el cual debe ser justificado por su contribución científica, técnica y social.

1.4.1. Justificación Social

Para diseñar estrategias de campañas de marketing que permitan ofrecer al consumidor productos que satisfagan sus necesidades, es necesario contar con un modelo NBO construido en base a modelos analíticos (modelos de propensión y modelos de rentabilidad). El modelo NBO permite definir un conjunto de productos financieros clasificados de acuerdo a las necesidades consumidor. Además, la utilización de este modelo incrementa los ingresos de los trabajadores de venta.

1.4.2. Justificación Técnica

El incremento de la oferta de productos financieros, demanda a las organizaciones del sector a rediseñar sus campañas comerciales (marketing visión cliente). Para lograrlo, deben construir un modelo analítico denominado NBO. La construcción de este modelo requiere desarrollar modelos estadísticos que calculen la probabilidad de toma del producto, los cuales deben ser estandarizados para que sean comparables entre sí. Además, se requiere construir modelos predictivos que calculen la rentabilidad esperada de acuerdo al perfil de la persona.

1.5. Objetivos

El objetivo central de esta investigación es desarrollar un modelo analítico (NBO) que permita construir campañas de marketing que satisfagan las necesidades del mercado e incrementen la productividad de las empresas del sector financiero.

1.5.1. Objetivo General

Implementar una metodología basada en la Analítica de Negocios (BA, por sus siglas del inglés) usando técnicas de aprendizaje automático e inteligencia artificial, esto optimiza la toma de decisiones en el contexto de las campañas comerciales.

1.5.2. Objetivos Específicos

- Utilizar el BA para la construcción de estrategias de marketing que optimicen el resultado de las campañas comerciales.
- Construir nuevos modelos analíticos que calculen la probabilidad de adquisición de un producto financiero, ya que los modelos actualmente usados tienen poca capacidad de predicción debido a su antigüedad y a las fluctuaciones del mercado.
- Incluir el concepto de rentabilidad en el diseño de las estrategias de las campañas comerciales.
- Determinar el mejor método de estandarización de las probabilidades de adquisición obtenidas de los modelos estadísticos.
- Construir una función de priorización que ordene un conjunto de productos financieros, ofertados al mercado, por su grado de aceptación.

Capítulo 2

Marco Teórico

En este capítulo, se realiza una descripción detallada de un conjunto de modelos predictivos usados para calcular la probabilidad de adquisición y la rentabilidad de un producto financiero. Asimismo, se analiza la utilización de la Red Neuronal y de los modelos ensamblados de predicción en la estandarización de las probabilidades que salen de los modelos predictivos.

2.1. Estado del Arte

Las organizaciones financieras para colocar un producto en el mercado construyen estrategias comerciales utilizando herramientas informáticas y modelos analíticos. Estas herramientas permiten comprender las necesidades de los clientes y no clientes a los que va dirigida la campaña de marketing [1].

Las herramientas informáticas se utilizan para consolidar la información que se encuentra dispersa en diferentes fuentes y formatos. Esta consolidación permite analizar diferentes tipos de variables (comerciales, transaccionales, demográficas, etc.), que son utilizadas en la construcción de los modelos analíticos.

Los modelos de propensión calculan la probabilidad de toma de un producto financiero. Además, si la persona a la cual se le ofrece el producto (a partir de ahora denominado *lead*) adquiere el producto, otro modelo calculará la rentabilidad.

Debido al constante decrecimiento de la capacidad de consumo de la población y al incremento de la competencia en la industria financiera, nace la necesidad de utilizar sistemas expertos (inteligencia artificial) que sean capaces, a partir de la gestión empresarial, recopilar en un programa informático aspectos relacionados con la auditoría, la fiscalidad, la planificación, el análisis financiero y la contabilidad financiera, con el fin de incrementar la productividad de la industria [2].

Las Redes Neuronales y los modelos ensamblados se utilizan para estandarizar las probabilidades provistas por los modelos analíticos, la estandarización es necesaria ya que los modelos son desarrollados utilizando diferentes algoritmos. El producto aritmético de estas probabilidades y la rentabilidad esperada nos da una función de priorización que determina el conjunto de productos priorizados que la organización debe ofrecer a un *lead*.

2.1.1. Aplicación de la minería de datos y del aprendizaje automático en el marketing financiero.

El avance de la tecnología ha permitido la aparición de nuevas técnicas de procesamiento y análisis de la información. Estos avances se extienden al campo de las decisiones empresariales, específicamente al campo del marketing. En el marketing tradicional las decisiones se basan fundamentalmente en un análisis descriptivo de la información, mientras que en el marketing moderno, la utilización de la Minería de Datos y del Aprendizaje Automático proveen de modelos cuantitativos que sirven para extraer el conocimiento de la información disponible.

Los modelos cuantitativos se utilizan para construir estrategias de marketing visión cliente. En esta perspectiva, el modelo NBO es la mejor herramienta para alcanzar este objetivo. Los modelos también calculan la rentabilidad esperada de la campaña comercial y la probabilidad de ocurrencia de los siguientes eventos: el contacto telefónico con el lead, el incumplimiento de pago y la fuga del cliente en un horizonte de tiempo. Estos participan en la elaboración de las estrategias utilizadas en las campañas de marketing [3].

Para calcular la probabilidad de ocurrencia de los eventos descritos en el párrafo anterior se utilizan modelos basados en los siguientes algoritmos: Regresión Logística, Random Forest, Arboles de Decisión con Aumento de Gradiente, Retro Propagación, entre otros; mientras que los modelos utilizados para predecir la rentabilidad esperada se basan en algoritmos de Regresión Lineal Regularizada, Regresión con Vectores de Apoyo, entre otros. La calidad de estos modelos depende del pre-procesamiento de las variables independientes, debido a que se elimina información redundante, se imputan valores nulos, se selecciona las variables principales y se eliminan variables altamente correlacionadas. Una vez identificadas las variables que intervienen en el modelo analítico se construyen diferentes modelos utilizando diferentes algoritmos y se selecciona aquel con mayor capacidad de generalización. Para medir la calidad de los modelos predictivos se usan diferentes métricas de medición como: el Área Bajo la Curva (AUC, por sus siglas del inglés), el coeficiente de determinación, el error cuadrático medio, entre otras [4].

En algunos casos, la variable dependiente (evento a predecir) se encuentra desbalanceada; es decir, la cantidad de leads que adquieren el producto financiero es

significativamente menor a la cantidad de aquellos que no lo adquieren. Esto da origen a que los modelos predictivos sean sesgados e inexactos.

Para resolver este problema se utilizan diversas técnicas analíticas, la mejor técnica sugerida en la literatura sugiere segmentar previamente la base de datos. En la Figura 1 se muestra los pasos que se deben seguir para elaborar un modelo con una variable dependiente no balanceada. En primer lugar, se segmenta la base de datos en tres conjuntos (C1, C2, C3), con cada conjunto de datos se entrena un modelo predictivo (M1, M2 y M3) y luego el resultado de la predicción se lo combina en una única tabla; de esta manera, la capacidad de predicción del evento es mayor a la que pudiésemos haber obtenido entrenando el modelo sin la segmentación previa. [5]

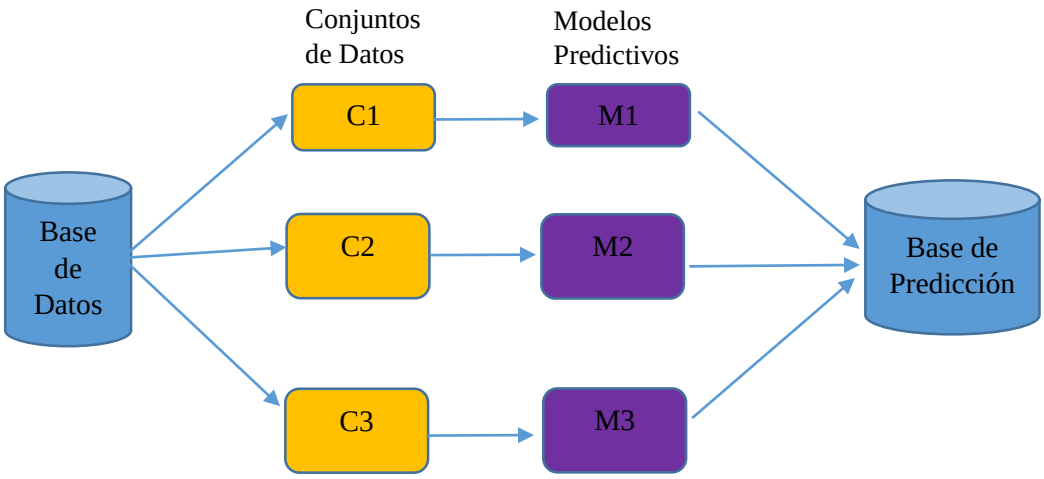


Figura 1: Diagrama de la metodología basada en conjuntos
Fuente: Guest Blog [5]

Este enfoque permite identificar grupos con mayor porcentaje de ocurrencia del evento a predecir, incrementando la capacidad de generalización del modelo.

2.1.2. Las Redes Neuronales Artificiales en el tratamiento de la información financiera.

Las variables que forman el modelo denominado Redes Neuronales se relacionan entre ellas utilizando relaciones matemáticas complejas y no tiene una forma determinada; es decir, no es paramétrico y se usa en finanzas y economía para calcular la probabilidad de adquisición de un producto financiero, la probabilidad de incumplimiento de pago (Riesgo Crediticio), la probabilidad de fraude financiero y la probabilidad de recuperar un crédito colocado. También son utilizadas para estandarizar las probabilidades obtenidas de los modelos estadísticos.

El modelo fue creado tratando de imitar la estructura y la capacidad de adaptarse del cerebro humano. Por esto se lo representa utilizando nodos interconectados que simulan a las neuronas y dendritas del cerebro. Poseen capas de entrada, de salida y ocultas. La capa de entrada se denomina sensorial y está formada por nodos o neuronas que reciben la información del entorno o de los modelos estadísticos (estandarización de probabilidades); las capas ocultas procesan la información y las de salida proveen las respuestas del sistema [6]

En la actualidad, los sistemas de ayuda a la decisión han reemplazado el término información por el de conocimiento, lo cual permite incluir en el proceso de toma de decisiones aspectos cualitativos en el tratamiento de datos, así como el saber acumulado de especialistas en el área de trabajo objeto del problema a resolver. Debido a esto, nace la necesidad de utilizar sistemas expertos (inteligencia artificial) que sean capaces, a partir de la gestión empresarial, de recopilar en un programa informático aspectos relacionados con la auditoría, la fiscalidad, la planificación, el análisis financiero y la contabilidad financiera, con el fin de incrementar la productividad de la industria. [7]

En muchas oportunidades, los resultados que dan los modelos estadísticos son los mismos a los que dan las redes neuronales artificiales. Esto se ve cuando la evaluación del comportamiento de las acciones en el mercado de valores se realiza utilizando modelos predictivos como el análisis discriminante, *lógit* y la partición recursiva, por un lado; y por otro, se evalúa utilizando un perceptrón multicapa (redes neuronales).

En muchos casos los excelentes resultados que da la utilización de las redes neuronales no se condice con la realidad (testeo), debido a que la información es sumamente heterogénea en lo cuantitativo e impredecible en lo cualitativo.

2.1.3. Optimización de los resultados de las campañas comerciales mediante la utilización de modelos analíticos.

Con el rápido avance de los sistemas computacionales y la creación de grandes bases de datos, emergen nuevas técnicas de Inteligencia Artificial adscritas a los sistemas expertos como la Minería de datos. Los avances en los sistemas han crecido a la par de la necesidad de extraer información valiosa para la toma de decisiones en un mundo globalizado.

Las campañas comerciales implementadas en la industria financiera utilizan los modelos segmentación para definir el público objetivo al cual se le va a ofrecer un producto financiero. Su eficacia se mide utilizando diferentes indicadores de rentabilidad (margen financiero, ingreso neto, saldos contables, etc.) y de gestión (efectividad y monto desembolsado) [8].

La existencia de monopolios permite que un marketing centrado en el producto conquiste mercados y optimice la rentabilidad de las empresas. El libre mercado conlleva a que existen otros operadores comerciales que incrementan la competencia. Esta realidad da origen al reemplazo del marketing centrado en el producto por un marketing centrado en el cliente, para lograr esto las organizaciones deben rediseñar sus campañas comerciales en función de las necesidades del mercado [9].

Se optimizan las campañas comerciales maximizando su rentabilidad y su efectividad. La rentabilidad se maximiza mediante un modelo de programación lineal entera donde la función objetivo tiene por variables la propensión, la contactabilidad, la productividad y los costos por canal. Esta función está sujeta a restricciones de negocio relacionadas con la capacidad de los canales de venta:

$$\text{Max} \sum_{(i\ in\ I,\ j\ in\ J)} \left(PotCanal_{ij} * GestCanal_j * ContCanal_j * Prob_{ij} * Rent_{ij} - CostCanal_j\right) * XCanal_{ij}$$

Donde:

i: Es el código identificador del lead.

j: Es el código identificador del producto.

$PotCanal_{ij}$: Indica si al lead "i" se le puede ofrecer el producto "j", a través de un determinado canal.

$GestCanal_j$: Porcentaje de gestión por canal y por producto "j", indica la productividad del canal.

$ContCanal_j$: Porcentaje de contacto efectivo por canal y por producto "j", indica el grado de contactabilidad del canal.

$Prob_{ij}$: Probabilidad del lead "i" hacía la adquisición del producto "j".

$Rent_{ij}$: Rentabilidad esperada del lead "i" en caso adquiera el producto "j".

$PotCanal_{ij}$: Indica si al lead "i" se le puede ofrecer el producto "j", a través de un determinado canal.

$XCanal_{ij}$: Variable de decisión por lead "i" y por producto "j", toma el valor de 1 o 0 e indica si al lead se le debe ofrecer o no el producto financiero.

La inclusión de otras variables (monto ofertado, tasa ofertada y probabilidad de contacto) en la función objetivo, mejora la performance del modelo.

2.2. Estrategias de marketing

Uno de los mayores desafíos a los que se enfrenta el marketing es intentar implementar estrategias centradas en el cliente y, al mismo tiempo, procesar la información utilizando conocimientos de Big Data. Evidentemente, muchas organizaciones tienen una cantidad excesiva de datos disponibles de diversas fuentes (sistemas internos, a través de la compra de información, interacción con los canales físicos y digitales, entre otras). Sin embargo, las organizaciones que pueden aprovechar estos datos y convertirlos en conocimiento, pueden lograr una mejor respuesta del servicio y, como resultado, pueden crear una ventaja competitiva, este proceso es denominado Gestión de la Relación con el Cliente (CRM, por sus siglas del inglés) [10].

La Estrategia de marketing es la metodología usada para entender las necesidades del cliente, llegar a él y crear oportunidades de venta. Sirve para comunicar y posicionar los productos y servicios de una empresa, y se traduce en líneas operativas que permiten llegar a un mercado meta por los canales de venta adecuados [11].

La alianza de herramientas tecnológicas y de marketing impulsa la toma de decisiones, mejora la productividad y la rentabilidad de las campañas comerciales.

Según las necesidades del entorno y de acuerdo con las características de las organizaciones financieras competidoras, se pueden desarrollar las siguientes estrategias:

- Estrategia basada en costes: Se busca minimizar los costos, maximizando la productividad y la efectividad de los funcionarios de venta, se debe también optimizar los resultados de las campañas de marketing.
- Estrategia de diferenciación: Busca mejorar la imagen de la marca ofreciendo productos y servicios de calidad.
- Estrategia de segmentación: Se trata de adecuar los productos financieros a cada segmento; se entiende por segmento a un conjunto de personas con características comunes. Esta estrategia permite implementar una campaña comercial centrada en las necesidades del mercado; es decir, un marketing visión cliente.

Al utilizar el modelo NBO en la construcción de estrategias comerciales, postulo la utilización de las estrategias de segmentación para optimizar el resultado de las campañas de marketing.

2.3. Modelos Predictivos

Son modelos estadísticos que se utilizan para predecir la ocurrencia de un fenómeno. En esta investigación, predicen la probabilidad de toma de un producto financiero y la rentabilidad esperada si el cliente adquiere el producto. Los algoritmos utilizados corresponden a modelos de regresión y de clasificación.

2.3.1. Regresión Lineal Simple

El modelo general de la regresión lineal es el siguiente: [12]

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

Y: Variable Dependiente

$X_1, X_2, \ldots, X_k$: Variables independientes o predictoras

$\beta_1, \beta_2, \ldots, \beta_k$: Parámetros del modelo

ε : Variable aleatoria, mide el error del modelo

Regresión lineal simple:

El modelo matemático que lo define es:

$$Y = a + bX + e$$

Y: Variable dependiente

X: Variable independiente

e: Variable aleatorio de error.

Esta variable aleatoria posee considerandos que constituyen los supuestos del modelo de regresión lineal simple, estos son: [12]

- Independencia: Los errores "e" son variables aleatorias estadísticamente independiente.
- Linealidad**:** Supone que la media de la distribución de probabilidades de "e" es cero en cada X_i.
- Igualdad de Varianzas**:** Supone que la variancia de la distribución de probabilidades de "e_i" es α^2 (variancia de la regresión).
- Normalidad**:** Supone que la distribución de probabilidades de "e_i" es normal.

Para obtener la estimación del modelo de regresión lineal simple ($\hat{Y} = a + bX$); aplicaremos el método de mínimos cuadrados a los datos de una muestra aleatoria.

Método de Mínimos Cuadrados:

En la ecuación: $Y_1 = a + b\,X_i + \ e_i$:

b: es la tangente de la recta de regresión.

a: es la intersección de la recta de regresión con el eje "Y".

En la Figura 2 se visibiliza cuatro puntos de la muestra representados en el plano (X, Y); además, se muestra la gráfica de la recta de regresión lineal y = a + bx.

$(\overline{x_k}, y_1)$, punto de la muestra representado en el plano (X, Y)

$(\overline{x_k}, \overline{y_k})$, pertenece a la recta de regresión lineal

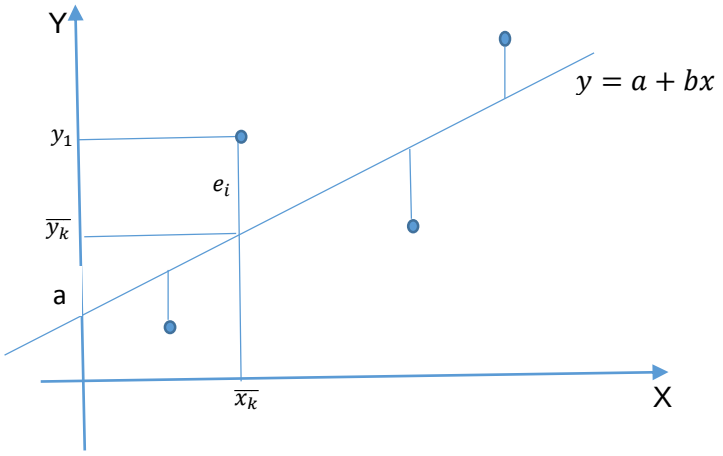


Figura 2: Método de mínimos cuadrados
Fuente: Elaboración Propia

Donde $e_k = y_1 - \ \overline{y_k}$, denominado error o residuo muestral, describe el error en el ajuste del modelo de regresión en el punto $(\overline{x_k}, \overline{y_k})$ de la recta de regresión. Los errores muestrales satisfacen la condición $\sum_{k=1}^{n} e_k = 0$.

La recta de regresión de mínimos cuadrados de Y en X es aquella que hace mínima la suma de los cuadrados de los errores (ESS, por sus siglas del inglés). [12]

$$ESS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \ \overline{y_l})^2 = \sum_{i=1}^{n} (y_i - a - \ bx_i)^2$$

La recta de regresión es aquella que minimiza la expresión anterior. Para esto se utiliza el teorema de Gas-Markov, "a" y "b" se obtienen resolviendo el siguiente sistema de ecuaciones:

$$an + b\sum x = \ \sum y \ \ldots\ldots\ldots (1)$$

$$a\sum x + b\sum x^2 = \ \sum xy \ \ldots\ldots\ldots (2)$$

Las ecuaciones (1) y (2) se obtienen al igualar a cero las derivadas parciales de la ESS con respecto a "a" y con respecto a "b"; resolviendo este sistema de ecuaciones para "b":

$$b = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} = \frac{\sum x_i y_i - n(\bar{x})(\bar{y})}{\sum x_i^2 - n\bar{x}^2}$$

Para calcular el valor de **"a",** se divide la ecuación (1) entre "n":

$$a = \bar{y} - b\,\bar{x}$$

Regresión lineal múltiple:

El modelo estadístico que define a la regresión lineal múltiple es:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_k X_k + \varepsilon$$

Y: Variable dependiente

$X_1, X_2, \ldots, X_k$ $(k \geq 2)$: Variables independientes

$\beta_0, \beta_1, \ldots, \beta_k$: Parámetros desconocidos.

ε : Variable aleatoria, define el término error**.**

La ecuación de regresión muestral es:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + \cdots + b_k X_k$$

Los coeficientes de la regresión muestral $b_0, b_1, b_2, \ldots, b_k$ se calculan, aplicando el método de mínimos cuadrados a los datos de una muestra aleatoria de tamaño "n", cuyos valores observados denotamos por: $(x_{1i}, x_{2i}, x_{3i}, \ldots, x_{ki}, y_i)$, i = 1, 2, 3 ,…, n y n > k, donde, y_i es la respuesta observada (valor de la variable dependiente Y) para los valores $x_{1i}, x_{2i}, x_{3i}, \ldots, x_{ki}$ de las "k" variables independientes respectivas $X_1, X_2, X_3, \ldots, X_k$.

Para cada i = 1, , 3, …, n los datos de la muestra satisfacen la ecuación:

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{3i} + \cdots + b_k x_{ki} + e_i$$

$e_i = y_i - \hat{y}_i$, es el error de la regresión.

$b_0, b_1, b_2, \ldots, b_k$ se calcula utilizando el método de mínimos cuadrados, que consiste en minimizar el valor de la métrica ESS mediante el teorema de Gauss – Markow.

$$ESS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_{1i} - b_2 x_{2i} - \cdots - b_k x_{ki})^2$$

Para construir el sistema de "k+1" ecuaciones lineales (ecuaciones de Gauss-Markow), que se necesitan para calcular los coeficientes $b_0, b_1, \dots, b_k$; se usan derivadas parciales aplicadas a la ecuación que define ESS**.**

$$\frac{\partial}{\partial b}(ESS) = 0$$

$$nb_0 + b_1 \sum x_1 + b_2 \sum x_2 + \dots + b_k \sum x_k = \sum y$$

$$b_0 \sum x_1 + b_1 \sum x_1^2 + b_2 \sum x_1 x_2 + \cdots + b_k \sum x_1 x_k = \sum x_1 y$$

$$\dots..$$

$$b_0 \sum x_k + b_1 \sum x_k x_1 + b_2 \sum x_k x_2 + \cdots + b_k \sum x_k^2 = \sum x_k y$$

Donde, $\sum x_j = \sum_{i=1}^{n} x_{ji}$, $\sum x_j y = \sum_{i=1}^{n} x_{ji} y_i$ para j = 1 , 2, … , k. [12]

Este sistema de ecuaciones se puede resolver utilizando el cálculo matricial: [12]

$$(X'X)b = X'Y \leftrightarrow b = (X'X)^{-1}X'Y$$

$$X'X = \begin{pmatrix} n & \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{2i} & \dots & \sum_{i=1}^{n} x_{ki} \\ \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{1i}^2 & \sum_{i=1}^{n} x_{1i}x_{2i} & \dots & \sum_{i=1}^{n} x_{1i}x_{ki} \\ & & \dots.. & & \\ \sum_{i=1}^{n} x_{ki} & \sum_{i=1}^{n} x_{ki}x_{1i} & \sum_{i=1}^{n} x_{ki}x_{2i} & \dots & \sum_{i=1}^{n} x_{ki}^2 \end{pmatrix} \quad X'Y = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_{1i}y_i \\ \dots.. \\ \sum_{i=1}^{n} x_{ki}y_i \end{pmatrix}$$

2.3.2. Regresión Logística

La regresión logística es una variante del modelo de regresión lineal. Se emplea para predecir el comportamiento de una variable categórica (variable dependiente Y), en función de otras variables independientes o predictoras. Este modelo usa como función de enlace, la función lógit.

Para iniciar el estudio del modelo, se supone que para cada caso determinado por el vector de variables independientes: $X^t = (X_1, X_2, \dots, X_p)$, la variable dependiente "Y" tiene los valores 1 si el evento a predecir se concretiza y 0 en caso contrario, con probabilidades "p" y "1 – p", respectivamente. De este modo, se utiliza el modelo de regresión logística binaria para calcular la probabilidad ("p") de ocurrencia del evento (adquiere el producto financiero):

$$p = f(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_q X_q)}}$$

La expresión matemática de la función logística simple es la siguiente f(x) = 1/(1+e^{-x}). Donde "x" es una variable cuyos valores pertenecen al conjunto de los números reales y van desde -∞ a +∞, "f(x)" es la probabilidad que va desde 0 a 1. En la Figura 3 se muestra la representación gráfica de la función logística simple. Al evaluar la función para "x ≤ -6" se obtiene valores cada vez más pequeños, lo que nos lleva a concluir que la función se convierte en asintótica en el eje y = 0. Para "x ≥ 6" los valores de f(x) tienden a convertir a la función en asintótica en el y = 1.

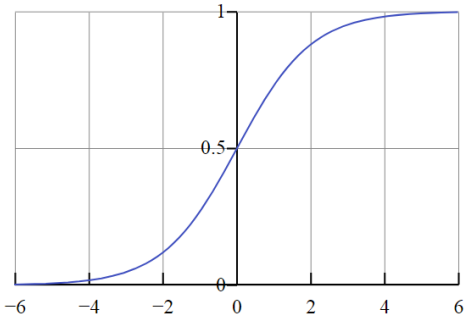


Figura 3: Gráfica de la función logística
Fuente: Elaboración Propia

La probabilidad de adquisición, utilizando el modelo de regresión logística, de un producto financiero se calcula de la siguiente forma:

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k)}}$$

$X_1, X_2, \ldots, X_k$ son variables independientes que definen el perfil de la persona, las cuales pueden ser variables categóricas (genero, nivel socio económico, grado de instrucción, segmento comercial, etc.) o variables continuas (ingreso salarial, deuda total en el sistema financiero, máxima línea de sus tarjetas de crédito, etc.).

Los parámetros desconocidos $\beta_1, \beta_2, \ldots, \beta_k$ son comúnmente estimadas usando el método de máxima verosimilitud, método habitual para ajustar un modelo. Además, el signo de los parámetros indica qué relación existe entre la variable independiente y el evento a predecir.

La regresión logística, con una variable explicativa, puede usarse para calcular la correlación entre la probabilidad de una variable dicotómica ("0" o "1"), con una variable escalar "x". La idea es que la regresión logística aproxime la probabilidad de obtener "0" (si no ocurre cierto suceso) o "1" (cuando ocurre el suceso) con el valor de la variable regresora x.

2.3.3. Árbol de Decisiones

Es un modelo de predicción que a partir de un conjunto de datos permite fabricar diagramas de construcciones lógicas, que sirven para representar y categorizar una serie de condiciones recurrentes, para la resolución de un problema.

En un conjunto de datos se realiza una partición recursiva para lograr subgrupos o patrones llamados nodos, que al ser representados gráficamente dan la idea de un sistema de árboles, que se utilizan en la toma de decisiones. Se usan en:

- En finanzas, se usa para identificar grupo de clientes propensos a entrar en mora y evitar el riesgo crediticio.
- En marketing financiero, identificar perfiles de clientes homogéneos con la finalidad de diseñar campañas tácticas de marketing.
- En control de calidad, para determinar los factores que influyen en la calidad de un producto.
- En recursos humanos, para construir procesos de captación de talentos.

Elementos de un Árbol de Decisión

En la Figura 4 se muestra la representación gráfica de un árbol de decisiones, aplicado a un caso práctico. El árbol de decisiones está conformado por nodos de decisión, nodos de probabilidad, ramificaciones y nodos terminales.

- Un nodo de decisión indica una decisión que se va a tomar.
- Un nodo de probabilidad muestra múltiples resultados inciertos.
- Las ramificaciones indica un posible resultado o acción.
- Un nodo terminal indica un resultado definitivo.

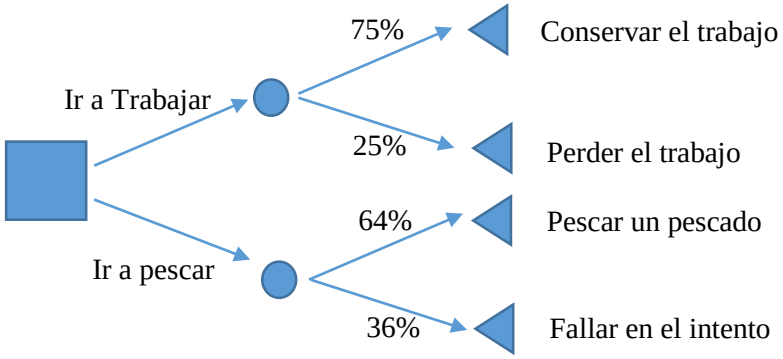


Figura 4: Gráfica de un árbol de decisiones
Fuente: Elaboración Propia

Árboles de Clasificación

Para formar un árbol de clasificación, se considera inicialmente un nodo padre o nodo raíz, formado por todas las instancias del universo de análisis, donde se encuentra definidas las variable predictoras $X_1, X_2, \dots, X_k$ y la variable categórica "Y".

Posteriormente, se selecciona una de las variables X_i para dividir el nodo raíz en dos subconjuntos distintos (nodos hijos) con la finalidad de encontrar grupos altamente homogéneos. Este proceso se repite recursivamente hasta llegar a los nodos terminales o hojas del árbol. Durante este proceso es necesario definir tres tareas:

- Para cada variable seleccionada "X_i" se define el valor del corte "s" a considerar para continuar con la partición del nodo.
- El establecimiento del criterio que se debe tomar en cuenta para considerar que un nodo es una hoja y para el proceso de partición.
- La asignación de la categoría "Y" que se debe fijar como predicción si el nodo resultante es un nodo terminal.

El índice Gini (IGini) sirve para calcular el grado de homogeneidad de los nodos:

$$IGini(q) = 1 - \sum_{i=1}^{h} r_i^2$$

Donde "h" es la cantidad de categorías presenten en la variable "Y" y r_i son las proporciones respectivas de cada categoría. Si la reducción del grado de impureza es significativa se realiza la partición de un nodo padre en dos nodos hijos, la reducción al pasar de un nodo padre q_i a los nodos hijos $q_{i+1}\ y\ q_{i+2}$, se mide:

$$\Delta(q_{i+1}, q_{i+2}) = IGini\ en\ el\ nodo \text{ padre } q_i -$$

$$[(IGIni\ en\ el\ nodo\ q_{i+1})\left({}^{Q_{i+1}}/_{Q_i}\right) + (IGIni\ en\ el\ nodo\ q_{i+2})\left({}^{Q_{i+2}}/_{Q_i}\right)]$$

Donde Q_i es el número de elementos en el nodo q_i.

2.3.4. Algoritmo de aumento de gradiente extremo

Es un algoritmo matemático, basado en árbol de decisiones, que tiene por finalidad menguar la incertidumbre a partir de la disminución de la gradiente de la curva de predicción. Se emplea cuando se tratan datos tabulares/estructurados medianos o pequeños, en tanto, proveen la mejor solución. Un grupo de científicos ha contribuido en el perfeccionamiento de este algoritmo, lo que ha permitido su utilización en

diferentes sectores de la industria. Este algoritmo es ampliamente utilizado en la industria financiera, por su capacidad de generalización para identificar a las personas propensas a adquirir un producto financiero, caer en mora o cometer un fraude crediticio [13]. Su mayor ventaja sobre los algoritmos de regresión tradicionales, es identificar relaciones lineales y no lineales entre las variables dependientes e independientes.

Características principales de este algoritmo:

- Penalización inteligente de árboles.
- Una contracción proporcional de los nodos de las hojas.
- Parámetro de aleatorización adicional.
- Selección automática de funciones.

Ventajas del algoritmo aumento de gradiente extremo:

- Puede utilizarse en una amplia gama de aplicaciones para resolver problemas de predicción, clasificación y regresión.
- Funciona sin problemas en diferentes sistemas operativos como Linux, OSX y Windows.
- Soporta la mayoría de lenguajes de programación como R, Python, Java, C++, Matlab, Julia y Scala.
- Permite la programación paralela sobre diversos ecosistemas de la nube como Spark y Flink.

2.3.5. Máquina de Soporte Vectorial

El modelo Máquina de Soporte Vectorial (SVM, por sus siglas del inglés) es un conjunto de algoritmos de aprendizaje supervisado que están relacionados con problemas de clasificación y regresión. A partir de una muestra cuyos elementos están etiquetados con clases predefinidas, podemos entrenar un modelo SVM que prediga a que clase pertenecen los elementos de una nueva muestra.

El modelo SVM identifica múltiples híper planos que permitan separar las clases en dos o más conjuntos, donde cada conjunto corresponde a alguna de las clases de la muestra. Este modelo es usado para clasificar elementos que pertenezcan a una nueva muestra, en función de los conjuntos a los que pertenezcan. [14]

Características principales del modelo SVM:

- Clasificación óptima: Encontrar el hiperplano que maximice el margen de separación entre las clases.
- Regularización: Generalización para la mayor cantidad de puntos de la muestra, dejando de lado unos pocos puntos incorrectamente clasificados
- Incluir una nueva dimensión ficticia (kernel) donde se pueda encontrar un híper plano que separe las clases.

Ventajas del modelo SVM:

- Eficaz en espacios de grandes dimensiones.
- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamados vectores de soporte), lo que lo hace eficiente en memoria.
- Versátil: se pueden especificar diferentes funciones del núcleo (kernels comunes y personalizados) para la función de decisión que permita optimizar el método de clasificación entre clases.

2.3.6. Análisis de Componentes Principales

Cuando a partir de un conjunto de variables correlacionadas, obtenidas del análisis de la información, se busca construir otro conjunto de componentes no correlacionadas (número limitado de variables), que capturen la mayor variabilidad de la información original. Se utiliza el concepto de análisis de componentes principales (PCA, por sus siglas del inglés). Los componentes se ordenan por la cantidad de la varianza original que describen.

Los componentes obtenidos están ordenados según la variabilidad que capturan de la información original. Por ejemplo, el primer componente principal captura la mayor varianza posible de los datos, el segundo componente captura la mayor variabilidad que no se pudo extraer con el primer componente y así sucesivamente. Estos componentes principales son combinaciones lineales de las variables originales.

Un resultado útil es el coeficiente de correlación múltiple entre cada variable observada (Xi) y todas las componentes principales. Su valor es 1, dado que toda variable Xi puede expresarse de modo exacto como combinación lineal de las componentes.

En síntesis, el objetivo primordial del PCA es resumir la información original, creando nuevas variables (componentes principales), tales que unas pocas sean capaces de reflejar casi toda la información registrada en los datos originales. [15]

Los métodos usados para conseguir las combinaciones lineales de las variables originales, son los siguientes:

- Método basado en la matriz de correlación; cuando los datos no son dimensionalmente homogéneos.
- Método basado en la matriz de covariancias, cuando los datos presentan una distribución homogénea.

Método Basado en la Matriz de Correlación

Consideremos el valor de cada una de las k variables aleatorias V_j. Para cada uno de las "n" individuos. El conjunto de datos en forma de Matriz:

$$(V_j^{\alpha})_{j=1,2,\ldots,n}^{\alpha=1,2,\ldots,n}$$

Obsérvese que cada conjunto de datos se puede expresar como:

$$M_j = \{V_j^{\alpha} \mid \alpha = 1,2,\ldots,n\}$$

Puede considerarse una muestra aleatoria para la variable V_j. A partir de los $k\ x\ n$ datos correspondientes a las "k" variables aleatorias, puede construirse la matriz de correlación muestral, que viene definida por:

$$R = [r_{ij}] \in M_{kxn} \text{ , donde } r_{ij} = \frac{cov(V_i,V_j)}{\sqrt{var(V_i)(V_j)}}$$

Puesto que la matriz de correlaciones es simétrica entonces resulta diagonalizable y sus valores propios λ_i, cumplen lo siguiente: $\sum_{i=1}^{k} \lambda_i = k$.

Los "k" valores propios reciben el nombre de pesos de cada uno de los "k" componentes principales. Cada una de las variables puede ser expresada como combinación lineal de los vectores propios o componentes principales.

Método Basado en la Matriz de Covariancias

El objetivo es transformar un conjunto "*X*" de datos de dimensión "$p\ x\ q$" a otro conjunto de datos "*Y*" de menor dimensión *"$p\ x\ l$",* con la menor perdida de información, utilizando para ello la matriz de covarianza.

Se parte de un conjunto *"p"* de muestras cada una de las cuales tiene "q" variables que las describen, el objetivo es que cada una de esas muestras sean descritas solo por "*I"* variables, donde $l < q$. Además, el número de componentes principales *"l"* tiene que ser inferior a la menor de las dimensiones de "*X*".

2.4. Redes Neuronales

La red neuronal artificial (ANN, por sus siglas del inglés) es un modelo analítico descrito con relaciones matemáticas complejas entre las variables que lo forman. Son usadas en diferentes aplicaciones relacionada con grandes volúmenes de datos que son analizados con modelos no lineales. Se utilizan en finanzas con fines predictivos. Este modelo imita la capacidad de aprender del cerebro humano, se representa usando nodos interconectados que simulan a las neuronas y dendritas del cerebro [16]. Como se muestra en la Figura 5 los nodos forman capas en niveles, en el primer nivel se encuentra la capa de entrada, en el nivel intermedio se ubican las capas ocultas y en el tercer nivel está la capa de salida. Las interrelaciones entre los nodos representan los pesos de la red y son calculados mediante el algoritmo de rentro propagación.

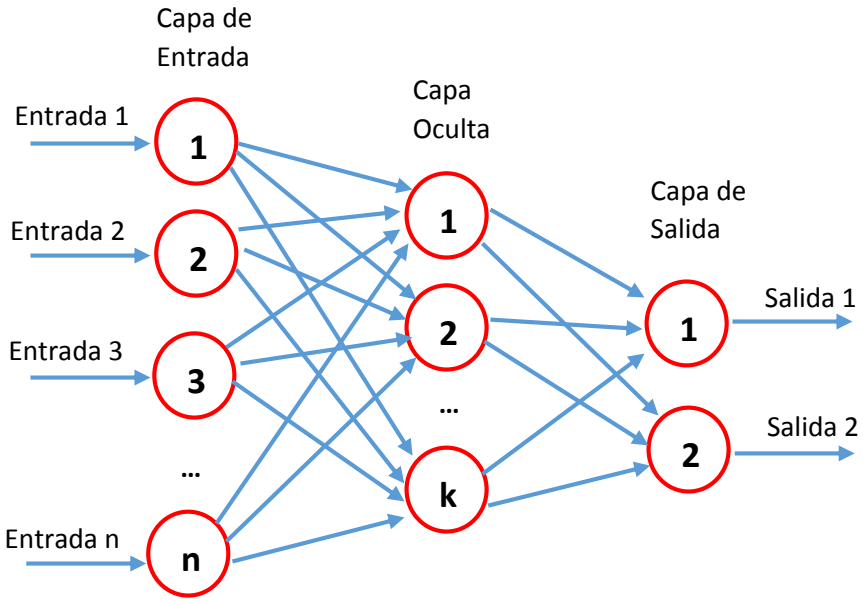


Figura 5: Estructura de una Red Neuronal
Fuente: Elaboración propia

Cada nodo de la ANN es una neurona conectada entre sí para transmitir señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida; los enlaces que interconectan a las neuronas poseen pesos, los que pueden incrementar o inhibir el estado de activación de las neuronas adyacentes.

2.4.1. Arquitectura de los sistemas neuronales artificiales

Según la cantidad de capas que contenga la red neuronal, se clasifica en:

Redes neuronales monocapa. - Es la red neuronal más sencilla tiene una capa de neuronas de entrada y una capa de neuronas de salida.

Redes neuronales multicapa. - Además de las capas de entrada y de salida, la ANN posee un conjunto de capas intermedias denominadas capas ocultas, donde la función de activación propia de cada neurona transforma los datos de entrada a valores cercanos a los que se busca predecir.

Según al tipo de conexiones que posee, se clasifica en las siguientes:

Redes neuronales no recurrentes. - En esta red, la propagación de las señales se produce en un sólo sentido, no existe retroalimentación y funcionan sin memoria.

Redes neuronales recurrentes. - Poseen lazos de retroalimentación, que pueden ser entre neuronas de diferentes capas, neuronas de la misma capa o entre una misma neurona. Esta estructura estudia principalmente la dinámica de sistemas no lineales.

Según a su grado de conexión, se clasifica en:

Redes neuronales totalmente conectadas. - En este caso todas las neuronas de una capa se encuentran conectadas con las de la capa siguiente (redes no recurrentes) o con las de la anterior (redes recurrentes).

Redes parcialmente conectadas. - No se da la conexión total entre neuronas de diferentes capas. Estas estructuras neuronales se podrían conectar entre sí para dar lugar a estructuras mayores. La conexión se puede llevar a cabo de diferentes formas siendo las más usuales las estructuras en paralelo y jerárquicas.

2.4.2. Algoritmo de retro propagación

El aprendizaje de la red consiste en la estimación de los pesos sinápticos y signos que minimizan el error entre las salidas que brinda el modelo y las respuestas que la realidad provee. El algoritmo de retro propagación es el siguiente:

A. Se define un vector aleatorio W para inicializar el proceso.
B. El vector W, de entrada, se introduce en la red hasta alcanzar la salida.
C. Calcular el error entre la salida de la red y la salida que se desea tener.
D. Redistribuir el error calculado en el punto C, en las capas ocultas de acuerdo a los pesos calculados.
E. Con los errores distribuidos, se actualizan los pesos.
F. Repetir los pasos B, C, D, E**.**

Cada iteración es llamada época y se repite k veces, hasta que el performance del modelo en el conjunto de datos de entrenamiento sea menor al performance en el conjunto de datos de evaluación.

Cuando el problema es de clasificación, con "n" instancias, y la variable dependiente "y" tiene "M" clases, la función error "E" se suele construir considerando la probabilidad de aparición de cada clase y el valor real de la variable dependiente, este error es el que se redistribuye en las neuronas de las capas ocultas:

$$E = -\sum_{i=1}^{n}\sum_{j=1}^{M}(y_{ij}\log(\widehat{y_{ij}}) + (1 - y_{ij})\log(1 - \widehat{y_{ij}}))$$

A esta función se le denomina también entropía cruzada.

2.4.3. Análisis de sensibilidad

En los modelos de regresión, es posible medir en forma aproximada la influencia que puede tener cada variable predictora "X_i" en el resultado de la regresión, debido a que podemos interpretar matemáticamente la relación que existe entre la variable dependiente e independiente; sin embargo, para el caso de una red neuronal no es tan fácil obtener esta interpretación de los resultados, en tanto la red neuronal consiste en el modelamiento de patrones no lineales.

El análisis de sensibilidad es un procedimiento que permite calcular un valor que define la influencia de las variables predictores en el modelo denominado red neuronal. Para calcular este valor definamos una red neuronal en la que intervienen 5 variables independientes (predictoras), éstas son:

- X1: Edad
- X2: Salario.
- X3: Cantidad de productos que el cliente tiene en la organización financiera.
- X4: Monto de la deuda del cliente en la organización.
- X5: Monto del ahorro .

Cada una de estas variables pertenecen a todos los clientes de la organización y tienen un valor mínimo, un valor máximo y un valor promedio. Estos valores, a manera de ejemplo, se muestran en la Tabla 1.

Algoritmo para calcular el valor de la influencia de la variable X1 en la red:

1. Se identifica el valor mínimo y el valor máximo de X1.
2. Se calcula el promedio de las variables X2, X3, X4 y X5.
3. Se introduce a la red neuronal, el valor mínimo de X1 y los valores promedio X2, X3, X4 y X5.
4. La red neuronal con estas cinco entradas tiene un valor de salida Y1.

5. Se introduce a la red neuronal, el valor máximo de X1 y los valores promedio de X2, X3, X4 y X5.
6. La red neuronal con estas cinco entradas tiene un valor de salida Y2.
7. El valor absoluto de (Y2 – Y1) indica el valor de la influencia de la variable X1 en la red neuronal.
8. El mismo procedimiento se sigue para calcular la influencia de X2, X3, X4 y X5 en la red neuronal.

Variable Predictora	Valor Mínimo	Valor Máximo	Valor Promedio
X1	18	65	44
X2	900	50,000	12,000
X3	1	6	3
X4	100	10,000	3,500
X5	300	200,000	50,000

Tabla 1: Valores mínimos, máximos y promedio de cinco variables predictoras
Fuente: Elaboración propia

Gracias al análisis de sensibilidad podemos identificar las variables independientes que tienen mayor influencia en el valor de las probabilidades, propensión a la adquisición, obtenidas por la red neuronal. Se realiza un análisis univariado de cada uno de estas variables para segmentar a los clientes que participan en las campañas comerciales. Esta segmentación permite elaborar estrategias comerciales de acuerdo a las características principales de cada segmento.

Además, el análisis de sensibilidad permite seleccionar las variables independientes principales que intervendrán en el modelo, descartando aquellas que no aportan significativamente en la calidad de la generalización de la red neuronal; evitando así, el sobreajuste del modelo debido a la multidimensionalidad de los datos.

El análisis de sensibilidad también evalúa si el orden de prioridad de las variables principales se mantiene al cierre de cada campaña comercial, validando así la estabilidad de la red neuronal a través del tiempo.[17]

2.5 Modelo ensamblado de predicción

Para mejorar el nivel de predicción de los modelos analíticos, se recurre a los métodos de conjuntos (modelo ensamblado) que utilizan múltiples algoritmos de aprendizaje provenientes de los modelos predictivos que lo conforman. Este modelo requiere mayores recursos computacionales. Esta característica limita su utilización en organizaciones que no cuentan con una sólida infraestructura tecnológica.

Los métodos de ensamble o métodos combinados son procesos mediante los cuales se construyen estratégicamente varios modelos de *Machine Learning* para luego ensamblarlos y utilizarlos para resolver un problema predictivo [18]. Los tipos de modelos ensamblados son los siguientes:

- **Agregación Bootstrap (Bootstrap AGGgregation) o BAGGing:**
  Es un tipo de modelo ensamblado que entrena en paralelo diferentes modelos predictivos, con el resultado (predicciones) de estos modelos se entrena un meta modelo, el cual calculará el valor final de la predicción. Como se muestra en la Figura 6, se extraen varias muestras formadas por un conjunto de diferentes variables, cada variable tiene la misma probabilidad de ser seleccionada. Una vez creadas las muestras, se entrenan los modelos unitarios que conforman el modelo ensamblado, las predicciones de cada modelo se convierten en las variables de entrada de un meta modelo que es utilizado para calcular la predicción final.

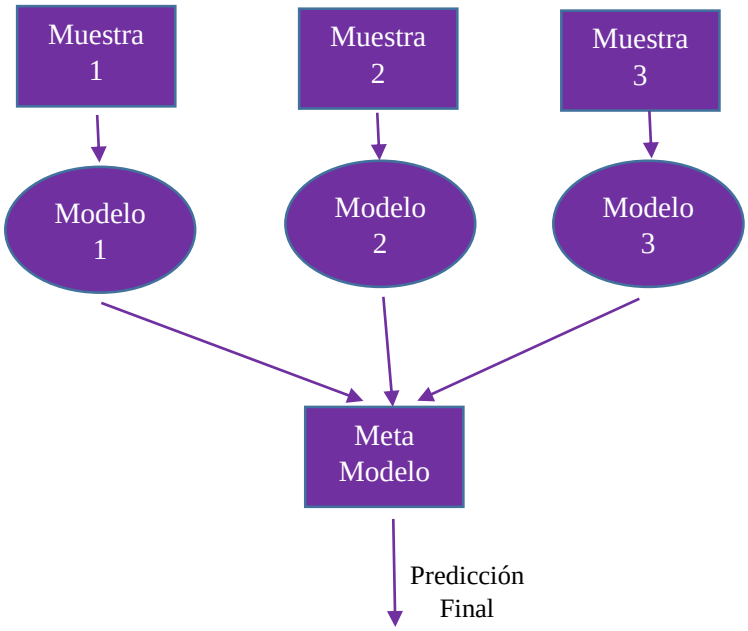


Figura 6: Modelo ensamblado Bagging
Fuente: Elaboración propia

- **Boosting:**

  Es una técnica de aprendizaje secuencial. Este algoritmo se caracteriza por el entrenamiento secuencial de cada modelo unitario que constituye el modelo ensamblado. De esta manera se va disminuyendo el error obtenido en las estimaciones del entrenamiento anterior.

  La Figura 7 muestra el entrenamiento secuencial de diversos modelos; las predicciones individuales de estos son ponderadas con las puntuaciones de generalización obtenidas en la etapa de entrenamiento, y por último estas ponderaciones son combinadas para generar una estimación final.

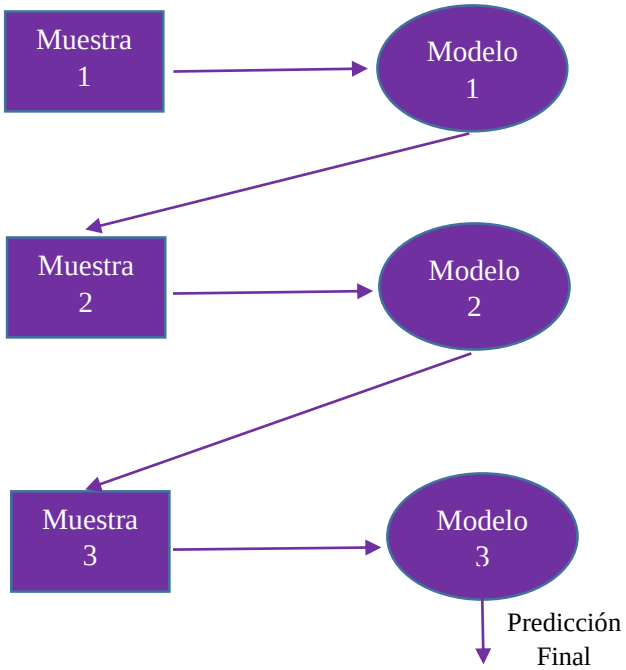


Figura 7: Modelo ensamblado Boosting
Fuente: Elaboración propia

Los métodos de ensamblaje Bagging y Boosting son los más utilizados en la industria financiera para construir conjuntos de modelos predictivos, debido a su alta capacidad de generalización. [18]

Capítulo 3

Desarrollo de los Modelos Analíticos

En este capítulo se discute la construcción de modelos de regresión para predecir el valor de la rentabilidad esperada en cada campaña comercial. También se elaboran modelos de propensión para calcular la probabilidad de adquisición de cada producto. Por último, se estandarizan estas probabilidades mediante dos métodos: Redes Neuronales y Modelos Ensamblados de Predicción.

3.1. Construcción de los modelos de rentabilidad

Generar ingresos económicos es una de las finalidades más importantes de todo proceso comercial; la rentabilidad, entendida como la cantidad de dinero que ingresa a la organización financiera debido a la colocación de un producto, permite un crecimiento sostenido de las empresas. Por tal motivo, es necesario predecir la rentabilidad de las campañas comerciales; para lograrlo se desarrollan modelos analíticos que calculen la rentabilidad esperada por producto financiero. Para desarrollar estos modelos es necesario construir un conjunto de datos con todas las posibles variables independientes y escoger las más importantes usando criterios de Selección de Características.

Luego de seleccionar las variables principales, se entrena diferentes modelos para cada producto financiero y se escoge el que tenga mayor coeficiente de determinación (R^2) y el menor error cuadrático medio (MSE, por sus siglas del inglés). En la Figura 8 se muestra los pasos que se deben seguir para desarrollar un modelo analítico, desde la comprensión del negocio hasta su despliegue. Estos pasos son:

- Consolidación de la información. - Se define todas las variables dependientes e independientes que serán utilizadas en la etapa de entrenamiento del modelo predictivo, luego, se construye un conjunto de datos que contenga a estas variables.
- Pre-procesamiento. – Consiste en procesar el conjunto de datos obtenido en la etapa anterior para eliminar la información redundante e inexacta. Este procesamiento incluye la eliminación de valores atípicos y nulos, incluye también el tratamiento de variables categóricas.

- Selección de variables. - Consiste en identificar las variables más importantes que caracterizan al cliente que adquiere un producto financiero. Este proceso sirve para reducir la dimensionalidad de los datos.
- Entrenamiento del modelo. – Se entrena los modelos para encontrar las mejores relaciones o patrones de comportamiento en la fuente de datos. El entrenamiento se realiza con el 70% del conjunto de datos, se utilizan métricas que midan el grado de generalización de la predicción.
- Evaluación de la calidad.- Se valida la calidad de la generalización del modelo con el 30% restante del conjunto de datos, el valor de la métrica debe ser muy parecida a la obtenida en la etapa de entrenamiento.
- Pase a producción del modelo.- Utilización del modelo predictivo en la creación de estrategias de marketing dentro de las campañas comerciales.
- Evaluación de los resultados.- Consiste en comparar los resultados obtenidos al término de la campaña, con la predicción inicial. Este proceso es constante y determina la estabilidad del modelo en el tiempo.

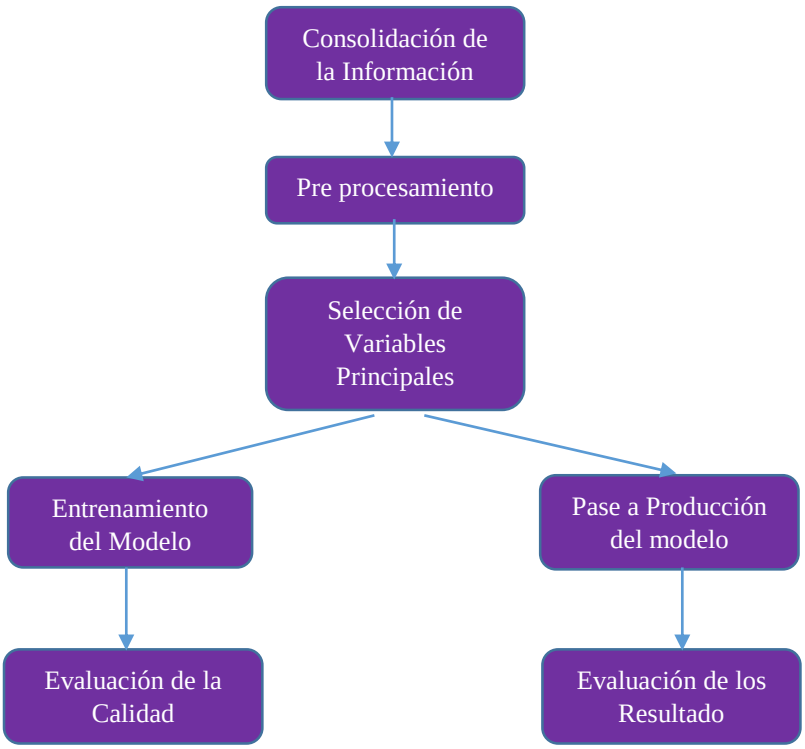


Figura 8: Pipeline utilizado para el desarrollo de los modelos analíticos
Fuente: Elaboración propia

En las siguientes secciones desarrollaremos cada una de las etapas descritas anteriormente, desde la consolidación de la información hasta el desarrollo y testeo de los modelos predictivos.

3.1.1. Consolidación de la información

La información se encuentra descentralizada en diferentes fuentes de datos. Por ello, es necesario consolidar en una sola tabla todas las variables independientes (cuantitativas o cualitativas) que serán evaluadas en la etapa de selección de características, con la finalidad de escoger aquellas que expliquen mejor el evento a predecir. Estas variables provienen de fuentes de datos tales como:

- Sistemas internos de la entidad financiera.
- Encuestas para conocer el nivel de satisfacción del cliente respecto a la atención brindada.
- Feedback de los canales de venta.
- Canales transaccionales (agencias, cajeros, agentes, etc.).
- Canales digitales como el aplicativo móvil y la web del banco.
- Fuente de datos provenientes de empresas externas; por ejemplo, información censal e información de contactabilidad.
- Información proveída por la Súper Intendencia de Banca y Seguros.

Estas fuentes son actualizadas mensualmente con la última información disponible. Las variables provenientes de estas fuentes pueden clasificarse en cinco grandes grupos:

- Variables Demográficas.
- Variables transaccionales que miden la relación del cliente con el banco.
- Variables que perfilan al cliente dentro de la entidad financiera.
- Variables que perfilan al cliente dentro del sistema financiero.
- Variables relacionadas al producto financiero.

Para validar la estabilidad de estas variables independientes se usa la métrica denominada Índice de Estabilidad de la Población (PSI, por sus siglas del inglés). El PSI es una estadística que mide cuanto ha cambiado una variable a lo largo del tiempo; en el sistema financiero se utiliza para cuantificar el cambio entre los datos que participan en el desarrollo del modelo y los datos actuales. Un PSI alto alerta a la organización de un cambio en las características del mercado.

Variables Demográficas:

Son aquellas que nos permiten segmentar la población en grupos más homogéneos, con el objetivo de definir acciones comerciales de acuerdo a cada perfil.

En la Tabla 2 se muestra el conjunto de variables demográficas que serán incluidas en el conjunto de datos. Estas variables son frecuentemente utilizadas en las diversas campañas de marketing y principalmente en la venta de seguros a personas naturales, ya que ayudan a identificar sus necesidades y su potencial según criterios de riesgo.

Variable	Descripción	Dominio
Edad	Edad del cliente.	Valores enteros entre 18 y 110
Genero	Genero del cliente.	Masculino o Femenino
EstadoCivil	Estado civil del cliente.	Sotero, Casado, Viudo y Divorciado
MacroZona	Zona de su ubicación geográfica.	Lima, Callao y Provincia
NivelEstudios	Grado de Instrucción.	Primaria, Secundaria, Técnica y Universitaria
SituacionLaboral	Indica la situación laboral de la persona	Valores: Trabaja y No Trabaja
Flag_Dependiente	Indica si es dependiente o independiente.	Valor binario: 1 o 0
IngresoNeto	Ingreso económico neto mensual	Valores reales entre 800 y 200,000
CantidadHijos	Cantidad de hijos del cliente.	Valores enteros 0 y 12
NSE	Nivel Socio Económico.	Valores entre: A, B, C, D y E
Nacionalidad	Nacionalidad de la Persona	Valores entre: Peruano o Extranjero
Tenencia_Inmueble	Indica si el cliente tiene vivienda propia.	Valor binario: 1 o 0
Tenencia_Vehiculo	Indica si el cliente cuenta con vehículo.	Valor binario: 1 o 0
Tipo de Vehiculo	Gama del vehículo en caso lo tenga.	Gama del vehículo: Baja, Media y Alta

Tabla 2: Variables Demográficas utilizadas para perfilar la Población
Fuente: Elaboración propia

Variables Transaccionales:

Son aquellas que cuantifican el nivel de interacción de la persona con los puntos de contacto. La Tabla 3 muestra la lista de variables transaccionales, muestra también la descripción de ellas y el conjunto de valores (dominio) que pueden tomar.

Variable	Descripción	Dominio
NroTrxAgencia	Cantidad de transacciones en Agencia	Valores enteros entre 0 y 100
NroTrxATM	Cantidad de transacciones hechas en el cajero automático	Valores enteros entre 0 y 100
NroTrxCorresponsal	Cantidad de transacciones hechas en agentes del banco	Valores enteros entre 0 y 100
NroTrxApp	Cantidad de transacciones hechas en el aplicativo móvil	Valores enteros entre 0 y 100
NroTrxWeb	Cantidad de transacciones hechas a través de la página web	Valores enteros entre 0 y 100
ImporteTrxAgencia	Importe monetario de las transacciones en Agencia	Valores enteros entre 0 y 100,000
ImporteTrxATM	Importe monetario de las transacciones hechas en cajero	Valores enteros entre 0 y 100,000
ImporteTrxCorresponsal	Importe monetario de las transacciones hechas en Agente	Valores enteros entre 0 y 100,000
ImporteTrxApp	Importe monetario de las transacciones a través del App	Valores enteros entre 0 y 100,000
ImporteTrxWeb	Importe monetario de las transacciones a través de la Web.	Valores enteros entre 0 y 100,000

Tabla 3: Variables Transaccionales
Fuente: Elaboración propia

Variables Internas del Banco:

Cuantifican el valor del cliente para el banco, los principales se muestran en la Tabla 4:

Variable	Descripción	Dominio
CantProd	Cantidad de Productos que tiene el cliente en el banco	Valores enteros entre 1 y 20
SegmentoComercial	Segmento comercial del cliente	Beyond, Premium, Preferente, Personal y Estándar
SaldoColocacion	Importe total de deuda del cliente con el banco	Valores enteros entre 0 y 1,000,000
SaldoTC	Importe total de Deuda con su Tarjeta de Crédito en el Banco	Valores enteros entre 0 y 50,000
SaldoPP	Importe total de Deuda en Préstamos Personales en el Banco	Valores enteros entre 0 y 150,000
SaldoVEH	Importe total de Deuda en Préstamo Vehicular en el Banco	Valores enteros entre 0 y 150,000
SaldoHip	Importe total de Deuda en Préstamo Hipotecario en el Banco	Valores enteros entre 0 y 800,000

Tabla 4: Variables Internas del Banco
Fuente: Elaboración propia

Variables del Sistema Financiero:

Son variables que explican la situación del cliente dentro del sistema financiero, estas se muestran en la Tabla 5 y se deben considerar en el conjunto de datos:

Variable	Descripción	Dominio
ClasificacionSBS	Clasificación de riesgo que otorga la SBS al cliente	Normal, CPP, Deficiente, Dudoso y Perdida
NumEnt	Cantidad de Entidades financieras donde el cliente tiene saldo de deuda	Valores entre 1 y 5
Max_LineaTC_SSFF	Máxima Línea de Tarjeta de Crédito del cliente en el sistema financiero	Valores enteros entre 0 y 100,000
Antigüedad_SSFF	Antigüedad del cliente en el sistema financiero	Valores entre 0 y 120
Entidad_Principal	Nombre de la entidad donde tiene su mayor saldo de Deuda	BCP, BBVA, IBK, SBP, etc.
SaldoTotal_SSFF	Importe total de deuda que tiene el cliente en el sistema financiero	Valores enteros entre 0 y 1,000,000
SaldoTC_SSFF	Importe total de Deuda con su Tarjeta de Crédito en el sistema financiero	Valores enteros entre 0 y 50,000
SaldoPP_SSFF	Importe total de Deuda en Préstamos Personales en el sistema financiero	Valores enteros entre 0 y 150,000
SaldoVEH_SSFF	Importe total de Deuda en Préstamo Vehicular en el sistema financiero	Valores enteros entre 0 y 150,000
SaldoHIP_SSFF	Importe total de Deuda en Préstamo Hipotecario en el sistema financiero	Valores enteros entre 0 y 150,000

Tabla 5: Variables del Sistema Financiero
Fuente: Elaboración propia

Variables del Producto Financiero:

Son las variables que caracterizan las aperturas de todos los productos financieros que han tenido lugar en los últimos doce meses. Estos se muestran en la Tabla 6:

Variable	Descripción	Dominio
MontoApertura	Monto de Apertura del producto financiero.	Valores entre 0 y 150,000
TasaInteres	Tasa de interés del producto financiero.	Valores entre 11.00 y 45.00.
MontoCuota	Cuota del préstamo personal	Valores entre 1 y 6
Plazo	Cantidad de cuotas por pagar	Valores entre 1 y 10
LineaTC	Línea de la tarjeta de crédito en moneda nacional	Valores entre 0, 1, 2 o 3

Tabla 6: Variables del Producto Financiero
Fuente: Elaboración propia

Arquitectura de la Base de Datos:

Para desarrollar los modelos analíticos debemos consolidar la información en una tabla que consolide todas las variables independientes y las variables dependientes que se desean predecir, el modelamiento estructural y organizado de las variables permite que éstas sean fácilmente utilizadas. Para este fin, en la Figura 9 se muestra un modelo relacional de datos que nos ayudará a identificar los atributos de cada entidad (variables) y las relaciones que existen entre ellas.

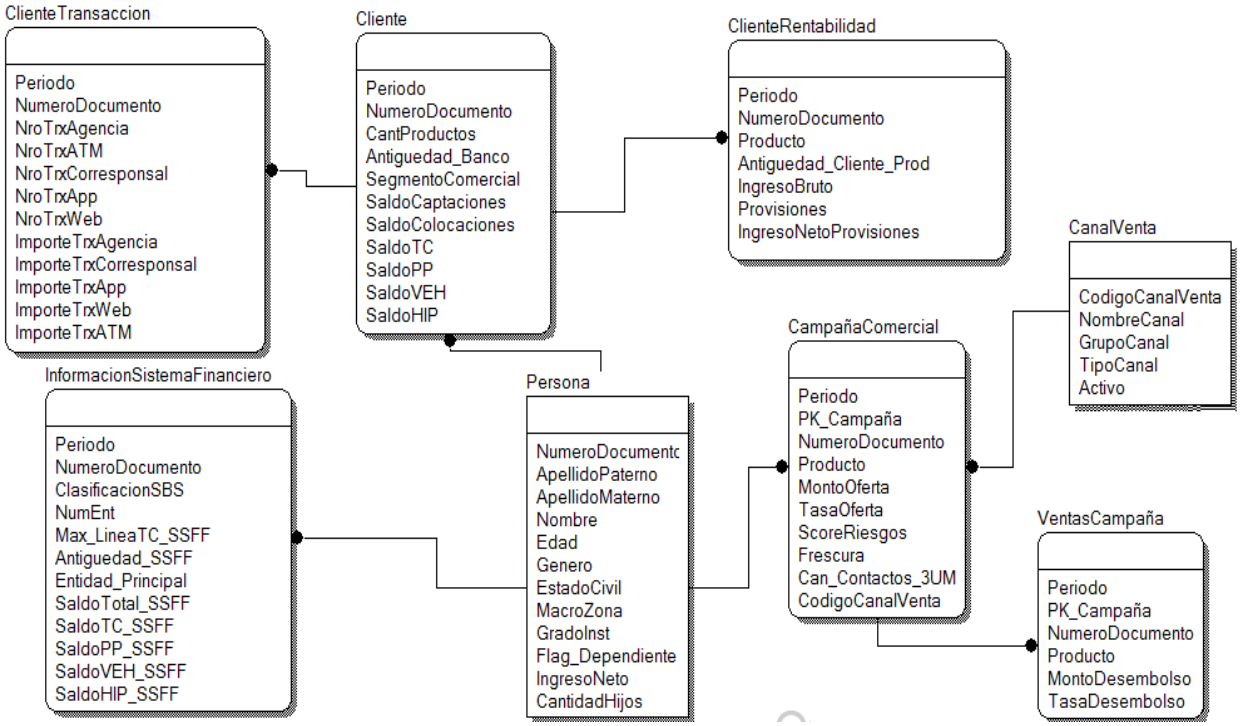


Figura 9: Modelo de Datos Relacional de las campañas de marketing
Fuente: Elaboración propia

En el modelo relacional de datos se muestra las principales entidades y las relaciones que existen entre ellas. Estas entidades son las que participan en los procesos de campaña; la entidad *Persona* representa a los clientes y no clientes de la entidad financiera. Para los clientes se tiene la información de su rentabilidad y de sus transacciones, y para los no clientes contamos con la información de su situación en el sistema financiero. Los canales de venta están representados en la entidad *CanalVenta* y las colocaciones del producto se identifican con la entidad *VentasCampaña*.

La información se almacena en un Data Lake utilizando herramientas de Big Data como PySpark. Esta información se actualiza mensualmente a través de procesos de extracción, transformación y carga de datos (ETL, por sus siglas del inglés).

En la Figura 10 se muestra todos los componentes del proceso de carga y análisis de la información de los diferentes eventos operativos e informativos dentro de un entorno de Big Data. Las fuentes de datos pueden ser los siguientes: archivos planos, aplicativos internos, páginas web y el aplicativo móvil; los cuales son utilizados para ingestar información en tiempo real.

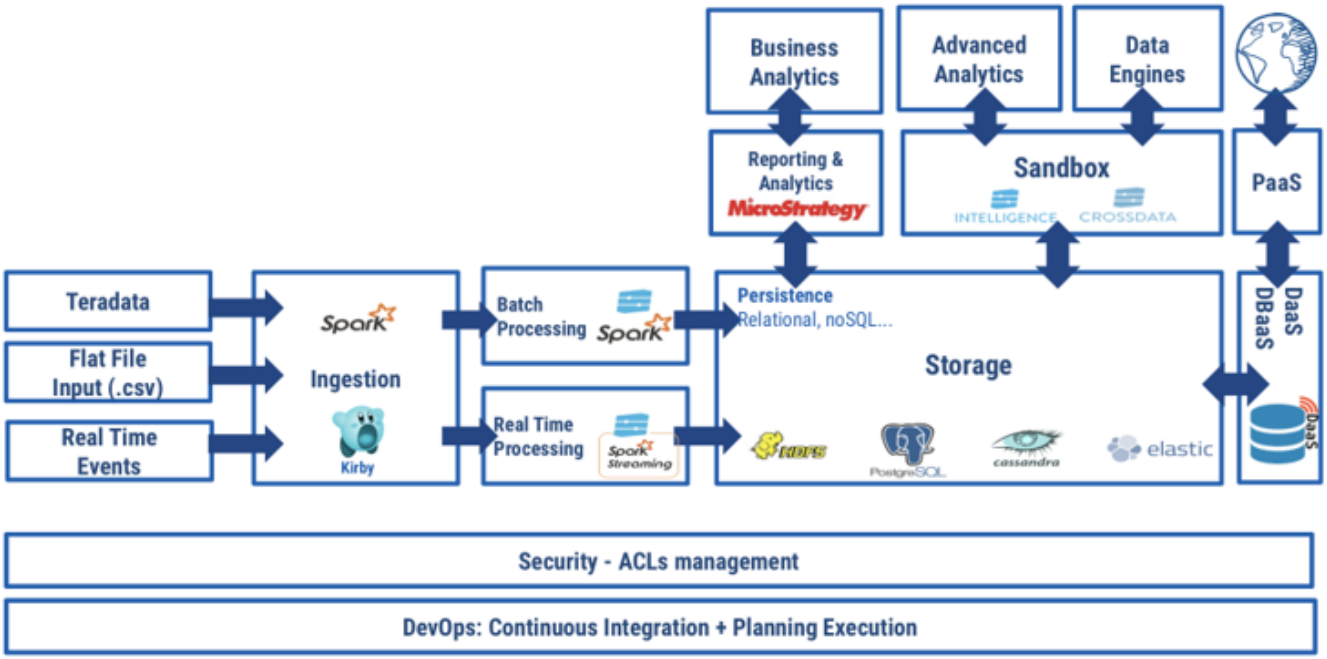


Figura 10: Arquitectura del flujo de datos en un entorno de Big Data
Fuente: Stone, M. D & Woodcock, N. D [18]

Actualmente, se cuenta con un entorno de desarrollo denominado *Sandbox* donde se realiza la consolidación y el análisis de la información con el objetivo de identificar variables que aportan en la predicción de la adquisición de un producto financiero. Además, se utiliza la herramienta *MicroStrategy* para construir tableros de control que nos permite llevar un seguimiento de las colocaciones mensuales y diarias durante las campañas comerciales.

Esta arquitectura de datos permite disponer de una visión integral de la información facilitando una responsabilidad compartida en las decisiones; por otro lado, homogeniza el valor de los indicadores utilizados a nivel global entre todas las áreas de negocio. Sin embargo, no todos los procesos de carga aseguran la integridad del dato. Por ello, en muchos casos encontramos valor nulos o variables con valores extremos que deben ser imputados. Por tanto, es necesario realizar un proceso de pre-procesamiento antes de desarrollar los modelos predictivos (modelos de rentabilidad y de propensión).

3.1.2. Preprocesamiento de la información

La Tabla 7 muestra un listado de las variables independientes (variables predictoras) y de las variables dependientes (rentabilidad por producto) que participan en la construcción de los modelos de rentabilidad para cada producto financiero.

El conjunto de datos conformado por estas variables contiene a todos los leads que han adquirido un producto financiero en los últimos 12 meses y la rentabilidad que han generado, utilizando esta información desarrollamos modelos de regresión que permitan predecir esta rentabilidad para futuras campañas comerciales.

Producto	Demográficas	Transaccionales	Internas	Sistema Financiero	Rentabilidad
MontoApertura	Edad	NumTrxAgencias	CantProd	ClasificacionSBS	Rent_TC
TasaInteres	Genero	NumTrxATM	SegmentoComercial	NumEnt	Rent_XL
MontoCuota	EstadoCivil	NumTrxCorresponsal	SaldoColocacion	Max_LineaTC_SSFF	Rent_CD
Plazo	Macrozona	NumTrxApp	SaldoCaptacion	Antiiguedad_SSFF	Rent_LD
LineaTC	Nacionalidad	NumTrxWeb	SaldoTC	Entidad_Principal	Rent_PA
	NivelEstudios	ImporteTrxAgencias	SaldoPP	SaldoTotal_SSFF	Rent_DXP
	SituacionLaboral	ImporteTrxATM	SaldoVEH	SaldoTC_SSFF	Rent_VEH
	IngresoNeto	ImporteTrxCorresponsal	SaldoHIP	SaldoPP_SSFF	Rent_HIP
	CantidadHijos	ImporteTrxApp		SaldoVEH_SSFF	Rent_FREE
	Tenencia_Inmueble	ImporteTrxWeb		SaldoHIP_SSFF	Rent_POW
	Tenencia_Vehiculo				Rent_SC
	Tipo_Vehiculo				Rent_CS
					Rent_FM
					Rent_DEP
					Rent_SEG_FR
					Rent_SEG_AC
					Rent_SEG_DE

Tabla 7: Variables del conjunto de datos utilizado para el desarrollo de los modelos
Fuente: Elaboración propia

Mediante el pre-procesamiento de los datos eliminamos la información redundante que afecta la capacidad de predicción de los modelos. Se divide en:

- Eliminación de las variables con un alto porcentaje de valores nulos.
- Imputación de los valores nulos.
- Imputación de los valores atípicos.
- Tratamiento de las variables categóricas.

El resultado de esta etapa es la base de datos utilizada para el entrenamiento de los modelos analíticos que calculan la rentabilidad esperada en caso de que el cliente adquiera un producto financiero.

Eliminación de las variables con alto porcentaje de nulos

Los valores nulos o ausentes son valores desconocidos de ciertas variables del conjunto de datos y se originan debido a la no disponibilidad de la información.

Las variables que superen el límite máximo de valores nulos no son consideradas en la etapa de selección de características principales; es decir, las que poseen más del 75% de valores nulos son eliminadas del conjunto de entrenamiento:

$$\%\, Valores\ Nulos\_V_k = \frac{Cantidad\ de\ registros\ con\ valor\ nulo\ para\ la\ variable\ V_k}{Cantidad\ de\ registros\ del\ Data\ Set}$$

El siguiente algoritmo explica los pasos a seguir:

Nrows = Cantidad de filas del Data Set
Cols = Arreglo que contiene el nombre de las columnas del Data Set
Matriz_Cols_PorMissings = ∅
Arreglo_Resultado = ∅
Para var ∈ Cols
  Cantidad_Missing_Values = Cantidad Valores nulos para la Variable "Var"
  Por_Missing_Values = Cantidad_Missing_Values/Nrows
  Ingresar en Matriz_Cols_PorMissings el vector ["Var", "Por_Missing_Values"]
  Si (Por_Missing_Values > 0.75)
    Ingresar en el arreglo Arreglo_Resultado la variable "var"
  Fin Si
Fin Para

En la Tabla 8 se muestran las siete variables independientes que deben ser excluidas del análisis, ya que su porcentaje de valores nulos es mayor al 75%:

Variable	Por Missing Value
Tenencia_Inmueble	99.24%
SaldoVEH	94.67%
SaldoHIP	91.19%
SaldoHIP_SSFF	88.96%
SaldoVEH_SSFF	85.72%
Tenencia_Vehiculo	81.11%
Tipo_Vehiculo	77.52%

Tabla 8: Variables con alto porcentaje de valores nulos
Fuente: Elaboración propia

Imputación de variables con valores nulos

Luego de excluir las variables que superen el umbral del 75%, aún quedan valores nulos que deben ser imputados. Para lograrlo, usaremos el algoritmo K vecinos más cercanos (KNN, por sus siglas del inglés) que consiste en reemplazar estos valores

nulos por el valor más frecuente entre sus vecinos (valores no nulos de las instancias más cercanas). Previamente, se define la cantidad de vecinos para la imputación y la métrica que se usará para calcular la distancia entre las instancias (generalmente se usa la distancia euclidiana y para el presente trabajo se eligió dicha métrica).

El siguiente algoritmo sirve para identificar las variables que aún contienen valores nulos:

```
Var_excluir = Arreglo de variables con un alto porcentaje de valores nulos
Nrows = Cantidad de filas del Data Set
Columns = Arreglo que contiene el nombre de las columnas del Data Set
Matriz_Resultado = ∅
Para var ∈ Columns
    Si var ∉ Var_excluir
        Cantidad_Missing_Values = Cantidad Valores nulos para la Variable "var"
        Tipo_Var = Tipo de la variable "var"
        Si Cantidad_Missing_Values > 0
            Ingresar en la Matriz_Resultado la variable "var" y la tipología "Tipo_Var"
        Fin Si
    Fin Si
Fin Para
```

En la Tabla 9 se muestran las cuatro variables que aún contienen valores nulos y que deben ser imputados (reemplazar el valor nulo) según el tipo de variable:

Variable	Por Missing Value	Tipo Variable
CantidadHijos	9.58%	Númerica
GradoInst	2.19%	Categórica
Edad	2.26%	Númerica
Genero	0.50%	Categórica

Tabla 9: Variables con valores nulos a imputar
Fuente: Elaboración propia

Remplazaremos los valores nulos de las variables categóricas Grado de Instrucción y Genero por el valor "NI" y para las variables numéricas usaremos la clase *KNNImputer* de la librería *Scikit-learn* de Python. Para cada valor nulo se identificará los 10 elementos más cercanos basándonos en la distancia euclidiana y se lo reemplazará por aquel que tenga mayor frecuencia (imputación por la moda).

Imputación de Valores Atípicos

Los valores atípicos (*outliers*, por sus siglas del inglés) son aquellos que están numéricamente más distantes del resto de los datos; existen dos tipos de valores atípicos: leves y extremos. En la Figura 11 se muestran los valores atípicos leves, los

cuales son aquellos que se encuentran 1.5 veces la distancia intercuartílica por encima del tercer cuartil y los valores extremos son aquellos que se encuentran 3 veces por encima del tercer cuartil.

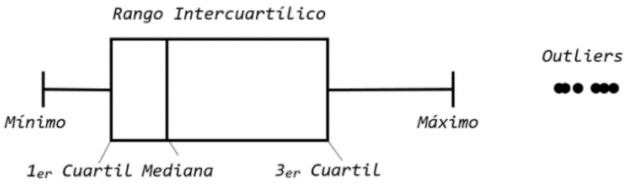


Figura 11: Representación gráfica de los valores atípicos
Fuente: Elaboración propia

El siguiente algoritmo sirve para imputar los valores atípicos extremos:

Var_imputar = Arreglo de variables definidas por el analista
Para var ∈ Var_imputar
Q1 = Primer cuartil de la variable "var"
Q3 = Tercer cuartil de la variable "var"
IQR = Q3 – Q1
Máximo = Q3 + 3*IQR
Valores_Variable = Arreglo de todos los valores de la variable "var"
Para valor ∈ Valores_Variable
Si valor >= Máximo
Actualizar el valor de la variable "var" con el Maxímo
Sino
No modificar el valor de la variable "var"
Fin Si
Fin Para
Actualizar el Data Set con los nuevos valores de la variable "var"
Fin Para

En la Figura 12 se analiza la variable Ingreso Neto, donde se muestra personas con salarios superiores a S/ 30,000 que son valores atípicos que deben ser imputados. Luego de aplicar el algoritmo anterior eliminamos los valores extremos y obtenemos:

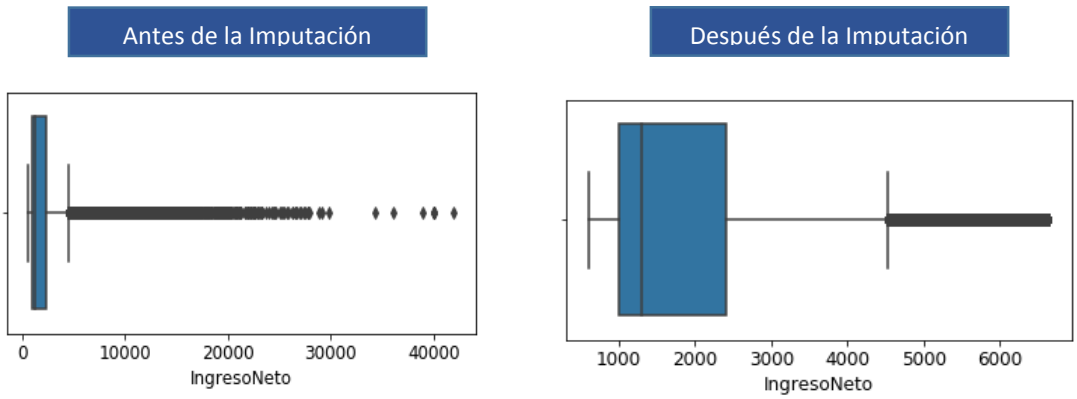
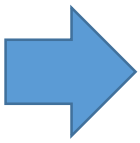


Figura 12: Efecto de la imputación de valores atípicos
Fuente: Elaboración propia

Tratamiento de Variables Categóricas

Las variables categóricas son aquellas que contienen un número finito de categorías o grupos distintos, para que puedan ser usadas en los modelos de Aprendizaje Automático deben ser transformadas a variables numéricas; la manera más sencilla de lograrlo es creando variables *dummy*. Las variables *dummy* son variables binarias creadas a partir de los valores de la variable original; por ejemplo, en la Tabla 10 se muestra la variable categórica Genero, la cual posee dos valores únicos: Masculino y Femenino, de estos dos valores creamos dos variables dicotómicas: Flag_Masculino, Flag_Femenino, las cuales tienen valores 1 o 0 de acuerdo al valor original. Para crear las variables *dummies* usaremos la función *get_dummies* de Python, el cual transforma los datos categóricos a numéricos:

ID Cliente	Género
1	Masculino
2	Femenino
3	Femenino
4	Masculino
5	Masculino
6	Masculino
7	Femenino
8	Femenino
9	Femenino
10	Masculino



ID Cliente	Flag Masculino	Flag Femenino
1	1	0
2	0	1
3	0	1
4	1	0
5	1	0
6	1	0
7	0	1
8	0	1
9	0	1
10	1	0

Tabla 10: Generación de las variables *dummys*
Fuente: Elaboración propia

Luego de eliminar las variables con alto porcentaje de valores nulos, imputar los valores nulos usando el algoritmo KNN, acotar los valores atípicos y tratar las variables categóricas mediante la creación de variables *dummies*, obtenemos 75 variables independientes utilizados en la etapa de entrenamiento de los modelos analíticos. En la siguiente sección, definiremos la variable dependiente de cada modelo de rentabilidad y el tipo de modelo a desarrollar (para variables dependientes que sean dicotómicos se desarrollan modelos de clasificación y para variables dependientes numéricas se desarrollan modelos de regresión). Según esta premisa, debemos usar un modelo de regresión para calcular la rentabilidad esperada y un modelo de clasificación para calcular la probabilidad de adquisición del producto financiero.

3.1.3. Definición de la variable dependiente del modelo de rentabilidad

La variable dependiente de cada modelo de rentabilidad es una variable cuantitativa que representa la cantidad de dinero (expresado en soles) que generará una persona al adquirir un producto financiero. La Tabla 11 muestra el número de ventas para cada uno de los 17 productos financieros a lo largo de 9 meses de campaña y la Tabla 12 muestra la rentabilidad promedio generada luego de 12 meses de colocado el producto en el mercado.

Las Tablas 11 y 12 se interpretan de la siguiente forma: en el primer mes de campaña 4,927 personas han adquirido una tarjeta de crédito y estos han generado en promedio S/118 de rentabilidad al usar su tarjeta en cada uno de los 12 meses posteriores a su apertura; la misma interpretación se aplica para los 17 productos.

Tipo	Producto	MES 1	MES 2	MES 3	MES 4	MES 5	MES 6	MES 7	MES 8	MES 9
Activos	Tarjetas	4,927	5,031	5,216	5,357	5,406	5,265	5,219	5,767	5,244
	Extralínea	1,733	1,377	1,857	1,769	1,735	1,362	1,896	1,498	1,696
	Compra de Deuda	133	120	124	144	124	132	125	128	127
	Libre Disponibilidad	3,498	3,316	3,604	3,574	4,125	2,302	3,474	2,622	3,477
	PrestaBono	2,312	2,458	4,290	3,120	4,415	2,382	4,315	3,139	3,503
	Descuento por Planilla	3,391	2,806	4,250	3,994	3,316	3,332	3,978	2,302	3,743
	Vehicular	124	120	122	130	144	129	124	131	121
	Hipotecario	1,774	2,018	2,042	2,126	2,156	1,408	1,720	2,186	2,342
Cuentas de Ahorro	Cuenta Free	17,718	17,335	14,405	14,070	14,175	13,991	15,026	14,044	18,145
	Cuenta Power	3,196	2,875	3,608	3,452	3,834	2,669	4,196	2,968	2,317
	Super Cuenta	2,865	2,302	3,136	2,701	3,452	4,297	3,974	2,589	2,989
	Cuenta Sueldo	2,192	2,426	1,972	1,900	2,149	2,155	2,422	2,185	1,969
	Fondos Mutuos	124	140	136	136	138	133	138	120	143
	Despositos a Plazo	191	134	176	237	191	158	129	182	157
Seguros	Seguro Fraude TC	12,167	12,164	11,815	10,357	11,069	10,891	11,180	11,305	9,705
	Seguro Accidentes Personales	235	232	218	215	210	207	218	221	182
	Seguro de Desempleo	230	213	235	223	214	216	189	224	213

Tabla 11: Numero de Ventas por producto financiero
Fuente: Elaboración propia

Tipo	Producto	MES 1	MES 2	MES 3	MES 4	MES 5	MES 6	MES 7	MES 8	MES 9
Activos	Tarjetas	S/. 118	S/. 108	S/. 97	S/. 124	S/. 113	S/. 110	S/. 86	S/. 86	S/. 125
	Extralínea	S/. 523	S/. 403	S/. 452	S/. 389	S/. 478	S/. 426	S/. 543	S/. 461	S/. 425
	Compra de Deuda	S/. 120	S/. 124	S/. 92	S/. 121	S/. 112	S/. 102	S/. 105	S/. 88	S/. 128
	Libre Disponibilidad	S/. 266	S/. 281	S/. 278	S/. 259	S/. 259	S/. 269	S/. 293	S/. 283	S/. 305
	PrestaBono	S/. 294	S/. 274	S/. 253	S/. 260	S/. 261	S/. 306	S/. 287	S/. 279	S/. 297
	Descuento por Planilla	S/. 304	S/. 303	S/. 270	S/. 280	S/. 279	S/. 278	S/. 263	S/. 277	S/. 256
	Vehicular	S/. 287	S/. 269	S/. 285	S/. 286	S/. 274	S/. 281	S/. 252	S/. 257	S/. 303
	Hipotecario	S/. 407	S/. 353	S/. 371	S/. 359	S/. 384	S/. 411	S/. 378	S/. 391	S/. 407
Cuentas de Ahorro	Cuenta Free	S/. 12	S/. 14	S/. 8	S/. 10	S/. 12	S/. 7	S/. 14	S/. 8	S/. 5
	Cuenta Power	S/. 44	S/. 38	S/. 49	S/. 45	S/. 52	S/. 34	S/. 47	S/. 34	S/. 47
	Super Cuenta	S/. 50	S/. 42	S/. 49	S/. 48	S/. 47	S/. 46	S/. 43	S/. 44	S/. 42
	Cuenta Sueldo	S/. 22	S/. 28	S/. 31	S/. 28	S/. 31	S/. 31	S/. 25	S/. 30	S/. 26
	Fondos Mutuos	S/. 87	S/. 110	S/. 122	S/. 112	S/. 88	S/. 126	S/. 108	S/. 89	S/. 92
	Despositos a Plazo	S/. 116	S/. 87	S/. 95	S/. 88	S/. 89	S/. 109	S/. 125	S/. 126	S/. 125
Seguros	Seguro Fraude TC	S/. 11	S/. 12	S/. 12	S/. 11	S/. 12	S/. 12	S/. 13	S/. 11	S/. 12
	Seguro Accidentes Personales	S/. 24	S/. 21	S/. 21	S/. 23	S/. 23	S/. 23	S/. 21	S/. 24	S/. 24
	Seguro de Desempleo	S/. 24	S/. 21	S/. 24	S/. 23	S/. 22	S/. 24	S/. 22	S/. 23	S/. 21

Tabla 12: Rentabilidad generada 12 meses después de la apertura
Fuente: Elaboración propia

3.1.4. Selección de variables principales (modelos de rentabilidad)

Esta etapa es necesaria para reducir la dimensionalidad de los datos al seleccionar las características más importantes para la etapa de entrenamiento del modelo. Existen diversos algoritmos de selección de características que identifican un subconjunto de variables regresoras que predicen de manera óptima la variable dependiente. Las principales ventajas de aplicar esta etapa son las siguientes:

- Mejora la generalización de la predicción.
- Elimina variables correlacionadas e información redundante.
- Mejora el entendimiento del comportamiento de las personas dentro del evento a predecir.

Usaremos el algoritmo Boques Aleatorios (*Random Forest*, en el idioma inglés) para seleccionar las variables principales de cada modelo predictivo, este algoritmo es un tipo de modelo ensamblado donde se entrena una serie de árboles de decisión y luego se promedian para reducir la variación de los mismos. Los parámetros de entrada de este algoritmo son la cantidad de árboles que entrenaremos, el criterio de validación que en nuestro caso es el valor de la métrica MSE y la profundidad máxima de los árboles de decisión. A manera de ejemplo, en la Figura 13 se muestran las 15 variables más importantes, ordenados por su orden de importancia, obtenidas al aplicar este algoritmo para la campaña de Libre Disponibilidad. Estas variables serán utilizadas en el desarrollo de los modelos de rentabilidad.

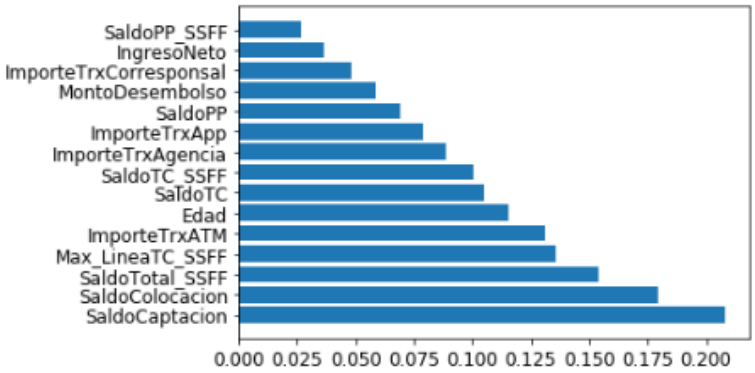


Figura 13: Selección de Variables Principales usando Random Forest
Fuente: Elaboración propia

Por último, se deben seleccionar aquellas variables con una correlación máxima entre sí del 40%, evitando así el sobreajuste del modelo; lo planteado se comprueba en la Figura 14 donde se muestra la matriz de correlaciones entre las variables finales que serán utilizadas para el modelo de rentabilidad del producto Libre Disponibilidad.

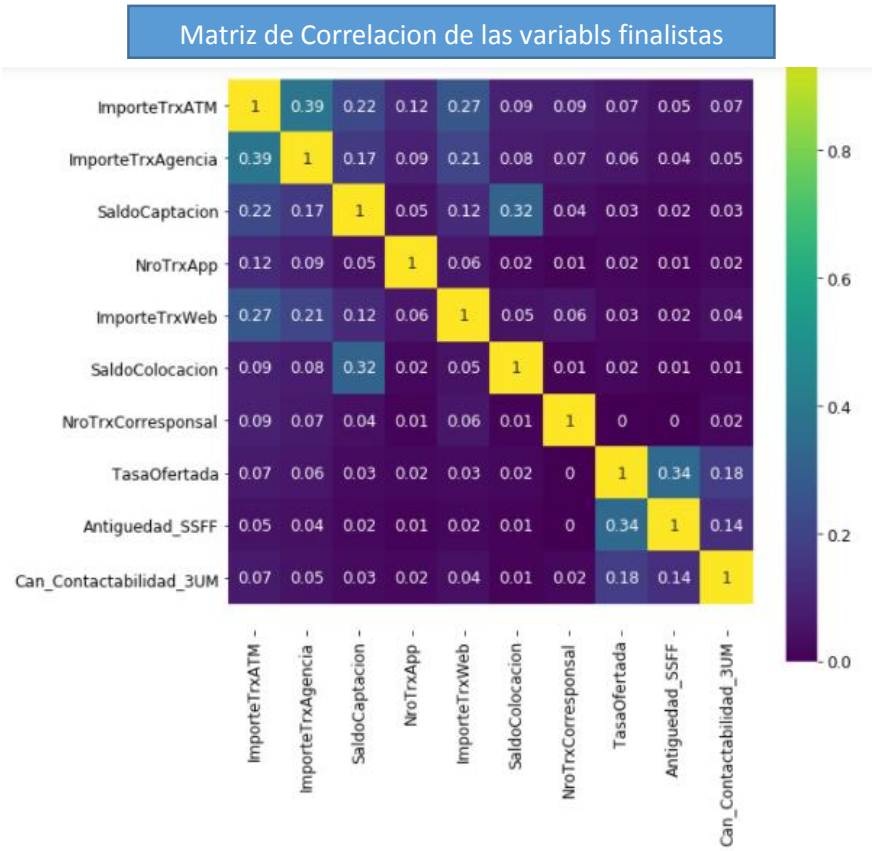


Figura 14: Matriz de Correlaciones de las variables finalistas
Fuente: Elaboración propia

Usaremos el siguiente algoritmo para identificar las variables principales del modelo de rentabilidad de cada producto financiero. Emplearemos, en cada caso el algoritmo Random Forest:

Var_Target = Arreglo de las variables dependientes en el conjunto de datos
Lista_Var_Ind = Arreglo de todas las variables independientes
Matriz_VarFinales = Matriz con las variables finalistas
Para var ∈ Var_Target
X_train = Tabla conformada por los valores de las variables "Lista_Var_Ind"
Y_train = Tabla conformada por los valores de la variable "var"
rf = RandomForesRegressor(radom_state = 0)
rf.fit(X_train,Y_train)
Importancia_Var = Arreglo con las 20 variables más importantes según "rf"
Var_Finalistas = Arreglo con las variables finales luego de eliminar correlaciones
Ingresar en la matriz "Matriz_VarFinales" las variables finales por cada "var"
Fin Para

El algoritmo anterior proporciona las variables principales (con una correlación máxima del 40%) que usaremos en el entrenamiento de cada modelo de regresión, este modelo servirá para calcular la rentabilidad esperada por producto financiero.

3.1.5. Desarrollo de los modelos de rentabilidad

En las secciones anteriores se ha preprocesado los datos, definido la variable dependiente y seleccionado las variables principales para cada producto financiero. El siguiente paso consiste en construir los modelos de rentabilidad.

Para alcanzar dicho objetivo, seguimos los siguientes pasos:

- Definimos la base de entrenamiento (Train) y la base de testeo (Test); para nuestro caso la base de entrenamiento corresponde al 70% de la población inicial, mientras que la base de testeo al 30% restante.
- Usaremos diferentes modelos de regresión y escogeremos el que tenga mayor capacidad de generalización, usando la métrica R^2.

  Para entrenar los modelos de rentabilidad se utilizan diferentes clases de Python pertenecientes a la librería *Sklearn*. Estas clases de Python son:

  - *LinearRegression*. – Clase de python que se utiliza para entrenar un modelo de regresión lineal múltiple.
  - *ElasticNet*. - Clase de python que se utiliza para entrenar un modelo de regresión que combina linealmente las penalizaciones L1 y L2 de los modelos de regresión Lasso y Ridge.
  - *AdaBoostRegressor*. – Clase de python que se utiliza para entrenar un meta estimador que comienza la predicción realizando una regresión en el conjunto de datos originales, luego realiza regresiones adicionales sobre copias del mismo conjunto de datos ajustando los pesos de los estimadores. Este algoritmo también es conocido como AdaBoost.
  - *RandomForesRegressor*. – Clase de python que se utiliza para entrenar un modelo ensamblado que entrena varios árboles de regresión creadas con diferentes grupos de estimadores y muestras de la información original, luego promedia los resultados para ajustar el error global.
  - *SVC*. – Clase de python que se utiliza para entrenar un modelo de regresión que combina tanto regresiones lineales como no lineales. La idea básica de este modelo es ajustar el error dentro de cierto umbral definido por el analista.
  - *r2_score*. – Métrica de python que es utilizada para medir la capacidad de generalización de cada modelo de regresión, a través del valor del coeficiente de determinación R^2, su valor se encuentra entre 0 y 1.

- Se crea una función en Python que inicialice todas las clases de Python utilizadas para entrenar los modelos de regresión que probaremos en la etapa de entrenamiento, cada uno de estas clases tiene un conjunto de parámetros tales como la cantidad de iteraciones, la máxima cantidad de árboles y la máxima profundidad de estos para modelos basados en arboles de regresión.
- Para cada producto financiero se utilizan las variables principales obtenidas en la sección anterior, se entrena cada modelo de regresión con la base de entrenamiento y se realiza la validación con la base de testeo, se elige el algoritmo que tenga el mayor R^2. La Tabla 13 muestra el algoritmo seleccionado para construir los modelos de rentabilidad para cada producto.

Producto Financiero	Algoritmo
Tarjetas de Crédito	AdaBoost
Xtralinea	Regresión lineal penalizada
Compra de Deuda TC	Regresión lineal penalizada
Libre Disponibilidad	AdaBoost
Prestabono	Regresión lineal
Descuento Planilla	Random Forest
Préstamo Vehicular	Regresión lineal penalizada
Préstamo Hipotecario	Regresión lineal
Cuenta Free	AdaBoost
Cuenta Power	AdaBoost
Super Cuenta	AdaBoost
Cuenta Sueldo	Random Forest
Fondos Mutuos	AdaBoost
Depósito a Plazo	Regresión lineal
Seguro Fraude Tarjeta	Regresión lineal
Seguro Accidentes Personales	Regresión lineal penalizada
Seguro de Desempleo	Regresión lineal

Tabla 13: Algoritmo seleccionado para cada Modelo de Rentabilidad
Fuente: Elaboración propia

- Los modelos entrenados y validados se utilizarán para calcular la rentabilidad esperada por producto en futuras campañas comerciales.

Los modelos de regresión captan el perfil de cada cliente que ha adquirido un producto financiero y de acuerdo a este perfil predice la rentabilidad que generará al adquirir dicho producto. En la sección 4.1 calcularemos el valor de la métrica R^2 para cada modelo de rentabilidad, ésta mide la capacidad de generalización del modelo. Un buen modelo es aquel que tiene un valor de R^2 que se encuentra entre [0.60 - 0.75].

3.2. Construcción de los modelos de propensión

Mensualmente, el área de Inteligencia Comercial elaboró estrategias de marketing en base a la probabilidad de adquisición a la toma del producto financiero, esta probabilidad es calculada utilizando modelos de propensión que actualmente se encuentran descalibrados debido a su antigüedad. Por tanto, es necesario medir la capacidad de generalización de los modelos que actualmente son utilizando por el área de Inteligencia Comercial. Para ello, utilizaremos la métrica AUC. En la Tabla 14 se muestra los AUC teóricos obtenidos en el desarrollo de los modelos actualmente usados y los AUC calculados a partir de los resultados de las campañas comerciales en los seis primeros meses del 2021:

Campaña Comercial de:	AUC Teórico	AUC Real (meses de campaña)					
		Enero	**Febrero**	**Marzo**	**Abril**	**Mayo**	**Junio**
Tarjetas de Crédito	0.75	0.63	0.67	0.67	0.67	0.64	0.68
Xtralinea	0.79	0.66	0.61	0.68	0.71	0.66	0.70
Compra de Deuda TC	0.77	0.69	0.68	0.71	0.70	0.64	0.68
Libre Disponibilidad	0.74	0.66	0.68	0.69	0.61	0.69	0.65
PrestaBono	0.79	0.67	0.66	0.68	0.66	0.69	0.61
Descuento Planilla	0.77	0.69	0.67	0.62	0.66	0.63	0.68
Vehicular	0.74	0.68	0.66	0.64	0.68	0.65	0.65
Hipotecario	0.75	0.69	0.69	0.70	0.61	0.65	0.67
Cuenta Free	0.74	0.61	0.67	0.66	0.66	0.68	0.71
Cuenta Power	0.73	0.61	0.70	0.66	0.63	0.69	0.62
Súper Cuenta	0.74	0.66	0.62	0.66	0.63	0.63	0.65
Cuenta Sueldo	0.76	0.66	0.66	0.63	0.67	0.66	0.66
Fondos Mutuos	0.75	0.64	0.66	0.61	0.63	0.66	0.70
Depósito a Plazo	0.75	0.61	0.66	0.64	0.61	0.61	0.66
Seguro Fraude Tarjetas	0.73	0.71	0.63	0.62	0.65	0.65	0.61
Seguro Accidentes Personales	0.76	0.71	0.61	0.69	0.61	0.66	0.67
Seguro De Desempleo	0.74	0.66	0.65	0.68	0.71	0.70	0.61

Tabla 14: Valor actual de la métrica AUC para los modelos de propensión actual
Fuente: Elaboración propia

Conclusión: La calidad de la predicción de los actuales de modelos de propensión, es muy baja, por lo que es necesario desarrollar nuevos modelos analíticos.

El conjunto de datos construido y procesado en la sección 3.1.2 de la presente tesis, será utilizada para el desarrollo de los nuevos modelos de propensión, los pasos a seguir son los siguientes:

- Definición de la variable dependiente.
- Selección de las variables principales.
- Desarrollo de los modelos de propensión (entrenamiento y validación).

3.2.1. Definición de la variable dependiente del modelo de propensión

La variable dependiente de los modelos de propensión es una variable dicotómica o binaria que puede tomar solo dos valores (1 o 0), esta variable representa el evento que se desea predecir. En nuestro caso este evento es la adquisición de un producto financiero y es representado de la siguiente forma:

$$y_i = \begin{cases} 1, si\ el\ lead\ \text{"i"}\ adquiere\ el\ producto\ financiero \\ 0, si\ el\ lead\ \text{"i" no}\ adquiere\ el\ producto\ financiero \end{cases}$$

Donde y_i es la variable dependiente que representa el evento a predecir.

Este evento es interpretado a partir de las variables que hemos construido en el conjunto de datos: mientras mayor es la cantidad y la calidad de las variables explicativas, el modelo tendrá mayor capacidad de predicción. Otro factor importante es la estabilidad del evento durante los periodos de entrenamiento; como ejemplo, en la Tabla 15 se muestra la estabilidad del evento: apertura de préstamos en la campaña de Libre Disponibilidad durante los seis primeros meses del año 2021:

	Enero	Febrero	Marzo	Abril	Mayo	Junio
Nº Leads	21,604	27,871	25,266	20,703	25,312	25,396
Nº Ventas	4,474	5,239	5,202	3,769	4,287	4,116
Efectividad	21%	19%	21%	18%	19%	18%

Tabla 15: Efectividad de la campaña de Libre Disponibilidad (variable dependiente)
Fuente: Elaboración propia

Se observa que la efectividad es lo suficientemente alta para que el modelo pueda identificar correctamente el perfil del cliente que adquiere este producto financiero. Los modelos analíticos que se van a desarrollar, como parte de la propuesta de solución de la presente tesis, buscan calcular la probabilidad de que este evento ocurra. Los modelos de regresión logística son comúnmente usados para tal fin:

$$\Pr(y_i = 1) = F(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_k X_{ki}\ ), donde\ k\ es\ la\ cantidad\ variables$$

$$\Pr(y_i = 1) = F(\beta^T X)\ \rightarrow\ F(\beta^T X) = \frac{e^{\beta^T X}}{1 + e^{\beta^T X}}$$

Aunque el modelo de regresión logística es el más usado, también existen otros modelos de clasificación que son utilizados para calcular dicha probabilidad.

3.2.2. Selección de las variables principales (modelo de propensión)

Actualmente, se cuenta con 75 variables independientes en el conjunto de datos, las cuales se usan para interpretar el evento de adquisición de un producto financiero; sin embargo, es necesario establecer un criterio que reduzca la cantidad de variables y seleccione las principales, en nuestro caso el criterio de selección será el índice Gini:

$$Gini = \frac{\sum_{i=1}^{m} |p_{gi} - q_{gi}|}{\sum_{i=1}^{m} q_{gi}}$$

Donde:

gi: Es el grupo generado a partir de la variable independiente var_i, comúnmente se usa quantiles para las variables numéricas y para las categóricas usamos sus valores únicos, "m" es la cantidad de grupos.

$$p_{gi} = \frac{Suma\ acumulativa\ de\ la\ cantidad\ de\ elementos\ hasta\ el\ grupo\ gi}{Cantidad\ Total\ de\ elementos}$$

$$q_{gi} = \frac{Suma\ acumulativa\ de\ eventos\ positivos\ (Y = 1)\ hasta\ el\ grupo\ gi}{Cantidad\ Total\ de\ eventos\ positivos}$$

El índice Gini permite calcular, para cada variable del conjunto de datos, el porcentaje de discriminación de los eventos positivos (Y=1) respecto de los negativos (Y=0). Este índice se calcula mediante el siguiente algoritmo:

```
Var_Originales = Arreglo de variables del Data Set
N = Cantidad total de registros del Data Set
V = Total de eventos positivos
Matriz_Var_Gini = ∅
Para var ∈ Var_Originales
  Si var es Numerico
      Tabla_Agrup = Matriz formada agrupando quintiles de la variable
  Sino
      Tabla_Agrup = Matriz formada agrupando valores únicos de la variable
  Fin Si
  Agregar a la Tabla_Agrup la columna "SumE" Suma Acumulativa de registros
  Agregar a la Tabla_Agrup la columna "SumV" Suma Acumulativa de eventos positivos
  Agregar a la Tabla_Agrup la columna "p" igual a "SumE/N"
  Agregar a la Tabla_Agrup la columna "q" igual a "SumV/V"
  Agregar a la Tabla_Agrup la columna "diff_pq" igual al valor absoluto de "p-q"
  SumDiff_pq = Suma de elementos de la columna "diff_pq" de la tabla "Tabla_Agrup"
  Sum_Q      = Suma de elementos de la columna "q" de la tabla "Tabla_Agrup"
  Gini       = SumDiff_pq / Sum_Q
  Agregar a la Matriz "Matriz_Var_Gini" el arreglo: [Var, Gini]
Fin Para
```

Este algoritmo identifica, para cada producto financiero, las variables que interpretan mejor el evento a predecir (la adquisición del producto).

3.2.3. Desarrollo de los modelos de propensión

En esta sección desarrollaremos y entrenaremos los modelos analíticos utilizando las variables principales identificadas en la sección anterior. Los pasos a seguir son:

1. Definimos la base de entrenamiento (Train) y la base de validación (Test); para nuestro caso la base de entrenamiento corresponde al 70% de la población inicial, mientras que la base de validación al 30% restante.
2. Seleccionamos para cada producto financiero las principales variables que utilizaremos en la etapa de entrenamiento del modelo de propensión.
3. Utilizamos la métrica AUC para medir la capacidad de generalización de cada modelo de propensión, debido a que nuestra variable dependiente es una variable dicotómica que toma el valor de "1" en caso el lead adquiera el producto financiero y el valor de "0" en caso contrario.
4. Para desarrollar el modelo predictivo se han entrenado cuatro modelos diferentes: Regresión Logística, Máquina Leve de Gradiente Ascendente (LGBM, por sus siglas del inglés), Aumento de Gradiente Extremo (XGB, por sus siglas del inglés) y el Clasificador de Vectores de Soporte (SVC, por sus siglas del inglés); en la Tabla 16 se muestra el AUC obtenido en la etapa de entrenamiento de cada algoritmo, de estos se selecciona el mejor:

Modelo Analítico	AUC
LGBM	**0.785**
XGB	0.771
Regresión Logística	0.753
SVC	0.726

Tabla 16: Comparación del valor de la métrica AUC por modelo
Fuente: Elaboración propia

5. El ejemplo anterior corresponde a la campaña del producto de libre disponibilidad; del testeo concluimos que el modelo LGBM tiene la mayor capacidad de generalización. El modelo seleccionado es exportado y guardado para futuras campañas comerciales.
6. Este procedimiento se aplica para desarrollar el modelo de propensión de los 17 productos financieros que participan en las campañas comerciales. De esta manera se utilizará un algoritmo diferente para calcular la probabilidad de adquisición de cada producto financiero.

3.3. Estandarización de las probabilidades

Las probabilidades de adquisición de cada producto financiero se calculan mediante los modelos de propensión previamente desarrollados. Sin embargo, estos han sido construidos utilizando diferentes algoritmos. Debido a esto, estas probabilidades no son comparables entre sí, por lo que es necesario estandarizarlas mediante un modelo analítico que tenga como inputs las probabilidades obtenidas de los modelos estadísticos y como salidas las probabilidades estandarizadas. La probabilidad estandarizada y la rentabilidad esperada serán las variables usadas para construir la función de priorización que permitirá ordenar los productos financieros de mayor a menor interés para el cliente (esta función es el output del modelo NBO).

Para la estandarización de las probabilidades utilizaremos dos modelos analíticos diferentes y luego compararemos el incremento de la generalización de las probabilidades originales luego de utilizar cada modelo analítico, los dos modelos son los siguientes:

- Una red neuronal que tenga como entradas las probabilidades que proveen los modelos estadísticos y como salidas las probabilidades estandarizadas.
- Modelos ensamblados que utiliza un meta modelo (Regresión Logística) que combina los resultados de diferentes modelos de clasificación para obtener una predicción tan buena o mejor que cualquier modelo único.

Previamente a la estandarización de las probabilidades, debemos validar la integridad de los datos de entrada; es decir, si existen valores nulos estos serán remplazados, las probabilidades deben estar entre 0 y 1. En la sección anterior se validó que la cantidad de eventos positivos para cada producto financiero es alta, por tanto, se puede evitar el balanceo de la variable dependiente.

El conjunto de entrenamiento contiene 17 variables independientes que representan las probabilidades de cada modelo estadístico y 17 variables dependientes que indican si la persona adquirió o no el producto financiero:

$$y_{ik} = \begin{cases} 1, si\ la\ persona\ \text{"i"}\ adquiere\ el\ producto\ financiero\ \text{"}k\text{"} \\ 0, si\ la\ persona\ \text{"i" no}\ adquiere\ el\ producto\ financiero\ \text{"}k\text{"} \end{cases}$$

La calidad de la predicción de las probabilidades ajustadas y estandarizadas lo calculamos utilizando la métrica AUC, por lo cual en la etapa de validación compararemos el AUC obtenido utilizando cada una de las metodologías citadas anteriormente para cada campaña comercial.

3.3.1. Estandarización mediante Redes Neuronales

Previamente al desarrollo de la red neuronal se prueban diferentes métodos de transformación de los datos de entrada para incrementar la capacidad de predicción de la red neuronal; los métodos probados son los siguientes: transformación min – max, método de Normalización, método de estandarización y el método de transformación Box – Cox, siendo este último la mejor técnica de transformación.

La Transformación Box - Cox permite corregir las varianzas desiguales y la no linealidad que existe entre las variables independientes, mejorando así la correlación entre ellas. Utilizando el algoritmo KNN imputamos los valores cero que puedan existir en el conjunto de datos, luego importamos la clase *PowerTransformer* de la librería *Sklearn* de Python para realizar la transformación propiamente dicha; los valores transformados serán el input para la red neuronal.

En la Figura 15, se muestra la curva Característica Operativa del Receptor (ROC, por sus siglas del inglés), ésta muestra como varia la sensibilidad de la predicción en función de la variación de especificidad para un clasificador binario:

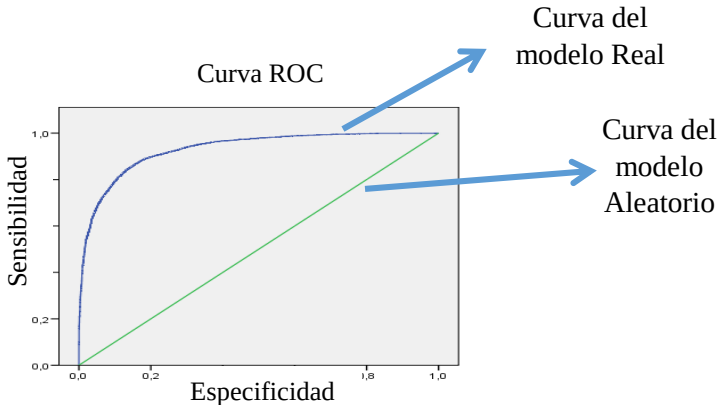


Figura 15: Curva ROC de un modelo predictivo
Fuente: Elaboración propia

En la Figura 16, se muestra la arquitectura de la red neuronal utilizada. Está tiene 17 neuronas de entrada (una neurona por cada probabilidad de adquisición de cada producto financiero), 4 capas ocultas con 60 neuronas cada, y una capa con 17 neuronas de salida (correspondientes a las variables dependientes binarias que indica si la persona adquirió o no el producto financiero). En cada una de las capas ocultas se utiliza una función de activación Sigmoidea o Relu; además, para evitar el sobreajuste se utiliza la técnica *Dropout*, ésta permite desactivar de manera aleatoria un porcentaje de las neuronas de la capa oculta, de acuerdo a una probabilidad de descarte definida previamente.

A continuación, se muestra la arquitectura de la red neuronal desarrollada:

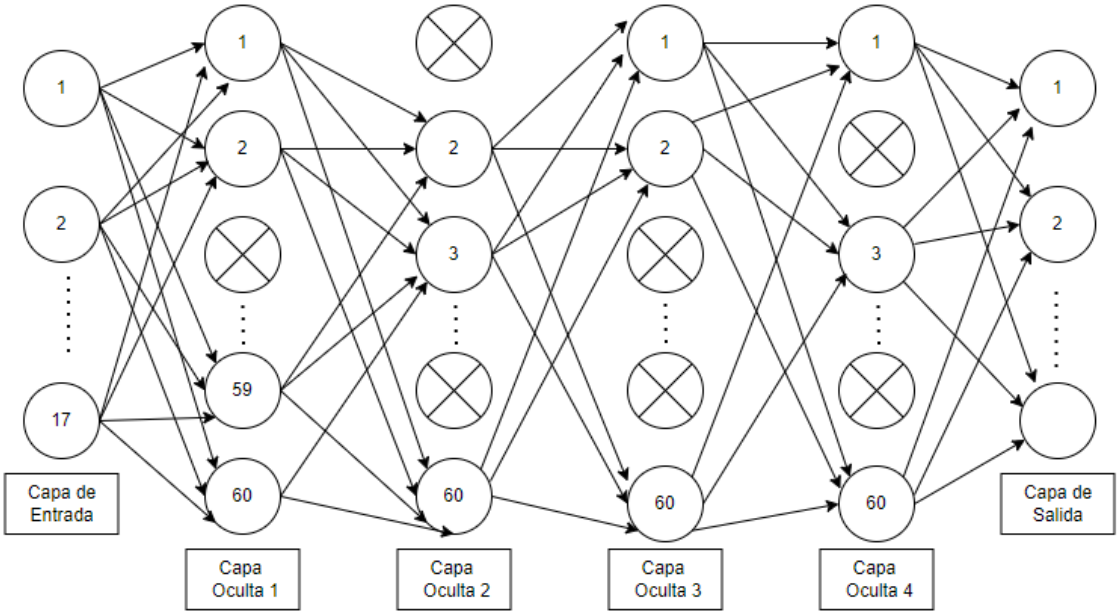


Figura 16: Arquitectura de la Red Neuronal desarrollada
Fuente: Elaboración propia

El entrenamiento de la red neuronal se realiza utilizando la librería de Python: *TensorFlow*, el optimizador *Adam SGD* y la métrica AUC. Se consideró 50 épocas para evitar el sobreajuste de los resultados.

En la Figura 17 se muestra cómo va incrementándose la métrica AUC para cada época del entrenamiento, al finalizar las 50 épocas, el valor de la métrica AUC asciende a 0.825.

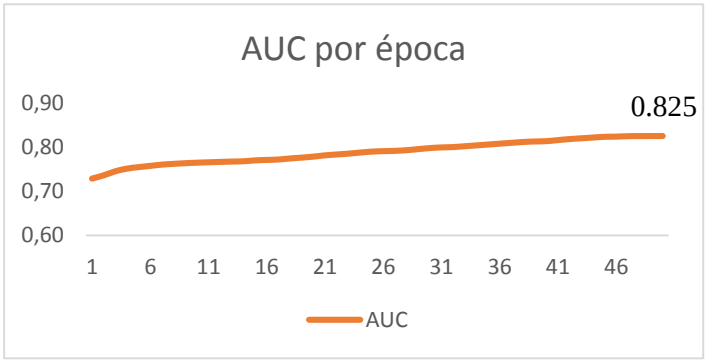


Figura 17: Incremento de la métrica AUC en el entrenamiento de la RN
Fuente: Elaboración propia

Las probabilidades obtenidas de las redes neuronales están estandarizadas y son comparables entre sí porque son obtenidas de un mismo algoritmo, estas se usarán en el modelo NBO para identificar el combo de productos que se le debe ofrecer a una persona dentro de las campañas comerciales.

3.3.2. Estandarización mediante un algoritmo ensamblado

Un modelo ensamblado es aquel que utiliza como variables las predicciones de los diferentes modelos analíticos para construir un solo meta modelo que predice la adquisición de un producto financiero.

La Figura 18 muestra un diagrama que representa al modelo ensamblado utilizado para estandarizar las probabilidades obtenidas de los modelos estadísticos. En la construcción de este modelo se han utilizado 4 algoritmos diferentes como modelos unitarios y el algoritmo de Regresión Logística como meta estimador. En nuestro caso, es necesario desarrollar 17 modelos ensamblados (uno para cada producto financiero). De esta manera estandarizamos e incrementamos el grado de generalización de las probabilidades obtenidas de los modelos estadísticos.

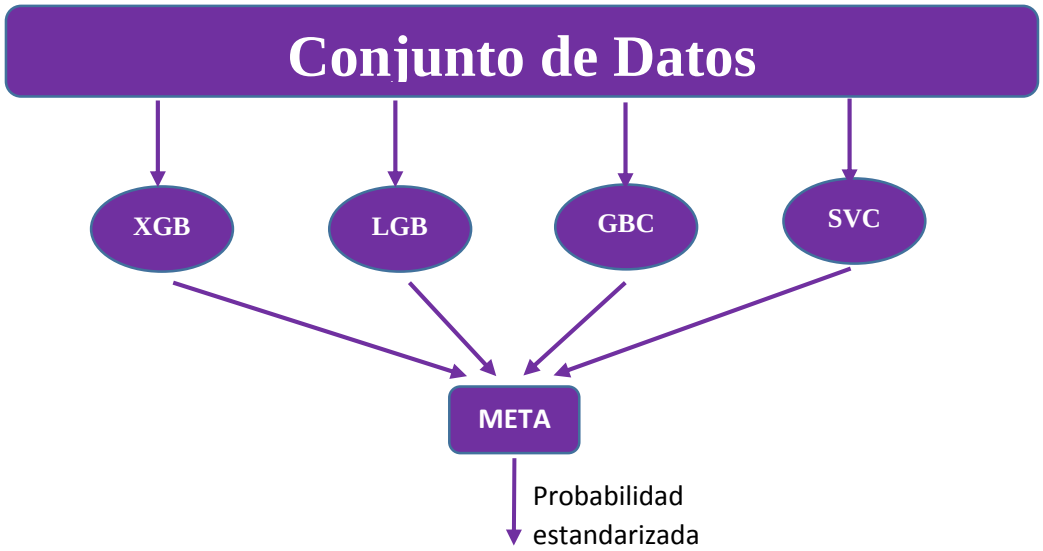


Figura 18: Modelo Ensamblado aplicado a la Banca
Fuente: Elaboración propia

Donde:

- El conjunto de datos es la tabla conformada por las probabilidades obtenidas de los modelos estadísticos y las variables dependientes que indican si una persona adquirió o no un producto financiero.
- XGB: Modelo de Aumento de Gradiente Extremo.
- LGB: Modelo de Maquina Leve de Gradiente Ascendente.
- GBC: Modelo de Clasificación de Gradiente Ascendente.
- SVC: Modelo de Clasificación de Soporte Vectorial.
- META: Meta modelo que calcula la probabilidad final de adquisición de un producto financiero.

Para desarrollar los modelos ensamblados, debemos crear un algoritmo que inicialice cada modelo unitario con sus respectivos parámetros de configuración. Para ello, utilizaremos las siguientes clases de Python:

- *XGBClassifier*: Clase de Python utilizada para entrenar modelos de clasificación XGB.
- *LGBMClassifier*: Clase de Python utilizada para entrenar modelos de clasificación LGBM.
- *GradientBoostingClassifier*: Clase de Python utilizada para entrenar modelos de Clasificación de Gradiente Ascendente.
- *SVC*: Clase de Python utilizada para entrenar modelos de clasificación Maquina de Soporte Vectorial.

Después de inicializado los modelos, se utiliza el 70% de la población para su entrenamiento y el 30% restante para su validación. Por último, los resultados de los modelos unitarios servirán para entrenar el meta estimador.

La Figura 19 muestra el diagrama de flujo que representa al algoritmo utilizado para desarrollar los modelos ensamblados de cada producto; es decir, el entrenamiento de los modelos unitarios y del meta modelo.

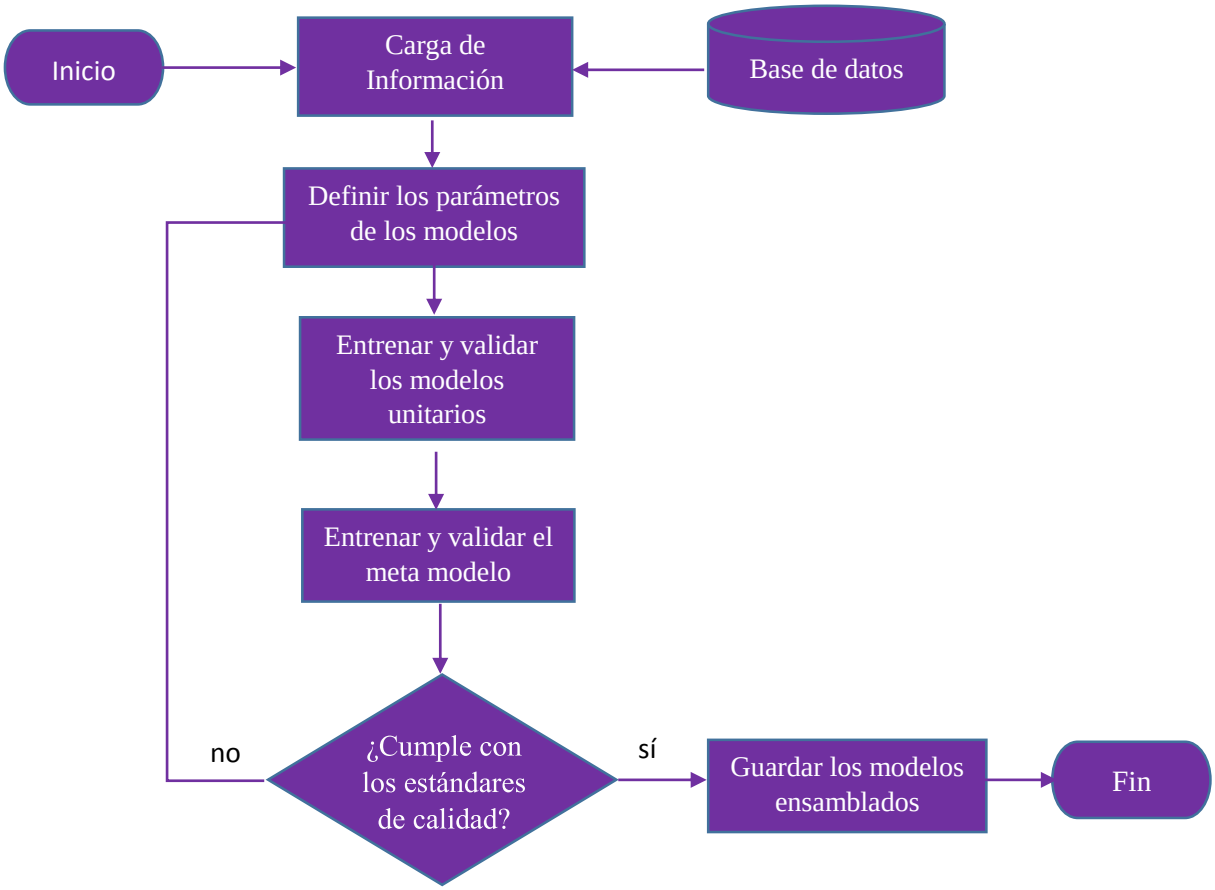


Figura 19: Diagrama de Flujo utilizado para desarrollar los modelos ensamblados
Fuente: Elaboración propia

Donde:

- Carga de Información.- Consiste en cargar al entorno de Python la entrada de los modelos unitarios, es decir, las probabilidades obtenidas de los modelos estadísticos.

- Definir los parámetros de los modelos.- Cada clase de Python descrita anteriormente utiliza múltiples parámetros de inicialización, los principales son: la profundidad máxima del árbol, la cantidad de iteraciones, la tasa de entrenamiento y el tipo de entrenamiento.

- Entrenar y validar los modelos.- Se entrena cada uno de los 4 modelos unitarios con el 70% de la población. Además, se utiliza la métrica AUC para medir el grado de generalización de la predicción, por último, se validan los resultados en el 30% de la población restante.

- Entrenar y validar el meta modelo.- Las probabilidades obtenidas de los modelos unitarios conforman el conjunto de datos de entrada del meta modelo (Regresión Logística), se entrena el modelo con el 70% de estos datos y se valida con el 30% restante. Se utiliza la métrica AUC para medir y validar los resultados de la generalización del meta modelo. Las salidas del meta modelo serán las probabilidades estandarizadas.

- Validación de estándares de calidad.- Un modelo es puesto en producción siempre y cuando cumpla con los siguientes requisitos:

    - Los valores de la métrica AUC deben ser similares en la etapa de entrenamiento y validación.
    - Si el valor del indicador se encuentra entre 0.7 y 0.85, se concluye que su capacidad de generalización es óptima.
    - Las variables independientes deben ser estables en el tiempo para preservar la capacidad de generalización del modelo.

- Guardar los modelo ensamblados.- Si los meta modelos cumplen con todos los requisitos anteriormente mencionados, estos son guardados para que sean aplicados en futuras campañas comerciales.

Capítulo 4

Construcción e Implementación del modelo NBO

Primero, se valida los resultados de los modelos de rentabilidad, y luego se mide la capacidad de generalización de cada método (Redes Neuronales y Modelos Ensamblados) utilizado para estandarizar las probabilidades; por último, se desarrolla el modelo NBO y se evalúa los resultados obtenidos de su puesta en producción.

4.1. Validación de los resultados de los modelos de rentabilidad

En la Tabla 17 validamos la calidad de cada modelo desarrollado calculando su coeficiente de determinación, el valor de este indicador debe estar en el rango [0.6 – 0.85]. La expresión matemática de este coeficiente es la siguiente:

$$R^2 = \frac{cov(o,m)^2}{sd(o)sd(m)}$$

$Donde$:

$cov(o,m)$: $es\ la\ covariancia\ entre\ los\ valores\ observados\ y\ los\ devueltos\ por\ el\ modelo$

$sd(o)$: $es\ la\ desvación\ tipica\ de\ los\ valores\ observados$

$sd(m)$: $es\ la\ desvación\ tipica\ de\ los\ resultados\ del\ modelo.$

Producto Financiero	Algoritmo	R2
Tarjetas de Crédito	AdaBoost	0.752
Xtralinea	Regresión lineal penalizada	0.814
Compra de Deuda TC	Regresión lineal penalizada	0.822
Libre Disponibilidad	AdaBoost	0.705
Prestabono	Regresión lineal	0.774
Descuento Planilla	Random Forest	0.757
Préstamo Vehicular	Regresión lineal penalizada	0.756
Préstamo Hipotecario	Regresión lineal	0.689
Cuenta Free	AdaBoost	0.750
Cuenta Power	AdaBoost	0.639
Super Cuenta	AdaBoost	0.714
Cuenta Sueldo	Random Forest	0.824
Fondos Mutuos	AdaBoost	0.717
Depósito a Plazo	Regresión lineal	0.685
Seguro Fraude Tarjeta	Regresión lineal	0.731
Seguro Accidentes Personales	Regresión lineal penalizada	0.782
Seguro de Desempleo	Regresión lineal	0.733

Tabla 17: Valores de la métrica R^2 de los modelos de rentabilidad
Fuente: Elaboración propia

Se verifica que los modelos tienen alto grado de generalización, sin sobreajuste.

4.2. Validación de los resultados de la Red Neuronal

Luego de entrenar la red neuronal en el 70% de la población, se valida si la capacidad de predicción de la red es la misma en el 30% de la población restante. Los pasos a seguir son los siguientes:

1. En la Tabla 18, se compara el valor de la métrica AUC obtenida en la etapa de entrenamiento con el valor de este indicador en la etapa de validación; estos valores deben ser similares para confiar en los resultados del modelo:

Modelo	AUC del Entrenamiento	AUC de Validación
Red Neuronal	**0.825**	0.812

Tabla 18: Validación de la Red Neuronal (Base Train vs Base Test)
Fuente: Elaboración propia

Conclusión: Los dos valores de la métrica AUC son aproximadamente iguales ($0.825 \approx 0.812$) y mayores a 0.8; entonces, la capacidad de la generalización de la red es confiable.

2. La Tabla 19 muestra la capacidad de generalización de las probabilidades obtenidas de los modelos de propensión y la capacidad de generalización de las probabilidades estandarizadas obtenidas con la red neuronal, concluyendo que la red neuronal tiene mayor capacidad de generalización en cada caso. La capacidad de generalización está determinada por el valor de la métrica AUC.

Producto Financiero	AUC Modelos Propensión	AUC Red Neuronal
Tarjetas de Crédito	0.685	**0.774**
Xtralinea	0.691	**0.746**
Compra de Deuda TC	0.724	**0.789**
Libre Disponibilidad	0.678	**0.728**
Prestabono	0.680	**0.754**
Descuento Planilla	0.691	**0.767**
Préstamo Vehicular	0.658	**0.720**
Préstamo Hipotecario	0.718	**0.806**
Cuenta Free	0.697	**0.776**
Cuenta Power	0.644	**0.703**
Súper Cuenta	0.697	**0.781**
Cuenta Sueldo	0.694	**0.767**
Fondos Mutuos	0.679	**0.756**
Depósito a Plazo	0.674	**0.729**
Seguro Fraude Tarjeta	0.651	**0.722**
Seguro Accidentes Personales	0.731	**0.789**
Seguro de Desempleo	0.621	**0.696**

Tabla 19: Valores de la métrica AUC obtenidos de la red neuronal y de los modelos de propensión.
Fuente: Elaboración propia

4.3. Validación de los resultados de los modelos ensamblados

La validación de estos modelos se realiza comparando la capacidad de generalización de las probabilidades obtenidas de los modelos de propensión con la capacidad de generalización de las probabilidades obtenidos de los modelos ensamblados. Se utiliza la métrica AUC para medir la capacidad de generalización. En la Tabla 20 se muestra, para cada producto financiero, los valores AUC obtenidos en la etapa de validación de los modelos de propensión y de los modelos ensamblados, buscando determinar para cada producto financiero el modelo que tenga mayor capacidad de generalización.

Producto Financiero	AUC Modelos Propensión	AUC Modelos Ensamblado
Tarjetas de Crédito	0.685	**0.803**
Xtralinea	0.690	**0.777**
Compra de Deuda TC	0.724	**0.837**
Libre Disponibilidad	0.678	**0.764**
Prestabono	0.680	**0.794**
Descuento Planilla	0.691	**0.801**
Préstamo Vehicular	0.658	**0.751**
Préstamo Hipotecario	0.718	**0.851**
Cuenta Free	0.697	**0.812**
Cuenta Power	0.644	**0.749**
Súper Cuenta	0.697	**0.805**
Cuenta Sueldo	0.694	**0.813**
Fondos Mutuos	0.679	**0.783**
Depósito a Plazo	0.674	**0.777**
Seguro Fraude Tarjeta	0.651	**0.752**
Seguro Accidentes Personales	0.730	**0.820**

Tabla 20: Valores de la métrica AUC obtenidos de los modelos de propensión y modelos ensamblados
Fuente: Elaboración propia

Todos los valores AUC de los modelos ensamblados son mayores al de los valores AUC de los modelos de propensión, para cada producto financiero. Un modelo óptimo es aquel cuya capacidad de generalización (valor de la métrica AUC), está en el intervalo [0.75 - 0.85]. En la Tabla 20 observamos que los valores de la métrica AUC de los modelos ensamblados se encuentran en este intervalo.

Conclusión: los modelos ensamblados tienen mayor capacidad de generalización que los modelos de propensión. Cuando los valores de la métrica AUC de los modelos ensamblados son mayores que 0.85, es necesario calibrar el modelo.

La Tabla 21 muestra las capacidades de generalización de la Red Neuronal y la de los modelos ensamblados; notamos que los valores de la métrica AUC, para cada producto financiero de los modelos ensamblados es mayor que los valores de la métrica AUC de la red neuronal. Esto nos permite concluir que el modelo óptimo para estandarizar probabilidades es el modelo ensamblado.

Otra bondad de este modelo es que no necesita calibración ya que el valor de la métrica AUC, para cada producto financiero, no supera el valor de 0.85.

Como las campañas comerciales, en el sistema financiero, se realizan todos los meses, cada fin de mes, el valor de la métrica AUC de estos modelos debe ser calculado para validar su estabilidad en el tiempo.

Producto Financiero	AUC Red Neuronal	AUC Modelos Ensamblado
Tarjetas de Crédito	0.774	**0.803**
Xtralinea	0.746	**0.787**
Compra de Deuda TC	0.789	**0.837**
Libre Disponibilidad	0.728	**0.764**
Prestabono	0.754	**0.794**
Descuento Planilla	0.767	**0.810**
Préstamo Vehicular	0.720	**0.751**
Préstamo Hipotecario	0.806	**0.851**
Cuenta Free	0.776	**0.812**
Cuenta Power	0.703	**0.779**
Súper Cuenta	0.781	**0.805**
Cuenta Sueldo	0.767	**0.813**
Fondos Mutuos	0.756	**0.783**
Depósito a Plazo	0.729	**0.787**
Seguro Fraude Tarjeta	0.722	**0.752**
Seguro Accidentes Personales	0.789	**0.820**

Tabla 21: Valores de la métrica AUC obtenidos de la red neuronal y de los modelos ensamblados
Fuente: Elaboración propia

Según los resultados observados los modelos ensamblados utilizados para calcular la probabilidad de adquisición de los productos Seguro Fraude Tarjeta y Préstamo Vehicular, tienen la menor capacidad de generalización. Por tanto, es probable que dentro de un corto periodo de tiempo estos modelos necesiten ser calibrados o rediseñados, porque, el valor de su métrica AUC está cercano a 0.75 que es el límite inferior del rango [0.75 - 0.85] en el que debe variar el valor de esta métrica.

4.4. Desarrollo e implementación del modelo NBO para la priorización de productos financieros.

A partir de la probabilidad de adquisición estandarizada del producto financiero y de la rentabilidad esperada, se construye el modelo NBO que sirve para priorizar los productos ofrecidos a los clientes y no clientes de la organización.

En las secciones anteriores, hemos desarrollado los modelos predictivos que calculan, por producto financiero, la rentabilidad esperada y la probabilidad de adquisición; además, se ha encontrado el mejor método para estandarizar dichas probabilidades. El producto aritmético de la probabilidad estandarizada y la rentabilidad esperada nos da una función de priorización que se utiliza para priorizar los productos que la organización debe ofertar.

A partir de un ejemplo explicaré la utilización del modelo NBO en la priorización de un conjunto de tres productos financieros, de acuerdo a su propensión y a su rentabilidad. Estos productos deben ser ofertados a un *lead* de la organización.

Para explicar el ejemplo, utilizaré el siguiente algoritmo:

1. Se calculan las rentabilidades esperadas para cada uno de los tres productos financieros utilizando los modelos de rentabilidad. En la Tabla 22 se muestra la rentabilidad esperada mensual de los productos: Tarjetas de Crédito, Libre Disponibilidad y PrestaBono.

Producto Financiero	Rentabilidad Esperada (R)
Tarjetas de Crédito	S/ 25.23
Libre Disponibilidad	S/ 35.87
PrestaBono	S/ 45.62

Tabla 22: Rentabilidades Esperadas de Tres Productos Financieros
Fuente: Elaboración propia

2. Se calcula las probabilidades de adquisición para cada uno de los tres productos financieros utilizando los modelos de propensión y se les estandariza utilizando los modelos ensamblados. La Tabla 23 muestra los valores de las probabilidades de adquisición estandarizadas de los productos: Tarjetas de Crédito, Libre Disponibilidad y PrestaBono.

Producto Financiero	Probabilidad (Modelos de Propensión)	Probabilidad Estandarizada (Modelos ensamblados)
Tarjetas de Crédito	79%	68%
Libre Disponibilidad	82%	72%
PrestaBono	72%	65%

Tabla 23: Probabilidades obtenidas de los modelos de propensión y de los modelos ensamblados
Fuente: Elaboración propia

Es necesaria la estandarización de las probabilidades obtenidas de los modelos de propensión, ya que estos modelos utilizan diferentes algoritmos de aprendizaje automático. Por tanto, luego de su estandarización es posible compararlas y utilizarlas en la priorización de los productos financieros.

3. Se calculan el valor de la función de priorización (f = R*P) multiplicando la rentabilidad esperada (R) y la probabilidad estandarizada (P). La Tabla 24 muestra los valores de la función de priorización para los tres productos.

Producto Financiero	Rentabilidad Esperada (R)	Probabilidad Estandarizada (P)	Valor de la Función de Priorización (R*P)
Tarjetas de Crédito	S/ 25.23	68%	17.15
Libre Disponibilidad	S/ 35.87	72%	25.82
PrestaBono	S/ 45.62	65%	29.65

Tabla 24: Valor de la función de priorización para tres productos financieros
Fuente: Elaboración propia

4. Por último, se ordena los productos financieros de acuerdo al valor de la función de priorización. La Tabla 25 muestra los productos financieros ordenados según su prioridad.

Orden de Prioridad	Producto Financiero	Función de Priorización
Prioridad 1	PrestaBono	29.65
Prioridad 2	Libre Disponibilidad	25.82
Prioridad 3	Tarjeta de Crédito	17.15

Tabla 25: Priorización de 3 productos financieros
Fuente: Elaboración propia

De acuerdo a la Tabla 25 se concluye que el producto PrestaBono es el primer producto a ofertar, el siguiente es el producto Libre Disponibilidad y por último el producto Tarjeta de Crédito.

Este ejemplo nos ha permitido comprender el algoritmo que se debe utilizar para priorizar los productos a ofertar por una institución financiero. Por lo tanto, podemos utilizar este algoritmo para priorizar la oferta de todos los productos financieros que la organización ofrece a todos los leads que participan en las campañas comerciales.

La Tabla 26, 27 y 28 muestran la priorización de los 17 productos financieros que se deben ofertar a cuatro leads que participan en las campañas comerciales. La Prioridad 1 es el primer producto a ofertar, la Prioridad 2 es el segundo producto y así sucesivamente hasta llegar a la Prioridad 17 que representa el último producto a ofertar. Se utiliza el modelo NBO para priorizar los 17 productos de acuerdo a su probabilidad de adquisición y a su rentabilidad.

ID Cliente	Prioridad 1	Prioridad 2	Prioridad 3	Prioridad 4	Prioridad 5	Prioridad 6
C001	Préstamo Hipotecario	Depósito a Plazo	Súper Cuenta	Cuenta Power	Tarjetas de Crédito	Seguro Fraude TC
C002	Cuenta Power	Prestabono	Cuenta Sueldo	Compra de Deuda TC	Seguro Fraude TC	Descuento Planilla
C003	Libre Disponibilidad	Súper Cuenta	Préstamo Hipotecario	Cuenta Power	Préstamo Vehicular	Depósito a Plazo
C004	Tarjetas de Crédito	Cuenta Power	Préstamo Vehicular	Súper Cuenta	Préstamo Hipotecario	Depósito a Plazo

Tabla 26: Prioridades del 1 al 6 de seis productos de acuerdo al modelo NBO
Fuente: Elaboración propia

ID Cliente	Prioridad 7	Prioridad 8	Prioridad 9	Prioridad 10	Prioridad 11	Prioridad 12
C001	Préstamo Vehicular	XTralínea	Compra de Deuda TC	Seguro contra Accidentes	Descuento Planilla	Cuenta Sueldo
C002	Cuenta Free	Fondos Mutuos	Préstamo Hipotecario	Depósito a Plazo	XTralínea	Súper Cuenta
C003	Compra de Deuda TC	Descuento Planilla	Cuenta Sueldo	Tarjeta de Crédito	Seguro de Fraude TC	Cuenta Free
C004	Fondos Mutuos	Seguro contra Accidentes	Descuento Planilla	Compra De Deuda TC	XTralínea	Cuenta Sueldo

Tabla 27: Prioridades del 7 al 12 de seis productos de acuerdo al modelo NBO
Fuente: Elaboración propia

ID Cliente	Prioridad 13	Prioridad 14	Prioridad 15	Prioridad 16	Prioridad 17
C001	Fondos Mutuos	PrestaBono	Cuenta Free	Libre Disponibilidad	Seguro de Desempleo
C002	Préstamo Vehicular	Tarjeta de Crédito	Seguro contra Accidentes	Seguro de Desempleo	Libre Disponibilidad
C003	Seguro de Desempleo	PrestaBono	Seguro contra Accidentes	Fondos Mutuos	XTralínea
C004	Cuenta Free	Seguro Fraude TC	Libre Disponibilidad	Seguro de Desempleo	PrestaBono

Tabla 28: Prioridades del 13 al 17 de cinco productos de acuerdo al modelo NBO
Fuente: Elaboración propia

Aplicando el modelo NBO al total de *leads* que participan en las campañas de marketing, se obtiene la priorización de los 17 productos ofertados para cada uno de los *leads*. Esto permite construir estrategias comerciales visión cliente.

4.5. Validación de los resultados del modelo NBO puesto en producción

En cada uno de los seis primeros meses del año 2021 se construyeron estrategias de marketing basadas en el análisis de la información y en los modelos de propensión que en la mayoría de los casos se encontraban descalibrados (Reglas de Experto); mientras, que en los meses restantes del año 2021 se utilizó el NBO para construir dichas estrategias. Se comparó la efectividad, la rentabilidad y el desembolso promedio realizado en cada semestre del año 2021, concluyendo que el modelo NBO ofrece una mejora sustancial de estos indicadores.

Para la medición de los resultados se consideraron 3 indicadores fundamentales:

- Efectividad: Nº de Ventas / Nº de Leads gestionados.
- Rentabilidad: Promedio de la rentabilidad esperada.
- Desembolso: Monto total de las colocaciones expresada en soles.

Detallaré, a modo de explicación, los resultados de estos tres indicadores referentes a las campañas de seis productos financieros a lo largo del año 2021. Las estrategias comerciales de los seis primeros meses se elaboraron utilizando reglas de experto; mientras, que en los últimos seis meses del 2021 se utilizó el modelo NBO.

Los productos financieros utilizados en el análisis de resultados son:

- Libre Disponibilidad.
- Tarjeta de Crédito.
- PrestaBono.
- Préstamo Vehicular.
- Préstamo Hipotecario.
- Descuento por Planilla.

Las Tabla 29, 31, 33, 35, 37 y 39 muestran los resultados de seis campañas comerciales a lo largo año 2021.

Las Tablas 30, 32, 34, 36, 38 y 40 muestran el promedio semestral de los resultados de estas campañas.

Al comparar el valor del promedio de estos indicadores, obtenidos en cada semestre del año 2021, podemos cuantificar el incremento de la efectividad y de la rentabilidad para cada campaña comercial gracias al uso del modelo NBO.

- **Para el producto Libre Disponibilidad:**

Indicadores	Reglas de Experto						Modelo NBO					
	Ene-21	Feb-21	Mar-21	Abr-21	May-21	Jun-21	Jul-21	Ago-21	Sep-21	Oct-21	Nov-21	Dic-21
Nro de Leads	21,063	24,516	22,285	25,208	24,237	21,956	20,010	25,496	22,730	24,955	24,479	22,615
Nro De Ventas	3,203	4,076	4,227	3,937	3,516	4,340	4,406	4,851	4,298	4,499	5,297	5,201
Efectividad	15.21%	16.63%	18.97%	15.62%	14.51%	19.77%	22.02%	19.03%	18.91%	18.03%	21.64%	23.00%
Rentabilidad	S/. 100	S/. 85	S/. 95	S/. 108	S/. 110	S/. 92	S/. 123	S/. 130	S/. 149	S/. 108	S/. 116	S/. 102
Desembolso (en miles)	S/. 70,703	S/. 82,820	S/. 92,022	S/. 91,984	S/. 83,674	S/. 98,783	S/. 106,334	S/. 119,252	S/. 111,731	S/. 111,071	S/. 123,245	S/. 135,699

Tabla 29: Resultados de las campañas comerciales del año 2021 del producto Libre Disponibilidad.
Fuente: Elaboración propia

Indicadores	Promedio de Indicadores		Incremento
	Reglas de Experto (6 Primeros Meses 2021)	Modelo NBO (6 Últimos Meses 2021)	
Nro de Leads	23,211	23,381	
Nro De Ventas	3,883	4,759	876
Efectividad	16.79%	20.44%	3.65%
Rentabilidad	S/. 98	S/. 121	S/. 23
Desembolso	S/. 86 MM	S/. 117 MM	S/ 31 MM

Tabla 30: Promedio semestral de los indicadores de las campañas del producto Libre Disponibilidad
Fuente: Elaboración propia

- **Para el producto Tarjeta de Crédito**

Indicadores	Reglas de Experto						Modelo NBO					
	Ene-21	Feb-21	Mar-21	Abr-21	May-21	Jun-21	Jul-21	Ago-21	Sep-21	Oct-21	Nov-21	Dic-21
Nro de Leads	23,050	27,069	25,978	21,325	23,388	21,270	23,510	28,151	25,198	20,898	24,557	20,631
Nro de Ventas	4,879	5,887	4,969	4,508	3,494	4,011	4,424	6,742	4,684	4,196	5,112	4,542
Efectividad	21.17%	21.75%	19.13%	21.14%	14.94%	18.86%	18.82%	23.95%	18.59%	20.08%	20.82%	22.02%
Rentabilidad	S/. 81	S/. 105	S/. 91	S/. 95	S/. 100	S/. 102	S/. 111	S/. 130	S/. 146	S/. 123	S/. 104	S/. 139

Tabla 31: Resultados de las campañas comerciales del año 2021 del producto Tarjeta de Crédito
Fuente: Elaboración propia

Indicadores	Promedio de Indicadores		Incremento
	Reglas de Experto (6 Primeros Meses 2021)	Modelo NBO (6 Últimos Meses 2021)	
Nro de Leads	23,680	23,824	
Nro de Ventas	4,625	4,950	325
Efectividad	19.50%	20.71%	1.21%
Rentabilidad	S/. 96	S/. 126	S/. 30

Tabla 32: Promedio semestral de los indicadores de las campañas del producto Tarjeta de Crédito
Fuente: Elaboración propia

- **Para el producto PrestaBono:**

Indicadores	Reglas de Experto						Modelo NBO					
	Ene-21	Feb-21	Mar-21	Abr-21	May-21	Jun-21	Jul-21	Ago-21	Sep-21	Oct-21	Nov-21	Dic-21
Nro de Leads	22,874	25,645	27,020	22,586	23,624	22,327	21,730	25,131	26,479	22,585	22,679	21,880
Nro De Ventas	3,241	5,577	5,441	4,713	5,031	3,190	4,111	4,943	4,983	5,400	4,143	4,111
Efectividad	14.17%	21.75%	20.14%	20.87%	21.30%	14.29%	18.92%	19.67%	18.82%	23.91%	18.27%	18.79%
Rentabilidad	S/. 96	S/. 90	S/. 86	S/. 94	S/. 90	S/. 91	S/. 109	S/. 112	S/. 123	S/. 105	S/. 124	S/. 101
Desembolso (en miles)	S/. 65,530	S/. 113,938	S/. 116,764	S/. 106,792	S/. 119,763	S/. 70,486	S/. 103,297	S/. 121,440	S/. 119,009	S/. 134,374	S/. 98,264	S/. 100,197

Tabla 33: Resultados de las campañas comerciales del año 2021 del producto PrestaBono
Fuente: Elaboración propia

Indicadores	Promedio de Indicadores		Incremento
	Reglas de Experto (6 Primeros Meses 2021)	Modelo NBO (6 Últimos Meses 2021)	
Nro de Leads	24,013	23,414	
Nro De Ventas	4,532	4,615	83
Efectividad	18.75%	19.73%	0.98%
Rentabilidad	S/. 91	S/. 112	S/. 21
Desembolso	S/. 98 MM	S/. 112 MM	S/ 13 MM

Tabla 34: Promedio semestral de los indicadores de las campañas del producto PrestaBono
Fuente: Elaboración propia

- **Para el producto Préstamo Vehicular**

Indicadores	Reglas de Experto						Modelo NBO					
	Ene-21	Feb-21	Mar-21	Abr-21	May-21	Jun-21	Jul-21	Ago-21	Sep-21	Oct-21	Nov-21	Dic-21
Nro de Leads	29,540	28,485	31,520	28,550	24,845	27,728	28,653	27,060	32,150	27,693	24,844	28,282
Nro De Ventas	4,693	4,170	5,563	4,279	3,736	4,128	6,263	6,402	7,185	6,488	4,588	5,905
Efectividad	15.89%	14.64%	17.65%	14.99%	15.04%	14.89%	21.86%	23.66%	22.35%	23.43%	18.47%	20.88%
Rentabilidad	S/. 92	S/. 106	S/. 107	S/. 105	S/. 110	S/. 85	S/. 113	S/. 128	S/. 139	S/. 115	S/. 101	S/. 132
Desembolso (en miles)	S/. 107,334	S/. 88,750	S/. 133,529	S/. 96,842	S/. 87,908	S/. 90,156	S/. 149,504	S/. 164,768	S/. 183,419	S/. 161,104	S/. 117,448	S/. 149,881

Tabla 35: Resultados de las campañas comerciales del año 2021 del producto Préstamo Vehicular
Fuente: Elaboración propia

Indicadores	Promedio de Indicadores		Incremento
	Reglas de Experto (6 Primeros Meses 2021)	Modelo NBO (6 Últimos Meses 2021)	
Nro de Leads	28,445	28,114	
Nro De Ventas	4,428	6,139	1,710
Efectividad	15.52%	21.78%	6.26%
Rentabilidad	S/. 101	S/. 121	S/. 20
Desembolso	S/. 100 MM	S/. 154 MM	S/. 53 MM

Tabla 36: Promedio semestral de los indicadores de las campañas del producto Préstamo Vehicular
Fuente: Elaboración propia

- **Para el producto Préstamo Hipotecario:**

Indicadores	Reglas de Experto						Modelo NBO					
	Ene-21	Feb-21	Mar-21	Abr-21	May-21	Jun-21	Jul-21	Ago-21	Sep-21	Oct-21	Nov-21	Dic-21
Nro de Leads	20,724	19,632	19,244	17,391	20,584	20,071	20,516	18,650	19,821	17,738	20,172	20,874
Nro de Ventas	445	301	311	285	351	364	477	435	358	371	385	416
Efectividad	21.51%	15.38%	16.21%	16.41%	17.06%	18.15%	23.28%	23.34%	18.08%	20.95%	19.12%	19.95%
Rentabilidad	S/. 109	S/. 101	S/. 90	S/. 86	S/. 106	S/. 104	S/. 115	S/. 118	S/. 144	S/. 109	S/. 102	S/. 140
Desembolso (en miles)	S/. 457,538	S/. 422,528	S/. 424,151	S/. 410,253	S/. 432,252	S/. 428,984	S/. 511,572	S/. 507,259	S/. 484,211	S/. 493,350	S/. 497,730	S/. 504,133

Tabla 37: Resultados de las campañas comerciales del año 2021 del producto Préstamo Hipotecario
Fuente: Elaboración propia

Indicadores	Promedio de Indicadores		Incremento
	Reglas de Experto (6 Primeros Meses 2021)	Modelo NBO (6 Últimos Meses 2021)	
Nro de Leads	19,608	19,629	
Nro De Ventas	343	407	64
Efectividad	17.45%	20.79%	3.33%
Rentabilidad	S/. 99	S/. 141	S/. 42
Desembolso	S/. 429 MM	S/. 499 MM	S/ 70 MM

Tabla 38: Promedio semestral de los indicadores de las campañas del producto Préstamo Hipotecario
Fuente: Elaboración propia

- **Para el producto Descuento por Planilla:**

Indicadores	Reglas de Experto						Modelo NBO					
	Ene-21	Feb-21	Mar-21	Abr-21	May-21	Jun-21	Jul-21	Ago-21	Sep-21	Oct-21	Nov-21	Dic-21
Nro de Leads	10,832	10,217	17,144	12,522	12,646	11,310	10,398	10,217	17,315	12,772	13,278	10,744
Nro de Ventas	1,945	1,688	2,688	2,082	2,687	2,015	2,480	2,049	3,379	2,642	2,658	2,543
Efectividad	17.96%	16.53%	15.68%	16.63%	21.25%	17.82%	23.86%	20.06%	19.52%	20.69%	20.02%	23.67%
Rentabilidad	S/. 104	S/. 103	S/. 110	S/. 106	S/. 85	S/. 101	S/. 101	S/. 105	S/. 107	S/. 102	S/. 142	S/. 125
Desembolso (en miles)	S/. 39,577	S/. 34,268	S/. 64,060	S/. 44,580	S/. 59,772	S/. 41,827	S/. 58,178	S/. 49,285	S/. 79,873	S/. 67,865	S/. 63,654	S/. 61,118

Tabla 39: Resultados de las campañas comerciales del año 2021 del producto Descuento Por Planilla
Fuente: Elaboración propia

Indicadores	Promedio de Indicadores		Incremento
	Reglas de Experto (6 Primeros Meses 2021)	Modelo NBO (6 Últimos Meses 2021)	
Nro de Leads	12,445	12,454	
Nro De Ventas	2,184	2,625	441
Efectividad	17.65%	21.30%	3.66%
Rentabilidad	S/. 102	S/. 114	S/ 12
Desembolso	S/. 47 MM	S/. 63 MM	S/ 15 MM

Tabla 40: Promedio semestral de los indicadores de las campañas del producto Descuento por Planilla
Fuente: Elaboración propia

- **Análisis de los resultados del modelo NBO**

La Tabla 41 muestra el incremento de la efectividad, de la rentabilidad y del desembolso, debido a la utilización del modelo NBO en la elaboración de las estrategias de Marketing que participan en el despliegue de las campañas comerciales de seis productos financieros correspondientes a los seis últimos meses del año 2021.

Producto	Reglas de Experto (6 Primeros Meses 2021)			Modelo NBO (6 Últimos Meses 2021)			Incremento		
	Efect.	Rent.	Desemb.	Efect.	Rent.	Desemb.	Efect.	Rent.	Desemb.
Libre Disponibilidad	16.79%	S/. 98	S/. 86 MM	20.44%	S/. 121	S/. 117 MM	3.65%	S/. 23	S/ 31 MM
Tarjeta de Crédito	19.50%	S/. 96	-	20.71%	S/. 136	-	1.21%	S/. 30	-
PrestaBono	18.75%	S/. 91	S/ 98 MM	19.73%	S/. 112	S/ 112 MM	0.98%	S/. 21	S/ 14 MM
Préstamo Vehicular	15.52%	S/. 101	S/ 100 MM	21.78%	S/. 121	S/ 154 MM	6.26%	S/. 20	S/ 54 MM
Préstamo Hipotecario	17.45%	S/. 99	S/ 429 MM	20.79%	S/. 141	S/ 499 MM	3.33%	S/. 42	S/ 70 MM
Descuento por Planilla	17.65%	S/. 102	S/ 47 MM	21.30%	S/. 114	S/ 63 MM	3.66%	S/. 12	S/ 15 MM

Tabla 41: Resumen del incremento de los indicadores analizados – campaña comercial del año 2021-
Fuente: Elaboración propia

La efectividad se define de la siguiente manera:

$$Efectividad = \frac{N^0\ de\ Ventas}{N^0\ de\ Leads\ Gestionados}$$

Como se observa en la Tabla 41 la efectividad de las campañas comerciales se ha incrementado en cada uno de los seis productos analizados; este incremento se debe a que el número de ventas ha aumentado debido a que la organización ha ofertado el producto que el mercado demanda.

El modelo NBO no solo selecciona el producto con la mayor probabilidad de adquisición; sino también, el producto que genera la mayor rentabilidad. Esta característica del modelo se demuestra en la Tabla 41.

La Tabla 41 muestra que el desembolso, monto total de las colocaciones expresadas en soles, se ha incrementado debido a la utilización del modelo NBO. El desembolso es directamente proporcional al número de ventas y al valor de éstas.

La capacidad de generalización de los modelos predictivos va disminuyendo con el tiempo. Por tanto, es necesario llevar un control mensual de la calidad de estos modelos y recalibrarlos en el caso de que sea necesario. En algunos casos estos modelos necesitan ser rediseñados.

Capítulo 5

En este capítulo se describen las principales conclusiones obtenidas de la utilización del modelo analítico NBO en las campañas de Marketing de los productos financieros que la organización oferta.

5.1 Conclusiones

- Esta investigación permite responder las siguientes preguntas:
  - ¿Cuáles son los impulsores e impedimentos para implementar el análisis predictivo en marketing?

    La creciente necesidad de comprender el comportamiento del mercado debido al incremento de la competencia y a la cada vez mayor exigencia de los clientes, lleva a las organizaciones a implementar estrategias de marketing centradas en el cliente. Para lograr este objetivo, las organizaciones financieras construyen sistemas de análisis de la información (procesamiento de los datos) y sistemas de utilización de modelos analíticos (modelos predictivos); que permitan incrementar la efectividad y la rentabilidad de las campañas comerciales. Los hallazgos de esta investigación refuerzan la centralidad en el cliente y permite a las organizaciones desenvolverse eficientemente en un entorno cambiante.

  - ¿Cómo gestionar el uso general y el desarrollo de un modelo analítico que es utilizado simultáneamente por múltiples unidades de negocio?

    Las actividades de marketing basadas en el comportamiento del cliente dependen en gran medida de la inteligencia empresarial, debido a esto, es necesario una estrecha cooperación entre los encargados de construir las estrategias de marketing y los encargados de administrar las Tecnologías de Información (IT, por sus siglas del inglés). Para conseguirlo, es necesario que la organización posea una sólida cultura de datos y que exista una estrecha colaboración entre la inteligencia empresarial, el marketing, el sistema de ventas y el uso del BI. Cuando se logra esta articulación, la construcción y la gestión de los modelos analíticos, entre ellos el modelo NBO (caso de estudio), sirven para optimizar los resultados de la organización.

- Para conseguir un resultado óptimo con la aplicación del modelo NBO, es necesario conseguir la mayor generalización (AUC) en el cálculo de las probabilidades de adquisición de cada producto; esto se consigue comparando el resultado obtenido al utilizar diferentes algoritmos de *Machine Learning* (Maquina Leve de Gradiente Ascendente, Aumento de Gradiente Extremo, Maquina de Soporte Vectorial, etc.) y seleccionando el mejor para cada producto.

- Se estandarizó las probabilidades (obtenidas al desarrollar los modelos de propensión), utilizando dos métodos:
  - Modelo ensamblado Bagging.
  - Redes Neuronales (Inteligencia Artificial)

  Al comparar el resultado de la estandarizacion de las probabilidades mediante estos dos métodos, el modelo ensamblado demostró tener mayor capacidad de generalización.

- Para incrementar la capacidad de generalización de la red neuronal se utilizó otro método de estandización denominado Red Neuronal Recurrente, está tiene la capacidad de retroalimentarse con las probabilidades de salida obtenidas en cada iteración del entrenamiento. El tiempo de procesamiento se incremento considerablemente, en tanto que la capacidad de generalización de la Red no fue sustantiva; por tal motivo este método se descartó

- El modelo NBO demuestra que es capaz de mejorar la efectividad y la rentabilidad de las campañas comerciales. El estudio se realizó para 17 productos financieros entre activos, pasivos, y de servicios; se implementó para las campañas comerciales de los siguientes productos: Libre Disponibilidad, Tarjeta de Crédito, PrestaBono, Préstamo Vehicular, Préstamo Hipotecario y Descuento por Planilla. Estas campañas se realizaron en el segundo semestre del año 2021, obteniendo un incremento respecto del primer semestre del mismo año de la efectividad, de la rentabilidad, y del desembolso. Esto demuestra que la utilización del modelo NBO en la construcción de estrategias para el despliegue de las campañas comerciales de los seis productos financieros, permite un salto de calidad respecto a la utilización de las reglas de experto en la elaboración de estas estrategias.

- El output del modelo NBO es una función de priorización que es el producto aritmético de las probabilidades estandarizadas y de la rentabilidad del proceso; debido a esto, los modelos estadísticos utilizados para calcular la rentabilidad deben ser permanenmente actualizados ya que los cambios del entorno en la que se implementan las estrategias de la campaña de marketing afectan la rentabilidad del proceso.

Referencias

[1] Sleep, S., Hulland, J., & Gooner, R.A. (2019). THE DATA HIERARCHY: factors influencing the adoption and implementation of data-driven decision mak- ing. AMS Review. pp. 1–19. Retrieved from doi.org/10.1007/s13162-019- 00146-8.

[2] Adams, J., Raeside, R., & Khan, H. T. A. (2014). Research Methods for Business and Social Science Students (Second edition). New Delhi: Sage Publications Pvt. Ltd.

[3] Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems, 22(3), pp. 48–55. Retrieved from doi.org/10.1109/MIS.2007.58.

[4] Ana María Huayna, Vanessa Calvo Huaraz, Juan Carlos Huiman Sánchez (2010). Modelo de evaluación de créditos financieros basado en Redes Neuronales orientado a EDPYMES. Universidad Nacional Mayor de San Marcos

[5] Guest Blog (2017). Imbalanced Data: How to handle imabalanced classification problems.

[6] Carlos Serrano Cinca y José Luis Gallizo Larraz (2001). Las Redes Neuronales Artificiales en el tratamiento de la información financiera. Facultad de Ciencias Económicas y Empresariales, Universidad Zaragoza.

[7] Junming Zhang (2010). Analysis of Neural Network on bank Marketing data. College of Computer Science – ANU.

[8] Stone, M. D., & Woodcock, N. D. (2014). Interactive, direct and digital marketing. A future that depends on better use of business intelligence. Journal of re- search in interactive marketing, 8(1), pp. 4–17. Retrieved from doi.org/10.1108/JRIM-07-2013-0046.

[9] Van Capelleveen, G., Amrit, C., Yazan, D. M., & Zijm. H. (2019). The recommender canvas: A model for developing and documenting recommender system design. Expert systems with applications, 129, pp. 97-117. Retrieved from doi.org/10.1016/j.eswa.2019.04.001.

[10] Andrés Mauricio Mendoza Espinoza (2014). Modelos de Clasificación en el Otorgamiento de Creditos Financieros: Comparación entre diferentes Técnicas de Machine Learning. Universidad de los Andes – Colombia.

[11] Ali Aydın Koç and Özgür Yeniay (2013). A comparative study of Artificial Neuronal Networks and Logistic Regression for classification of Marketing Campaign Results. Hacettepe University - Department of Statistics and Mathematics.

[12] Manuel Córdova Zamora (2008), Libro de Estadística Aplicada, Primera Edición.

[13] Abbass Ziad (2018). Implementing a bank sales analytics solution and a predictive model for the next best offer, Universidad Nova de Lisboa.

[14] Pavlo Delias, Athanasios Lagopoulos, Grigorio Tsoumakas, Daniela Grigor (2017). Using Multi-Target Feature Evaluation to discover factors that Affect Business Process Behavior. Eastern Macedonia and Thrace Institute of Technology, Kavala, Greece.

[15] Barton, D., & Court, D. (2012). Making advanced analytics work for you. Harvard Business Review, 90(10), pp. 78–83. Retrieved from search-proquestcom. ezproxy.jyu.fi/docview/1113988190.

[16] Carlos Serrano Cinca y José Luis Gallizo Larraz (2001). Redes Neuronales Recurrentes que optimicen los modelos de propensión aplicados en la Industria Financiera. Facultad de Ciencias Económicas y Empresariales, Universidad Zaragoza.

[17] Fredy Pérez Ramírez y Horacio Fernández Castaño (2007). Redes Neuronales y la evaluación del Riesgos de Crédito, Revista de Ingeniería, Universidad de Medellín.

[18] Astudillo, Gustavo, Saenz, Cecilia 2018. Sistemas Ensambladores de Objetos de Aprendizaje. Facultad de Ciencias Exactas y Naturales, Universidad Nacional de la Plata.

Anexos

Anexo 1: Conceptos Teóricos

- Lead. – Personas que pueden ser clientes o no clientes del banco, a las cuales se les ofrece un producto financiero; son seleccionados para participar en las campañas comerciales por el **Gerencia de Riesgos.**
- Modelo de Propensión. - Es un modelo estadístico que permite calcular la probabilidad de que un lead adquiera un producto financiero.
- Modelo Ensamblados. - Es un modelo que calcula la probabilidad de que ocurra un evento, a partir de la predicción de modelos unitarios.
- Estandarización de Probabilidades. – Estandarización y ajuste de las probabilidades obtenidas de los modelos estadísticos. Este proceso se realiza utilizando una red neuronal o modelos ensamblados.
- Modelo de Rentabilidad. - Es un modelo analítico que calcula la rentabilidad promedio esperada en caso que el cliente adquiera el producto.
- Modelo NBO. - Es un modelo analítico que determina el combo de productos que se le debe ofrecer a un cliente según su propensión y la rentabilidad esperado hacia el producto.
- Campaña comercial. – Es un conjunto de procesos cuya finalidad es colocar un producto financiero en el mercado. Los procesos son:
  - Generación de un base de datos preliminar para definir un conjunto de personas (leads) a los cuales se les debe ofertar un producto financiero.
  - Calcular la probabilidad (propensión) de adquisición de un producto, para cada uno de los leads que intervienen en la campaña.
  - Calcular la rentabilidad esperada en el caso de que un producto sea adquirido por un lead.
  - Construcción de estrategias comerciales que permitan convertir en realidad lo planeado en la campaña comercial; es decir, seleccionar un sub conjunto de leads con mayor propensión y rentabilidad a los cuales debe ir dirigida la campaña (actualmente este proceso se realiza según criterios empíricos de negocio, el objetivo del proyecto es darle un sustento matemático).
  - Análisis de los ratios de conversión de la campaña, es decir, determinar la cantidad de ventas por producto.

Anexo 2: Código en Python

PREPROCESAMIENTO DE LOS DATOS

```
# Eliminar variables con alto porcentaje de valores missing (>80%)
nrows = df_final_consolidado_rent.shape[0]
columns = df_final_consolidado_rent.columns
df_missing_value = pd.DataFrame()
matriz_ayuda = []
for var in columns:
    num_missing_value = df_final_consolidado_rent[var].isnull().sum()
    if (num_missing_value/nrows > 0.75):
        matriz_ayuda.append([var, num_missing_value/nrows])
df_missing_value = pd.DataFrame(matriz_ayuda,columns = ['Variable', 'Por_Missing_Value'])
df_missing_value = df_missing_value.sort_values(by='Por_Missing_Value', ascending = False)
df_missing_value.head(10)
```

```
# Actualizar los valores nulos de las variables
var_excluir = list(df_missing_value['Variable'])
nrows = df_final_consolidado_rent.shape[0]
columns = df_final_consolidado_rent.columns
matriz_ayuda = []
for var in columns:
    if var not in var_excluir:
        num_missing_value = df_final_consolidado_rent[var].isnull().sum()
        tipo_variable = ''
        if str(df_final_consolidado_rent[var].dtypes) == 'object':
            tipo_variable = 'Categorica'
            df_final_consolidado_rent[var] = df_final_consolidado_rent[var].replace(np.nan, 'NI')
        else:
            tipo_variable = 'Numerica'
            #df_final_consolidado_rent[var] = df_final_consolidado_rent[var].replace(np.nan, 0)
        if (num_missing_value > 0):
            matriz_ayuda.append([var, num_missing_value/nrows, tipo_variable])
df_imputar_missing_value = pd.DataFrame(matriz_ayuda,columns = ['Variable', 'Por_Missing_Value', 'Tipo_Variable'])
df_imputar_missing_value = df_imputar_missing_value.sort_values(by='Por_Missing_Value', ascending = False)
df_imputar_missing_value.head(10)
```

Imputacion KNN

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import KNNImputer
%matplotlib inline

x_train = df_final_consolidado_rent[['CantidadHijos','Edad']]
n=5000
m=0
max_iter = int(x_train.shape[0]/n)
columnasOriginales = x_train.columns
dfImputado = pd.DataFrame()

while(m<max_iter):
    if m==max_iter:
        x_prob = x_train.iloc[n*m:x_train.shape[0],:]
    else:
        x_prob = x_train.iloc[n*m:n*(m+1),:]
    imputer = KNNImputer(n_neighbors=10)
    x1= imputer.fit_transform(x_prob)
    df1=pd.DataFrame(x1)
    dfImputado = dfImputado.append(df1,ignore_index=True)
    m=m+1

dfImputado_2 = pd.DataFrame(np.array(dfImputado), columns = ['CantidadHijos','Edad'])
df_final_consolidado_rent['CantidadHijos'] = dfImputado_2['CantidadHijos']
df_final_consolidado_rent['Edad'] = dfImputado_2['Edad']
```

SELECCIÓN DE VARIABLES PRINCIPALES

```
# Seleccion de Variables usando Random Forrest
from sklearn.ensemble import RandomForestRegressor
x_train = df_copy[lista_variables_finales]
y_train = df_copy['Rent_LD']
x_train = x_train.fillna(0)
y_train = y_train.fillna(0)
rf = RandomForestRegressor(n_estimators = 10)
rf.fit(x_train,y_train)
f_i = list(zip(lista_variables_finales,rf.feature_importances_))
f_i.sort(key = lambda x : x[1], reverse = True)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=10, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

```
def genrar_ginis_individuales(df_input,var_target,var_numericas, var_no_numericas):
    df = df_input.copy() df['Flag_Obj'] = df[var_target]
    N = df.shape[0] V = df['Flag_Obj'].sum()
    gini = 0 df_ginis = pd.DataFrame(columns = ['Tipo_Variable','Variable','Gini'])
    columns = set(set(var_numericas)|set(var_no_numericas))
    for var in columns:
        if var in var_numericas and var != var_target:
            if len(df[var].unique()) >= 20:
                df = Categorize(df,var,5)
                new_var = 'Q_' + var
                df_quintiles = df.groupby(new_var).agg({'Id_Cliente': 'count', 'Flag_Obj': 'sum'})
                df_quintiles = df_quintiles.rename(columns = {'Id_Cliente':'Cantidad'})
                df_quintiles = df_quintiles.rename(columns = {'Flag_Obj':'Ventas'})
                df_quintiles['SumE'] = df_quintiles['Cantidad'].cumsum()
                df_quintiles['SumV'] = df_quintiles['Ventas'].cumsum()
                df_quintiles['p'] = df_quintiles['SumE']/N
                df_quintiles['q'] = df_quintiles['SumV']/V
                df_quintiles['diff_pq'] = abs(df_quintiles['p'] - df_quintiles['q'])
                gini = df_quintiles['diff_pq'].sum()/df_quintiles['q'].sum()
                df_ginis = df_ginis.append({"Tipo_Variable":  'Numerica' ,"Variable": var,"Gini" : gini}, ignore_index = True)
            else:
                df_quintiles = df.groupby(var).agg({'Id_Cliente': 'count', 'Flag_Obj': 'sum'})
                df_quintiles = df_quintiles.rename(columns = {'Id_Cliente':'Cantidad'})
                df_quintiles = df_quintiles.rename(columns = {'Flag_Obj':'Ventas'})
                df_quintiles['SumE'] = df_quintiles['Cantidad'].cumsum()
                df_quintiles['SumV'] = df_quintiles['Ventas'].cumsum()
                df_quintiles['p'] = df_quintiles['SumE']/N
                df_quintiles['q'] = df_quintiles['SumV']/V
                df_quintiles['diff_pq'] = abs(df_quintiles['p'] - df_quintiles['q'])
                gini = df_quintiles['diff_pq'].sum()/df_quintiles['q'].sum()
                df_ginis = df_ginis.append({"Tipo_Variable":  'Numerica' ,"Variable": var,"Gini" : gini}, ignore_index = True)
        elif var in var_categoricas and var != var_target:
            df_quintiles = df.groupby(var).agg({'Id_Cliente': 'count', 'Flag_Obj': 'sum'})
            df_quintiles = df_quintiles.rename(columns = {'Id_Cliente':'Cantidad'})
            df_quintiles = df_quintiles.rename(columns = {'Flag_Obj':'Ventas'})
            df_quintiles['SumE'] = df_quintiles['Cantidad'].cumsum()
            df_quintiles['SumV'] = df_quintiles['Ventas'].cumsum()
            df_quintiles['p'] = df_quintiles['SumE']/N
            df_quintiles['q'] = df_quintiles['SumV']/V
            df_quintiles['diff_pq'] = abs(df_quintiles['p'] - df_quintiles['q'])
            gini = df_quintiles['diff_pq'].sum()/df_quintiles['q'].sum()
            df_ginis = df_ginis.append({"Tipo_Variable":  'Categorica' ,"Variable": var,"Gini" : gini}, ignore_index = True)
    df_ginis = df_ginis.sort_values(by = 'Gini', ascending = False)
    return df_ginis
```

```
    dfImputado = dfImputado.append(df1,ignore_index=True)
    m=m+1

dfImputado_2 = pd.DataFrame(np.array(dfImputado), columns = ['CantidadHijos','Edad'])
df_final_consolidado_rent['CantidadHijos'] = dfImputado_2['CantidadHijos']
df_final_consolidado_rent['Edad'] = dfImputado_2['Edad']
```

ENTRENAMIENTO DE LOS MODELOS PREDICTIVOS

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import ElasticNet
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
import joblib
```

```
def get_models():
    models = list()
    nombre_models = list()
    models.append(LinearRegression())
    nombre_models.append('LinearRegression')
    models.append(ElasticNet())
    nombre_models.append('ElasticNet')
    models.append(DecisionTreeRegressor())
    nombre_models.append('DecisionTreeRegressor')
    models.append(KNeighborsRegressor())
    nombre_models.append('KNeighborsRegressor')
    models.append(AdaBoostRegressor())
    nombre_models.append('AdaBoostRegressor')
    models.append(BaggingRegressor(n_estimators=10))
    nombre_models.append('BaggingRegressor')
    models.append(RandomForestRegressor(n_estimators=10))
    nombre_models.append('RandomForestRegressor')
    models.append(ExtraTreesRegressor(n_estimators=10))
    nombre_models.append('ExtraTreesRegressor')
    return models, nombre_models
```

```
lista_productos = list(Tabla_Vars_Seleccionadas['Var_Target'].unique())
df_input = df_final_consolidado_rent.copy()
models, nombre_models = get_models()
df_models_rentabilidad = pd.DataFrame(columns = ['Producto', 'Modelo', 'R2'])
for var_target in lista_productos:
    nombre_archivo_modelo = 'Modelo_'
    variables_seleccionadas = list(Tabla_Vars_Seleccionadas[Tabla_Vars_Seleccionadas['Var_Target'] == var_target]['Variable'])
    vars_in = variables_seleccionadas
    df_copy = df_input.copy()
    train_x, valid_x, train_y, valid_y = train_test_split(df_copy[vars_in],
                                                          df_copy[var_target],
                                                          test_size=0.3,
                                                          shuffle=True,
                                                          random_state=123)

    print(var_target)
    r2_maximo = 0
    model_optimo = None
    nombre_model_optimo = ''
    i=0
    for model in models:
        model.fit(train_x, train_y)
        y_pred = model.predict(valid_x)
        r2 = r2_score(valid_y, y_pred)
        if r2 >= r2_maximo:
            r2_maximo = r2
            model_optimo = model
            nombre_model_optimo = nombre_models[i]
        #print(nombre_models[i])
        i=i+1
    df_models_rentabilidad = df_models_rentabilidad.append({'Producto': var_target, 'Modelo': nombre_model_optimo,
                                                            'R2': r2_maximo},ignore_index = True)
    nombre_archivo_modelo = nombre_archivo_modelo + var_target + '.pkl'
    joblib.dump(model_optimo, nombre_archivo_modelo)
```

ESTANDARIZACIÓN DE PROBABILIDADES (REDES NEURONALES)

```
#Transformacion Box Cox para mejorar los resultados de la Red Neuronal
import tensorflow as tf
import keras
from tensorflow.python.keras.models import Sequential
from keras.layers import Dropout, Dense, Embedding, LSTM, SpatialDropout1D
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import PowerTransformer
from sklearn.model_selection import train_test_split

df_copy = df_final_consolidadd[['Prob_TC','Prob_XL','Prob_CD','Prob_LD','Prob_PA','Prob_DXP']].copy()
features = df_copy.copy()
#instatiate
pt = PowerTransformer(method='box-cox', standardize=True,)
#Fit the data to the powertransformer
skl_boxcox = pt.fit(features)
#Lets get the Lambdas that were found
#print (skl_boxcox.lambdas_)
calc_lambdas_bc = skl_boxcox.lambdas_
#Transform the data
skl_boxcox = pt.transform(features)
#Pass the transformed data into a new dataframe
df_features = pd.DataFrame(data=skl_boxcox, columns = ['Prob_TC', 'Prob_XL', 'Prob_CD', 'Prob_LD', 'Prob_PA', 'Prob_DXP'] )
# Pass to the original dataframe the transform columns
df_copy.drop(['Prob_TC'], axis=1, inplace=True)
df_copy.drop(['Prob_XL'], axis=1, inplace=True)
df_copy.drop(['Prob_CD'], axis=1, inplace=True)
df_copy.drop(['Prob_LD'], axis=1, inplace=True)
df_copy.drop(['Prob_PA'], axis=1, inplace=True)
df_copy.drop(['Prob_DXP'], axis=1, inplace=True)
# Concatenar ambos dataframes
df_boxcox = pd.concat([df_copy, df_features], axis=1)
```

```
# Entrenamiento de la Red Neuronal
x_train = df_boxcox[['Prob_TC','Prob_XL','Prob_CD','Prob_LD','Prob_PA','Prob_DXP']]
y_train = df_Affluent_v2[['Flag_Apertura_TC','Flag_Apertura_XL','Flag_Apertura_CD',
                          'Flag_Apertura_LD','Flag_Apertura_PA','Flag_Apertura_DXP']]
x_train_1, x_test_1, y_train_1, y_test_1 = train_test_split(x_train, y_train, train_size=0.80, random_state=0)

#### Modelo de Redes Neuronales
## Probar Drop Out en cada
model = Sequential()
# Input put Layer and First Hidden Layer
model.add(Dense(40, input_dim=x_train_1.shape[1], activation='sigmoid'))
model.add(Dense(30, activation ='tanh', kernel_initializer = 'uniform'))
model.add(Dense(20, activation ='sigmoid', kernel_initializer = 'uniform'))
model.add(Dense(10, activation ='tanh', kernel_initializer = 'uniform'))
model.add(Dense(10, activation ='sigmoid', kernel_initializer = 'uniform'))

# Out put layer
model.add(Dense(y_train_1.shape[1], activation='sigmoid'))


# compile model
model.compile(
              loss='binary_crossentropy',
              optimizer='Adam',
              metrics=['AUC']
             )

# fit the model
model.fit(x_train_1, y_train_1, epochs=100, batch_size=32)

# evaluate the model
train_scores = model.evaluate(x_test_1, y_test_1, verbose=0)
train_scores
```

```
#predicciones de la red Neuronal
y_pred_RN = model.predict(x_train)
print(y_pred_RN.shape)
print("Cantidad de Probs < 0 en el DataFrame original: ", (y_pred_RN <= 0.0).astype(float).sum(axis=0))
y_pred_train = y_pred_RN
y_pred_train

#DataFrame Final con predicciones de la red neuronal
df_ypred = pd.DataFrame(y_pred_train, columns = ["ProbTC_AJUSTADA",
                                                 "ProbXL_AJUSTADA",
                                                 "ProbCD_AJUSTADA",
                                                 "ProbLD_AJUSTADA",
                                                 "ProbPA_AJUSTADA",
                                                 "ProbDXP_AJUSTADA" ,
                                                 ])
df_final = pd.merge(df_Affluent, df_ypred, how="left", left_index=True, right_index=True)
df_final.head(10)
```

ESTANDARIZACIÓN DE PROBABILIDADES (MODELOS ENSAMBLADOS)

```
def get_models():
    models = list()
    models.append(XGBClassifier(
                  max_depth=4,
                  learning_rate=0.1,
                  n_estimators=80,
                  objective='binary:logistic',
                  eval_metric = 'logloss',
                  booster='gbtree'))
    models.append(LGBMClassifier(
                    n_estimators=80,
                    learning_rate=0.015,
                    boosting_type= 'gbdt',
                    objective='binary',
                    colsample_bytree=.8,
                    subsample=.8,
                    max_depth=4))
    models.append(GradientBoostingClassifier(
                                            n_estimators=80,
                                            criterion='friedman_mse',
                                            learning_rate=0.1,
                                            max_depth=3,
                                            min_samples_leaf=1,
                                            min_samples_split=2,
                                            min_weight_fraction_leaf=0.0
                                           ))
    models.append(SVC(
                      max_iter=100,
                      degree=3,
                      probability=True,
                      decision_function_shape='multinomial',
                      gamma='scale',
                      kernel='sigmoid'
                  ))
```

```
from sklearn.model_selection import KFold
from numpy import hstack
from numpy import vstack
from numpy import asarray
# collect out of fold predictions form k-fold cross validation
def get_out_of_fold_predictions(X, y, models, num_splits):
    meta_X, meta_y = list(), list()
    # define split of data
    kfold = KFold(n_splits=num_splits, shuffle=True)
    # enumerate splits
    for train_ix, test_ix in kfold.split(X):
        fold_yhats = list()
        # get data
        train_X, test_X = X.iloc[train_ix,], X.iloc[test_ix,]
        train_y, test_y = y.iloc[train_ix,], y.iloc[test_ix,]
        meta_y.extend(test_y)
        # fit and make predictions with each sub-model
        for model in models:
            model.fit(train_X, train_y)
            yhat = model.predict_proba(test_X)[:,1]
            # store columns
            fold_yhats.append(yhat.reshape(len(yhat),1))
        # store fold yhats as columns
        meta_X.append(hstack(fold_yhats))
    return vstack(meta_X), asarray(meta_y)
```

```
import statsmodels.formula.api as smf
import statsmodels.api as sm

# fit all base models on the training dataset
def fit_base_models(X, y, models):
    for model in models:
        model.fit(X, y)

# fit a meta model
def fit_meta_model(X, y):
    X = pd.DataFrame(X,columns=['XGB','LGBM','GBC','SVC','RFC'])
    y = pd.DataFrame(y,columns=['Target'])
    num_variables = len(np.array(X.columns))
    var_target = y.columns[0]
    formula = str(var_target) + '~'
    i=1
    for col in X.columns:
        if i<num_variables:
            formula = formula + col + " + "
        else:
            formula = formula + col
        i=i+1

    X[var_target] = y
    model = smf.glm(formula = formula, data = X, family=sm.families.Binomial()).fit()
    return model
```

```
# evaluate a list of models on a dataset
from sklearn.metrics import roc_auc_score
def evaluate_models(X, y, models):
    for model in models:
        yhat = model.predict_proba(X)[:,1]
        # Calculando el score AUC
        AUC = roc_auc_score(np.asarray(y),yhat)
        print('%s: AUC %.3f' % (model.__class__.__name__, AUC))
```

```
# make predictions with stacked model
def super_learner_predictions(X, models, meta_model):
    meta_X = list()
    i=0
    for model in models:
        yhat = model.predict_proba(X)[:,1]
        meta_X.append(yhat.reshape(len(yhat),1))
        i=i+1
    meta_X = hstack(meta_X)
    meta_X = pd.DataFrame(meta_X,columns=['XGB','LGBM','GBC','SVC','RFC'])
    # predict
    return meta_model.predict(meta_X)
```

```
from sklearn.model_selection import train_test_split
X = df_v2.iloc[:,2:19]
var_target = 'Flag_Apertura_SEG_AC'
y = y_train[var_target]
X, X_val, y, y_val = train_test_split(X, y, test_size=0.50)
print('Train', X.shape, y.shape, 'Test', X_val.shape, y_val.shape)
```

```
Train (28110, 17) (28110,) Test (28110, 17) (28110,)
```

```
import warnings
warnings.filterwarnings("ignore")
# get models
models = get_models()
# get out of fold predictions
meta_X, meta_y = get_out_of_fold_predictions(X, y, models, 5)
print('Meta ', meta_X.shape, meta_y.shape)
# fit base models
fit_base_models(X, y, models)
# fit the meta model
meta_model = fit_meta_model(meta_X, meta_y)
# evaluate base models
evaluate_models(X_val, y_val, models)
# evaluate meta model
yhat = super_learner_predictions(X_val, models, meta_model)
print('Super Learner: AUC %.3f' % (roc_auc_score(np.asarray(y_val),yhat)))
print('Se busca predecir: ' + var_target)
```

```
Meta  (28110, 5) (28110,)
XGBClassifier: AUC 0.821
LGBMClassifier: AUC 0.807
GradientBoostingClassifier: AUC 0.816
SVC: AUC 0.303
RandomForestClassifier: AUC 0.793
Super Learner: AUC 0.813
Se busca predecir: Flag_Apertura_SEG_AC
```

```
import joblib
nombre_models = ''
nombre_meta_model = ''
prod = "Seguro Accidentes Personales"
if prod == "Tarjetas":
    nombre_models = 'Models_Tarjeta.pkl'
    nombre_meta_model = 'Meta_Model_Tarjeta.pkl'

if prod == "Xtralinea":
    nombre_models = 'Models_Xtralinea.pkl'
    nombre_meta_model = 'Meta_Model_Xtralinea.pkl'

if prod == "Compra Deuda TC":
    nombre_models = 'Models_CD_Tarjeta.pkl'
    nombre_meta_model = 'Meta_Model_CD_Tarjeta.pkl'

if prod == "Libre Disponibilidad":
    nombre_models = 'Models_LD.pkl'
    nombre_meta_model = 'Meta_Model_LD.pkl'

if prod == "PrestaBono":
    nombre_models = 'Models_PA.pkl'
    nombre_meta_model = 'Meta_Model_PA.pkl'

if prod == "Descuento Planilla":
    nombre_models = 'Models_DXP.pkl'
    nombre_meta_model = 'Meta_Model_DXP.pkl'

if prod == "Prestamo Vehicular":
    nombre_models = 'Models_VEH.pkl'
    nombre_meta_model = 'Meta_Model_VEH.pkl'
```

CONSTRUCCION DEL MODELO NEXT BEST OFFER

Modelo NBO (Proceso Final)

```
df_dataset_NBO = pd.DataFrame()
df_dataset_NBO = pd.merge(df_probabilidades_ajustadas, df_rentabilidades, how="left", left_index=True, right_index=True)

campos_probabilidades = ["ProbTC_AJUSTADA", "ProbXL_AJUSTADA", "ProbCD_AJUSTADA", "Prob_LD_AJUSTADA", "Prob_PA_AJUSTADA",
                         "ProbDXP_AJUSTADA", "ProbVEH_AJUSTADA", "ProbHIP_AJUSTADA", "ProbFree_AJUSTADA", "ProbPow_AJUSTADA",
                         "ProbSC_AJUSTADA", "ProbCS_AJUSTADA", "ProbFM_AJUSTADA", "ProbDEP_AJUSTADA",
                         "ProbFR_AJUSTADA", "ProbDE_AJUSTADA", "ProbAC_AJUSTADA" ]

campos_rentabilidades = ['Rent_TC_Esperada','Rent_XL_Esperada','Rent_CD_Esperada','Rent_LD_Esperada','Rent_PA_Esperada',
                         'Rent_DXP_Esperada', 'Rent_VEH_Esperada','Rent_HIP_Esperada','Rent_FREE_Esperada','Rent_POW_Esperada',
                         'Rent_SC_Esperada','Rent_CS_Esperada','Rent_FM_Esperada','Rent_DP_Esperada','Rent_SEG_FR_Esperada',
                         'Rent_SEG_AC_Esperada','Rent_SEG_DE_Esperada']

campos_priorudad      = ['Prioridad_TC','Prioridad_XL','Prioridad_CD','Prioridad_LD','Prioridad_PA',
                         'Prioridad_DXP', 'Prioridad_VEH','Prioridad_HIP','Prioridad_FREE','Prioridad_POW',
                         'Prioridad_SC','Prioridad_CS','Prioridad_FM','Prioridad_DEP','Prioridad_SEG_FR',
                         'Rent_SEG_AC_Esperada','Rent_SEG_DE_Esperada']
productos_financieros = ['Tarjeta de Credito', 'InstaCash','Compra de Deuda','Libre Disponibilidad','Presta Bono',
                         'Descuento Planilla','Prestamo Vehicular','Prestamo Hipotecario','Cuenta Free','Cuenta Power',
                         'Super Cuenta','Cuenta Sueldo','Fondos Mutuos','Deposito a Plazo','Seguro Fraude TC',
                         'Seguro Accidentes Personales','Seguro Desempleo']
df_resultados_prioridad = pd.DataFrame(columns = ['Id_Cliente','Valor_Prioridad','Producto'])
for i in range(16):
    var_prioridad = campos_priorudad[i]
    var_probabilidad = campos_probabilidades[i]
    var_rentabilidad = campos_rentabilidades[i]
    producto = productos_financieros[i]
    df_dataset_NBO[var_prioridad] = df_dataset_NBO[var_probabilidad] * df_dataset_NBO[var_rentabilidad]
    df_filtrado = df_dataset_NBO[['Id_Cliente',var_prioridad]]
    df_filtrado.rename(columns={var_prioridad: 'Valor_Prioridad'}, inplace=True)
    df_filtrado['Producto'] = producto
    df_resultados_prioridad = pd.concat([df_resultados_prioridad, df_filtrado])
df_dataset_NBO.head(10)
```



```
df_resultados_prioridad['rank'] = df_resultados_prioridad.groupby('Id_Cliente')['Valor_Prioridad'].rank(method='first')
for i in range(17):
    var = 'Prior_' + str(i+1)
    df_filtrado = df_resultados_prioridad[df_resultados_prioridad['rank'] == (16-i)]
    df_filtrado = df_filtrado[['Id_Cliente','Producto']]
    df_dataset_NBO = pd.merge(df_dataset_NBO, df_filtrado, how="left", on  = 'Id_Cliente')
    df_dataset_NBO.rename(columns={'Producto': var}, inplace=True)
df_dataset_NBO[['Id_Cliente', 'Prior_1', 'Prior_2', 'Prior_3', 'Prior_4', 'Prior_5', 'Prior_7', 'Prior_8', 'Prior_9', 'Prior_10',
```

	Id_Cliente	Prior_1	Prior_2	Prior_3	Prior_4	Prior_5	Prior_7	Prior_8	Prior_9	Prior_10	Prior_11	Prior_12
0	0	Libre Disponibilidad	Prestamo Hipotecario	Compra de Deuda	Tarjeta de Credito	Presta Bono	Seguro Fraude TC	InstaCash	Fondos Mutuos	Cuenta Power	Cuenta Sueldo	Cuenta Free
1	1	Super Cuenta	Presta Bono	Compra de Deuda	InstaCash	Tarjeta de Credito	Seguro Fraude TC	Prestamo Hipotecario	Fondos Mutuos	Descuento Planilla	Cuenta Free	Cuenta Power
2	2	Compra de Deuda	Super Cuenta	Prestamo Hipotecario	Tarjeta de Credito	InstaCash	Libre Disponibilidad	Fondos Mutuos	Presta Bono	Descuento Planilla	Seguro Fraude TC	Cuenta Power
3	3	Compra de Deuda	Presta Bono	Libre Disponibilidad	Tarjeta de Credito	Super Cuenta	Prestamo Hipotecario	Seguro Fraude TC	Fondos Mutuos	Descuento Planilla	Cuenta Free	Deposito a Plazo
4	4	InstaCash	Cuenta Free	Compra de Deuda	Tarjeta de Credito	Super Cuenta	Descuento Planilla	Prestamo Hipotecario	Seguro Fraude TC	Presta Bono	Fondos Mutuos	Cuenta Sueldo