

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE
SISTEMAS



TESIS
“APLICACIÓN DE MODELOS ANALÍTICOS PARA LA
OPTIMIZACIÓN DE LA ASIGNACIÓN DE LOS PRODUCTOS EN
UNA ENTIDAD FINANCIERA”

PARA OBTENER EL TÍTULO PROFESIONAL DE

INGENIERO DE SISTEMAS

ELABORADO POR:

EDMUNDO DE ELVIRA MORI ORRILLO

ASESOR:

DR. GLEN DARIO RODRIGUEZ RAFAEL

LIMA - PERÚ

2022

DEDICATORIA

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.

AGRADECIMIENTO

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, personalizada en la Facultad de Ingeniería Industrial y de Sistemas y en la Facultad de Ciencias, se ha encargado de proveerme la sólida formación académica que poseo. A mis profesores de estas dos facultades mi sincero agradecimiento.

La calidad académica y profesional del doctor Glen Rodríguez, mi asesor de tesis, ha hecho posible la exitosa finalización de esta tesis. 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

Esta investigación busca optimizar la asignación de productos a los leads que participan en 6 campañas comerciales correspondientes a 6 productos financieros (Tarjeta de Crédito (TC), Insta Cash (XL), Compra de Deuda (CD), Libre Disponibilidad (LD), Presta Bono (PA) y Descuento por Planilla (DXP)) y determinar el canal óptimo para ofrecer estos productos. Se utilizan 2 canales de venta (Red de Agencias (RED) y Call Center (CALL)).

Las variables utilizadas para la asignación de productos son la probabilidad de adquisición y la rentabilidad. La optimización se realiza construyendo una función objetivo sujeta a restricciones de negocio (Programación Lineal Entera), esta función objetivo está formada por 12 variables de decisión $X_{i,j,k}$; donde i es el lead, j es la campaña comercial y k es el canal de venta, es decir $j \in \{ TC, XL, CD, LD, PA, DXP \}$ y $k \in \{ RED, CALL \}$. Los coeficientes de la función objetivo son:

- Potencial de cada campaña comercial.
- Porcentaje promedio de la gestión en el canal (RED o CALL).
- Porcentaje promedio de la contactabilidad en el canal (RED o CALL).
- Probabilidad de adquisición de un producto financiero.
- Rentabilidad del producto financiero.
- Costos de gestión por canal de cada lead asignado.

Las restricciones de la función objetivo están en función de la máxima cantidad de productos a ofertar a un lead y de la capacidad de gestión de los canales de venta.

La probabilidad de adquisición de un producto se obtiene a partir de un Modelo Predictivo y la estandarización de estas probabilidades se realiza usando el concepto de Redes Neuronales.

La rentabilidad esperada es el promedio de la rentabilidad generada por los clientes que poseen el producto financiero, luego de doce meses de haberlo adquirido.

La investigación consta de dos grandes etapas:

- Etapa descriptiva. - Se efectúa el planteamiento del problema, para lo cual se define los objetivos de la investigación y se realiza un análisis de la literatura y de los conceptos teóricos que participan en la solución del problema.
- Etapa aplicativa. - Se desarrolla la solución propuesta, es decir, se construyen los modelos predictivos para calcular las probabilidades de adquisición del producto, se construye una red neuronal para estandarizar estas probabilidades y se usa el concepto de Programación Lineal Entera para encontrar la asignación óptima de los leads por producto y por canal, maximizando así la rentabilidad y la efectividad de las campañas comerciales. Finalmente, se utiliza una librería de Python para encontrar la solución óptima global del problema de programación lineal entera.

También se puede usar el concepto de algoritmo genético para encontrar una **solución óptima local** del problema de optimización. Al poner en práctica estas dos soluciones, la solución óptima global ha demostrado generar mejores resultados de rentabilidad y efectividad en las campañas comerciales.

ABSTRACT

This research seeks to optimize the allocation of products to the leads that participate in 6 commercial campaigns corresponding to 6 financial products (Credit Card (TC), Insta Cash (XL), Debt Purchase (CD), Free Availability (LD), Loan Bonus (PA) and Payroll Discount (DXP)) and determine the optimal channel to offer these products. 2 sales channels are used (Agencies Network (RED) and Call Center (CALL)). The variables used for the allocation of products are the probability of acquisition and profitability. The optimization is done by building an objective function subject to business restrictions (Integer Linear Programming), this objective function is made up of 12 decision variables $X_{(i,j,k)}$; where i is the lead, j is the commercial campaign and k is the sales channel, that is, $j \in \{ TC, XL, CD, LD, PA, DXP \}$ and $k \in \{ RED, CALL \}$.

The coefficients of the objective function are:

- Potential of each commercial campaign.
- Average percentage of channel management (RED or CALL).
- Average percentage of channel contactability (RED or CALL).
- Probability of acquisition a financial product.
- Profitability of the financial product.
- Management costs per channel for each assigned lead.

The restrictions of the objective function are based on the maximum number of products to offer to a lead and the ability to manage sales channels.

The probabilities of acquiring a product is obtained from the Predictive Models and for the their standardization use the concept of Neural Networks.

The expected profitability's the average of the profitability generated by the clients who have the financial product, after twelve months of having acquired.

The research that will be developed in this thesis consists of two main stages:

- Descriptive stage. - The problem statement is made, for which the objectives of the research are defined, an analysis of the existing literature and of the theoretical concepts that participate in the solution of the problem is carried out.
- Application stage. - The proposed solution is developed, that is, predictive models are built to calculate the probabilities of product acquisition, a neural network is built to standardize these probabilities and the concept of Integer Linear Programming is used to find the optimal allocation of leads by product and by channel, thus maximizing the profitability and effectiveness of commercial campaigns; a Python library is used to find the global optimal solution of the integer linear programming problem.

The concept of a genetic algorithm can also be used to find a local optimal solution of the optimization problem. By putting these two solutions into practice, the global optimal solution has been shown to generate better profitability and effectiveness results in commercial campaigns.

PRÓLOGO

En el sistema financiero, debido a la gran cantidad de datos que constantemente se genera y se procesa, así como a la complejidad y a la tasa exponencial de crecimiento de estos, es posible extraer información precisa que genere conocimiento para la toma de decisiones, si somos capaces de construir modelos analíticos utilizando los conceptos de: Programación Lineal Entera, Aprendizaje Automático y de Inteligencia Artificial. Estos modelos se utilizan para construir estrategias de marketing que permitan ofrecer a los leads un conjunto de productos acorde con sus necesidades. El análisis de la información es la piedra angular del desarrollo de todos los campos de la actividad económica; por lo que 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. Sin embargo, la mayoría de las empresas modernas, específicamente las pequeñas y medianas, sólo están interesadas en tener potentes capacidades de almacenamiento con fines estadísticos; sin tener en cuenta que el aprendizaje automático entendido como una ciencia de los algoritmos les permite extraer patrones (minería de datos) y realizar análisis predictivos mediante la utilización de modelos analíticos.

Para incrementar la productividad de las Campañas de Marketing y construir estrategias para la retención y reactivación de los clientes, es necesario construir modelos analíticos que reemplacen a los métodos empíricos que actualmente son utilizados en el área de Inteligencia Comercial de las empresas del sector financiero.

ÍNDICE

DEDICATORIA	I
AGRADECIMIENTO	II
RESUMEN.....	III
ABSTRACT	V
PRÓLOGO	VII
ÍNDICE.....	VIII
LISTA DE FIGURAS	X
LISTA DE TABLAS	XII
ÍNDICE DE ACRONIMOS	
XV CAPÍTULO I	
INTRODUCCIÓN	1
1.1. MARCO CONTEXTUAL	1
1.2. DESCRIPCIÓN DE LA PROBLEMÁTICA	1
1.3. FORMULACIÓN DEL PROBLEMA	3
1.4. OBJETIVOS	3
1.4.1. OBJETIVO GENERAL	3
1.4.2. OBJETIVOS ESPECÍFICOS	3
1.5. JUSTIFICACIÓN	4
1.6. ANTECEDENTES DE LA INVESTIGACION BIBLIOGRÁFICA	5
CAPÍTULO II	13
MARCO TEÓRICO	13
2.1. CONCEPTOS PREVIOS.....	13
2.2. MARKETING FINANCIERO	15
2.2.1. MARKETING CENTRADO EN EL PRODUCTO	16
2.2.2. MARKETING CENTRADO EN EL CLIENTE	17
2.2.3. ESTRATEGIAS DE MARKETING	18
2.3. MODELOS PREDICTIVOS	20
2.3.1. APRENDIZAJE SUPERVISADO	21
2.3.2. APRENDIZAJE NO SUPERVISADO	31
2.3.3. APRENDIZAJE POR REFORZAMIENTO	36
2.4. REDES NEURONALES	39
2.4.1. CONCEPTOS Y DESCRIPCIÓN	39
2.4.2. ARQUITECTURA DE LOS SISTEMAS NEURONALES ARTIFICIALES	40
2.4.3. APRENDIZAJE DE LA RED. ALGORITMO DE RETROPROPAGACION	41
2.4.4. ANALISIS DE SENSIBILIDAD	42
2.4.5. APLICACIÓNES DE LAS REDES NEURONALES	44
2.5. PROGRAMACIÓN LINEAL ENTERA	45
2.5.1. CONCEPTOS GENERALES	45
2.5.2. METODO DE RAMIFICACIÓN Y ACOTAMIENTO	46

2.5.3. METODO DE ENUMERACIÓN IMPLICITA	48
2.5.4. ALGORITMO DE PLANO DE CORTE	
50	
2.6. ALGORITMOS GENÉTICOS	52
CAPÍTULO III	
53 DESARROLLO DEL TRABAJO DE INVESTIGACIÓN	53
3.1. FASE I: CENTRALIZACION Y EXPLOTACION DE LOS DATOS	53
3.1.1. LEVANTAMIENTO DE LA INFORMACIÓN	53
3.1.2. CONSOLIDACIÓN DE LA INFORMACIÓN	57
3.1.3. INFORMACIÓN HISTORICA DE LA GESTIÓN DE CAMPAÑAS	58
3.1.4. ANALISIS DE LA RENTABILIDAD ESPERADA.....	60
3.2. FASE II: DESARROLLO DE LOS MODELOS ANALÍTICOS	61
3.2.1. CONTRUCCIÓN DE LOS MODELOS DE PROPENSIÓN	61
3.2.2. ESTANDARIZACIÓN DE PROBABILIDADES	72
3.2.3. VALIDACION DE LOS RESULTADOS DE LA RED NEURONAL	73
3.3. FASE III: OPTIMIZACIÓN DE LA ASIGNACIÓN DE LEADS	74
3.3.1. PLANTEAMIENTO MATEMÁTICO	74
3.3.2. SOLUCION DEL PROBLEMA USANDO PROGRAMACION LINEAL ENTERA 77	
3.3.3. SOLUCION DEL PROBLEMA MEDIANTE ALGORITMOS GENÉTICOS	80
CAPÍTULO IV	83
ANÁLISIS Y VALIDACION DE RESULTADOS	83
CONCLUSIONES.....	94
RECOMENDACIONES	97
REFERENCIAS BIBLIOGRÁFICAS	99
ANEXO 01: PRUEBAS DE LOS MODELOS ESTADISTICOS	102
ANEXO 02: CODIGO PYTHON	105

LISTA DE FIGURAS

- Figura 1: Demandas, componentes, capacidades y resultados potenciales de marketing centrado en el cliente. Fuente: The Data Hierarchy: Factors influencing the adaptation and implementation of data driven decisión in Marketing. By Sleep, Hulland & Gooner, 2019.....21
- Figura 2: Método de mínimos cuadrados. Fuente: Estadística Aplicada, Manuel

Córdova Zamora, 2008.....	24
Figura 3: Grafica de la función logística. Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018.....	27
Figura 4: Grafica de un árbol de decisiones. Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018.....	29
Figura 5: Gráfica del método KMeans. Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018.....	34
Figura 6: Conjunto de datos por agrupar. Fuente: Clustering Jerárquico, Departamento de Ciencias de la Computación e IA, Universidad de Granda, por Fernando Berzal 2018.....	38
Figura 7: Dendrograma del agrupamiento jerárquico. Fuente: Clustering Jerárquico, Departamento de Ciencias de la Computación e IA, Universidad de Granda, por Fernando Berzal 2018.....	38
Figura 8: Estructura de una Red Neuronal. Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018.....	42
Figura 9: Región Factible para una PE y su relajación de la PL. Fuente: Investigación de Operaciones, por Wayne L. & Winston, Cuarta Edición.....	48
Figura 10: Modelo de Datos de las Campañas de Marketing. Fuente: Institución Financiera, elaboración propia.....	59
Figura 11: Leads por producto y Canal. Fuente: Institución Financiera, elaboración propia.....	61
Figura 12: % Gestión por producto y Canal. Fuente: Institución Financiera, elaboración propia.....	61
Figura 13: % Contactabilidad por producto y Canal. Fuente: Institución Financiera, elaboración propia.....	61
Figura 14: % Efectividad por producto y Canal. Fuente: Institución Financiera, elaboración propia.....	61
Figura 15: Diagrama de Flujo para el Desarrollo de un Modelo Analítico. Fuente: Institución Financiera, elaboración propia.....	63
Figura 16: Esquema gráfico de los valores atípicos. Fuente: Institución Financiera, elaboración propia.....	66
Figura 17: Imputación de valores atípicos. Fuente: Institución Financiera, elaboración propia.....	67
Figura 18: Matriz de correlaciones de las variables seleccionadas. Fuente:	

Institución Financiera, elaboración propia.....	71
Figura 19: Curva ROC del modelo LightGBM. Fuente: Institución Financiera, elaboración propia.....	72
Figura 20: Código en Lenguaje Python para el entrenamiento de una Red Neuronal. Fuente: Institución Financiera, elaboración propia.....	74
Figura 21: Diagrama de Flujo de una Algoritmo Genético. Fuente: Institución Financiera, elaboración propia.....	82
Figura 22: Graficas que muestran la calidad de los modelos estadísticos. Fuente: Institución Financiera, elaboración propia.....	105
Figura 23: Métricas Recall y Precisión de los modelos estadísticos. Fuente: Institución Financiera, elaboración propia.....	106

LISTA DE TABLAS

Tabla 1: Comparación de la precisión de los Modelos de Clasificación. Fuente: Modelos de Clasificación en el Otorgamiento de Créditos Financieros, por Andrés Mauricio Mendoza, 2014.....	8
Tabla 2: Métricas usadas en el agrupamiento jerárquico. Fuente: Big Data analytics and the transformation of Marketing, Journal of Business Research, by Erevelles S., Fukawa N. & Swayne L. 2016.....	37
Tabla 3: Ejemplo para el Análisis de Sensibilidad de una Red Neuronal. Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018.....	45
Tabla 4: Factibilidad de un nodo que satisfaga una restricción dada. Fuente: Investigación de Operaciones, por Wayne L. & Winston, Cuarta	

Edición.....	51
Tabla 5: Tablero Óptimo para la relajación de la PL. Fuente: Investigación de Operaciones, por Wayne L. & Winston, Cuarta Edición.....	52
Tabla 6: Variables Demográficas para perfilar Clientes. Fuente: Institución Financiera, elaboración propia.....	56
Tabla 7: Variables Transaccionales. Fuente: Institución Financiera, elaboración propia.....	57
Tabla 8: Variables Internas del Banco. Fuente: Institución Financiera, elaboración propia.....	57
Tabla 9: Variables del Sistema Financiero. Fuente: Institución Financiera, elaboración propia.....	58
Tabla 10: Variables de la Campaña Comercial. Fuente: Institución Financiera, elaboración propia.....	58
Tabla 11: Indicadores de Seguimiento de las Campañas de Marketing. Fuente: Institución Financiera, elaboración propia.....	60
Tabla 12: Rentabilidad por Producto Financiero. Fuente: Institución Financiera, elaboración propia.....	62
Tabla 13: Rentabilidad en nuevos soles por Producto y Perfil financiero. Fuente: Institución Financiera, elaboración propia.....	62
Tabla 14: Variables del DataSet para el Desarrollo de los Modelos. Fuente: Institución Financiera, elaboración propia.....	64
Tabla 15: Variables con alto porcentaje de valores nulos. Fuente: Institución Financiera, elaboración propia.....	65
Tabla 16: Variables con valores nulos a imputar. Fuente: Institución Financiera, elaboración propia.....	66
Tabla 17: Transformación de variables categóricas. Fuente: Institución Financiera, elaboración propia.....	68
Tabla 18: Target del modelo predictivo. Fuente: Institución Financiera, elaboración propia.....	69
Tabla 19: Las 20 variables independientes con mayor índice Gini. Fuente: Institución Financiera, elaboración propia.....	71
Tabla 20: Tabla de las variables Seleccionadas. Fuente: Institución Financiera, elaboración propia.....	71
Tabla 21: Comparación de AUCs por algoritmo. Fuente: Institución Financiera, elaboración propia.....	72
Tabla 22: Validación del modelo LightGBM. Fuente: Institución Financiera, elaboración propia.....	73
Tabla 23: Deciles del Modelo LightGBM. Fuente: Institución Financiera, elaboración propia.....	73

Tabla 24: Ginis de los modelos Estadísticos vs Ginis de la Red Neuronal. Fuente: Institución Financiera, elaboración propia.....	75
Tabla 25: Muestra del Data Set que contiene la información de los leads. Fuente: Institución Financiera, elaboración propia.....	80
Tabla 26: Muestra de la Solución Final usando Programación Lineal Entera. Fuente: Institución Financiera, elaboración propia.....	81
Tabla 27: Validación de la calidad de predicción de la red neuronal. Fuente: Institución Financiera, elaboración propia.....	85
Tabla 28: Estratificación de la base de leads correspondiente al segmento comercial “Beyond”. Fuente: Institución Financiera, elaboración propia.....	87
Tabla 29: Estratificación de la base de leads correspondiente al segmento comercial “Premium”. Fuente: Institución Financiera, elaboración propia.....	88
Tabla 30: Estratificación de la base de leads correspondiente al segmento comercial “Preferente”. Fuente: Institución Financiera, elaboración propia.....	89
Tabla 31: Estratificación de la base de leads correspondiente al segmento comercial “Personal”. Fuente: Institución Financiera, elaboración propia.....	90
Tabla 32: Estratificación de la base de leads correspondiente al segmento comercial “Estándar”. Fuente: Institución Financiera, elaboración propia.....	91
Tabla 33: Resultados de la implementación del modelo de optimización de la campaña Libre Disponibilidad. Fuente: Institución Financiera, elaboración propia...94 Tabla 34: Resultados de la implementación del modelo de optimización de la campaña Tarjetas de Crédito. Fuente: Institución Financiera, elaboración propia...94	
Tabla 35: Resultados de la implementación del modelo de optimización de la campaña Xtralinea. Fuente: Institución Financiera, elaboración propia.....94	
Tabla 36: Resultados de la implementación del modelo de optimización de la campaña Compra de Deuda. Fuente: Institución Financiera, elaboración propia....94	
Tabla 37: Resultados de la implementación del modelo de optimización de la campaña Presta Bono. Fuente: Institución Financiera, elaboración propia.....95	
Tabla 38: Resultados de la implementación del modelo de optimización de la campaña Descuento Planilla. Fuente: Institución Financiera, elaboración propia95	
Tabla 39: Matriz de Confusión. Fuente: Institución Financiera, elaboración propia.....104	
Tabla 40: Métricas Recall y Precisión de los modelos estadísticos. Fuente: Institución Financiera, elaboración propia.....106	

ÍNDICE DE ACRONIMOS

CRM Customer RelationShip Management

RRHH Recursos Humanos

SBS Súper Intendencia de Banca y Seguros

BA Business Analytics

BI Business Intelligence

TI Tecnologías de Información

SCE Suma de los Cuadrados de los Errores

SVM Support Vector Machine

ACP Análisis de Componentes Principales

AUC Area Under Curve

ROC Receiver Operating Characteristic

MSE Mean Square Error

XGBoost Xtreme Gradient Boost

LGBM Light Gradient Boosting Model **GBC**

Gradient Boosting Classifier.

SVC Support Vector Classifier

CAPÍTULO I

INTRODUCCIÓN

1.1. MARCO CONTEXTUAL

El mercado al cual va dirigido los productos que el sistema financiero oferta se caracteriza por ser inestable debido a las permanentes crisis a las cuales está sometido; esto obliga a las empresas del sector a diseñar estrategias de marketing que utilicen modelos predictivos para identificar a los clientes que son propensos a adquirir el producto. También es necesidad de las empresas determinar el canal óptimo de venta para ofertar los productos financieros.

Las empresas que van a utilizar estos modelos ofertan mensualmente un mínimo 20 productos financieros e implementan una Campaña de Marketing para colocarlos en el mercado; al término de éstas se lleva un control de las colocaciones y se las compara con la producción del resto del sistema financiero.

1.2. DESCRIPCIÓN DE LA PROBLEMÁTICA

En el contexto de la proliferación de tecnologías, canales de comercialización y dispositivos digitales, se ha multiplicado la demanda de los clientes por servicios más innovadores y adaptados a sus necesidades; como resultado de esto, la participación del cliente se ha convertido en uno de los principales factores de éxito para las organizaciones (Clow & Baack, 2016).

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 (Sleep & Hulland, 2019).

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 lo que es obligatorio que la infraestructura tecnológica dentro de las empresas esté fuertemente adaptada para respaldar un crecimiento continuo; además, la utilización del aprendizaje automático, entendido como una ciencia de los algoritmos, les permite 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 campañas de marketing basadas en el cliente; esta visión obliga a construir un modelo analítico (visión cliente) que a partir de modelos predictivos que proveen las probabilidades de adquisición estandarizadas utilizando redes neuronales (Inteligencia Artificial) y de variables numéricas de rentabilidad, gestión y contactabilidad, permiten construir una función objetivo que maximice la efectividad y la rentabilidad de estas campañas. Esta función objetivo está sujeta a restricciones del negocio, por ejemplo, la cantidad de leads (lead es la persona a la cual se le oferta un producto) asignados no puede superar el capacity de los canales de venta; además, un lead no puede ser enviado a dos canales diferentes para ofrecerles el mismo producto.

1.3. FORMULACIÓN DEL PROBLEMA

¿En cuánto aumentará los ratios de conversión y de rentabilidad de las campañas comerciales, a partir del uso de un modelo de Programación Lineal Entera que optimice la asignación de leads por producto y canal?

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Aplicar el concepto de Programación Línea Entera para construir una función objetivo sujeta a un conjunto de restricciones (éstas son ecuaciones e inecuaciones lineales), cuya solución permita incrementar los ratios de conversión y de rentabilidad de las campañas comerciales.

1.4.2. OBJETIVOS ESPECÍFICOS

- Actualmente las estrategias de marketing en el sector financiero se construyen en base a la experiencia empírica del analista de campañas comerciales, se busca utilizar el Data Analytics para optimizar las estrategias de marketing.
- En la actualidad, los modelos analíticos usados tienen poco poder de predicción, debido a su antigüedad y a las fluctuaciones del mercado, para resolver este problema desarrollaremos modelos predictivos para calcular la propensión de adquisición de cada producto financiero.
- El concepto de rentabilidad no interviene en el diseño de las campañas comerciales, debido a esto, en esta investigación el indicador rentabilidad es un coeficiente de la función objetivo del problema a maximizar.
- El concepto de **Redes Neuronales** se utiliza para estandarizar las probabilidades de adquisición obtenidas de los modelos estadísticos; de esta manera incrementamos la efectividad de las campañas comerciales.
- Para optimizar el resultado del proceso de marketing es necesario definir correctamente el producto a ofrecer y el canal de contacto a utilizar.

1.5. JUSTIFICACIÓN

La utilización de los modelos analíticos se está generalizando en las campañas de marketing con la finalidad de incrementar los ratios de conversación y aumentar la rentabilidad de este proceso comercial. Como afirman Sleep, Hulland y Gooner

(2019), el desarrollo continuo del marketing hacia modelos más centrados en el cliente, que enfatizan la toma de decisiones basada en datos y en los avances tecnológicos, así como el acceso de los profesionales del marketing a una cantidad y variedad de datos cada vez más complejos y en constante crecimiento exponencial (BIG DATA), permite optimizar la productividad de la campaña y construir estrategias para la retención y reactivación de los clientes, también identifica perfiles diferenciados dentro de la cartera de clientes mediante la operación de segmentación. Los perfiles obtenidos de la segmentación tienen características intrínsecas que permiten implementar acciones tácticas que coadyuvan a incrementar la relación con el cliente.

Campbell, Sands, Ferraro, Tsao y Mavrommatis (2019) han estudiado ampliamente las diversas posibilidades que tienen los especialistas en marketing para aprovechar la inteligencia artificial y el aprendizaje automático en la construcción de modelos analíticos que permiten construir estrategias y realizar actividades que conllevan a optimizar las operaciones de marketing. Las tecnologías del Big Data aplicadas al marketing han sido estudiadas por Erevelles, Fukawa y Swayne (2016), Mithas et al. (2013) y Bose (2008).

Estas tecnologías acompañadas con los aportes que da la Programación Lineal Entera conducen al éxito de las campañas de Marketing, logrando incrementar las ganancias de la organización.

1.6. ANTECEDENTES DE LA INVESTIGACIÓN BIBLIOGRÁFICA

EXPLOITING RESPONSE MODELS – OPTIMIZING CROSS – SELL AND UP – SELL OPPORTUNITIES IN BANKING.

SAS Intitute Inc. , Scotiabank Toronto Canada, 2015

Andrew Storey, Marc – David Cohen

El paradigma del marketing moderno consiste en seleccionar un conjunto de productos financieros, priorizados de acuerdo a su propensión de adquisición, que se le debe ofertar a los clientes en el momento adecuado; este objetivo es difícil de lograr debido a que las empresas financieras tienen múltiples productos y operan bajo un conjunto complejo de restricciones comerciales.

Elegir que productos se debe ofrecer a los clientes y en que canales de contactabilidad, con el fin de maximizar la rentabilidad del proceso de marketing, conlleva a utilizar los conceptos de Programación Lineal Entera, construyendo una función objetivo sujeta a restricciones de negocio; la solución de esta ecuación permite incrementar la efectividad de las campañas comerciales.

Una versión muy simplificada del problema se puede expresarse de la siguiente manera:

$$\text{Max} \sum_{ij} x_{ij} r_{ij}$$

Sujeto A:

$$\sum_{ij} x_{ij} c_{ij} \leq \text{Presupuesto de la Campaña}$$

$$\sum_i x_{ij} \geq \text{Minima cantidad de ofertas por producto } j$$

$$\sum_{ij} x_{ij} r_{ij} \geq \sum_{ij} x_{ij} c_{ij} (1 + R) \text{ (Rentabilidad mínimo esperado)}$$

Donde $x_{ij} = 1$ si al cliente “i” se le ofrece el producto “j”, r_{ij} es el beneficio esperado en caso el cliente “i” adquiera el producto “j”, c_{ij} es el costo incurrido al ofrecerle al cliente “i” el producto “j” y R es el porcentaje de ganancia mínima que se espera a final de la campaña comercial.

Es útil para comprender el marco de la solución comprender los datos que están disponibles para la planificación de campañas de marketing. Comprender los datos ayudará a que el problema sea más concreto.

El análisis de los datos de las Campañas de Marketing permite calcular los indicadores de control como la efectividad de la campaña, el porcentaje de gestión y contactabilidad de los canales de venta, la rentabilidad de la campaña, etc. La utilización de los conceptos de efectividad, rentabilidad, gestión y contactabilidad en el diseño del problema de optimización, permite acercar a este sistema a la realidad.

La calidad del resultado del modelo optimización dependerá del grado de precisión de las probabilidades de adquisición obtenidas de los modelos estadísticos; de esta condición nace la necesidad de utilizar el concepto de Redes Neuronales, para ajustar y estandarizar las propensiones de toma de la oferta financiera.

**OPTIMIZATION MODELS FOR TARGETED OFFERS IN DIRECT MARKETING:
EXACT AND HEURISTIC ALGORITHMS.**

European Journal of Operational Research

Fabrice Nobibon, Roel Leus, Frits C.R.Spieksma

Cuando el objetivo es construir un modelo de optimización para la selección de conjuntos de clientes que recibirán una oferta de uno o más productos durante una campaña de Marketing, es necesario construir modelos computacionales que se basan en algoritmos exactos y heurísticos; este artículo trata sobre la construcción de estos modelos; esta construcción es fuertemente NP-Hard (conjunto de los problemas de decisión que tienen una solución compleja). Una solución de algoritmos de aproximación de factores constantes no puede ser usada para resolver este problema. El artículo propone una formulación alternativa de cobertura de conjuntos y desarrolla un algoritmo Branch-and-Price para resolver este problema; describe también 8 heurísticas que permiten una aproximación a una solución óptima.

La solución óptima es utilizada en instancias pequeñas y medianas, mientras que la solución heurística se emplea en casos donde se analiza grandes volúmenes de datos y el factor tiempo es preponderante.

La solución óptima permite mejorar el beneficio económico de una empresa, ya sea adquiriendo nuevos clientes o generando ingresos adicionales. Reinartz señala que “cuando las empresas negocian entre los gastos de adquisición y los de retención, una asignación sub óptima de los gastos de retención tendrá un mayor impacto en la rentabilidad del cliente a largo plazo que los gastos de adquisición sub óptimos”.

MODELOS DE CLASIFICACIÓN EN EL OTORGAMIENTO DE CRÉDITOS FINANCIEROS: COMPARACIÓN ENTRE DIFERENTES TÉCNICAS DE MACHINE LEARNING Y MODELOS DE REGRESIÓN MULTIPLE.

Universidad de los Andes – Colombia

Andrés Mauricio Mendoza Espinoza, 2014

La investigación se basa en el análisis minucioso de diferentes modelos de credit scoring, planteando las ventajas y desventajas de cada uno de ellos y determinando, en base al análisis realizado, cuál debe ser la técnica óptima en la resolución del problema de otorgamiento de crédito.

El Python es el lenguaje utilizado en la realización del proyecto, las librerías que provee contienen algoritmos optimizados que disminuyen los tiempos de ejecución de los modelos durante el desarrollo del trabajo.

El análisis del otorgamiento del crédito utiliza modelos estadísticos de clasificación (regresión logística y el modelo probit) y modelos no estadísticos como las redes neuronales, AdaBoost y Support Vector Machine, se desarrolla el algoritmo de entrenamiento para cada una de las técnicas anteriormente mencionadas. Previamente se realiza un análisis descriptivo de las variables independientes cuantificando la relación que existe entre la variable dependiente e independiente.

Finalmente se comparan estos modelos usando métricas, como el error y el grado de precisión con un nivel de confianza del 95%. La Tabla 1 muestra el error y la precisión de los modelos, concluyendo que el modelo SVM Gaussiano es el seleccionado.

Tabla 1: Comparación de la precisión de los Modelos de Clasificación

	Error	Precisión (95% de confianza)	Complejidad	Overfitting
Logit	22,30%	3,52%	Baja	No
Redes Neuronales	23,30%	3,52%	Media	No
Adaboost	23,00%	3,52%	Media-Alta	Si
SVM Kernel gaussiano	20,77%	3,52%	Media-Alta	No
SVM Kernel polinomial	20,33%	3,52%	Alta	No

Fuente: Modelos de Clasificación en el Otorgamiento de Créditos Financieros, por Andrés Mauricio Mendoza, 2014

IMPLEMENTING A BANK SALES ANYTICS SOLUTION AND A PREDICTIVE MODEL FOR THE NEXT BEST OFFER

Universidad Nova de Lisboa

Ziad El Abbass, 2018

Una campaña de marketing es una acción que organiza una institución financiera con el objetivo de colocar un producto o servicio. También se puede realizar para fines que no estén relacionados con la venta cruzada, como campañas de marketing informativo u otras campañas utilizadas para aumentar la lealtad del cliente y evitar el desgaste del producto (campañas de retención).

Una campaña de marketing tiene más de un canal de contacto, dependiendo de la naturaleza del producto (Ziad El Abbass & Mauro Castelli, 2018).

Colocar un producto financiero en el mercado implica la construcción e implementación de estrategias de marketing cuyo objetivo es el conocimiento (awareness) del cliente, utilizando modelos de segmentación que identifiquen perfiles diferenciados de los leads que participan en las campañas comerciales. Estas estrategias van acompañadas por otras que coadyuvan a profundizar la relación con

el cliente para entender mejor sus necesidades. Existen otras campañas que buscan retener en la organización al cliente y reactivar su fidelidad.

ANALISIS PREDICTIVO: TÉCNICO Y MODELOS UTILIZADOS Y APLICACIONES DEL MISMO

Universitat Oberta de Catalunya

Carlos Espino Timón, 2017

El análisis predictivo es una disciplina del análisis de datos que usa técnicas de estadística, como el aprendizaje automático, para desarrollar modelos que predicen eventos futuros o conductas. Estos modelos predictivos permiten aprovechar los patrones (minería de datos) de comportamiento encontrados en los datos actuales e históricos para identificar riesgos y oportunidades.

El análisis predictivo se fundamenta en la identificación de relaciones entre variables en eventos pasados, para luego explotar dichas relaciones y predecir posibles resultados en futuras situaciones. Además, hay que tener en cuenta que la precisión de los resultados obtenidos depende mucho de cómo se ha realizado el análisis de los datos, así como de la calidad de las suposiciones (Carlos Espino, 2017).

El empobrecimiento de los mercados y el incremento de la competencia, ha traído consigo que el marketing tradicional (basado en el producto) sea desplazado por el marketing basado en el cliente.

ANALYTICS OF NEURAL NETWORK ON BANK MARKETING DATA

College of Computer Science - ANU

Junming Zhang, 2010

El empobrecimiento de los mercados y el crecimiento de la competencia, ha traído consigo que el Marketing Tradicional (basado en el producto) sea reemplazado por el Marketing basado en el cliente; en este proceso de cambio, la utilización de

herramientas, provistas por la inteligencia artificial (redes neuronales), permiten diseñar estrategias de Marketing que a partir de probabilidades de adquisición de una oferta financiera, se logra definir un conjunto de productos priorizado por su grado de aceptación, con esto se consigue optimizar el ratio de conversión de las campañas comerciales, incrementando así los ingresos de la organización. La estandarización de las probabilidades se realiza mediante la implementación de una Red Neuronal que utiliza como input las probabilidades obtenidas de los modelos estadísticos y como variable dependiente (output) múltiples salidas binarias correspondiente a cada producto financiero. La variable dependiente de cada producto financiero, toma los valores de “1” y “0”, si el cliente adquirió el producto tiene el valor “1” y el valor de “0” en caso contrario.

La etapa de entrenamiento de la red neuronal utiliza como input las probabilidades de los modelos estadísticos y como output las variables dependientes (cuyos valores son 1 y 0), una vez entrenada y validada la red neuronal se la utiliza para calcular las probabilidades estandarizadas utilizadas en la función objetivo.

A COMPARATIVE STUDY OF ARTIFICIAL NEURAL NETWORKS AND LOGISTIC REGRESSION FOR CLASSIFICATION OF MARKETING CAMPAIGN RESULTS.

Hacettepe University - Department of Statistics and Mathematics

Ali Aydın Koç and Özgür Yeniay, 2013

A partir de variables: edad, ingreso, grado instrucción, saldo en su cuenta de ahorro, deuda en el sistema financiero, cantidad de productos que tiene con el banco, etc. y usando el concepto de Redes Neuronales se calcula la probabilidad de que un cliente de una institución financiera adquiera un producto. Esta probabilidad también es calculada mediante el algoritmo de Regresión Logística.

Para las Redes Neuronales

Ventajas

Al tener mayor cantidad de parámetros (cantidad de capas escondidas, número de neuronas por capa, funciones de activación, técnicas de reducción para evitar el overfitting, etc.), las redes neuronales obtienen mejores resultados que otros métodos estadísticos.

Desventajas

Luego de cierta cantidad de iteraciones (épocas), la velocidad de ajuste disminuye drásticamente haciendo que el tiempo de procesamiento sea relativamente alto.

Para la Regresión Logística

Ventajas

Es uno de los modelos más fáciles de interpretar ya que las variables independientes (variables regresoras) tienen una relación directa o indirecta, dependiendo de su naturaleza, con la variable dependiente.

Desventajas

En los modelos de regresión logística las variables utilizadas no deben estar correlacionadas entre sí, esto limita la capacidad de precisión del modelo.

EXPLOITING RESPONSE MODELS – OPTIMIZING CROSS-SELL AND UP –SELL OPPORTUNITIES IN BANKING

ScienceDirect, Volume 29, Page 327 – 341

Andrew Storey and Mar David Cohen, 2008

El sector financiero implementa periódicamente campañas para mejorar la relación del cliente con la organización ofreciéndole nuevos productos. En los últimos años, esta forma de actuar ha ganado un impulso significativo debido a la creciente disponibilidad de información y a la mejora de las capacidades de análisis debido a la utilización del Big Data y de la minería de datos. Para maximizar el retorno de los recursos empleados en las campañas de marketing, las organizaciones financieras rediseñan sus procesos y mejoran continuamente el grado de predicción de sus modelos estadísticos. Ofertar a los leads de la organización un conjunto de productos

priorizados de acuerdo a su probabilidad de aceptación a través de un canal adecuado al perfil del cliente, maximiza la rentabilidad de las campañas de marketing. Se utilizan los conceptos de la Programación Lineal Entera para construir una función objetivo, sujeta a restricciones comerciales, que permita seleccionar un conjunto de leads a los cuales se les ofertará un producto financiero utilizando el mejor canal de contacto. La solución de este problema de optimización maximiza la rentabilidad, minimizando los costos de implementación de las campañas a través de los canales de venta, por lo tanto, es una herramienta eficaz para la ejecución de campañas tácticas y estratégicas.

CAPÍTULO II

MARCO TEÓRICO

2.1. CONCEPTOS PREVIOS

- Lead. – Personas que pueden ser clientes o no clientes del banco, a las cuales se les puede ofrecer un producto financiero; son seleccionados para participar en las campañas comerciales por el Gerencia de Riesgos.
- Canal de Venta. – Es el medio que utiliza la organización financiera para contactar al lead con la finalidad de ofrecerle un producto financiero; en el caso de estudio puede ser el Call Center o la Red de Agencias.
- Gestión de Leads. - Es un proceso mediante el cual el canal de ventas se comunica con los leads de la campaña comercial.

- Costo de la gestión de los leads. - Es el costo total en el que incurre el canal al contactar a los leads de las campañas comerciales, depende de la cantidad de leads a contactar y de la comisión por venta.
- Campaña Comercial. – Es un conjunto de procesos cuya finalidad es colocar un producto financiero en el mercado. Los procesos son:
 - Generación de una base de datos preliminar para definir un conjunto de personas (leads) a los cuales se les debe ofrecer 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 subconjunto 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 del feedback de la respuesta de cada lead contactado en la campaña comercial, con la finalidad de obtener los resultados de la contactabilidad.
 - Análisis de los ratios de conversión de la campaña, es decir, determinar la cantidad de ventas por producto.
- Indicadores de Campaña. – son valores que cuantifican el resultado de la campaña comercial, los principales son:
 - Porcentaje promedio de gestión de un canal. – Se calcula dividiendo la cantidad de leads que fueron gestionados por el canal entre la cantidad de leads enviados a dicho canal al inicio de mes.

- Porcentaje promedio de contactabilidad. - Se calcula dividiendo la cantidad de leads contactados entre la cantidad de leads gestionados, este indicador determina el nivel de contactabilidad dentro de la gestión de leads.
- Ratio de conversión. - Se calcula dividiendo la cantidad de colocaciones entre la cantidad de leads contactados, este indicador determina la efectividad de las campañas comerciales.
- Rentabilidad de un producto financiero. - Es el promedio de la rentabilidad generada por los clientes que tienen el producto financiero, luego de doce meses de haberlo adquirido.
- Modelo de optimización. – Modelo que permite determinar el producto que se le debe ofertar a un lead y definir el canal de contacto, sus componentes:
 - Función objetivo (F.O). - Modelo matemático a optimizar.
 - Restricciones. - Restricciones de Negocio a las cuales está sujeta la F.O.
 - Plano de Corte. - Algoritmo usado para encontrar la solución óptima global.

2.2. MARKETING FINANCIERO

Hoy en día, la mayoría de las interacciones con los clientes se llevan a cabo en canales digitales, generando continuamente cantidades significativas de datos que permiten a las empresas obtener una mejor percepción del cliente e integrarla para atraerlos a lo largo de su experiencia de compra (Hartman, 2014). Los datos de los perfiles del consumidor y de los clientes individuales se pueden capturar en tiempo real con la ayuda de tecnologías modernas, que han convertido a los consumidores en un generador constante de datos estructurados, transaccionales y de comportamiento. Luego, el marketing puede convertir los datos del comportamiento del consumidor en conocimiento al combinar los datos recopilados con la percepción humana para hacerlos efectivos, lo que eventualmente podría generar una ventaja de mercado. (Galvin, 2013; Erevelles, Fukawa y Swayne, 2016.).

Como resultado del cambio en el entorno empresarial y la demanda cambiante de los clientes, las organizaciones están invirtiendo en bases de datos que contengan información del lead. El análisis de esta información permite comprender, monitorear e influir en el comportamiento del cliente. Además, la utilización de nuevas tecnologías permite atraer nuevos consumidores, reducir los costos de gestión y realizar ventas cruzadas (Verhoef y Lemon, 2013.). Las organizaciones que pueden comprender las necesidades del consumidor mediante la creación de una vista única del cliente mediante la recopilación de datos de varias fuentes, pueden interactuar con el mercado eficientemente, por ejemplo, a través de una estrategia de análisis predictivo de la próxima mejor oferta (Deloitte MCS Limited, 2013, Sleep 2019).

Teniendo en cuenta los aportes anteriores, podemos definir al marketing financiero como una rama del marketing que se encarga de las campañas de publicidad, promoción e imagen de las empresas del sector, así como también del posicionamiento, la determinación de precios, diseño de canales de distribución de los productos y servicios que ofertan las entidades financieras.

2.2.1. MARKETING CENTRADO EN EL PRODUCTO

En tiempos pasados, las empresas utilizaban la publicidad para posicionar sus productos en el mercado y diseñaban sus políticas de precios basándose únicamente en sus costes de producción y en sus ganancias; esto les llevaba a implementar campañas de marketing centrándose únicamente en el producto ofertado; esta política les daba resultados en la medida en que el mercado no era competitivo y variado (existencia de monopolios).

En la medida en que la competencia crece en forma sostenida y los mercados se empobrecen, las organizaciones necesitan utilizar la información de los clientes, obtenida a partir del Big Data, para mejorar continuamente las actividades de marketing y, finalmente, innovar y diseñar nuevas formas de relacionarse con el cliente (Tellis, Prabhu y Chandy, 2009; Story, O'Malley y Hart, 2011; Erevelles, Fukawa y Swayne, 2016). El análisis sistemático de datos y la toma de decisiones basada en datos permiten a las organizaciones pasar de una estrategia de marketing centrada en el producto a una centrada en el cliente, esto permite construir relaciones sólidas y de mayor valor con el mercado (Lee et al. 2012; Deloitte MSC Limited, 2013; Leeflang et al., 2014).

2.2.2. MARKETING CENTRADO EN EL CLIENTE

Hoy en día, los consumidores esperan soluciones personalizadas para satisfacer sus necesidades de productos y servicios (Clow y Baack, 2016). Debido al constante cambio de las tendencias de consumo y de la estimativa de los consumidores, se prevé que retener clientes de alto valor se convierta en prioridad para las empresas, especialmente en la industria financiera (Deloitte MCS Limited, 2013). Por lo tanto, mejorar y enriquecer la experiencia del cliente a largo plazo se ha vuelto esencial para las organizaciones (Goldenberg, 2017).

El actual entorno económico, caracterizado por la incertidumbre y las cambiantes demandas de los clientes, conlleva a las organizaciones a implementar un marketing con una visión hacia una experiencia personalizada del cliente. Al mismo tiempo, la gestión rentable de clientes demanda optimizar el presupuesto de marketing. La inversión publicitaria se ha convertido en temas emergentes en las organizaciones. (Deloitte MCS Limited, 2013; Clow y Baack, 2016.) Por lo tanto, se requieren estrategias de crecimiento rentables que den como resultado una mayor satisfacción del cliente y una estrategia eficaz de análisis del cliente que convierta los conocimientos en crecimiento de las ventas (Teerlink & Haydock, 2012).

Según Sleep & Hulland (2019), uno de los mayores desafíos a los que se enfrenta el marketing es intentar implementar estrategias centradas en el cliente y, al mismo tiempo, manejar la información utilizando herramientas como el Big Data.

Las organizaciones que pueden aprovechar los datos y convertirlos en conocimientos del cliente, pueden lograr una mejor respuesta del servicio ofertado y, como resultado, pueden crear una ventaja competitiva (Manyika et al. 2011; Schroect, Shockley, Smart, Romero Morales y Tufan, 2012). Goldenberg (2017) afirman que la clave para lograr relaciones redituales, sostenidas y mejoradas a través del tiempo con los clientes, es interactuar con ellos y aprender continuamente de cada interacción. Por lo tanto, es necesario aplicar la analítica de negocios para generar ofertas recomendadas según el perfil del cliente.

Las campañas de Marketing visión cliente priorizan la estimativa y las necesidades del público objetivo, permitiendo ofertar productos financieros de acuerdo a las necesidades del cliente.

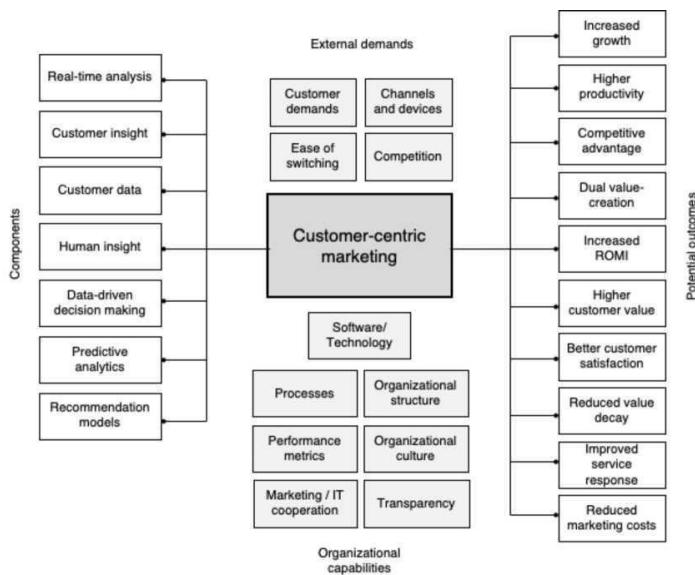
2.2.3. ESTRATEGIAS DE MARKETING

Según Sleep & Hulland (2019), 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 de clientes disponibles de diversas fuentes (sistemas CRM, transacciones, redes sociales, compras en línea e interacción cara a cara) (Teerlink & Haydock, 2012; Sleep & Hulland, 2019). Sin embargo, las organizaciones que pueden aprovechar los datos y convertirlos en conocimiento del cliente, pueden lograr una mejor respuesta del servicio y, como resultado, pueden crear una ventaja competitiva (Manyika et al. 2011; Schroect, Shockley, Smart, Romero Morales y Tufan, 2012).

Como se indicó, se espera que las actividades de datos y análisis, generen una ventaja competitiva y mejoren la experiencia del cliente. Por lo tanto, desarrollar e implementar estrategias basadas en datos se convertirá en un activo comercial cada vez más importante (Barton & Court, 2012; Brown & Gottlieb, 2016). Además, la alianza de herramientas tecnológicas y de marketing impulsa la toma de decisiones, mejora la productividad y la rentabilidad (Manyika et al. 2011; Schroeck et al., 2012). Stone & Woodcock, 2014; Sleep, Hulland & Gooner, 2019 sostienen que el éxito al pasar de una estrategia de marketing centrada en el producto a una estrategia centrada en el cliente, depende de la capacidad de la organización para conocer las necesidades de sus clientes a través de un análisis proactivo en tiempo real. Entre todos los tipos de estrategias de marketing, el emailing marketing ha demostrado ser el más efectivo año tras año, ya que este sirve para establecer un canal de comunicación a través del cual vamos alimentando la relación que existe entre la organización y sus clientes.

En la Figura 1 se resume: las demandas externas, los componentes, las capacidades organizativas y los resultados potenciales del marketing centrado en el cliente, conceptos derivados de la literatura de investigación. Las demandas externas para ejecutar una estrategia de marketing centrada en el cliente provienen tanto de los clientes como del entorno competitivo, los componentes de la ejecución de actividades de marketing se enumeran a la izquierda; las capacidades y requisitos organizativos para implementar y gestionar con éxito la estrategia de marketing se mencionan a continuación. Finalmente, los posibles resultados y beneficios de una estrategia de marketing se enumeran a la derecha.

Figura 1: Demandas, componentes, capacidades y resultados potenciales de marketing centrado en el cliente.



Fuente: The Data Hierarchy: Factors influencing the adaptation and implementation of data driven decisión in Marketing. By Sleep, Hulland & Gooner, 2019

Según las necesidades del entorno y de acuerdo con las características de las organizaciones financieras, 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 y mejorando, cada vez más, la atención al cliente.
- Estrategia de segmentación: Se trata de adecuar los productos financieros a cada segmento; se entiende por segmento a un conjunto de clientes con características comunes.

2.3. MODELOS PREDICTIVOS

Estos modelos analíticos predicen la probabilidad de adquisición de un producto financiero, tomando en cuenta variables que definen el perfil del cliente; las más importantes son: variables demográficas (edad, ingreso salarial, nivel socio económico, grado de instrucción y de digitalización, etc.), variables del sistema financiero (deuda total en el sistema, máxima línea de tarjeta de crédito, calificación

de la SBS, etc.), variables transaccionales (nivel de facturación con su tarjeta de crédito y débito, número y monto de transacciones hechas en agencia, etc.); existen otras variables, pero estas son las más importantes.

Los principales modelos predictivos son: Regresión Lineal Simple y Múltiple, Regresión Logística, Árbol de Decisiones, Modelos XGBoost y LightGBM, Redes Neuronales, entre otras.

2.3.1. APRENDIZAJE SUPERVISADO

2.3.1.1. REGRESIÓN LINEAL SIMPLE Y MULTIPLE

La Ecuación (1) representa al modelo general de la regresión lineal:

$$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \dots + \beta_kX_k + \varepsilon \quad (1)$$

Y: Variable dependiente

X_1, X_2, \dots, X_k : Variables independientes o predictoras

$\beta_1, \beta_2, \dots, \beta_k$: Parámetros del modelo ε : Variable

aleatoria, mide el error del modelo

Regresión lineal simple:

La Ecuación (2) representa el modelo utilizado para la regresión lineal simple:

$$Y = a + bX + e \quad (2)$$

Y: Variable dependiente X:

Variable independiente e:

Variable aleatorio de error.

Según (Manuel Córdova Zamora, 2008, en Estadística Aplicada), esta variable aleatoria posee considerandos que constituyen los supuestos del modelo de regresión lineal simple, estos son:

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

Método de Mínimos Cuadrados:

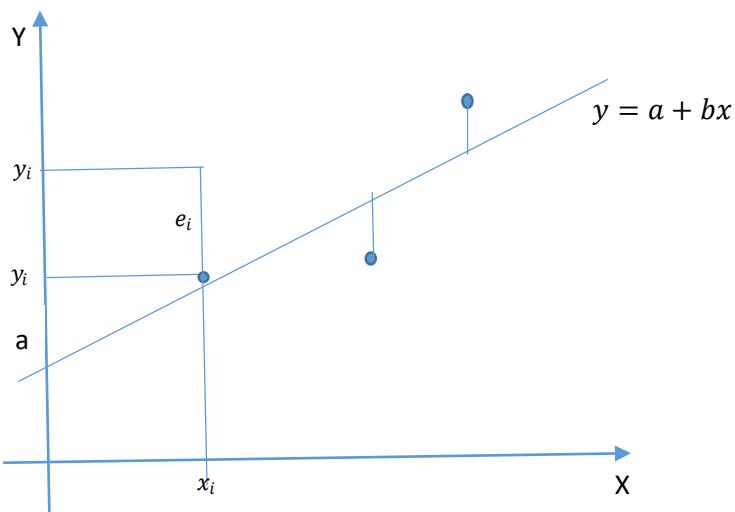
Manuel Córdova Zamora, en su libro Estadística Aplicada, propone:

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

b: es la pendiente de la recta de regresión. **a:** en la intersección de la recta de regresión con el eje “Y”.

La Figura 2 muestra la gráfica de la recta de regresión lineal utilizada para el cálculo de las constantes **a** y **b**, utilizando el método de los mínimos cuadrados:

Figura 2: Método de mínimos cuadrados



Fuente: Estadística Aplicada, Manuel Córdova Zamora, 2008

Donde $e_i = y_i - \bar{y}_i$, denominado, error o residuo muestral, describe el error en el ajuste del modelo de regresión en el punto “i” de los datos. Los errores muestrales satisfacen la condición $\sum_{i=1}^n e_i = 0$.

La recta de regresión de mínimos cuadrados de Y en X, es aquella que hace mínima la suma de los cuadrados de los errores (SCE) representada en la Ecuación (3) (Manuel Córdova Zamora, 2008).

$$SCE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \bar{y}_i)^2 = \sum_{i=1}^n (y_i - a - bx_i)^2 \quad (3)$$

La recta de regresión es aquella que minimiza la expresión anterior; para esto se utiliza el teorema de Gas-Markov, donde “a” y “b” se obtienen resolviendo el siguiente sistema de ecuaciones:

$$\begin{aligned} an + b \sum x &= \sum y & (4) \quad a \sum x + b \sum x^2 &= \sum xy \\ (5) \end{aligned}$$

Las Ecuaciones (4) y (5) se obtienen al igualar a cero las derivadas parciales de la “SCE” (Ecuación (3)) con respecto a “a” y con respecto a “b”; resolviendo este sistema de ecuaciones para “b” se obtiene la Ecuación (6):

$$b = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sum x_i^2 - (\sum x_i)^2} = \frac{n \sum x_i y_i - n(\bar{x})(\bar{y})}{\sum x_i^2 - n(\bar{x})^2} \quad (6)$$

El valor de “a” ($a = \bar{y} - b \bar{x}$) se obtiene dividiendo la Ecuación (4) entre “n”.

Regresión lineal múltiple:

La Ecuación (7) muestra el modelo estadístico que define la regresión lineal múltiple:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_k X_k + \varepsilon \quad (7)$$

Y: Variable dependiente

X_1, X_2, \dots, X_k ($k \geq 2$) : Variables independientes $\beta_0, \beta_1, \dots,$

β_k : Parámetros desconocidos.

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

La ecuación de regresión muestral (Manuel Córdova Zamora, Estadística Aplicada):

$$Y = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_kX_k \quad (8)$$

Los coeficientes de la Ecuación (8) de regresión muestral $b_0, b_1, b_2, \dots, b_k$ se calculan, aplicando el método de **mínimos cuadrados** a los datos de una muestra aleatoria de tamaño “n”, cuyos valores observados denotamos por: $(x_{1i}, x_{2i}, x_{3i}, \dots, x_{ki}, y_i)$, $i = 1, 2, 3, \dots, n$ y $n > k$, donde, y_i es la respuesta observada (valor de la variable dependiente Y) para los valores $x_{1i}, x_{2i}, x_{3i}, \dots, x_{ki}$ de las “k” variables independientes respectivas $X_1, X_2, X_3, \dots, X_k$.

Para $i = 1, 2, \dots, n$ los datos de la muestra satisfacen la Ecuación (9):

$$y_i = b_0 + b_1x_{1i} + b_2x_{2i} + b_3x_{3i} + \dots + b_kx_{ki} + e_i \quad (9)$$

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

b_0, b_1, \dots, b_k se calcula utilizando el método de **mínimos cuadrados**, que consiste en minimizar la suma del cuadrado de los errores (**SCE**) definida con la Ecuación (10):

$$SCE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - b_0 - b_1x_{1i} - b_2x_{2i} - \dots - b_kx_{ki}) \quad (10)$$

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

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

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

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

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

Donde, $\sum x_j = \sum_{i=1}^n x_{ji}$, $\sum x_j y = \sum_{i=1}^n x_{ji} y_i$ para $j = 1, 2, \dots, k$

Este sistema de ecuaciones se puede resolver utilizando la Ecuación (11) matricial

(Manuel Córdova Zamora, 2008, Estadística Aplicada):

$$(X'X)\mathbf{b} = X'Y \leftrightarrow \mathbf{b} = (X'X)^{-1}X'Y \quad (11)$$

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

2.3.1.2. REGRESIÓN LOGÍSTICA

La regresión logística es una variante del modelo de regresión lineal, se usa para predecir el comportamiento de una variable categórica (variable dependiente Y), esta predicción se realiza determinando la probabilidad de que suceda cualquiera de las categorías de la variable en función de las variables independientes o predictoras. Se utiliza para modelar la probabilidad de un evento en función de otros factores. Este modelo usa como función de enlace la función lógit y se le define como un modelo probabilístico de clasificación supervisada. Para iniciar el estudio del modelo, se supone que para cada caso determinado por el vector de variable $X^t =$

(X_1, X_2, \dots, X_p) , la variable dependiente "Y" tiene los valores 1 y 0 con probabilidades "p" y "1 - p". El modelo de regresión logística binaria a cada vector "X" le hace corresponder un $f(X)$ (función logística), cuyo valor se expresa en la Ecuación (12):

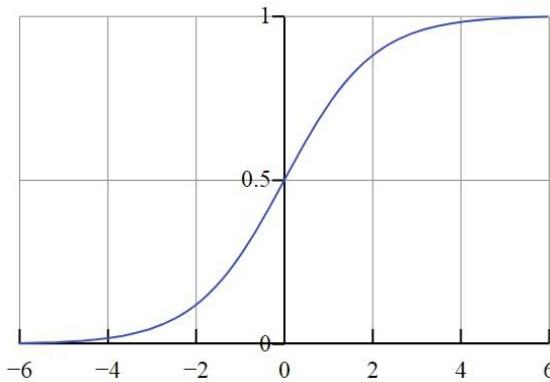
$$1 \quad (12)$$

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

(Carlos Veliz Capuñay, 2018, Aprendizaje Automático)

La Figura 3 muestra la gráfica de la **función logística**, cuando $X = X_1$:

Figura 3: Grafica de la función logística



Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018

El modelo tiene múltiples usos, en finanzas se emplea para determinar el riesgo crediticio y en medicina para calcular la probabilidad de contraer una enfermedad. La probabilidad de adquisición se calcula utilizando la Ecuación (13):

$$p^i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k)}} \quad (13)$$

X_1, X_2, \dots, X_k son variables 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 $\beta_1, \beta_2, \dots, \beta_k$ son comúnmente estimadas usando el método de **máxima verosimilitud**, el cual es un método habitual para ajustar un modelo. Además, el signo de los parámetros indica la relación directa o inversamente proporcional que existe entre la variable dependiente e independiente.

La regresión logística (Ecuación (14)), con una variable explicativa, puede usarse para calcular la correlación entre la probabilidad de una variable dicotómica, que puede tomar dos valores "0" y "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. Por tanto, la probabilidad del suceso se calculará mediante una función logística:

$$p(x) = \frac{e^{(\beta_0 + \beta_1 x)}}{e^{(\beta_0 + \beta_1 x)} + 1} = \frac{1}{e^{-(\beta_0 + \beta_1 x)} + 1} \quad (14)$$

La cual puede reducirse al cálculo de una regresión lineal, utilizando la Ecuación (15), para la función "logit" de la probabilidad:

$$h(x) = \ln \frac{p(x)}{1 - p(x)} = \beta_0 + \beta_1 x \quad (15)$$

2.3.1.3. ÁRBOL DE DECISIONES

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

En un conjunto de datos se realiza una partición recursiva para lograr sub grupos o patrones llamados nodos, que al ser representados gráficamente dan la idea de un sistema de árboles. Se usa:

- En Finanzas, para identificar grupo de clientes propensos a entrar en mora y evitar el riesgo crediticio.
- En Marketing Financiero, para identificar perfiles de clientes homogéneos con la finalidad de diseñar campañas tácticas de marketing.

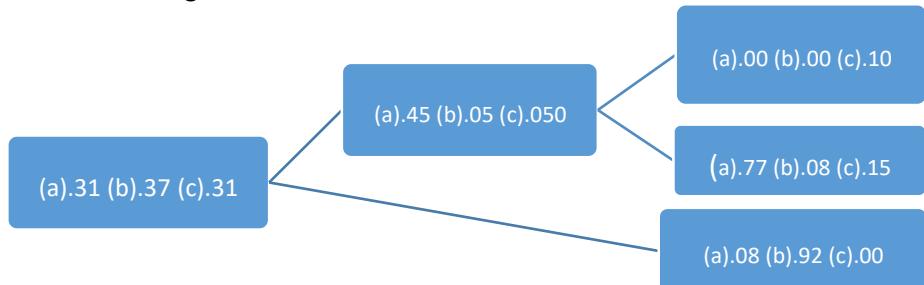
- En Control de Calidad, para determinar los factores que influyen en la calidad de un producto.

Elementos de un Árbol de Decisión

En la Figura 4 se muestra la representación gráfica de un árbol de decisiones, que está conformado por nodos de decisión, de probabilidad, terminales y ramificaciones:

- Un nodo define el momento en el que se toma una decisión.
- Los vectores de números son la solución final.
- Las flechas son las uniones entre nodos y representan cada acción distinta.
- Las etiquetas están en cada nodo y flecha y dan nombre a cada acción.

Figura 4: Grafica de un árbol de decisiones



Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018

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, en este nodo se encuentra definidas las variables 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.
- Establecer un criterio de parada donde se define un nodo como una hoja.

- La asignación de la categoría “Y” que se debe fijar como predicción si el nodo resultante es un nodo terminal.

Utilizando la Ecuación (16), modelo matemático del índice Gini, calculamos el grado de homogeneidad de los nodos:

$$IGini(q) = 1 - \sum_{i=1}^h p_i^2 \quad (16)$$

Donde “h” es la cantidad de categorías presentes en la variable “Y” y p_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, se usa la Ecuación (17) para calcular la reducción del grado de impureza al pasar de un nodo padre q_i a los nodos hijos q_{i+1} y q_{i+2} :

$$\Delta(q_{i+1}, q_{i+2}) = IGini \text{ nodo "padre"} q_i - \quad (17)$$

$$[(IGini \text{ nodo } q_{i+1})(Q_{i+1}/Q_i) + (IGini \text{ nodo } q_{i+2})(Q_{i+2}/Q_i)]$$

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

2.3.1.4. ALGORITMO XGBOOST

Es un algoritmo matemático, basado en árbol de decisiones, que tiene por finalidad menguar la incertidumbre a partir de la disminución de la pendiente de la curva de predicción; se usa cuando se tratan datos tabulares/estructurados medianos o pequeños, en tanto, proveen la mejor solución. Un grupo de científicos contribuye en diferentes proyectos relacionados con XGBoost, lo que hace posible su utilización en varias aplicaciones de la industria. Este algoritmo es ampliamente utilizado en la industria financiera, por su grado de precisión para identificar a las personas propensas a adquirir un producto financiero; su mayor ventaja sobre los algoritmos de regresión tradicionales, es identificar relaciones lineales y no lineales entre las variables dependientes e independientes.

Características principales del algoritmo XGBoost:

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

Ventajas del algoritmo XGBoost:

- Puede utilizarse en una amplia gama de aplicaciones para resolver problemas de predicción, clasificación y regresión.
- Se implementa en diferentes sistemas operativos como Linux, OSX y Windows.
- Soporta la mayoría de lenguajes de programación como R, Python, Java, C++.
- Permite la programación paralela sobre diversos ecosistemas de la nube.
- Identifica relaciones lineales y no lineales entre las variables independientes y las dependientes, logrando identificar la combinación óptima de variables que minimicen el error de predicción.

2.3.1.5. SUPPORT VECTOR MACHINE

La máquina de vector soporte (del inglés Support Vector Machine, SVM) 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 Support Vector Machine, que prediga a qué clase pertenecen los elementos de una nueva muestra.

El modelo SVM identifica múltiples hiperplanos 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.

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 hiperplano que separe las clases.

Ventajas del modelo Maquina de Soporte Vectorial:

- 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.

2.3.2. APRENDIZAJE NO SUPERVISADO

2.3.2.1. CLUSTERIZACIÓN USANDO KMEANS.

Cluster es un vocablo en inglés que significa grupo; la clusterización es el agrupamiento automático de los elementos que conforman un grupo. En la industria financiera se clusteriza con la finalidad de obtener conjuntos de leads (prospectos de cliente) y de clientes con perfiles específicos, por ejemplo, segmentos diferenciados por sus ingresos económicos.

K-medias es un método de clusterización, que tiene como objetivo la partición de un conjunto de n elementos en k sub conjuntos cuyos elementos pertenecen se caracterizan por tener un perfil relativamente homogéneo.

Dado un conjunto de “n” elementos (x_1, x_2, \dots, x_n), donde cada elemento tiene un serie de atributos que definen su perfil, el algoritmo KMeans busca agrupar estos

elementos en k conjuntos ($k \leq n$), minimizando la suma del cuadrado de la distancia de cada uno de estos elementos al centro del grupo $S = \{S_1, S_2, \dots, S_k\}$:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - u_i\|^2$$

Donde u_i es la media de puntos en S_i .

Dado un conjunto inicial de k centroides $m_1^{(1)}, m_2^{(1)}, \dots, m_k^{(1)}$, el algoritmo sigue alternando entre dos pasos:

- Se utiliza la Ecuación (18) para asignar cada observación al grupo con la media más cercana:

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k\} \quad (18)$$

Donde cada x_p va exactamente dentro de un $S_i^{(t)}$.

- Se utiliza la Ecuación (19) para calcular los nuevos centroides de cada grupo:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (19)$$

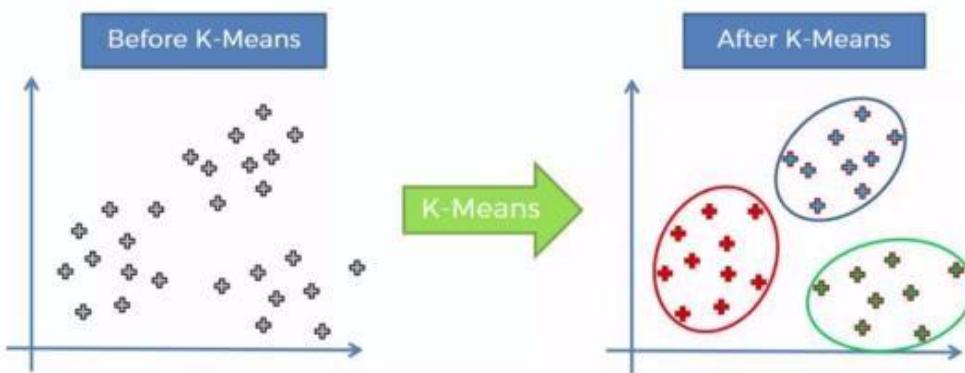
- Cuando las asignaciones ya no cambian, el algoritmo ha convergido.

Para definir inicialmente los centroides, se utilizan comúnmente los métodos Forgy y Partición Aleatoria. El método Forgy elige aleatoriamente k observaciones del conjunto de datos y las utiliza como centroides iniciales. El método de Partición Aleatoria, primero, asigna aleatoriamente un cluster para cada uno de los elementos, los centroides iniciales serán el centro de gravedad de cada cluster creado al azar.

El método Forgy tiende a dispersar los centroides iniciales, mientras que la Partición Aleatoria ubica los centroides cerca del centro del conjunto de datos. El método de Partición Aleatoria general, es preferible para los algoritmos tales como los KMeans armonizadas y fuzzy KMeans (Según Hamerly, 2013).

En la Figura 5 se muestra el resultado de aplicar el algoritmo KMeans:

Figura 5: Gráfica del método KMeans



Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018

Según la Figura 5 se concluye que las técnicas de Clustering nos permiten descubrir el mejor método de agrupamiento de los datos. Algunas de estas técnicas necesitan que especifiquemos previamente la cantidad de grupos.

2.3.2.2. 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 (ACP)**. 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

(X_i) y todas las componentes principales. Su valor es 1, dado que toda variable X_i puede expresarse de modo exacto como combinación lineal de las componentes.

En síntesis, el objetivo primordial del ACP es crear nuevas variables (componentes principales) que sean capaces de reflejar casi toda la información registrada en los datos originales.

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.

2.3.2.3. AGRUPACIÓN JERÁRQUICA

Una de las herramientas que utiliza el Aprendizaje no Supervisado, es un algoritmo denominado “Agrupación Jerárquica”, sirve para agrupar puntos de datos no etiquetados; también agrupa los puntos de datos con características similares. Tiene alguna similitud con el algoritmo K Means.

Existen dos tipos de Agrupación Jerárquica: aglomerativa y divisoria:

- Agrupación Aglomerativa: Los puntos de datos se agrupan utilizando un enfoque ascendente que comienza con puntos de datos individuales.
 - Agrupación Divisoria: El criterio de agrupamiento es descendente donde todos los puntos se tratan como un gran conjunto de datos y el proceso de agrupación implica la partición de este gran conjunto en varios sub conjuntos pequeños.
- Para realizar la agrupación jerárquica de los datos observados, es necesario definir una métrica de similitud entre los conjuntos por agrupar (Agrupación Aglomerativa) o por dividir (Agrupación Divisoria) y un criterio de enlace que especifica la similitud entre ellos, la métrica escogida influenciará en el resultado del agrupamiento ya que algunos puntos pueden ser cercanos o lejanos unos de

otros dependiendo de la métrica utilizada. En la Tabla 2 se indican las métricas comúnmente usadas para el agrupamiento jerárquico.

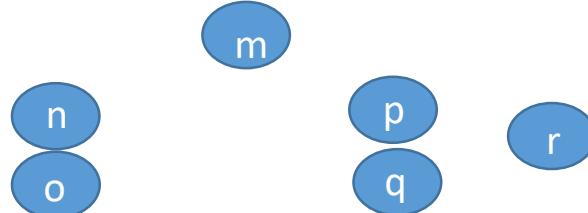
Tabla 2: Métricas usadas en el agrupamiento jerárquico

Métricas	Formulas
Distancia Euclíadiana	$\ a - b\ _2 = \sqrt{\sum_i (a_i - b_i)^2}$
Distancia Euclíadiana al cuadrado	$\ a - b\ _2^2 = \sum_i (a_i - b_i)^2$
Distancia de Manhattan	$\ a - b\ _1 = \sum_i a_i - b_i $
Distancia de Mahalanobis	$\sqrt{(a - b)^T S^{-1} (a - b)}$ donde S es la matriz de covarianza
Similitud Coseno	$\frac{a \cdot b}{\ a\ \ b\ }$

Fuente: Big Data analytics and the transformation of Marketing, Journal of Business Research, pp. 897-904, by Erevelles S., Fukawa N. & Swayne L. (2016)

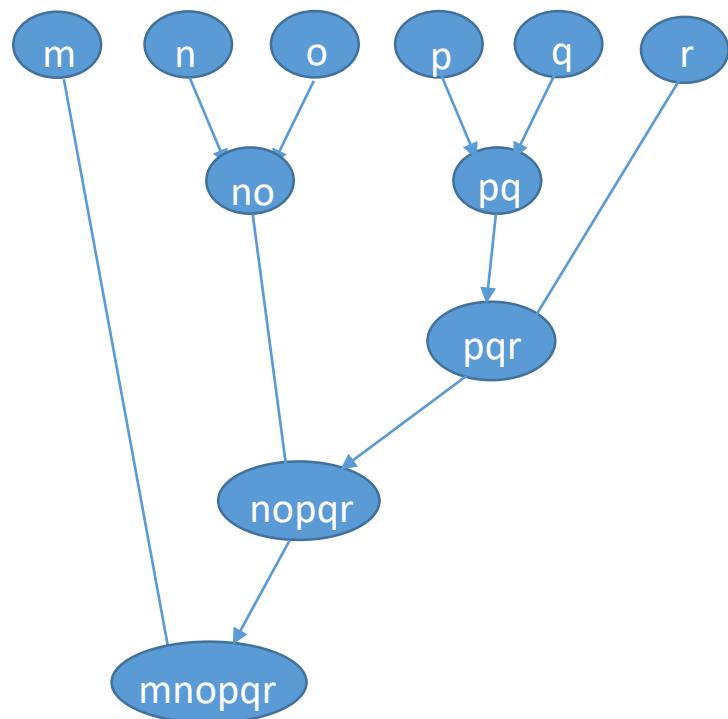
Los resultados del agrupamiento jerárquico son usualmente presentados en un dendrograma. Las Figuras 6 y 7 muestran un ejemplo de agrupación jerárquica:

Figura 6: Conjunto de datos por agrupar



Fuente: Clustering Jerárquico, Departamento de Ciencias de la Computación e IA, Universidad de Granda, por Fernando Berzal 2018

Figura 7: Dendrograma del agrupamiento jerárquico



Fuente: Clustering Jerárquico, Departamento de Ciencias de la Computación e IA, Universidad de Granda, por Fernando Berzal 2018

2.3.3. APRENDIZAJE POR REFORZAMIENTO

2.3.3.1. SIMULACIÓN MONTECARLO

La Simulación Montecarlo es un método estadístico, usado para aproximar expresiones matemáticas complejas a partir de la generación de variables aleatorias. Realizar una simulación consiste en repetir, o duplicar, las características y comportamientos de un sistema real. Así pues, el objetivo principal de la simulación de Montecarlo es intentar imitar el comportamiento de variables reales para analizar o predecir cómo van a evolucionar.

A través de la simulación, se pueden resolver desde problemas muy sencillos, hasta problemas muy complejos; dependiendo de la complejidad se usan programas informáticos como Excel, R Studio o Matlab para encontrar la solución.

Algunos usos de la simulación de Montecarlo

- En la industria: se utiliza para optimizar las operaciones y los procesos unitarios que intervienen en la producción y comercialización de bienes.
- En finanzas, se emplea para:
 - Crear, valorar y analizar carteras de inversión.
 - Valorar productos financieros como las opciones financieras.
 - Creación de modelos de gestión de riesgo.

En la bolsa de valores, los movimientos de una acción no se pueden predecir. Se pueden estimar, pero es imposible hacerlo con exactitud. Por ello, mediante la simulación de Montecarlo, se intenta imitar el comportamiento de una acción o de un conjunto de ellas para analizar cómo podrían evolucionar. Una vez que se realiza la simulación de Montecarlo se simulan una gran cantidad de escenarios posibles.

2.3.3.2. Q – LEARNING

El aprendizaje por refuerzo es una técnica de inteligencia artificial que permite entrenar a las máquinas con la ayuda de un algoritmo de aprendizaje automático. El aprendizaje supervisado y el aprendizaje por refuerzo son técnicas diferentes. En el aprendizaje supervisado, los modelos se entrena utilizando una variable dependiente (target del modelo) la cual es la respuesta al evento; por ejemplo, en finanzas, si se busca calcular la probabilidad de que un cliente adquiera un producto financiero se debe considerar una variable que califique al cliente con “1” si toma el producto financiero y con “0” en caso contrario; con este criterio, se desarrolla un modelo utilizando el aprendizaje supervisado. En el aprendizaje por refuerzo, el algoritmo no incluye las respuestas correctas (“1 y 0” en el aprendizaje supervisado), para mejorar el rendimiento del modelo se realizan varias actividades a medida que se maximiza la recompensa hasta llegar a un valor óptimo de rendimiento.

Q-learning es una técnica de aprendizaje por refuerzo utilizada en aprendizaje automático. Tiene por objetivo aprender una serie de normas que le diga a un agente qué acción debe tomar bajo determinadas circunstancias.

También se le puede definir como un algoritmo de aprendizaje basado en valores, cuyo objetivo es la optimización de la función de valor según el entorno o el problema. La Q en el Q-learning representa la calidad con la que el modelo encuentra su próxima acción mejorando la calidad. El modelo almacena todos los valores en una tabla, que es la Tabla Q, esta tabla es actualizada mediante el algoritmo Q – Learning buscando la mejor solución posible.

En conclusión, el aprendizaje por refuerzo trata de resolver cómo un agente aprenderá en un entorno incierto tomando una secuencia de decisiones. Existen numerosas técnicas y métodos que permiten al agente determinar su trayectoria y realizar acciones progresivas.

Algoritmo Q-Learning

- Se inicializa una matriz de estados por cada posible acción a realizar, donde cada valor será actualizado con el entrenamiento del algoritmo. Los valores iniciales altos también conocidos como “valores optimistas” promueven la exploración, estos valores permiten que las siguientes actualizaciones tengan valores menores a los iniciales.
- Se inicializa un valor aleatorio para la variable “Q” a maximizar y un estado inicial “S”, así como el valor de los siguientes parámetros:
 - Tasa de aprendizaje: Es una constante que permite generar nuevos valores para “Q” basándose en la información anterior, este valor determina el grado de exploración del algoritmo.
 - Tasa de descuento: Determina la importancia de futuras recompensas en las siguientes iteraciones, esta constante ayuda a equilibrar el efecto de estas nuevas recompensas.

- El agente elige aleatoriamente una acción “ a_t ” por realizar, calcula la recompensa “ r_t ” ganada por realizar esta acción y llega a nuevo estado “ S_{t+1} ”. Mediante la Ecuación (20) se actualiza el valor de “Q”, en este nuevo estado:

$$Q^{new}(s_{t+1}, a_t) \leftarrow (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max Q(s_{t+1}, a_t)) \quad (20)$$

Donde “ α ” es la tasa de aprendizaje y “ γ ” es la tasa de descuento.

- El algoritmo termina cuando el estado “ s_{t+1} ” llega a un estado final establecido por el usuario o cuando el ratio de mejora es menor a un mínimo permisible.

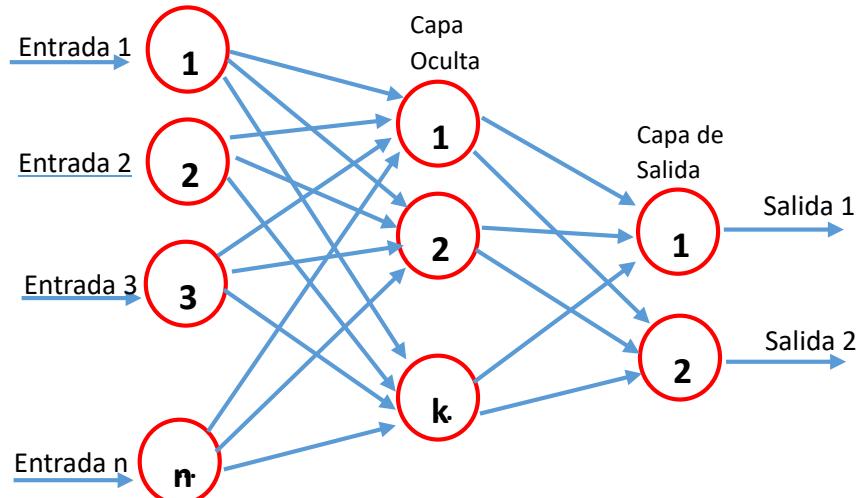
Este algoritmo le permite al agente encontrar la secuencia optima de acciones que maximiza la recompensa total del evento.

2.4. REDES NEURONALES

2.4.1. CONCEPTOS Y DESCRIPCIÓN

La red neuronal es un modelo analítico descrito con relaciones matemáticas complejas entre las variables que lo forman, son usadas en la actualidad en diferentes aplicaciones relacionada con grandes cantidades de datos que son analizados con modelos no lineales; se utilizan en finanzas y economía con fines predictivos. Este modelo trata de imitar la estructura y la capacidad de aprender del cerebro humano, por esto se representa usando nodos interconectados que simulan a las neuronas y dendritas del cerebro. Como se muestra en la Figura 8, 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.

Figura 8: Estructura de una Red Neuronal



Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018

Cada nodo es una neurona artificial conectada entre sí para transmitir señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo 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.

2.4.2. ARQUITECTURA DE LOS SISTEMAS NEURONALES ARTIFICIALES

- Segundo el número de capas

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 red neuronal posee un conjunto de capas intermedias denominadas capas ocultas, en éstas la función de activación propia de cada neurona transforma los datos de entrada a valores cercanos a los que se busca predecir.

- Segundo el tipo de conexiones

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.

➤ Segundo el grado de conexión

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. En la primera estructura se plantea un “consenso” entre las diferentes redes para obtener la salida mientras que en la estructura jerárquica existen redes subordinadas a otras que actúan como elementos centrales en la salida final de la red.

2.4.3. APRENDIZAJE DE LA RED. ALGORITMO DE RETROPROPAGACIÓN

La red “aprende o se entrena” a través de la experiencia lograda al iterar los mecanismos que transforman la información de las instancias buscando una relación entre los datos de entrada y de salida. 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.

En resumen, el algoritmo de retropopagación es:

- 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** determinada por 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:

$$E = - \sum_{i=1}^n \sum_{j=1}^M (y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})) \quad (21)$$

La Ecuación (21) es también denominada “**entropía cruzada**” y es comúnmente usada por su velocidad de convergencia y la mejora en la generalización.

2.4.4. ANALISIS 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 3.

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.

Tabla 3: Ejemplo para el Análisis de Sensibilidad de una 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

Fuente: Aprendizaje Automático, análisis para la minería de datos y Big Data, por Carlos Veliz 2018

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.

2.4.5. APLICACIONES DE LAS REDES NEURONALES

Categorías generales de aplicación:

- Análisis de regresión y modelos de predicción.
- Reconocimiento de patrones para apoyar la toma de decisiones.
- Procesamiento de datos: clusterización y separación de señales.
- Robótica: en drones, prótesis y otros.
- Ingeniería de control, incluyendo control numérico por computadora.

Áreas de aplicación: ➤

Predicción de trayectorias

- Control de procesos.
- Identificación de sistemas y control (vehículo autónomo).
- Manejo de recursos naturales.
- Química cuántica.
- Reconocimiento de patrones: reconocimiento facial, clasificación de señales, sistemas de radar, reconocimiento de objetos, etc.
- Diagnóstico médico: diagnóstico de varios tipos de cáncer (cáncer pulmón, cáncer colon rectal, cáncer de mama).

- Aplicaciones Financieras: construcción de sistemas automatizados de comercio electrónico, construcción de modelos predictivos, estandarización de probabilidades, estimación de fraude crediticio, diseño de estrategia de campañas comerciales, en modelos de pricing, etc.
- Procesamiento de lenguaje natural: visualización y traducción automática, análisis de datos no estructurados (redes sociales), reconocimiento de patrones en textos y/o documentos.
- Reconocimiento de secuencias: gesto, voz y reconocimiento de texto escrito y a mano.

2.5. PROGRAMACIÓN LINEAL ENTERA

2.5.1. CONCEPTOS GENERALES

La Programación Lineal Entera es uno de los conocimientos más utilizados de la Investigación de Operaciones, debido a su flexibilidad para describir situaciones reales. Ha sido desarrollado para representar y solucionar problemas de decisión que implican la optimización (maximización o minimización) de una función lineal, denominada función objetivo, de tal forma que las variables de dicha función estén sujetas a una serie de restricciones expresadas mediante un sistema de ecuaciones o inecuaciones también lineales. Un modelo PE es aquel que tiene una o más variables de decisión que son de tipo entero.

Tipos de Programación Lineal Entera (PE)

- Modelo PE Puro: Es aquel en el cual todas sus variables son enteras.
- Modelo PE Binario: Es aquel que tiene variables enteras que solo toman valores 0 o 1 (binario). Estas variables son muy importantes porque permiten expresar situaciones complejas.
- Modelo PE Mixto: Es aquel que contiene variables enteras, binarias, continuas, etc.

Estructura de un modelo de Programación Lineal Entera:

- **Variables de decisión:** son aquellas definidas por el analista, sus valores constituyen la solución del problema. Los valores de estas variables son positivos ($X_i \geq 0$).
- **Función Objetivo (FO):** es aquella función lineal que se desea optimizar
- **Restricciones:** representan las limitaciones que tiene el problema y están expresadas mediante un sistema de ecuaciones e inecuaciones lineales. Existen muchas técnicas para encontrar la solución óptima de la función objetivo, la más usada es la **técnica de Ramificación y Acotamiento (Branch and Bound)**.

2.5.2. METODO DE RAMIFICACIÓN Y ACOTAMIENTO

Consiste en construir un árbol de soluciones, donde cada rama nos lleva a una posible solución, que es sub siguiente a la inicial. El algoritmo se encarga de detectar en cada ramificación las soluciones que se alejan de la solución óptima (soluciones que contienen valores no enteros), estas ramas se podan del árbol. En el caso en el que se encuentre dos o más soluciones enteras, se escoge la solución óptima. Esta es la solución final de la **Programación Entera**.

Como ejemplo explicativo (Investigación de Operaciones – Wayne L. Winston – Cuarta Edición):

$$\text{Max } z = 3x_1 + 2x_2$$

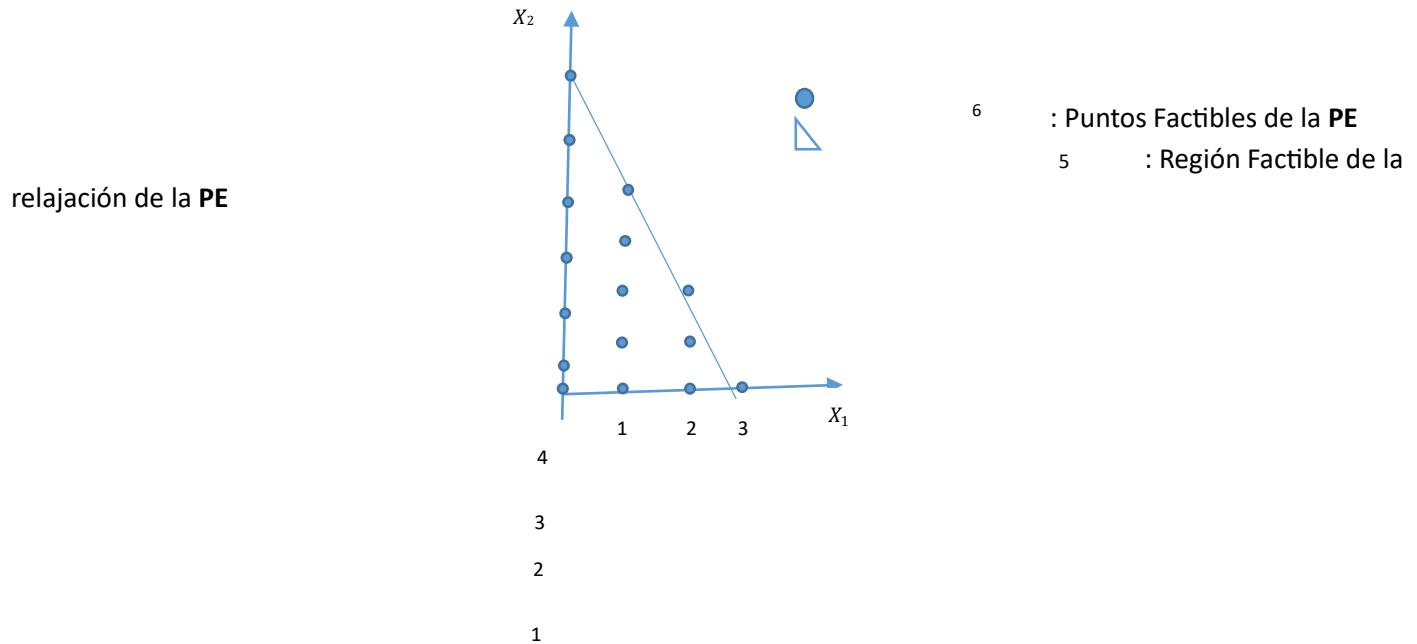
$$\text{s. a: } 2x_1 + x_2 \leq 6$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2: \text{enteros}$$

La solución óptima para la relajación de la Programación Lineal de esta Programación Entera es: $x_1 = 0$, $x_2 = 6$, $z = 12$; como todas las variables tienen valores enteros, entonces, esta solución es también la solución óptima de la Programación Entera; la gráfica de la solución se muestra en la Figura 9:

Figura 9: Región Factible para una **PE** y su relajación de la **PL**



Fuente: Investigación de Operaciones, por Wayne L. & Winston, Cuarta Edición
 El algoritmo del método de Ramificación y Acotamiento es el siguiente:

1. Encontrar la solución óptima de la relajación del problema lineal de la programación entera.
2. Si todas las variables de decisión, de esta solución, asumen valores enteros, entonces, esta es la solución del problema de programación entera.
3. Si una o más variables de decisión, de la solución parcial analizada, asumen valores no enteros, se procede de la siguiente forma:
 - Si existe una sola variable que toma un valor no entero, se crea dos restricciones adicionales, a partir de los valores enteros más próximos al valor decimal de la variable, es decir, este valor es menor o igual a la parte entera del valor de la variable y es mayor o igual al número entero inmediatamente superior al valor de la variable.
 - Si existen más de una variable que toma un valor no entero, se selecciona aleatoriamente uno de estos valores y se procede como en el caso anterior.

4. Estas dos restricciones adicionales que se han creado generan dos sub problemas.
5. Se encuentra la solución de cada sub problema, descartando la solución no factible.
6. Se regresa al paso 3 hasta lograr que todas las soluciones parciales sean enteras; la solución que maximiza la función objetivo es la solución óptima de la programación entera. (Investigación de Operaciones – Wayne L. Winston – Cuarta Edición)

Cuando la solución óptima puede incluir valores decimales (programación mixta), se modifica el método de ramificación y acotamiento, descrito anteriormente, ramificando solo sobre las variables que deban asumir necesariamente valores enteros. La solución final es aquella que maximiza el valor de la función objetivo.

2.5.3. METODO DE ENUMERACIÓN IMPLICITA

Este método se utiliza con frecuencia para resolver los problemas de Programación Entera, donde las variables de la función objetivo sólo asumen valores 0 o 1; a este tipo de programación se le denomina Programación Binaria. Este modelo de programación permite simplificar tanto la ramificación como los elementos del acotamiento del proceso, y para determinar de manera eficiente cuando es o no factible un nodo (Investigación de Operaciones – Wayne L. Winston – Cuarta Edición). Los conceptos involucrados son los siguientes:

- Nodo: Cada nodo está compuesto por la función objetivo con sus restricciones.
- Ramificación de un nodo: Cuando aleatoriamente se selecciona una variable libre y se le asigna un valor fijo, se consigue ramificar un nodo.
- Variables Libres: Son aquellas que pueden tomar indistintamente dos valores (0 o 1).
- Variables Fijas: Son aquellas que en el desarrollo del proceso se les ha

asignado un valor fijo (0 o 1).

- Mejor terminación de un nodo: Son los valores (0 o 1) que optimice la función objetivo sin tomar en cuenta las restricciones.
- Terminación no factible de un nodo: Es aquella que no satisface las restricciones de la función objetivo.
- Factibilidad de un nodo: Un nodo es factible cuando todas las restricciones tienen una terminación que las satisface.
- Factibilidad de la solución: Se consigue cuando la solución obtenida en un nodo satisface las restricciones de la función objetivo.

Como se muestra en la Tabla 4, para el desarrollo del proceso de ramificación y acotamiento es necesario construir una tabla que permite determinar si un nodo tiene una terminación que satisface una restricción dada.

Tabla 4: Factibilidad de un nodo que satisface una restricción dada.

Tipo de Restricción	Signo del coeficiente de la variable libre en la restricción	Valor asignado a la variable libre en la restricción de la factibilidad
\leq	+	0
\leq	-	1
\geq	+	1
\geq	-	0

Fuente: Investigación de Operaciones, por Wayne L. & Winston, Cuarta Edición

Algoritmo de la Enumeración Implícita:

1. Se comprueba si la mejor terminación del nodo 1 es factible:
 - 1.1. Si es factible esta terminación se convierte en solución óptima (solución final).
 - 1.2. Si no es factible, se valida la factibilidad del nodo:

1.2.1. Si el nodo es factible, se da inicio a la ramificación (cada ramificación origina dos nodos, es decir, para $X_i = 0$ y $X_i = 1$).

1.2.2. Si el nodo no es factible, el problema no tiene solución.

2. De los nodos ramificados se selecciona uno aleatoriamente y volvemos al paso 1 hasta que la mejor terminación del nodo satisfaga todas las restricciones de la función objetivo y lo optimice (maximice o minimice).

Para disminuir la cantidad de nodos que se tienen que examinar antes de que se encuentre la solución óptima, se recurre al uso de pruebas de infactibilidad denominadas restricciones subrogadas o sustitutas. (Investigación de Operaciones – Wayne L. Winston – Cuarta Edición).

2.5.4. ALGORITMO DE PLANO DE CORTE

Es un procedimiento utilizado para encontrar soluciones enteras de un problema lineal. Funciona resolviendo un programa lineal no entero, después comprobando si la optimización encontrada es también una solución entera. Si no es así, es añadida una nueva restricción que corta la solución no entera pero no corta ningún otro punto de la región factible. Esto se repite hasta que se encuentra la solución entera óptima.

Para la mejor comprensión del método es necesario explicar:

- Los planos de corte se generan introduciendo una nueva restricción al modelo, es por eso que a una restricción añadida se denomina corte.
- Cualquier punto factible para el problema de Programación Entera satisface el corte.
- La solución óptima para la relajación del problema lineal, no satisface el corte. Por tanto, un corte limita la solución óptima para la relajación del problema lineal, pero no limita las soluciones factibles del problema de programación entera.

Se necesita construir un Tableau Óptimo para la Relajación del PL; para:

$$\begin{aligned}
 \text{Max } z &= 8x_1 + 5x_2 \\
 \text{s. a: } x_1 + x_2 &\leq 6 \quad 2 \\
 9x_1 + 5x_2 &\leq 45 \\
 x_1, x_2 &\geq 0; x_1, x_2: \text{enteros}
 \end{aligned}$$

La Tabla 5 muestra el tablero óptimo para la relajación del problema de PL:

Tabla 5: Tablero Óptimo para la relajación de la PL.

Z	X ₁	X ₂	S ₁	S ₂	Sol
1	0	0	1.25	0.75	41.25
0	0	1	2.25	-0.25	2.25
0	1	0	-1.25	0.25	3.75

Fuente: Investigación de Operaciones, por Wayne L. & Winston, Cuarta Edición

Algoritmo del Plano de Corte (Investigación de Operaciones – Wayne L. Winston):

1. Encuentre el Tableau óptimo para la relajación del problema lineal (mediante el método Simplex). Si todas las variables de la solución óptima asumen valores enteros, entonces se ha encontrado una solución óptima para el problema de programación entera; en caso contrario, siga con el paso 2.
2. Elija una restricción en el Tableau óptimo de la relajación del PL, cuyo lado derecho tiene la parte fraccionaria más cercana a $\frac{1}{2}$. Esta restricción se usa para generar un corte.
 - 2.a. Para la restricción obtenida en el paso 2, escriba su lado derecho y cada coeficiente de las variables en la forma $[x] + f$, donde $0 \leq f \leq 1$.
 - 2.b. Vuelva a escribir la restricción usada para generar el corte como: Todos los términos con coeficiente entero = todos los términos con coeficientes fraccionario.

Entonces el corte es:

Todos los términos con coeficiente fraccionario ≤ 0

3. Encuentre la solución óptima para la relajación del PL, con el corte como una restricción adicional, mediante el algoritmo simplex dual. Si todas las variables asumen valores enteros en la solución óptima, ha encontrado una solución óptima para el PE. En caso contrario, escoja la restricción cuyo lado derecho tenga la mayor fracción cercana a $\frac{1}{2}$ y úsela para generar otro corte, el cual se suma al Tableau óptimo. Continúe este proceso hasta que obtenga una solución en el cual todas las variables sean enteras. Esta será una solución óptima para el problema de Programación Entera.

El Tableau óptimo de la relajación del problema lineal se obtiene mediante el método simplex, éste es un método iterativo que permite ir mejorando la solución paso por paso, maximizando o minimizando la función objetivo.

2.6. ALGORITMOS GENÉTICOS

Son llamados así porque se inspiran en la evolución biológica y su base genéticomolecular. Estos algoritmos se enmarcan dentro de los algoritmos evolutivos, que incluyen también las estrategias evolutivas, la programación evolutiva y la programación genética.

Si bien es cierto, que los algoritmos genéticos se pueden usar para calcular el valor de cualquier función, su uso es preferible cuando la función matemática no es derivable (o de derivación muy compleja). Además, debe tenerse en cuenta también las siguientes consideraciones:

- Si la función a optimizar tiene muchos máximos/mínimos locales se requerirán más iteraciones del algoritmo para "asegurar" el máximo/mínimo global.
- Si la función a optimizar contiene varios puntos muy cercanos en valor al óptimo, solamente podemos "asegurar" que encontraremos uno de ellos (no necesariamente el óptimo).

Desventajas y limitaciones:

- Para problemas de alta complejidad la función de evaluación puede tornarse demasiado costosa en términos de tiempo y recursos.

- Puede haber casos en los cuales dependiendo de los parámetros que se utilicen para la evaluación, el algoritmo podría no llegar a converger en una solución óptima.
- La "mejor" solución lo es solo en comparación a otras soluciones por lo que no se tiene demasiado claro un criterio de cuándo detenerse.

El pseudocódigo de este algoritmo lo he desarrollado en capítulo III de la propuesta de solución.

CAPÍTULO III

DESARROLLO DEL TRABAJO DE INVESTIGACIÓN

3.1. FASE I: CENTRALIZACIÓN Y EXPLOTACIÓN DE LOS DATOS

En el sistema financiero, encontramos diversa y cuantiosa información acerca del perfil y comportamiento del mercado, estos datos son captados a través de la interacción de los diferentes canales de contacto de las organizaciones financieras (digitales, canales de venta, canales de atención al cliente, entre otros) con los clientes; sin embargo, este sistema carece de organización y centralización en la gestión de la información, lo cual impide la correcta explotación de los datos. El objetivo de la primera fase de la propuesta de solución, es desarrollar un modelo de

datos que consolide la información obtenida de la gestión de las campañas de marketing. Se busca también calcular la rentabilidad de estas campañas.

3.1.1. LEVANTAMIENTO DE LA INFORMACIÓN

La información disponible corresponde a diferentes tipos de variables (cuantitativas o cualitativas), las cuales provienen de diversas fuentes de datos, tales como:

- 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, POS, etc.).
- Canales digitales como el aplicativo móvil y la web del banco.
- Fuente de datos provenientes de empresas externas, por ejemplo, información del censo o información de contactabilidad.
- Información provista por la Súper Intendencia de Banca y Seguros.

Esta información es utilizada en la gestión de las campañas de marketing, que por lo general es actualizada mensualmente y en otros casos día tras día, para llevar un seguimiento de la producción y rentabilidad de la campaña.

De cada fuente de datos se puede extraer un conjunto de variables que serán utilizadas para el desarrollo de los modelos predictivos que intervienen en la construcción de estrategias de despliegue en las campañas de marketing, estas variables pueden clasificarse en cinco grandes grupos.

Variables Demográficas:

La Tabla 6 muestra las principales variables demográficas que permiten perifilar la cartera de clientes del banco, cada perfil define una acción comercial.

Tabla 6: Variables Demográficas

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
GradoInst	Grado de Instrucción	Primaria, Secundaria, Técnica y Universitaria
Flag_Dependiente	Indica si es dependiente	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
Tenencia_Inmueble	Indica si el cliente tiene vivienda	Valor binario: 1 o 0
Tenencia_Vehiculo	Indica si el cliente tiene vehículo	Valor binario: 1 o 0
Tipo de Vehiculo	Gama del vehículo que posee	Gama del vehículo: Baja, Media y Alta

Fuente: Institución Financiera, elaboración propia

Variables Transaccionales:

Cuantifican el nivel de interacción del cliente con los diferentes puntos de contacto (red de agencias, agentes, cajeros automáticos, aplicativo móvil y web).

La Tabla 7 muestra las principales variables transaccionales.

Tabla 7: Variables Transaccionales

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
ImporteTrxWeb	Importe monetario de las transacciones a través de la Web.	Valores enteros entre 0 y 100,000

Fuente: Institución Financiera, elaboración propia

Variables Internas de la institución financiera:

Cuantifican el valor del cliente para el banco. La Tabla 8 muestra las principales variables internas de la institución financiera:

Tabla 8: Variables Internas de la institución financiera

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 dentro del 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

Fuente: Institución Financiera, elaboración propia

Variables del Sistema Financiero:

Son variables que explican la situación del cliente dentro del sistema financiero, suelen ser usados para evaluar el riesgo crediticio asociada a cada cliente:

La Tabla 9 muestra las principales variables del sistema financiero.

Tabla 9: Variables del Sistema Financiero

Variable	Descripción	Dominio
ClasificacionSBS	Clasificación de riesgo que otorga la SBS al cliente	Normal, CPP, Deficiente, Dudos 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
----------------------	---	-----------------------------------

Fuente: Institución Financiera, elaboración propia

Variables de la Campaña:

Explican las características de la campaña de marketing, se muestran en la Tabla 10

Tabla 10: Variables de la campaña comercial

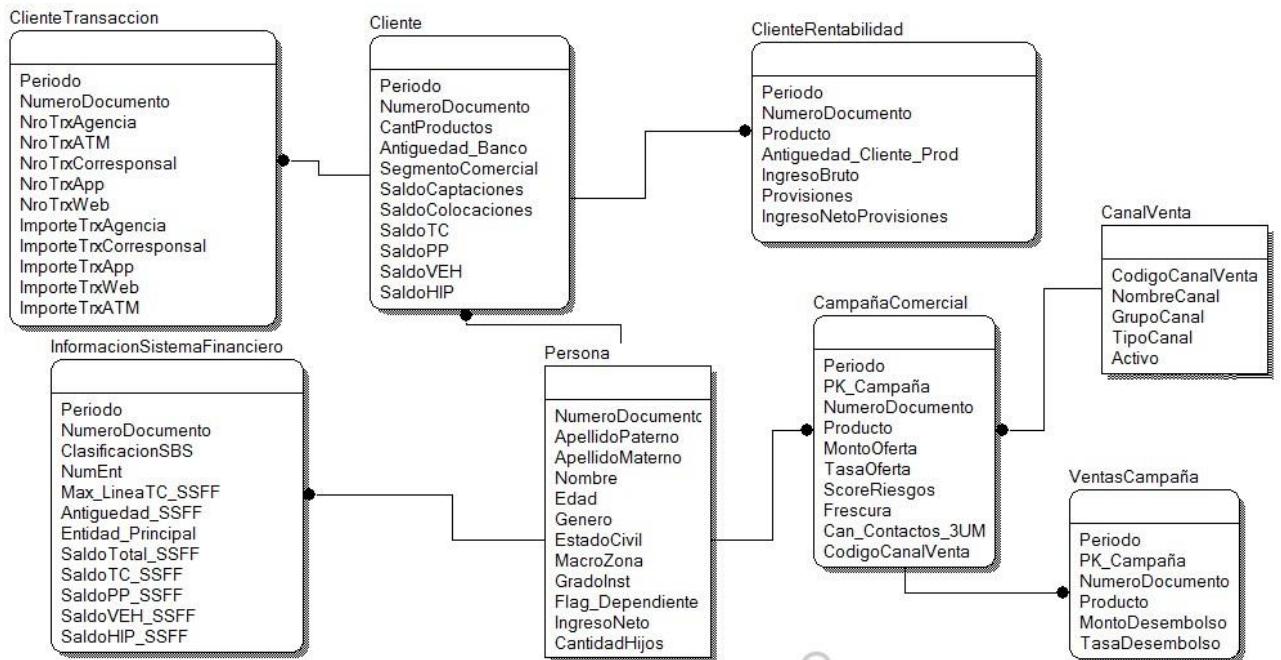
Variable	Descripción	Dominio
MontoOferta	Monto del producto a ofrecer	Valores entre 0 y 150,000
TasaOferta	Tasa del producto a ofrecer	Valores entre 11.00 y 45.00.
Frescura	Cantidad de veces que el cliente ha estado en las últimas 6 campañas.	Valores entre 1 y 6
ScoreRiesgos	Score que determina el nivel de riesgo del cliente	Valores entre 1 y 10
Can_Contactabilidad_3UM	Cantidad de veces que ha sido contactado el cliente en las 3 últimas campañas	Valores entre 0, 1, 2 o 3

Fuente: Institución Financiera, elaboración propia

3.1.2. CONSOLIDACIÓN DE LA INFORMACIÓN

Para desarrollar los modelos analíticos debemos consolidar la información en un DataSet que contenga todas las variables independientes y la variable dependiente (Target) que se desea predecir; para que las variables puedan ser utilizadas fácilmente es necesario modelarlas de manera estructurada y organizada. Para este fin, en la Figura 10 se muestra un modelo relacional de datos que nos ayuda a identificar los atributos de cada entidad (variables) y sus relaciones.

Figura 10: Modelo de Datos de las campañas de marketing.



Fuente: Institución Financiera, elaboración propia

Según el modelo de datos, la entidad “Persona” representa a los leads que son considerados para una o más campañas comerciales, y cada campaña puede ser desplegada por uno o más canales de venta; la información de las ventas sirve para llevar el seguimiento diario de la colocación de los productos financieros. Los leads pueden ser clientes o no clientes del banco, para los clientes se tiene la información de su rentabilidad y de sus transacciones, pero para los no clientes solo se tiene la información de su situación en el sistema financiero.

3.1.3. INFORMACIÓN HISTORICA DE LA GESTIÓN DE CAMPAÑAS

Antes de comenzar a desarrollar los modelos analíticos es necesario analizar la información histórica disponible de las campañas de marketing, el objetivo es calcular los indicadores de efectividad, gestión y contactabilidad de las diferentes campañas para cada producto financiero: Tarjetas, Instacash, Compra de Deuda, Libre Disponibilidad, Presta Bono y Descuento Planilla, desplegadas por canal de venta: Red de Agencias y Call Center.

La Tabla 11 muestra el valor de estos indicadores por producto financiero:

Tabla 11: Indicadores de Seguimiento de las campañas de marketing

Indicadores	Periodos									Promedio
	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	
Leads										
1. Tarjetas	127,470	125,751	131,199	126,673	127,102	121,970	129,310	133,914	125,069	127,606
2. InstaCash	67,951	73,864	70,103	66,138	74,014	72,562	70,349	75,086	69,765	71,092
3. Compra Deuda	37,813	39,450	35,486	44,587	32,840	39,022	39,118	43,879	38,907	39,011
4. Libre Disponibilidad	136,199	119,181	135,473	135,504	133,000	119,848	129,061	133,765	140,109	131,349
5. Presta Bono	85,393	91,215	95,212	91,625	95,276	90,268	87,610	94,595	92,548	91,527
6. Descuento Planilla	46,698	49,728	45,577	45,588	48,582	44,089	40,100	43,422	48,571	45,817
% Gestión										
1. Tarjetas	91%	90%	94%	90%	92%	90%	88%	86%	88%	90%
2. InstaCash	87%	87%	85%	85%	87%	87%	87%	82%	87%	86%
3. Compra Deuda	83%	79%	79%	78%	78%	78%	84%	84%	80%	80%
4. Libre Disponibilidad	86%	86%	84%	87%	89%	90%	88%	90%	88%	88%
5. Presta Bono	87%	84%	88%	83%	87%	84%	85%	85%	85%	85%
6. Descuento Planilla	89%	86%	88%	81%	86%	86%	83%	83%	84%	85%
% Contactabilidad										
1. Tarjetas	62%	59%	59%	61%	58%	63%	57%	63%	61%	60%
2. InstaCash	62%	59%	59%	61%	63%	58%	60%	62%	64%	61%
3. Compra Deuda	59%	62%	63%	62%	62%	64%	60%	60%	61%	61%
4. Libre Disponibilidad	56%	57%	63%	63%	58%	63%	63%	61%	61%	60%
5. Presta Bono	60%	63%	58%	63%	60%	58%	60%	62%	58%	60%
6. Descuento Planilla	63%	60%	58%	61%	59%	60%	63%	62%	60%	61%
% Efectividad										
1. Tarjetas	11%	12%	11%	11%	13%	13%	12%	15%	13%	12%
2. InstaCash	19%	15%	19%	17%	14%	15%	15%	13%	19%	16%
3. Compra Deuda	16%	18%	18%	20%	18%	15%	19%	15%	17%	17%
4. Libre Disponibilidad	11%	16%	14%	13%	18%	19%	11%	12%	17%	14%
5. Presta Bono	19%	12%	18%	18%	14%	12%	11%	17%	14%	15%
6. Descuento Planilla	19%	19%	16%	18%	16%	16%	12%	19%	16%	17%
Total Leads	501,524	499,189	513,050	510,115	510,814	487,759	495,548	524,661	514,969	
Total % Gestión	87.61%	86.22%	87.36%	85.51%	88.00%	87.13%	86.57%	85.85%	86.17%	
Total % Contactabilidad	60.08%	59.35%	59.96%	61.84%	59.27%	60.96%	60.32%	61.70%	60.79%	
Total % Efectividad	14%	14%	15%	15%	15%	15%	13%	15%	16%	

Fuente: Institución Financiera, elaboración propia

- % Gestión: Porcentaje de la cantidad de leads que fueron gestionados por los canales de venta.
- % Contactabilidad: Porcentaje de los leads gestionados que resultaron un contacto efectivo.
- % Efectividad: Es el porcentaje de los leads que aceptaron el producto financiero. El promedio de estos indicadores por producto y por canal, en los 9 meses, es:

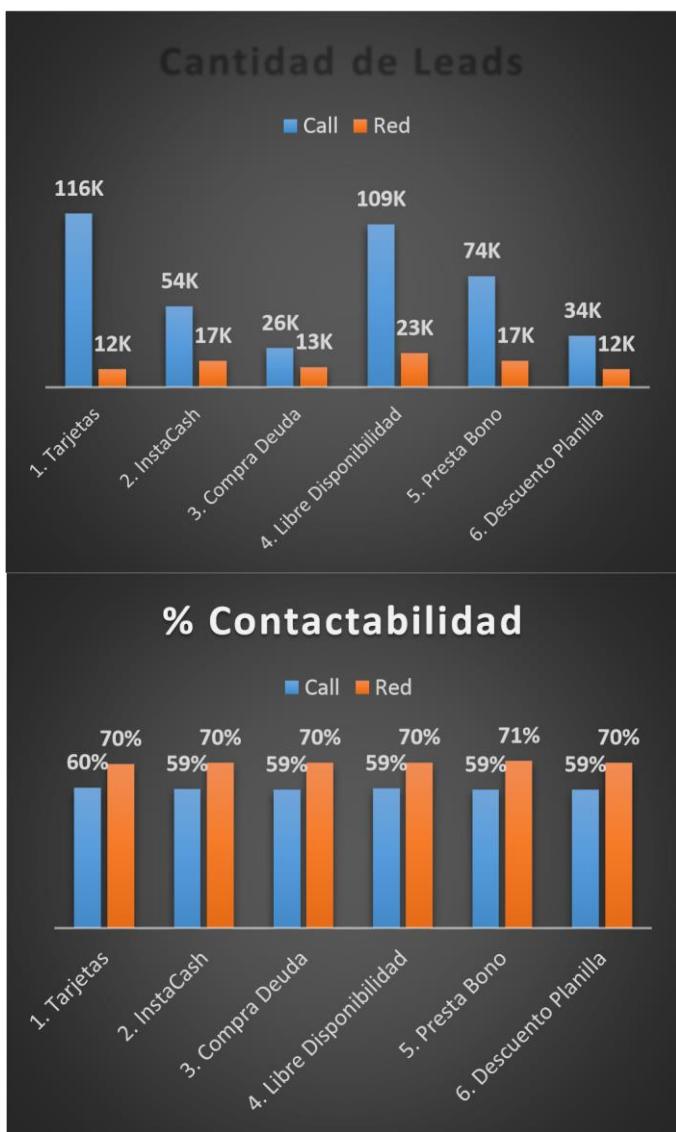
Figura 11: Cantidad de Leads por producto y Canal
producto y Canal

Figura 12: % Gestión por

Cantidad de Leads

Fuente: Institución Financiera, elaboración propia

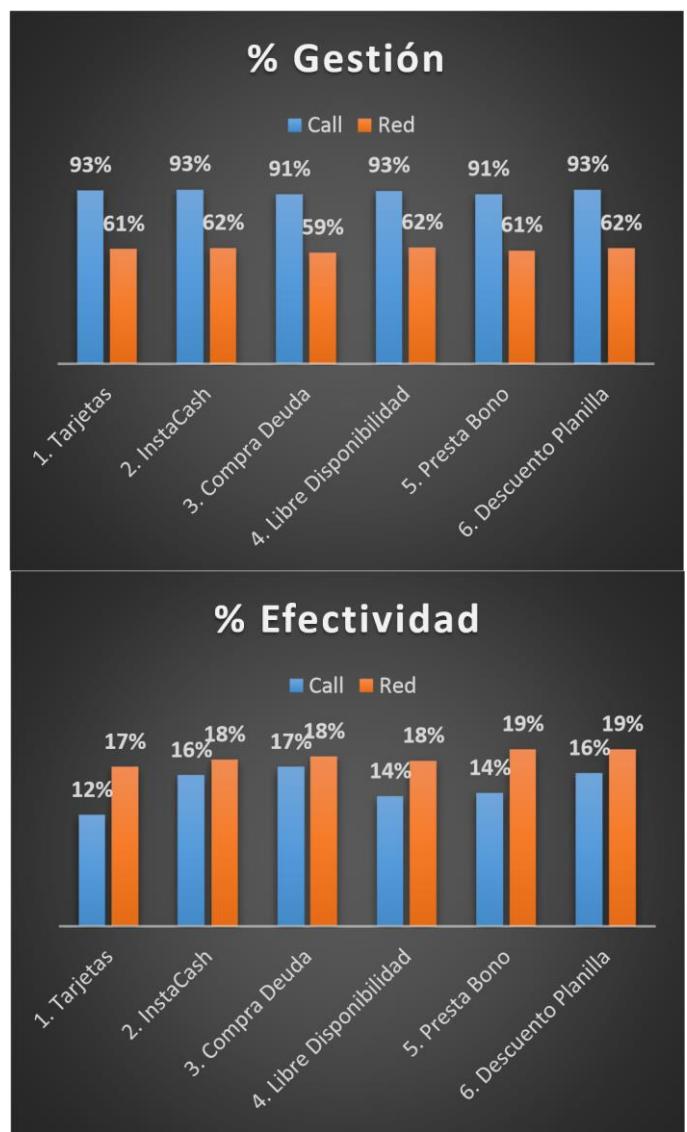
Figura 13: % Contactabilidad por producto y Canal



Fuente: Institución Financiera, elaboración propia

Fuente: Institución Financiera, elaboración propia

Figura 14: % Efectividad por producto y Canal



Fuente: Institución Financiera, elaboración propia

De las Figuras 11, 12, 13 y 14 se concluye que el “Call Center” es el canal que gestiona la mayor cantidad de leads, pero con menor porcentaje de contactabilidad respecto al canal “Red de Agencias”, lo que genera una menor efectividad. Estos indicadores serán considerados en el planteamiento del problema de optimización, ya que identifica el producto que se le debe ofrecer a un lead, a través del mejor canal de venta.

3.1.4. ANALISIS DE LA RENTABILIDAD ESPERADA

La rentabilidad esperada se calcula en base a la rentabilidad generada por los clientes que tienen el producto financiero, luego de doce meses de haberlo adquirido.

Por ejemplo, para el producto “Tarjetas” se tomó la información histórica correspondientes a los meses de marzo, abril, mayo, junio, julio y agosto del 2020, y se calculó el promedio de la rentabilidad correspondiente a 12 meses después de su adquisición (en el mes de agosto 2020 su rentabilidad fue de S/71.27). La Tabla 12 muestra la rentabilidad en nuevos soles de los meses citados anteriormente:

Tabla 12: Rentabilidad por Producto Financiero.

Producto	Rentabilidad por producto y por meses de campaña					
	Marzo	Abril	Mayo	Junio	Julio	Agosto
1. Tarjetas	65.24	82.11	68.51	70.35	54.94	71.27
2. InstaCash	163.31	143.57	137.80	188.65	160.00	137.97
3. Compra Deuda	46.54	60.67	55.87	59.11	53.12	49.20
4. Libre Disponibilidad	181.89	156.80	160.24	164.52	152.64	141.24
5. Presta Bono	200.02	185.48	210.30	173.88	148.66	187.47
6. Descuento Planilla	116.72	126.52	139.37	147.61	125.08	129.19
Promedio general	128.95	125.86	128.68	134.02	115.74	119.39

Fuente: Institución Financiera, elaboración propia

Otra variable que influye en la rentabilidad es el perfil del cliente, los clientes con un mejor perfil financiero generarán mayor rentabilidad, por lo tanto, es necesario calcular este indicador por producto y perfil. Por lo general, las entidades financieras agrupan a sus clientes, según su valor, en conjuntos relativamente homogéneos denominados segmentos. La Tabla 13 muestra la rentabilidad promedio de los últimos seis meses por producto financiero y segmento comercial.

Tabla 13: Rentabilidad en nuevos soles por Producto y Perfil financiero

Producto	Rentabilidad por producto y por segmento comercial				
	1. Beyond	2. Premium	3. Preferente	4. Personal	5. Estandar
1. Tarjetas	121.96	99.12	70.87	39.21	12.55
2. InstaCash	276.89	225.16	158.59	86.88	28.57
3. Compra Deuda	96.33	79.12	55.32	29.91	9.75
4. Libre Disponibilidad	278.75	232.22	165.78	90.78	30.25
5. Presta Bono	328.27	269.07	185.61	103.41	35.15
6. Descuento Planilla	230.75	189.48	135.29	74.48	23.73
Promedio general	222.16	182.36	128.58	70.78	23.33

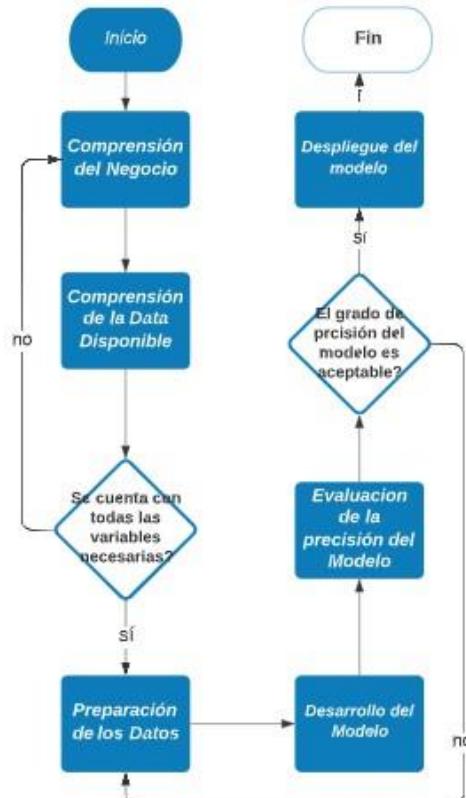
Fuente: Institución Financiera, elaboración propia

3.2. FASE II: DESARROLLO DE LOS MODELOS ANALÍTICOS

3.2.1. CONTRUCCIÓN DE LOS MODELOS DE PROPENSIÓN

Las estrategias implementadas en las campañas de marketing usan modelos estadísticos para identificar a los leads más propensos a adquirir un producto. En el flujo de la Figura 15 se muestra los pasos que se deben seguir para desarrollar un modelo analítico, desde la comprensión del negocio hasta su despliegue.

Figura 15: Diagrama de Flujo para el desarrollo de un modelo analítico.



Fuente: Institución Financiera, elaboración propia

En la FASE I, para comprender el negocio se han analizado los principales indicadores y entidades que intervienen en las campañas de marketing, además se ha diseñado un modelo de datos que consolide las principales variables. La FASE II de la propuesta de solución tiene por objetivo desarrollar los modelos analíticos que calculan la probabilidad de adquisición de un producto financiero, además construir una red neuronal que estandarice estas probabilidades.

3.2.1.1. PRE PROCESAMIENTO DE LOS DATOS

La Tabla 14 contiene todas las variables (demográficas, transaccionales, internas y del sistema financiero) utilizadas en el desarrollo de los modelos predictivos: Tabla 14: Variables del DataSet para el Desarrollo de los Modelos

Variables del DataSet para el desarrollo de los modelos				
Variables de Campaña	Demograficas	Transaccionales	Internas	SistemaFinanciero
MontoOferta	Edad	NroTrxAgencia	CantProd	ClasificacionSBS
TasaOferta	Genero	NroTrxATM	SegmentoComercial	NumEnt
Frescura	EstadoCivil	NroTrxCorresponsal	SaldoColocacion	Max_LineaTC_SSFF
ScoreRiesgos	MacroZona	NroTrxApp	SaldoCaptacion	Antiguedad_SSFF
Can_Contactabilidad_3UM	GradoInst	NroTrxWeb	SaldoTC	Entidad_Principal
	Flag_Dependiente	ImporteTrxAgencia	SaldoPP	SaldoTotal_SSFF
	IngresoNeto	ImporteTrxATM	SaldoVEH	SaldoTC_SSFF
	CantidadHijos	ImporteTrxCorresponsal	SaldoHip	SaldoPP_SSFF
	NSE	ImporteTrxApp		SaldoVEH_SSFF
	Tenencia_Inmueble	ImporteTrxWeb		SaldoHIP_SSFF
	Tenencia_Vehiculo			
	Tipo de Vehiculo			

Fuente: Institución Financiera, elaboración propia

Mensualmente se despliegan más de 400 mil leads a los canales de venta para que sean gestionados dentro de las diferentes campañas de marketing; para el desarrollo de los modelos analíticos incluiremos la información de los 9 últimos meses de campaña seleccionando una muestra representativa del 10% del universo de leads.

El pre procesamiento de los datos es importante ya que permite eliminar información redundante e irrelevante, datos ruidosos y poco confiables que afectan la capacidad de predicción de los modelos. Este proceso se divide en cuatro pasos:

- 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.

Luego de realizar el pre procesamiento de las variables independientes, se definirá la variable dependiente a predecir (Target), es decir, si una persona adquiere o no un producto financiero. El resultado de esta etapa dará como resultado el conjunto de entrenamiento para el desarrollo del modelo analítico.

Eliminación de las variables con alto porcentaje de nulos

Se busca eliminar del Data Set las variables que superen un límite máximo de valores nulos, por lo general, este límite en términos porcentuales es del 80%. Se calcula el porcentaje de valores nulos para cada variable V_i utilizando la Ecuación (22):

$$\% \text{ Valores Nulos } V_i = \frac{\text{Cantidad de registros con valor nulo para la variable } V_i}{\text{Cantidad de registros del Data Set}} \quad (22)$$

Los pasos a seguir se presentan en el siguiente algoritmo:

```

Nrows = Cantidad de filas del Data Set
Columns = Arreglo que contiene el nombre de las columnas del Data Set
Arreglo_Resultado = Ø
For var ∈ Columns
    Cantidad_Missing_Values = Cantidad Valores nulos para la Variable "var"
    Si (Cantidad_Missing_Values/Nrows > 0.8)
        Ingresar en el arreglo Arreglo_Resultado la variable "var"
    Fin Si Fin
For

```

La Tabla 15 muestra las cinco variables que superan el 80% de valores nulos. Estas variables se han identificado utilizando el algoritmo anterior (la variable Tenencia de Inmueble tiene el 99.24% de valores nulos - Por_Missing_Value -).

Tabla 15: Variables con alto porcentaje de valores nulos.

Variable	Por_Missing_Value
Tenencia_Inmueble	0.99240
SaldoVEH	0.89872
SaldoVEH_SSFF	0.86764
Tenencia_Vehiculo	0.81868
Tipo_Vehiculo	0.81868

Fuente: Institución Financiera, elaboración propia

Imputación de Valores Nulos

Luego de excluir las variables que superen el umbral del 80%, aún quedan valores nulos en el Data Set que deben ser reemplazados; esto se consigue reemplazando los valores nulos de las variables numéricas por cero y los valores nulos de las variables categóricas por el valor “NI” (No Identificado).

Utilizamos el siguiente algoritmo 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 = Ø  
For 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 For
```

La Tabla 16 muestra las cinco variables que aún contienen valores nulos que deben ser reemplazados según el tipo de variable.

Tabla 16: Variables con valores nulos a imputar.

Variable	Por_Missing_Value	Tipo_Variable
CantidadHijos	0.10190	Numerica
ScoreRiesgos	0.06014	Categorica
GradoInst	0.02194	Categorica
Edad	0.00226	Numerica
Genero	0.00002	Categorica

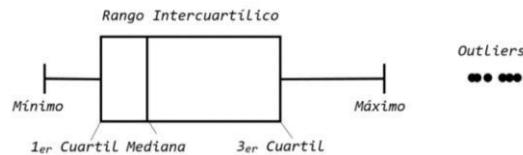
Fuente: Institución Financiera, elaboración propia

Imputación de Valores Atípicos (outliers)

Los valores atípicos son observaciones que están numéricamente más distantes al resto de los datos. En la Figura 16 se muestra la representación gráfica de los dos

tipos de valores atípicos: leves y extremos, los valores leves 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 la distancia intercuartílica.

Figura 16: Representación gráfica de los valores atípicos



Fuente: Institución Financiera, elaboración propia

El siguiente algoritmo se utiliza para imputar los valores atípicos extremos de las variables definidas por el analista:

```

Var_imputar = Arreglo de variables definidas por el analista
For 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”
    For valor ∈ Valores_Variable
        Si valor >= Máximo
            Actualizar el valor de la variable “var” con el Máximo
        Else
            No modificar el valor de la variable “var”
        Fin Si
    Fin for
    Actualizar el Data Set con los nuevos valores de la variable “var”
Fin For

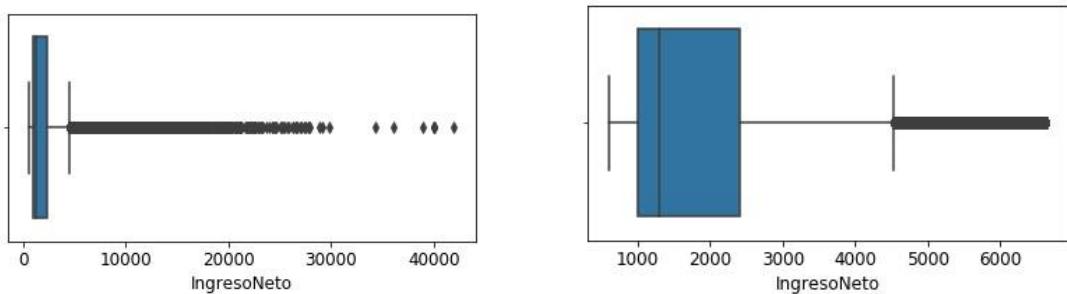
```

Por ejemplo, para la variable “IngresoNeto”, existe personas con salarios superiores a S/ 30,000, los cuales son valores atípicos ya que la mayor parte de las personas tienen salarios menores a S/ 10,000. Estos valores atípicos son imputados, como se muestra en la Figura 17, con el algoritmo descrito anteriormente:

Antes de la Imputación

Después de la Imputación

Figura 17: Efecto de la imputación de valores atípicos
 Fuente: Institución Financiera, elaboración propia



Tratamiento de las variables categóricas

Las variables categóricas son utilizadas para nombrar o categorizar una característica de una instancia; sin embargo, para que puedan ser usadas en los modelos de machine learning deben ser transformadas a variables numérica y la manera más sencilla de lograrlo es creando variables dummy. Las variables dummy son binarias y se crean a partir de los valores de la variable original. Para crear las variables dummies podemos usar el método “`get_dummies`” de Python.

Otra técnica usada para transformar los datos categóricos en valores numéricos S_i es usar la Ecuación (23):

$$S_i = f(n_i) \frac{n_{iY}}{n_i} + (1 - f(n_i)) \frac{n_Y}{n_T} \quad (23)$$

- S_i valor numérico de la variable categórica.
- n_{iY} es la cantidad de casos $Y=1$, tal que $X = X_i$.
- n_i es el total de casos $X = X_i$.
- n_Y es el total de casos $Y = 1$. n_T es el total de registros del Data Set.
- $f(n_i)$ es una función paramétrica cuyos valores se encuentran entre [0 - 1].

$f(n_i)$ es una función que permite calcular la probabilidad de que “ Y ” sea igual a 1 para cualquier valor de la variable categórica X_i . El valor de $f(n_i)$ se calcula a partir de la Ecuación (24):

$$f(n_i) = \frac{1}{1 + e^{-\frac{(n_i - k)}{f}}} \quad (24)$$

Donde k y f son parámetros pre definidos por el analista. En la Tabla 17 se muestra el resultado de aplicar este método para transformar las variables categóricas del Data Set (Segmento Comercial, Grado de Instrucción y Estado Civil) en valores numéricos que representan la probabilidad de adquisición del producto.

Tabla 17: Transformación de variables categóricas

SegmentoComercial	T_SegmentoComercial	GradolInst	T_GradolInst	EstadoCivil	T_EstadoCivil
1. Beyond	0.536656	Superior	0.456276	SOLTERO	0.448138
2. Premium	0.485080	Tecnica	0.449458	CASADO	0.442049
3. Preferente	0.436201	Secundaria	0.441053	SEPARADO	0.435248
4. Personal	0.392466	Primaria	0.406192	VIUDO	0.431795
5. Estandar	0.344029	Illetrado	0.401261	DIVORCIADO	0.428916

Fuente: Institución Financiera, elaboración propia

3.2.1.2. DEFINICIÓN DEL TARGET DEL MODELO

El target de los modelos predictivos es una variable dicotómica o binaria; esta variable representa al evento que se desea predecir. En nuestro caso este evento es la adquisición de un producto financiero y es representado de la siguiente forma:

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

Donde y_i es la variable dependiente (TARGET) que representa el evento a predecir. Para determinar si es necesario aplicar técnicas de balanceo a la variable dependiente, debemos medir la estabilidad del evento durante seis meses de campaña, calculando la efectividad (Efectividad = Ventas/Leads).

La Tabla 18 muestra la estabilidad de la efectividad en el tiempo y permite validar si el porcentaje de unos y ceros (Efectividad) se encuentra correctamente balanceado; debido a que este porcentaje tiene un alto valor el modelo podrá diferenciar

correctamente el perfil del lead que adquiere el producto. Las efectividades de las campañas comerciales en el sistema financiero están entre 5% y el 25%.

Tabla 18 : Efectividad de una campaña de marketing

	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Nº Leads	21,604	27,871	25,266	20,703	25,312	25,396
Nº Ventas	3,888	5,295	3,519	3,933	4,809	4,317
Efectividad	18%	19%	17%	19%	19%	17%

Fuente: Institución Financiera, elaboración propia

Los modelos analíticos, que se van a desarrollar como parte de la solución, buscan calcular la probabilidad de que el evento ocurra. Los modelos de regresión logística son comúnmente usados para tal fin.

Aunque el modelo de regresión logística es el más usado, también se usan otros modelos para calcular dicha probabilidad, tales como los modelos XGBoost, LightGBM, RandomForest, etc.

3.2.1.3. SELECCIÓN DE VARIABLES PRINCIPALES

Actualmente se cuenta con 45 variables independientes en el Data Set, las cuales se usarán para explicar el evento de adquisición de un producto financiero, sin embargo, es necesario establecer un criterio que reduzca la cantidad de variables y permita seleccionar las principales. El criterio utilizado es el índice Gini (Ecuación (25)):

$$Gini = \frac{\sum_{i=1}^m |p_{gi} - q_{gi}|}{\sum_{i=1}^m q_{gi}} \quad (25)$$

Donde:

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

- $p_{gi} = \frac{\text{Suma acumulativa de la cantidad de elementos hasta el grupo } gi}{\text{Cantidad Total de elementos}}$
- $q_{gi} = \frac{\text{Suma acumulativa de eventos positivos (Y=1) hasta el grupo } gi}{\text{Cantidad Total de eventos positivos}}$

El índice gini permite calcular, para cada variable del Data Set, el porcentaje de discriminación de los eventos positivos (Y=1) respecto de los negativos (Y=0), para calcular este índice usaremos 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 = Ø
For var ∈ Var_Originales
    Si var es Numerico
        Tabla_Agrup = Matriz formada agrupando quintiles de la variable
    Else
        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 For

```

La Tabla 19 muestra las 20 variables independientes con mayor índice Gini, seleccionadas utilizando el algoritmo anterior:

Tabla 19: Las 20 variables independientes con mayor índice Gini

Tipo_Variable	Variable	Gini
Numerica	ImporteTrxATM	0.496962
Numerica	ImporteTrxApp	0.377282
Numerica	ImporteTrxAgencia	0.326370
Numerica	ImporteTrxCorresponsal	0.248445
Numerica	NroTrxATM	0.210148
Numerica	SaldoCaptacion	0.203629
Numerica	CantProd	0.183823
Numerica	NroTrxApp	0.175924
Categorica	SegmentoComercial	0.159351
Numerica	ImporteTrxWeb	0.155311
Numerica	SaldoTC	0.133015
Numerica	SaldoColocacion	0.105285
Numerica	NroTrxAgencia	0.087502
Numerica	SaldoPP	0.067264
Numerica	NroTrxCorresponsal	0.059948
Numerica	TasaOfertada	0.054538
Numerica	NroTrxWeb	0.051213
Categorica	NSE	0.050866

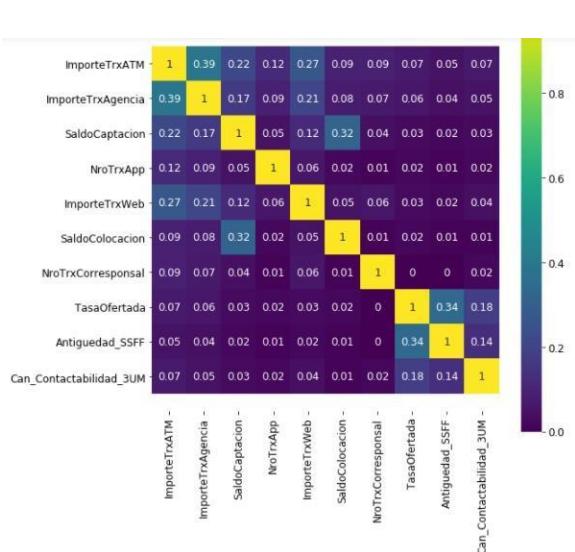
Fuente: Institución Financiera, elaboración propia

En la Tabla 20 se muestran las variables finales con mayor índice Gini y con una correlación máxima entre sí del 40%. En la Figura 18 se muestra la matriz de correlación entre las variables finales seleccionadas.

Figura 18: Matriz de correlaciones de las variables seleccionadas

Tabla 20: Tabla de las variables Seleccionadas

Tipo_Variable	Variable	Gini
Numerica	ImporteTrxATM	0.496962
Numerica	ImporteTrxAgencia	0.326370
Numerica	SaldoCaptacion	0.203629
Numerica	NroTrxApp	0.175924
Categorica	SegmentoComercial	0.159351
Numerica	ImporteTrxWeb	0.155311
Numerica	SaldoColocacion	0.105285
Numerica	NroTrxCorresponsal	0.059948
Numerica	TasaOfertada	0.054538
Categorica	NSE	0.050866
Numerica	Antiguedad_SSFF	0.039865
Numerica	Can_Contactabilidad_3UM	0.039555



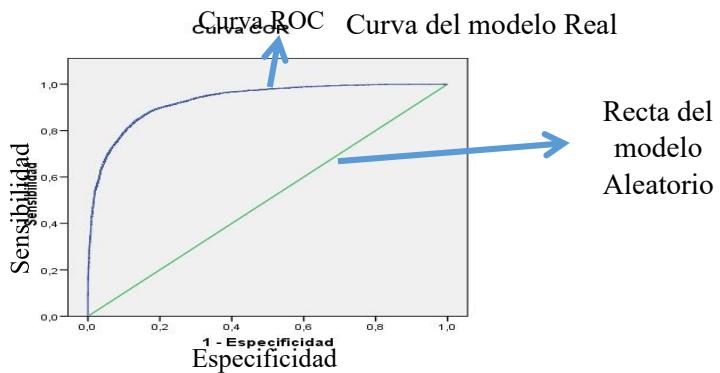
Fuente: Institución Financiera, elaboración propia

Fuente: Institución Financiera, elaboración propia

3.2.1.4. DESARROLLO DE LOS MODELOS PREDICTIVOS

En las secciones anteriores se pre procesan los datos, se define el Target del modelo y se selecciona las variables principales, en esta sección desarrollamos los modelos predictivos. La metodología para desarrollar los modelos predictivos es la siguiente:

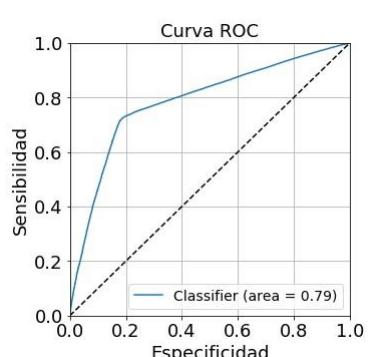
1. Definimos la base de entrenamiento (Train) correspondiente al 70% de la población y la base de validación (Test) correspondiente al 30% restante.
2. Seleccionamos una métrica para medir la precisión de los modelos analíticos, en el caso de estudio la métrica seleccionada es el “Área bajo la Curva ROC” (AUC, por sus siglas en inglés):



3. La Tabla 21 muestra el AUC de cuatro algoritmos diferentes: Regresión Logística, LightGBM (LGBM), XGBoost (XGB) y Support Vector Classifier (SVC), Gradient Boosting Classifier (GBC). La Figura 19 muestra la curva ROC del modelo LightGBM, seleccionado por tener el mayor AUC.

Figura 19: Curva ROC del modelo
Tabla 21: Comparación de AUCs por algoritmo

Modelo	AUC
Modelo LightGBM	0.785380
Modelo XGBoost	0.770975
Regresion Logistica	0.752801
Super Vector Classifier	0.725670



Fuente: Institución Financiera, elaboración propia

Fuente: Institución Financiera, elaboración propia

3.2.1.5. VALIDACIÓN DEL MODELO PREDICTIVO

De la sección anterior podemos concluir que el algoritmo LGBM es el que tiene mayor precisión al comparar los AUCs de los 4 modelos desarrollados, el siguiente paso es la validación de los resultados usando dos métodos diferentes (métodos 1 y 2):

1. Comparamos las métricas AUC obtenidas con las bases de entrenamiento y de validación, estos valores deben ser similares para confiar en los resultados del modelo; si no lo son, incurrimos en un error de sobre ajuste.

En la Tabla 22 se demuestra que el modelo LightGBM debe ser usado para cálculo de la probabilidad de adquisición; en tanto el AUC del Train es aproximadamente igual al AUC del Test.

Tabla 22: Validación del modelo LightGBM

Modelo	AUC del Train	AUC del Test
Modelo LGBM	78.53%	76.23%

Fuente: Institución Financiera, elaboración propia

2. En base a cortes de probabilidad, se divide el universo del Data Set en diez conjuntos iguales denominados deciles de 5000 elementos; donde los elementos del décimo decil tienen la mayor probabilidad de apertura. La Tabla 23 demuestra que para cada decil existe una correspondencia entre la probabilidad obtenida del modelo y la efectividad real de la campaña.

Tabla 23: Deciles del Modelo LightGBM

Decil	Cantidad	Apertura	Prob	Efec	Lift	Acumulado %
10	5,000	4,420	81%	88%	1.90	18%
9	5,000	4,056	79%	81%	1.74	36%
8	5,000	3,684	78%	74%	1.58	53%
7	5,000	3,417	76%	68%	1.47	69%
6	5,000	2,420	41%	48%	1.04	78%
5	5,000	1,200	24%	24%	0.52	83%
4	5,000	1,166	22%	23%	0.50	87%
3	5,000	1,046	22%	21%	0.45	92%
2	5,000	988	21%	20%	0.42	96%
1	5,000	901	21%	18%	0.39	100%

Fuente: Institución Financiera, elaboración propia

3.2.2. ESTANDARIZACIÓN DE PROBABILIDADES

En secciones anteriores hemos descrito los pasos a seguir para desarrollar los modelos predictivos para cada producto financiero; sin embargo, para que las probabilidades sean usadas en el problema de optimización de asignación de leads, es necesario estandarizar estas probabilidades para que sean comparables. Usaremos el algoritmo de redes neuronales para estandarizar las probabilidades obtenidas de los modelos estadísticos. La Figura 20 muestra el código en Python utilizado para entrenar una red neuronal con la siguiente arquitectura: 6 neuronas de entrada (una neurona por cada probabilidad de adquisición de cada producto financiero), 4 capas ocultas con 30 neuronas cada uno y con funciones *Sigmoideas* o *Relu* de activación, la capa de salida tendrá 6 neuronas de salida (correspondientes a los Targets binarios que indica si la persona adquirió o no el producto financiero).

Figura 20: Código en Lenguaje Python para el entrenamiento de una Red Neuronal

```
#### Modelo de Redes Neuronales
## Probar Drop Out en cada
model = Sequential()
# Input put layer and First Hidden Layer
model.add(Dense(30, input_dim=x_train_1.shape[1], activation='sigmoid'))
model.add(Dense(30, activation ='relu', kernel_initializer = 'uniform'))
model.add(Dense(30, activation = 'sigmoid', kernel_initializer = 'uniform'))
model.add(Dense(30, activation ='relu', kernel_initializer = 'uniform'))
# Out put layer
model.add(Dense(y_train_1.shape[1], activation='sigmoid'))

# compile model
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['AUC']
)

# fit the model
model.fit(x_train_1, y_train_1, epochs=80, batch_size=32)

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

Fuente: Institución Financiera, elaboración propia

Las probabilidades obtenidas de las redes neuronales están ajustadas a la realidad y están estandarizadas ya que son obtenidas de un mismo algoritmo, estas probabilidades se usarán en el planteamiento del problema de optimización.

3.2.3. VALIDACIÓN DE LOS RESULTADOS DE LA RED NEURONAL

Para validar la eficiencia de la red neuronal se compara el Gini y el AUC de las probabilidades obtenidas a partir de los modelos predictivos (en la Tabla 24 se lo denomina AUC y Gini Original), con el Gini y el AUC de las probabilidades obtenidas como output de la red neuronal (en la Tabla 24 se lo denomina AUC y Gini Ajustado).

El Gini y el AUC de los dos métodos se calculan mediante el siguiente algoritmo:

```

Arreglo_Productos = Arreglo que contiene los nombres de los 6 productos
Y_Target = Ø
Prob_Rn = Ø
Prob_Orig = Ø
Matriz_Resultado = Ø
For prod ∈ Arreglo_Productos
    Si prod = "Tarjetas"
        Y_Target = Target binario de la adquisición de Tarjeta de Crédito (TC)
        PROB_Rn = Probabilidades de adquisición de TC obtenidas de la Red Neuronal
        PROB_Orig = Probabilidades originales de adquisición de TC
    Fin Si
    Si prod = "Instacash"
        Y_Target = Target binario de la adquisición del producto Instacash (XL)
        PROB_Rn = Probabilidades de adquisición de XL obtenidas de la Red Neuronal
        PROB_Orig = Probabilidades originales de adquisición de XL
    Fin Si
    Si prod = "Libre Disponibilidad"
        Y_Target = Target binario adquisición del producto Libre Disponibilidad (LD)
        PROB_Rn = Probabilidades de adquisición de LD obtenidas de la Red Neuronal
        PROB_Orig = Probabilidades originales de adquisición de LD
    Fin Si
    Si prod = "PrestaBono"
        Y_Target = Target binario de la adquisición del producto PrestaBono (PA)
        PROB_Rn = Probabilidades de adquisición de PA obtenidas de la Red Neuronal
        PROB_Orig = Probabilidades originales de adquisición de XL
    Fin Si
    Si prod = "DescuentoPlanilla"
        Y_Target = Target binario de adquisición del producto DescuentoPlanilla (DXP)
        PROB_Rn = Probabilidades de adquisición de DXP obtenidas de Red Neuronal
        PROB_Orig = Probabilidades originales de adquisición de DXP
    Fin Si
    Valor_AUC_RN = AUC(Y_Target, PROB_Rn)
    Valor_AUC_Orig = AUC(Y_Target, PROB_Orig)
    Agregar en Matriz_Resultado el Arreglo {prod, Valor_AUC_Orig, Valor_AUC_RN}
Fin For
```

Tabla 24: Ginis de los modelos Estadísticos vs Ginis de la Red Neuronal

Producto Financiero	AUC_Original	GINI_Original	AUC_Ajustada	GINI_Ajustada
Tarjetas	0.6892564772016592	0.37851295440331834	0.745312712869579	0.490625425739158
Xtralinea	0.8166965414521468	0.6333930829042935	0.9282681441941151	0.8565362883882301
Compra Deuda TC	0.6331281724665915	0.26625634493318295	0.7449051635025267	0.48981032700505334
Libre Disponibilidad	0.6963799082325058	0.3927598164650117	0.7971721160455323	0.5943442320910646
PrestaBono	0.7223939816807984	0.44478796336159676	0.806624199812013	0.6132483996240261
Descuento Planilla	0.5679528614327631	0.13590572286552627	0.852606821127405	0.7052136422548101

Fuente: Institución Financiera, elaboración propia

La Tabla 24 muestra que el AUC y el Gini de la red neuronal es mayor.

3.3. FASE III: OPTIMIZACIÓN DE LA ASIGNACIÓN DE LEADS

El proceso de asignación de leads, dentro de las campañas comerciales, consiste en generar una base de datos que contenga la información de las personas a los cuales se les ofrece un producto financiero a través de los canales de venta.

Para generar esta base de datos es necesario calcular la probabilidad de adquisición y la rentabilidad esperada para cada uno de los leads que intervienen en la campaña comercial, según estas dos variables seleccionamos un sub conjunto de leads con mayor propensión y rentabilidad a los cuales se dirige la campaña.

3.3.1. PLANTEAMIENTO MATEMÁTICO

Actualmente, el proceso de asignación de leads se realiza según criterios empíricos de negocio, el objetivo del proyecto es darle un sustento matemático.

3.3.1.1. FUNCIÓN OBJETIVO

La función objetivo (Ecuación (26)) maximiza la rentabilidad y la conversión de las campañas comerciales, sujeto a restricciones de negocio, en una empresa del sector financiero, nuestro problema tiene 6 productos financieros (Tarjetas - TC, InstaCash - XL, Compra Deuda - CD, Libre Disponibilidad - LD, PrestaBono – PA y Descuento Planilla – DXP) y 2 canales de venta (Red y Call).

$$\text{Max } \sum_{(i \text{ in } I, j \text{ in } J)} (\text{PotCanal}_{ij} * \text{GestCanal}_j * \text{ContCanal}_j * \text{Prob}_{ij} * \text{Rent}_j - \text{CostCanal}_j) * X\text{Canal}_{ij} \quad (26)$$

Donde:

PotCanal_{ij} : Potencial por producto y canal, da origen a 12 coeficientes:

PotRedTC, PotCallTC, PotRedXL, ..., PotCallDXP.

GestCanal_j : Porcentaje de gestión por canal y producto, genera 12 coeficientes:

GestRedTC, GestCallTC, GestRedXL, ..., GestCallDXP.

ContCanal_j : Porcentaje de contacto por canal y producto, genera 12 coeficientes:

ContRedTC, ContCallTC, ContRedXL, ..., ContCallDXP.

$Prob_j$: Probabilidad de adquisición por producto financiero, genera 6 coeficientes:

ProbTC, ProbXL, ..., ProbDXP.

Ing_j : Ingreso esperado por producto financiero, genera 6 coeficientes:

IngTC, IngXL, ..., IngDXP.

$CostCanal_j$: Costo de cada Canal por gestionar leads, genera 2 coeficientes:

CostRed, CostCall.

Los coeficientes de la función objetivo han sido calculados en las secciones anteriores de la propuesta de solución.

3.3.1.2. VARIABLES DE DECISIÓN

Son variables binarias ($XCanal_{ij}$) que pueden tomar dos valores 1 o 0, el valor de 1 indica que el lead debe asignarse a la campaña comercial y enviarse al canal (Red o Call), el valor de 0 indica que el lead no debe ser asignado ni enviado.

$XCanal_{ij}$: Variable de decisión por producto y canal, genera 12 variables binarias:

XRedTC, XCallTC, XRedXL, ..., XCallDXP. (12 variables)

3.3.1.3. RESTRICCIONES DE NEGOCIO

1. La cantidad de leads asignadas por producto y canal debe estar dentro de un mínimo y máximo permisible (Inecuaciones (27) y (28)):

$$\sum XCanal_{ij} \geq MinAsignadosCanal_j \quad (27)$$

$$\sum XCanal_{ij} \leq MaxAsignadosCanal_j \quad (28)$$

La capacidad de los canales por producto son los límites superiores y el mínimo de las cantidades enviadas en los tres meses anteriores determina los límites inferiores de las restricciones; estos valores son inputs del modelo. Cada variable da origen a 2 restricciones, como son 12 variables la cantidad total de restricciones serán 24.

2. A cada lead se le debe asignar un producto financiero (Inecuación (29)), siempre y cuando el perfil del lead determine que puede ser incluido en la campaña:

$$(X_{Canal_{ij}} \leq PotCanal_{ij}) \quad (29)$$

Por ejemplo, $PotRedTC$ (Canal Red y Producto TC) = 0 quiere decir que el lead no puede ser asignado a la Red ni a la campaña de Tarjetas, en cambio si $PotRedTC$ (Canal Red y Producto TC) = 1 significa que el lead si puede ser asignado a la Red y a la campaña de Tarjetas, esto da origen a 12 restricciones para cada lead que participa en la campaña.

3. Cada lead no puede superar una cantidad máxima de campañas comerciales:

$$\sum_{(j \text{ in } J)} X_{Canal_{ij}} \leq Max\ CampañasPorLead \quad (30)$$

La cantidad máxima de campañas por lead es determinada por el analista y es directamente proporcional a la cantidad de llamadas que recibirá el lead, esta restricción evita el hostigamiento excesivo de los canales hacia los leads. Esto da origen a una restricción por lead procesado (Inecuación (30)).

4. Cada lead, por campaña, puede ser enviado a un solo canal:

$$X_{RedTC} + X_{CallTC} \leq 1 \quad (31)$$

$$X_{RedXL} + X_{CallXL} \leq 1 \quad (32)$$

....

$$X_{RedDXP} + X_{CallDXP} \leq 1 \quad (38)$$

Por ejemplo, La restricción ($X_{RedTC} + X_{CallTC} \leq 1$) quiere decir que si un lead es seleccionado para la campaña de Tarjetas de Call Center, este lead no podría ser seleccionado para la misma campaña dentro la Red de Agencias. Esto da origen a 6 restricciones por lead procesado (de la Inecuación (31) hasta la Inecuación (32)).

3.3.2. SOLUCIÓN DEL PROBLEMA USANDO PROGRAMACIÓN LINEAL ENTERA

El resumen del planteamiento matemático detallado en la sección anterior es:

Función Objetivo:

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

Sujeto A:

1. Para todo Lead i y Campaña j :

$$\sum XCanal_{ij} \geq \text{MinAsignadosCanal}_j \text{ y}$$

$$\sum XCanal_{ij} \leq \text{MaxAsignadosCanal}_j$$

2. Para todo Lead i y Campaña j :

$$(XCanal_{ij} \leq PotCanal_{ij})$$

3. Para todo Lead i :

$$\sum_{(j \in J)} XCanal_{ij} \leq \text{Max CampañasPorLead}$$

4. A cada lead i y campaña j , le corresponde un solo canal:

$$XRedTC + XCallTC \leq 1 \dots$$

$$XRedDXP + XCallDXP \leq 1$$

Variables de Decisión:

$XRed_{ij}$:

Si el Lead i es asignado a la campaña j y desplegado a la Red.

$XCall_{ij}$:

Si el Lead i es asignado a la campaña j y desplegado al canal Tele-Marketing.

El input del modelo de optimización es un DataSet que contiene la información de 500 mil leads de las 6 campañas comerciales, este DataSet brinda lo siguiente:

- 1 columna que representa el identificador de cada registro.
- 12 columnas que representan el potencial por producto y canal, estas columnas toman el valor de 1 o 0, indicando si al lead “ i ” se le puede ofrecer el producto “ j ” y puede ser gestionado por canal “ k ”.
- 6 columnas que representan las probabilidades de adquisición de cada lead “ i ” para el producto “ j ”.
- 6 columnas que representan la rentabilidad esperada en caso el lead “ i ” adquiera el producto “ j ”, esta rentabilidad depende del segmento de mercado al cual pertenece el lead.

La Tabla 25 muestra el DataSet utilizado en el modelo de optimización para 10 leads.

Tabla 25: Muestra del Data Set que contiene la información de los leads

Id_Cliente	Prob_TC	Prob_XL	Prob_CD	Prob_LD	Prob_PA	Prob_DXP	PotTC_Red	PotTC_Call	...	RevenueCD	RevenueLD	RevenuePA	RevenueDXP
1	0.299127	0.301270	0.297424	0.305120	0.313034	0.295540	0	1	...	90	300	380	250
2	0.286997	0.298490	0.269392	0.296178	0.291184	0.281376	1	1	...	10	80	120	80
3	0.285065	0.304365	0.278457	0.300852	0.294672	0.271029	1	0	...	10	80	120	80
4	0.704251	0.673227	0.740139	0.704888	0.706290	0.708963	0	0	...	40	160	240	130
5	0.284767	0.296625	0.269880	0.293737	0.291110	0.275214	0	0	...	90	300	380	250
6	0.282606	0.302375	0.273394	0.293051	0.297418	0.280619	1	1	...	90	300	380	250
7	0.709316	0.694136	0.725154	0.695498	0.681337	0.729743	1	1	...	40	160	240	130
8	0.341641	0.354282	0.333523	0.322282	0.335633	0.355081	0	1	...	10	80	120	80
9	0.721192	0.714179	0.746387	0.725895	0.692824	0.719154	0	1	...	90	300	380	250
10	0.284906	0.306757	0.282346	0.301025	0.307505	0.278715	1	1	...	40	160	240	130

Fuente: Institución Financiera, elaboración propia

Se utiliza también un archivo de texto que contiene la siguiente información:

- El porcentaje de gestión y contactabilidad por producto y canal.
- Las cantidades mínimas y máximas de leads que deben ser asignados por producto y canal.
- El costo de gestionar un lead por producto y canal.

Para resolver el problema de programación lineal entera usaremos la librería de Python PULP la cual encuentra la solución óptima global mediante el algoritmo de plano de corte. Detallaremos los pasos del algoritmo:

1. Importamos la librería PULP.
2. Cargamos el Data Set con la información de los leads y el archivo de texto con los valores de los coeficientes en la función objetivo.
3. Realizamos un pre procesamiento para adecuar el formato de la información de los leads (Data Set).
4. Definimos un nombre adecuado para el problema y además se especifica el objetivo, en nuestro caso es maximizar.
5. Definimos las LpVariables que son usadas en la función objetivo y en las restricciones del problema.
6. Codificamos la función objetivo en términos de las LpVariables y los coeficientes de gestión, contactabilidad y Rentabilidad.

7. Codificamos las restricciones, en función al planteamiento matemático del problema.
8. Usamos al método “Solve” para encontrar la solución óptima del problema, validamos el status de la solución y calculamos el tiempo de ejecución.
9. Si el status de la solución es igual a “Optimal”, calculamos el valor de la función objetivo y guardamos la solución (asignación de los leads) en un archivo texto.
La Tabla 26 muestra la solución del problema de optimización, ésta es un Data Set que contenga “1” y “0”, donde el valor de “1” significa que al lead “i” se le debe ofrecer el producto “j” mediante el canal de venta “k” y “0” es caso contrario.

Tabla 26: Muestra de la Solución usando Programación Lineal Entera

XRed_TC	XCall_TC	XRed_XL	XCall_XL	XRed_CD	XCall_CD	XRed_LD	XCall_LD	XRed_PA	XCall_PA	XRed_DXP	XCall_DXP	ID_Cliente
0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	0	0	1	0	0	3
0	0	0	0	1	0	0	1	0	0	0	0	4
0	0	0	0	0	0	1	0	0	0	0	1	5
0	0	0	0	0	0	0	0	1	0	0	1	6
0	0	0	1	0	0	0	0	0	0	1	0	7
0	0	1	0	0	0	0	0	0	0	0	1	8
0	0	0	1	0	0	1	0	0	0	0	0	9

Fuente: Institución Financiera, elaboración propia

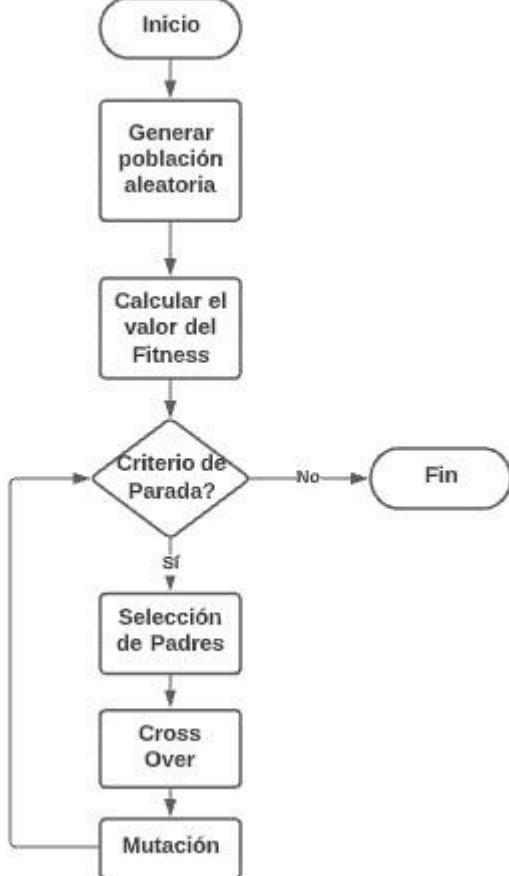
Actualmente el proceso de asignación de leads se realiza de acuerdo a reglas de experto, nuestro objetivo es reemplazar este método empírico de asignación por un modelo de optimización que use conceptos de programación lineal entera.

3.3.3. SOLUCIÓN DEL PROBLEMA MEDIANTE ALGORITMOS GENÉTICOS

Un algoritmo genético es un algoritmo de búsqueda basado en la mecánica de la selección natural; estos algoritmos hacen evolucionar una población de individuos a través de acciones aleatorias semejantes a las que actúan según la teoría de evolución biológica como mutaciones y recombinaciones genéticas. Además, se

realiza una selección de acuerdo con alguna función de aptitud que determina cuales son los individuos mas aptos para sobrevivir, los cuales se convertirán en los nuevos padres y estos a su vez crean nuevos hijos. En la Figura 21 se muestran los pasos a seguir para implementar una solución basada en algoritmos genéticos. Figura 21:

Diagrama de Flujo de una Algoritmo Genético



Fuente: Institución Financiera, elaboración propia
 Primero se construye un algoritmo que calcule el valor de la función fitness, el cual nos servirá para identificar a los mejores individuos de la población:

```

Input_Leads = Matriz con la información de los leads
Producto_Canal = Matriz con las combinaciones de producto y canal
Input_parametros = Matriz con los valores de los indicadores
Input_Solucion = Matriz con la solución propuesta
FuncionObejtivo = 0
For lead ∈ Input_Leads
  For ProdCanal ∈ Producto_Canal
    Pot = valor del potencial por ProdCanal para el lead
    Rent = valor de la rentabilidad por ProdCanal para el lead
    Prob = valor de la rentabilidad por ProdCanal para el lead
    Costo = valor del costo por ProdCanal
  
```

```

X = valor de la variable de decisión por ProdCanal para el lead
FuncionObjetivo = FuncionObjetivo + (Pot * Rent * Prob * X – Costo)
Fin For
Fin For
Devolver FuncionObjetivo

```

El siguiente algoritmo valide la factibilidad de una solución verificando si cumple con todas las restricciones:

```

Cumple_Min_Max = Valida si la solución cumple con los límites por producto y canal.
Cumple_Potencial = Verifica si la solución cumple con los criterios de potencial por producto, canal y lead.
Cumple_MaxOfertas = Verifica si la solución cumple con la cantidad máxima de productos a ofrecer por lead.
Cumple_UnCanal_UnLead = Verifica si la solución cumple con la condición de un canal por lead.
Cumple_Restricciones = False
Si Cumple_Min_Max = True and Cumple_Potencial = True and Cumple_MaxOfertas
= True and Cumple_UnCanal_UnLead = True
    Cumple_Restricciones = True
Fin Si
Devolver Cumple_Restricciones

```

Luego se construye un algoritmo que crea una población de soluciones aleatorias que cumplen con las restricciones del negocio:

```

Padres = Ø
CantidadPadres = Cantidad de individuos de la población
I = 1
While I ≤ CantidadPadres
    Padre = Generar solución aleatoria factible
    Agregar “Padre” en la Matriz de “Padres”
    I = I + 1
Fin While
Devolver Padres

```

Para evaluar la función fitness de cada individuo de la población de soluciones aleatorias, se utiliza el siguiente algoritmo:

```

Cantidad_Padres = Cantidad de individuos de la población
Matriz_Fitness = 0
For Padre ∈ Padres
    ValorFitness = Calcula el valor del fitness del padre
    Agregar en la Matriz Fitness el valor “ValorFitness”
Fin For
    Matriz_Mejores_Padres = Seleccionamos los padres con los mejores fitness
Devolver Matriz_Mejores_Padres

```

El algoritmo que acontinuación se indica detalla los pasos a seguir para el Cross Over de dos padres de la población de individuos:

```
Padre1_Partial_Izquierda = son las columnas de lado izquierdo del padre 1  
Padre1_Partial_Derecho = son las columnas de lado Derecho del padre 1  
Padre2_Partial_Izquierda = son las columnas de lado izquierdo del padre 2  
Padre2_Partial_Derecho = son las columnas de lado Derecho del padre 2  
NuevoPadre1= Juntamos Padre1_Partial_Izquierda y Padre2_Partial_Derecho  
NuevoPadre2= Juntamos Padre2_Partial_Izquierda y Padre1_Partial_Derecho  
Devolver NuevoPadre1 y NuevoPadre2
```

Los algoritmos anteriores nos servirán para construir el siguiente algoritmo basado en evolución genética:

```
NumIteraciones = Cantidad de Iteraciones  
Fitness_Solucion = 0 y Iteracion = 1  
Solucion = Ø  
Padres = Crear una población de soluciones factibles  
While Iteracion <= NumIteraciones  
    Mejores_Padres = Matriz de los padres con los mejores fitness  
    Fitness_Temp = 0  
    NuevaPoblacion = Ø  
    Indice_Padres = Lista de índice de los mejores padres  
    For index ∈ (Indice_Padres)  
        Padre1 = Padre con el índice “index”  
        Padre2 = Padre con el índice “index + 1”  
        NuevoPadre1, NuevoPadre2 = Cross Over de Padre 1 y Padre 2  
        Agregamos a la Matriz “NuevaPoblacion” el NuevoPadre1 y NuevoPadre2  
    End For  
    For individuo ∈ (NuevaPoblacion)  
        Mutamos cada individuo según una probabilidad de mutación  
    End For  
    Fitness_Temp=Fitness del mejor individuo dentro de la “NuevaPoblacion”  
    Cumple_Restricciones = Valida si el mejor individuo cumple con todas las restricciones.  
    Si Fitness_Temp > Fitness_Solucion and Cumple_Restricciones = True  
        Solucion = Mejor Individuo de la población  
        Fitness_Solucion = Fitness_Temp  
    Fin Si  
    Padres = NuevaPoblacion  
    Iteracion = Iteracion + 1  
Fin While
```

ANÁLISIS Y VALIDACIÓN DE RESULTADOS

La validación de la eficacia del “**Modelo de Optimización**” se realiza por comparación entre los resultados de la aplicación del **modelo** en el despliegue mensual de las campañas comerciales y los resultados de la utilización de las

“**Reglas de Experto**” en la implementación de estas campañas. Pasos a seguir:

Paso 1: Validación de la calidad de predicción de la red neuronal.- Para realizar esta validación se utiliza la métrica GINI (el GINI es un indicador que mide la precisión de los modelos analíticos); si el GINI de la red neuronal se mantiene aproximadamente constante en los meses de puesta en producción (octubre, noviembre y diciembre), entonces la calidad de predicción de la red es constante en el tiempo.

La Tabla 27 muestra el GINI teórico obtenido en el entrenamiento de la red neuronal y los GINIs calculados con los resultados de cada campaña comercial en los meses de implementación. Se observa que el GINI es constante en el tiempo:

Tabla 27: Validación de la calidad de predicción de la red neuronal

Campaña Comercial de:	GINI Teórico	GINI Real (Meses de Implementación)		
		Octubre	Noviembre	Diciembre
Tarjetas	74%	72%	74%	76%
Xtralinea	86%	87%	88%	85%
Compra de Deuda TC	75%	73%	74%	74%
Libre Disponibilidad	79%	77%	78%	76%
PrestaBono	82%	80%	81%	82%
Descuento Planilla	85%	82%	84%	85%

Fuente: Institución Financiera, elaboración propia

Paso 2: El área de riesgos, de su base de datos, seleccionó para el mes de:

- Octubre: 454,826 leads
- Noviembre: 416,324 leads
- Diciembre: 484,256 leads

El área de Riesgos envía al área de Inteligencia Comercial 454,826 leads, que constituyen el universo de leads (correspondientes a mes de octubre), este universo

se divide en partes iguales: en el “conjunto 1” (227,413 leads) y en el “conjunto 2” (227,413 leads); para lograr que estos dos conjuntos tengan las mismas características se utilizó la técnica del muestreo estratificado. **Técnica del muestreo estratificado: Explicación del método**

Conceptos previos:

- Segmento Comercial: El universo de leads se segmenta en varios segmentos comerciales utilizando como variables el salario neto mensual, la edad, grado de instrucción, zona de residencia, etc; en el caso de estudio los segmentos comerciales son: Beyond, Premium, Preferente, Personal y Estándar.
- Grado de Contactabilidad: Existen tres grados de contactabilidad (alto, medio y bajo), estos grados están definidos por su probabilidad de contacto.
- Rango Monto Oferta: Cada lead tiene un rango de oferta de dinero definido. El muestreo estratificado consiste en dividir el universo de 454,826 leads (que envía el área de Riesgos en el mes de octubre) en grupos (estratos) caracterizados por su segmento comercial (se definió 5 cinco segmentos: Beyond, Premium, Preferente, Personal y Estándar), por su grados de contactabilidad (Alto, Medio y Bajo) y por sus rangos monto oferta; se definió 8 rangos: 1. [2k - 5k], 2. <5k - 15k], 3. <15k - 30k], 4. <30k - 45k], 5. <45k - 60 k], 6. <60k - 80k], 7. <80k - 100k],
- 8. <100k A MAS>; esto trae consigo que por cada segmento comercial se generen 24 grupos de leads (3 grados de contactabilidad X 8 rangos de oferta = 24 grupos) y para los 5 segmentos 120 grupos (Tablas 28, 29, 30, 31 y 32), estos grupos serán divididos en dos sub grupos de igual número de elementos (el sub grupo 1 para las **Reglas de Experto**, y el sub grupo 2 para el **Modelo de Optimización**). Las tablas 28, 29, 30, 31 y 32 muestran los 24 grupos (estratos) para cada segmento comercial:

Tabla 28: Estratificación de la base de leads correspondiente al segmento comercial “Beyond”

Nro de Segmento	Segmento Comercial	Grado de Contactabilidad	Rango de Oferta	Nro Leads del Grupo	Sub Grupo 1 (Para Reglas de Experto)	Sub Grupo 2 (Para Modelo de Optimización)
1	Beyond	Alto	1. [2k - 5k]	6,173	3,086	3,087
	Beyond	Alto	2. <5k - 15k]	5,663	2,831	2,832
	Beyond	Alto	3. <15k - 30k]	3,952	1,976	1,976
	Beyond	Alto	4. <30k - 45k]	4,419	2,209	2,210
	Beyond	Alto	5. <45k - 60k]	1,942	971	971
	Beyond	Alto	6. <60k - 80k]	1,281	640	641
	Beyond	Alto	7. <80k - 100k]	773	386	387
	Beyond	Alto	8. <100k A MAS>	531	265	266
	Beyond	Medio	1. [2k - 5k]	6,605	3,302	3,303
	Beyond	Medio	2. <5k - 15k]	6,229	3,114	3,115
	Beyond	Medio	3. <15k - 30k]	3,991	1,995	1,996
	Beyond	Medio	4. <30k - 45k]	4,552	2,276	2,276
	Beyond	Medio	5. <45k - 60k]	2,175	1,087	1,088
	Beyond	Medio	6. <60k - 80k]	1,383	691	692
	Beyond	Medio	7. <80k - 100k]	834	417	417
	Beyond	Medio	8. <100k A MAS>	600	300	300
	Beyond	Bajo	1. [2k - 5k]	7,133	3,566	3,567
	Beyond	Bajo	2. <5k - 15k]	6,728	3,364	3,364
	Beyond	Bajo	3. <15k - 30k]	4,350	2,175	2,175
	Beyond	Bajo	4. <30k - 45k]	5,098	2,549	2,549
	Beyond	Bajo	5. <45k - 60k]	2,197	1,098	1,099
	Beyond	Bajo	6. <60k - 80k]	1,508	754	754
	Beyond	Bajo	7. <80k - 100k]	935	467	468
	Beyond	Bajo	8. <100k A MAS>	672	336	336
Total				79,724	39,862	39,862

Fuente: Institución Financiera, elaboración propia

Tabla 29: Estratificación de la base de leads correspondiente al segmento comercial “Premium”

Nro de Segmento	Segmento Comercial	Grado de Contactabilidad	Rango de Oferta	Nro Leads del Grupo	Sub Grupo 1 (Para Reglas de Experto)	Sub Grupo 2 (Para Modelo de Optimización)
2	Premium	Alto	1. [2k - 5k]	7,015	3,507	3,508
	Premium	Alto	2. <5k - 15k]	5,899	2,949	2,950
	Premium	Alto	3. <15k - 30k]	4,116	2,058	2,058
	Premium	Alto	4. <30k - 45k]	4,701	2,350	2,351
	Premium	Alto	5. <45k - 60k]	2,002	1,001	1,001
	Premium	Alto	6. <60k - 80k]	1,307	653	654
	Premium	Alto	7. <80k - 100k]	849	424	425
	Premium	Alto	8. <100k A MAS>	597	298	299
	Premium	Medio	1. [2k - 5k]	7,576	3,788	3,788
	Premium	Medio	2. <5k - 15k]	5,958	2,979	2,979
	Premium	Medio	3. <15k - 30k]	4,363	2,181	2,182
	Premium	Medio	4. <30k - 45k]	4,701	2,350	2,351
	Premium	Medio	5. <45k - 60k]	2,062	1,031	1,031
	Premium	Medio	6. <60k - 80k]	1,438	719	719
	Premium	Medio	7. <80k - 100k]	909	454	455
	Premium	Medio	8. <100k A MAS>	639	319	320
	Premium	Bajo	1. [2k - 5k]	7,879	3,939	3,940
	Premium	Bajo	2. <5k - 15k]	6,316	3,158	3,158
	Premium	Bajo	3. <15k - 30k]	4,843	2,421	2,422
	Premium	Bajo	4. <30k - 45k]	4,842	2,421	2,421
	Premium	Bajo	5. <45k - 60k]	2,289	1,144	1,145
	Premium	Bajo	6. <60k - 80k]	1,581	790	791
	Premium	Bajo	7. <80k - 100k]	936	468	468
	Premium	Bajo	8. <100k A MAS>	709	354	355
Total				83,527	41,764	41,764

Fuente: Institución Financiera, elaboración propia

Tabla 30: Estratificación de la base de leads correspondiente al segmento comercial “Preferente”

Nro de Segmento	Segmento Comercial	Grado de Contactabilidad	Rango de Oferta	Nro Leads del Grupo	Sub Grupo 1 (Para Reglas de Experto)	Sub Grupo 2 (Para Modelo de Optimización)
3	Preferente	Alto	1. [2k - 5k]	7,708	3,854	3,854
	Preferente	Alto	2. <5k - 15k]	6,555	3,277	3,278
	Preferente	Alto	3. <15k - 30k]	4,677	2,338	2,339
	Preferente	Alto	4. <30k - 45k]	4,749	2,374	2,375
	Preferente	Alto	5. <45k - 60k]	2,176	1,088	1,088
	Preferente	Alto	6. <60k - 80k]	1,405	702	703
	Preferente	Alto	7. <80k - 100k]	875	437	438
	Preferente	Alto	8. <100k A MAS>	671	335	336
	Preferente	Medio	1. [2k - 5k]	7,940	3,970	3,970
	Preferente	Medio	2. <5k - 15k]	6,620	3,310	3,310
	Preferente	Medio	3. <15k - 30k]	4,958	2,479	2,479
	Preferente	Medio	4. <30k - 45k]	4,939	2,469	2,470
	Preferente	Medio	5. <45k - 60k]	2,415	1,207	1,208
	Preferente	Medio	6. <60k - 80k]	1,433	716	717
	Preferente	Medio	7. <80k - 100k]	910	455	455
	Preferente	Medio	8. <100k A MAS>	677	338	339
	Preferente	Bajo	1. [2k - 5k]	8,892	4,446	4,446
	Preferente	Bajo	2. <5k - 15k]	6,686	3,343	3,343
	Preferente	Bajo	3. <15k - 30k]	5,008	2,504	2,504
	Preferente	Bajo	4. <30k - 45k]	5,383	2,691	2,692
	Preferente	Bajo	5. <45k - 60k]	2,560	1,280	1,280
	Preferente	Bajo	6. <60k - 80k]	1,519	759	760
	Preferente	Bajo	7. <80k - 100k]	965	482	483
	Preferente	Bajo	8. <100k A MAS>	745	372	373
Total				90,466	45,233	45,233

Fuente: Institución Financiera, elaboración propia

Tabla 31: Estratificación de la base de leads correspondiente al segmento comercial “Personal”

Fuente: Institución Financiera, elaboración propia

Tabla 32: Estratificación de la base de leads correspondiente al segmento comercial “Estándar”

Nro de Segmento	Segmento Comercial	Grado de Contactabilidad	Rango de Oferta	Nro Leads del Grupo	Sub Grupo 1 (Para Reglas de Experto)	Sub Grupo 2 (Para Modelo de Optimización)
5	Estandar	Alto	1. [2k - 5k]	8,634	4,317	4,317
	Estandar	Alto	2. <5k - 15k]	7,119	3,559	3,560
	Estandar	Alto	3. <15k - 30k]	5,407	2,703	2,704
	Estandar	Alto	4. <30k - 45k]	4,894	2,447	2,447
	Estandar	Alto	5. <45k - 60k]	2,628	1,314	1,314
	Estandar	Alto	6. <60k - 80k]	1,754	877	877
	Estandar	Alto	7. <80k - 100k]	980	490	490
	Estandar	Alto	8. <100k A MAS>	759	379	380
	Estandar	Medio	1. [2k - 5k]	8,893	4,446	4,447
	Estandar	Medio	2. <5k - 15k]	7,760	3,880	3,880
	Estandar	Medio	3. <15k - 30k]	5,570	2,785	2,785
	Estandar	Medio	4. <30k - 45k]	4,993	2,496	2,497
	Estandar	Medio	5. <45k - 60k]	2,654	1,327	1,327
	Estandar	Medio	6. <60k - 80k]	1,860	930	930
	Estandar	Medio	7. <80k - 100k]	1,020	510	510
	Estandar	Medio	8. <100k A MAS>	858	429	429
	Estandar	Bajo	1. [2k - 5k]	9,960	4,980	4,980
	Estandar	Bajo	2. <5k - 15k]	7,993	3,996	3,997
	Estandar	Bajo	3. <15k - 30k]	5,960	2,980	2,980
	Estandar	Bajo	4. <30k - 45k]	5,592	2,796	2,796
	Estandar	Bajo	5. <45k - 60k]	2,946	1,473	1,473
	Estandar	Bajo	6. <60k - 80k]	2,065	1,032	1,033
	Estandar	Bajo	7. <80k - 100k]	1,020	510	510
	Estandar	Bajo	8. <100k A MAS>	901	450	451
Total				102,220	51,110	51,110

Fuente: Institución Financiera, elaboración propia

El *Nº* de leads del conjunto universo se obtiene al sumar el *Nº* de leads del segmento Beyond (79,724) + el *Nº* de leads del segmento Premium (83,527) + el *Nº* de leads del segmento Preferente (90,466) + el *Nº* de leads del segmento Personal (98,888) + el *Nº* de leads del segmento Estándar (102,220):

Nº de leads del conjunto universo = 454,826

En la primera fila de la Tabla 27 se observa que el primer grupo (estrato) del segmento comercial Beyond tiene 6,173 leads, este conjunto se divide aleatoriamente en dos sub grupos “de igual número de leads” (6,173 no es un número par): 3,086 leads pertenecientes al sub grupo 1 a los cuales se les aplicará las **Reglas de Experto** y 3,087 leads pertenecientes al sub grupo 2 a los cuales se les aplicará el **Modelo de Optimización**; este procedimiento se realiza con las restantes 24 filas de la Tabla 27.

Al unir los 24 sub grupos 1 de la tabla 27 se obtiene un conjunto de 39,862 leads (Reglas de Experto) y al unir los 24 sub grupos 2 de la misma tabla se obtiene 39,862 leads (Modelo de Optimización).

Realizando el mismo análisis para los segmentos comerciales: Premium (Tabla 28), Preferente (Tabla 29), Personal (Tabla 30), Estándar (Tabla 31); podemos generalizar lo siguiente:

Segmento Comercial Beyond: 39,862 leads (Reglas de Experto) y 39,862 leads (Modelo de Optimización).

Segmento Comercial Premium: 41,764 leads (Reglas de Experto) y 41,764 leads (Modelo de Optimización).

Segmento Comercial Preferente: 45,233 leads (Reglas de Experto) y 45,233 leads (Modelo de Optimización).

Segmento Comercial Personal: 49,444 leads (Reglas de Experto) y 49,444 leads (Modelo de Optimización).

Segmento Comercial Estándar: 51,110 leads (Reglas de Experto) y 51,110 leads (Modelo de Optimización).

Al unir los cinco sub grupos 1 y 2 (Tablas del 27 al 31) obtenemos dos conjuntos de igual numero de leads: el **Conjunto 1** de 227,413 leads (Reglas de Experto) y el

Conjunto 2 de 227,413 leads (Modelo de Optimización).

Paso 3:

Del **conjunto 1** (227,413 leads) se seleccionó un grupo de 29,131 leads utilizando las **Reglas de Experto** (este proceso se realizó tomando como variables de selección: la probabilidad de toma del producto, grado de contactabilidad, monto de la oferta y la tasa ofertada; además, se construye un query SQL que le permite a la computadora realizar la selección) y del **conjunto 2** (227,413 leads) se seleccionó un grupo de 29,131 leads utilizando el **Modelo de Optimización**; uniendo estos dos grupos se formó un solo conjunto de 58,262 leads, que fue enviado a los canales de venta para su gestión (Nota: Los ejecutivos de ventas no tienen conocimiento de este procedimiento, esto evita todo tipo de direccionamiento en la gestión).

Paso 4:

Al finalizar el mes, los resultados de los procesos de venta son analizados por el área de Inteligencia de Negocios (esta área tiene la relación de leads que fueron seleccionados mediante las Reglas de Experto y mediante el Modelo de Optimización) para medir la efectividad y rentabilidad de la campaña comercial. Se consideró 3 indicadores para esta medición:

- Efectividad: Nº de Aperturas del Producto/ Nº de Leads gestionados.
- Ticket_Desembolso: Para el caso de Prestamos Personales es el promedio del monto desembolsado al adquirir el préstamo y para el caso de tarjetas es el promedio de la linea de crédito de las tarjetas aperturadas.
- Rentabilidad: Promedio de la rentabilidad esperada de aquellos leads que adquirieron el producto.

Al comparar el valor de estos indicadores podemos calcular el incremento en la efectividad y en la rentabilidad de cada campaña comercial debido a la utilización del Modelo de Optimización. Las Tablas 33, 34, 35, 36, 37 y 38 muestran los resultados de la campañas de los productos Libre Disponibilidad, Tarjeta de Crédito, XTralínea, Compra de Deuda, Presta Bono y Descuento Planilla, respectivamente:

Campaña de Presta Bono					Campaña de Descuento Planilla				
Mes de Campaña	Indicadores	Modelo de Optimización	Reglas de Experto	Incremental	Mes de Campaña	Indicadores	Modelo de Optimización	Reglas de Experto	Incremental
Octubre	Nro Leads Gestionados	33,214	33,214		Octubre	Nro Leads Gestionados	32,866	32,866	
	Efectividad	12.2%	10.4%	1.80%		Efectividad	17.2%	15.1%	2.10%
	Nro Ventas	4,062	3,466	596		Nro Ventas	5,653	4,963	690
	Ticket_Desembolso	S/. 17,345	S/. 13,456	S/. 3,889		Ticket_Desembolso	S/. 21,234	S/. 19,134	S/. 2,100
	Desembolso	S/. 70,456,642	S/. 46,635,106	S/. 23,821,537		Desembolso	S/. 120,034,481	S/. 94,957,326	S/. 25,077,155
	Rentabilidad	S/. 134.2	S/. 112.7	S/. 21.53		Rentabilidad	S/. 167.2	S/. 145.5	S/. 21.76
Noviembre	Nro Leads Gestionados	34,210	34,210		Noviembre	Nro Leads Gestionados	31,551	31,551	
	Efectividad	15.2%	12.9%	2.27%		Efectividad	15.2%	12.9%	2.27%
	Nro Ventas	5,200	4,425	775		Nro Ventas	4,796	4,081	715
	Ticket_Desembolso	S/. 18,345	S/. 16,456	S/. 1,889		Ticket_Desembolso	S/. 19,872	S/. 17,091	S/. 2,781
	Desembolso	S/. 95,393,704	S/. 72,817,487	S/. 22,576,216		Desembolso	S/. 95,302,031	S/. 69,748,917	S/. 25,553,114
	Rentabilidad	S/. 124.2	S/. 92.7	S/. 31.53		Rentabilidad	S/. 178.2	S/. 152.7	S/. 25.53
Diciembre	Nro Leads Gestionados	37,631	37,631		Diciembre	Nro Leads Gestionados	34,391	34,391	
	Efectividad	13.2%	11.1%	2.08%		Efectividad	16.9%	14.7%	2.22%
	Nro Ventas	4,971	4,190	781		Nro Ventas	5,802	5,038	763
	Ticket_Desembolso	S/. 16,498	S/. 13,784	S/. 2,714		Ticket_Desembolso	S/. 20,156	S/. 18,416	S/. 1,740
	Desembolso	S/. 82,013,474	S/. 57,756,514	S/. 24,256,960		Desembolso	S/. 116,939,955	S/. 92,784,711	S/. 24,155,244
	Rentabilidad	S/. 144.2	S/. 132.7	S/. 11.53		Rentabilidad	S/. 167.2	S/. 143.7	S/. 23.53

Fuente: Institución Financiera, elaboración propia

Fuente: Institución Financiera, elaboración propia

El análisis de la información, proveniente de las Tablas 33, 34, 35, 36, 37 y 38, respecto a los indicadores en ellas mencionados, nos permite concluir que la utilización del Modelo de Optimización, en la construcción de las estrategias de Marketing, optimiza los resultados de estas campañas; como ejemplo citaremos que en la campaña del producto de Libre Disponibilidad, hemos incrementado las colocaciones en 31 millones de soles por mes, y en la campaña del producto Tarjeta de Crédito el incremento en las ventas es de 1,500 tarjetas por mes, con líneas de créditos mayores a las de las tarjetas aperturadas con el método tradicional. También la efectividad y la rentabilidad de cada campaña comercial de estos productos se ha visto incrementada. Esta realidad también se observa en los demás productos financieros (Xtralinea, Compra de Deuda, Presta Bono, Descuento por Planilla) donde se implementó el Modelo de Optimización.

CONCLUSIONES

1. Esta investigación permite responder la siguiente pregunta: ¿Cuáles son los impulsores e impedimentos para implementar el análisis predictivo en marketing?. La creciente necesidad de comprender el comportamiento del mercado debido al aumento de la competencia y a la cada vez mayor exigencia de los clientes; lleva a las organizaciones financieras a implementar estrategias de marketing centradas en el cliente; estas estrategias permiten incrementar la efectividad y la rentabilidad de las campañas comerciales, en tanto las organizaciones son capaces de construir y utilizar sistemas de información (procesamiento de datos) y modelos analíticos que optimizan el resultado de estas campañas. Los hallazgos de esta investigación refuerzan la centralidad en el cliente y permiten a las organizaciones desenvolverse eficientemente en un entorno empresarial en constante cambio.

2. Esta investigación también permite responder la siguiente pregunta: ¿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 (TI).

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 de servicio al cliente y el uso del BI; cuando se logra esta articulación, la construcción y la gestión de los modelos analíticos, entre ellos el modelo de optimización (caso de estudio), sirven para optimizar los resultados de las campañas de marketing.

3. Cuando el Modelo de Optimización es utilizado, demuestra que es capaz de mejorar la efectividad, la rentabilidad y otros indicadores de las campañas de marketing de 6 productos finaicerios (Prestamos Personales, Tarjeta de Crédito, Xtralinea, Compra de Deuda, Presta Bono y Descuento Planilla), esta afirmación se demuestra en las Tablas 32, 33, 34, 35, 36 y 37.
4. Para conseguir un resultado óptimo al utilizar el modelo de optimización, es necesario conseguir la mayor precisión (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 (LightGBM, XGBoost, Regresión Logística y Super Vector Classifier); en el estudio el algoritmo LightGBM es el que dio mejores resultados en la predicción de adquisición del producto Prestamo Personal. Para los demás productos se siguió la misma metodología y se seleccionó el algoritmo con el que se obtuvo mayor precisión; los resultados se muestran en la tabla 39 descrita en el anexo de la presente tesis.
5. La utilización de la inteligencia artificial (Redes Neuronales), permitió ajustar y estandarizar las probabilidades obtenidas de los modelos estadísticos; la eficiencia de la red esta en función de su configuración; es decir, debemos definir correctamente el numero de capas ocultas y el numero de neuronas de cada uno de ellas, la función de activación (sigmoidea, relu, linear o tanh) y el mejor método de transformación de los datos (normalización, MinMax, estandarización o Transformación BoxCox).
6. Existen dos soluciones de la función objetivo, una solución óptima global utilizando una librería de Python (PULP) y una solución local que utiliza el metodo de los Algorítmogenéticos. La primera solución necesita un mayor tiempo de procesamiento y mayores recursos tecnológicos, la segunda requiere un mayor tratamiento matemático, es menos óptima y se utiliza cuando el factor tiempo de procesamiento es relevante.

RECOMENDACIONES

1. Para medir y evaluar el desempeño del modelo de optimización es necesario complementarlo con una eficiente **analitica de marketing**, lo que permite

optimizar la toma de decisiones y evaluar los impactos del modelo en la efectividad y rentabilidad de las campañas comerciales.

2. En la función objetivo del proceso de optimización participa el indicador rentabilidad como coeficiente de sus variables; en esta primera versión del trabajo de investigación se calcula el valor de este indicador analizando la rentabilidad histórica de las aperturas (adquisición) de cada producto financiero a través de un periodo de tiempo; para mejorar la eficiencia del modelo de optimización es necesario construir un modelo de rentabilidad que calcule este indicador por producto y para cada lead de la campaña comercial, este modelo debe considerar las siguientes variables independientes:

- Variables del Sistema Financiero: Determinan la situación de los leads dentro del sistema bancario.
- Variables del Internas del Banco: Explican el perfil dentro del banco.
- Variables de Campaña: Son las características del producto que se ofrece al lead en cada campaña comercial, por ejemplo el monto ofertado, la tasa de interés, el plazo del crédito, etc.

Estas variables permiten calcular la rentabilidad esperada, en caso el lead adquiera el producto (target del modelo).

3. Actualmente el modelo de optimización considera 2 restricciones generales, el capacity (máximo número de leads que pueden ser gestionados por el canal) y la cantidad máxima de productos que se puede ofertar a cada lead. Se recomienda incluir en las construcciones de las restricciones de la función objetivo las siguientes variables:

- Monto Oferta: Monto que se le va a ofertar al lead.
- Frescura: Cantidad de veces que el lead se encuentra en la misma campaña comercial, dentro de los 6 últimos meses.
- Score de Contactabilidad: La probabilidad de que el cliente sea contactado por el canal de venta.

4. Para estandarizar las probabilidades que participan en la función objetivo hemos usado un modelo basado en Inteligencia Artificial (Redes Neuronales); sin embargo, existen otros modelos MultiTarget de Clasificación (Support Vector Classifier, Gradient Boosting Classifier, Random Forest Classifier) que pueden ser usados con el mismo propósito. La calidad de los resultados del modelo de optimización depende del grado de precisión de las probabilidades usadas, debido a esto recomendamos escoger el mejor modelo para estandarizar dichas probabilidades luego de comparar el grado de precisión del proceso de estandarización de los modelos MultiTarget.
5. Esta tesis puede servir de base para resolver otros problemas de optimización que se encuentran en el sector financiero como por ejemplo:
 - Maximizar la producción de las campañas comerciales, minimizando el riesgo crediticio.
 - Maximizar la rentabilidad asignando la tasa de interés óptima, teniendo en cuenta que al aumentar la tasa de interés, la probabilidad de aceptación disminuye pero la rentabilidad aumenta.

REFERENCIAS BIBLIOGRÁFICAS

Ali Aydin Koç, Özgür Yeniay (2013). A comparative study of Artificial Neural Networks and Logistic Regression for classification of marketing campaign results. Hacettepe University - Department of Statistics and Mathematics.

Andrés Mauricio Mendoza Espinoza (2014). Modelos de clasificación en el otorgamiento de créditos financieros: comparación entre diferentes técnicas de Machine Learning y modelos de regresión múltiple. Universidad de los Andes.

Andrew Storey, Marc David Cohen (2015). Exploiting Response Models – Optimizing Cross – sell and up – sell opportunities in Banking. SAS Institute Inc. Scotiabank Toronto Canada.

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

Brown, B., & Gottlieb, J. (2016). The Need to Lead in Data and Analytics. McKinsey&Company. Retrieved from mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/The%20need%20to%20lead%20in%20data%20and%20analytics.ashx

Carlos Espino Timón (2017). Análisis predictivo: técnico y modelos utilizados y aplicaciones del mismo. Universitat Oberta de Catalunya.

Clow, K. E., & Baack, D. (2016). Integrated Advertising, Promotion, and Marketing Communications, 7th Global Edition. Pearson.

Deloitte MCS Limited. (2013). Next Best Action driving customer value through a rich and relevant multichannel experience in Financial Services. Deloitte analytics. Retrieved from www2.deloitte.com/content/dam/Deloitte/uk/Documents/consultancy/deloitte-uk-con-next-best-action.pdf

Erevelles, S., Fukawa, N., & Swayne, L. (2016). Big data analytics and the transformation of marketing. *Journal of Business Research*, 69(2), pp. 897-904. Retrieved from doi.org/10.1016/j.jbusres.2015.07.001

Fabrice Nobibon, Roel Leus, & Frits Spieksma (2018). Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms. *European Journal of Operational Research*.

Fernández, W., & Thomas, G. (2008). Success in IT projects: ¿A matter of definition? *International journal of project management*, 26(7), pp. 733–742. Retrieved from doi.org/10.1016/j.ijproman.2008.06.003

Galvin, C. (2013.) Making data work. B&T Weekly. Retrieved from search-proquestcom.ezproxy.jyu.fi/docview/1464856126?accountid=11774

Goldenberg, B. (2017). Make your customer engagement a closed loop. Customer Relationship Management: CRM 2017, 21(10), p.6(2). Retrieved from search-proquest-com.ezproxy.jyu.fi/docview/1951492738? accountid=11774

Junming Zhang (2010). Analytics of neural network on bank marketing data. College of Computer Scienie – MIT. P203-253.

Lee, J., Deloitte, Limited, (2013). Effect of cus- tomer-centric structure on firm performance. Marketing Science Institute Working Paper Series, 34(2), pp. 250-268. Retrieved from doi.org/10.1287/mksc.2014.0878

Leeflang, P. S. H., Verhoef, P. C., Dahlström, P., & Freundt, T. (2014). Challenges and solutions for marketing in digital era. European management journal 32, pp. 112. Retrieved from doi.org/10.1016/j.emj.2013.12.001

Manuel Córdova Zamora, 2008. Libro de Estadística Aplicada. Primera Edición.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity.

Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., & Tufan, P. (2012). Analytics: The real-world use of big data. IBM Institute for Business Value, pp. 1–20. Retrieved from bdvc.nl/images/Rapporten/GBE03519USEN.PDF

Sleep, S., & Hulland, J. (2019). Is big data driving cooperation in the c-suite? The evolving relationship between the chief marketing officer and chief information officer. Journal of Strategic Marketing, 27(8), pp. 666-678. Retrieved from doi.org/10.1080/0965254X.2018.1464496

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

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

Story, V., O'Malley, L., & Hart, S. (2011). Roles, role performance, and radical innovation competencies. *Industrial Marketing Management*, 40(6), pp. 952–966. Retrieved from doi.org/10.1016/j.indmarman.2011.06.025

Tellis, G. J., Prabhu, J. C., & Chandy, R. K. (2009). Radical innovation across nations: The preeminence of corporate culture. *The Journal of Marketing*, 73(1), pp. 3–23. Retrieved from doi.org/10.1509/jmkg.73.1.3

Teerlink, M., & Haydock, M. (2012). Customer analytics pays off: Driving top-line growth by bringing science to the art of marketing. IBM Global Business Services.

Verhoef, P. C., & Lemon, K. N. (2013). Successful customer value management: Key lessons and emerging trends. *European Management Journal*, 31(?), pp. 1–15. Retrieved from doi.org/10.1016/j.emj.2012.08.001

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

ANEXO 01: PRUEBAS DE LOS MODELOS ESTADISTICOS

En la propuesta de solución se ha calculado las probabilidades de adquisición de los productos financieros a través de modelos analíticos que han sido elegidos al comparar la precisión de los 4 algoritmos utilizados (LightGBM, XGBoost, Regresión Logística, Support Vector Classifier), siguiendo esta metodología para cada producto se ha seleccionado un algoritmo diferente.

Para calcular la precisión del modelo elegido se utilizan, además del AUC (empleado en la tesis) otras métricas como: Recall y Precisión. La Tabla 39 muestra la matriz de confusión, la cual es utilizada para calcular los indicadores Recall y Precisión.

Tabla 39: Matriz de confusión

		Condición Predicha	
Población Total		Codición Predecida Positiva	Codición Predecida Negativa
Verdadera Condición	Codición Positiva	True Positive	False Negative
	Codición Negativa	False Positive	True Negative

Fuente: Institución financiera, elaboración propia

RECALL

Recall indica basicamente que tanto por ciento, de los leads que adquirieron el producto financiero (Condición Positiva), fueron correctamente identificados. Se define de la siguiente manera:

$$Recall = \frac{TP}{TP + FN}$$

PRECISIÓN

Precisión mide la calidad de la predicción, es decir que porcentaje de las ventas predichas son efectivamente ventas reales. Se define de la siguiente manera:

$$Precision = \frac{TP}{TP + FP}$$

Donde TP: True Positive, FN: False Negative y FP: False Positive

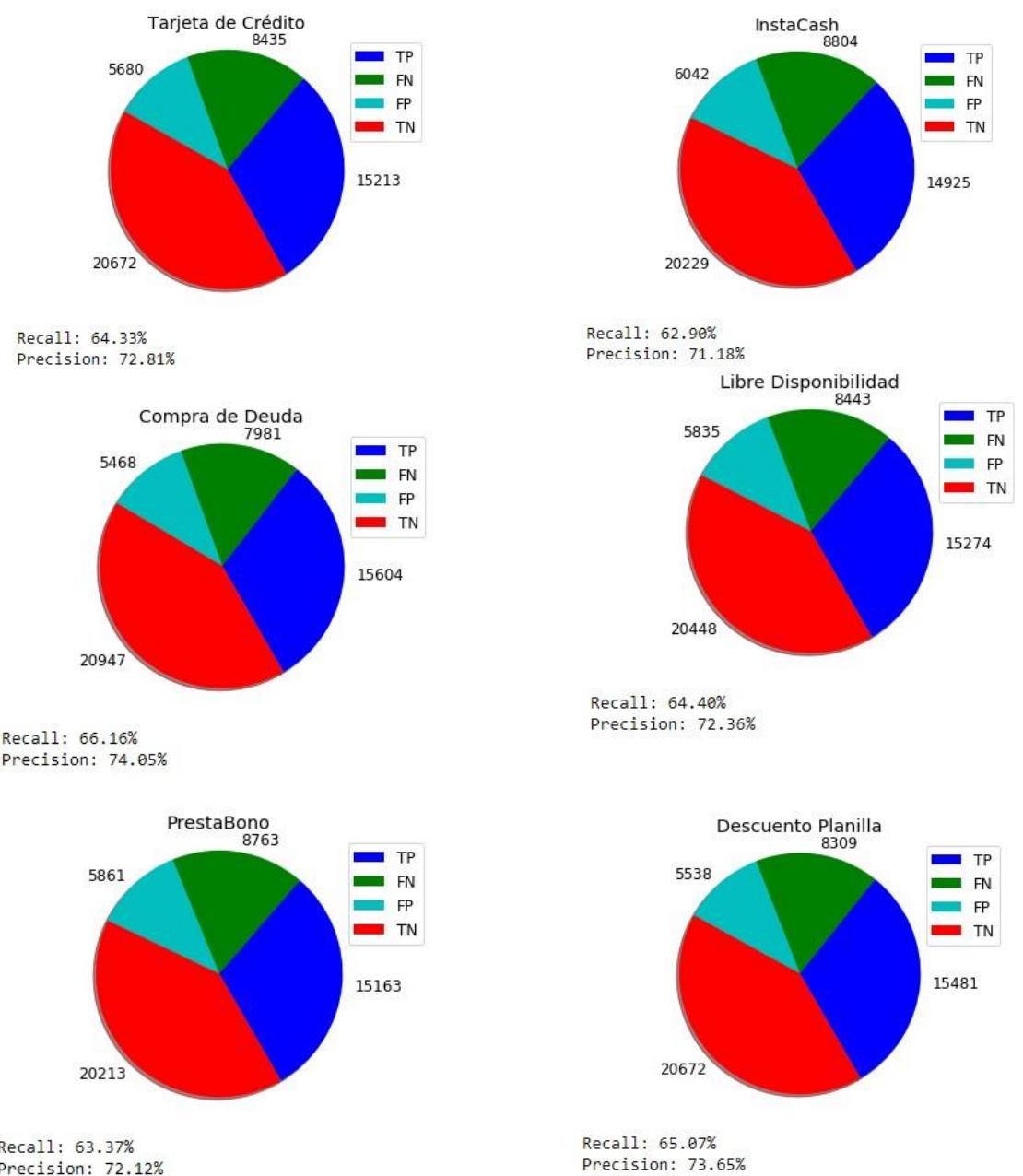
Los parámetros usados para el proceso de pruebas de los modelos estadísticos son:

- Número Total de Leads: 50,000
- Umbral de probabilidad para predecir una venta: 0.65

RESULTADOS OBTENIDOS POR MODELO ESTADÍSTICO

Las gráficas de la Figura 22 muestran los valores del Recall y Precision para cada campaña comercial, estos valores demuestran la calidad de los modelos estadísticos.

Figura 22: Gráficas que muestran la calidad de los modelos estadísticos



Fuente: Institución Financiera, elaboración propia

Area Under Curve (AUC)

Es una métrica utilizada para medir la capacidad de generalización de los modelos de clasificación (modelos de propensión), dentro de la etapa de validación de los modelos analíticos. Se define de la siguiente manera:

$$AUC = \frac{1}{2} (GINI + 1)$$

Donde, GINI: Índice Gini del modelo analítico

La Tabla 39 muestra el resumen de las métricas mencionadas por modelo.

Tabla 39: Métricas Recall, Precisión y AUC de los modelos estadísticos

Campaña	Algoritmo Utilizado	Precisión	Recall	AUC
Tarjeta de Crédito	XGBoost	72.81%	64.33%	82.25%
Insta Cash	Regresión Logística	71.18%	62.90%	77.62%
Compra de Deuda	Support Vector Classifier	74.05%	66.16%	73.86%
Préstamo Personal	LightGBM	72.36%	64.40%	78.87%
Presta Bono	XGBoost	72.12%	63.37%	77.54%
Descuento Planilla	LightGBM	73.65%	65.07%	79.78%

Fuente: Institución Financiera, elaboración propia

En la Figura 23 se muestra el valor del Recall y de la Precisión por modelo estadístico.

Figura 23: Métricas Recall y Precisión de los modelos estadísticos



Fuente: Institución Financiera, elaboración propia

ANEXO 02: CODIGO PYTHON

SELECCIÓN DE VARIABLES PRINCIPALES

```

def Categorize(df_input, in_var, cantidad_quantiles):
    in_var = in_var
    df_input = df_input.copy()
    #Calculamos la cantidad de elementos unicos
    cat_counts = df_input[in_var].value_counts(dropna = False).reset_index().sort_values(['index'])

    if len(cat_counts)>10:
        #df_input[in_var]= df_input[in_var].replace(np.nan, df_input[in_var].mean())
        df_input[in_var]= df_input[in_var].replace(np.nan, 0)
        categories, Maximos = pd.qcut(df_input[in_var],cantidad_quantiles, duplicates = 'drop', retbins = True)
        out_var = "Q_" + in_var
        i=0
        for maximo in Maximos:
            if i == 1:
                df_input[out_var] = np.where(df_input[in_var]<maximo, i,0)
            elif i>1:
                df_input[out_var] = np.where((df_input[in_var]<maximo) & (df_input[out_var] == 0),i, df_input[out_var])
            i=i+1
        df_input[out_var] = np.where(df_input[out_var]==0, 1, df_input[out_var])
    else:
        print('La variable: '+in_var+' no debe ser categorizada')

    return df_input

def generar_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

var_numericas = columns.symmetric_difference(var_no_numericas)
df_ginis_INI = generar_ginis_individuales(df_final_consolidado,'Es_Venta',var_numericas,var_categoricas)
df_ginis_TC = generar_ginis_individuales(df_final_consolidado,'Flag_Apertura_TC',var_numericas,var_categoricas)
df_ginis_XL = generar_ginis_individuales(df_final_consolidado,'Flag_Apertura_XL',var_numericas,var_categoricas)
df_ginis_CD = generar_ginis_individuales(df_final_consolidado,'Flag_Apertura_CD',var_numericas,var_categoricas)
df_ginis_LD = generar_ginis_individuales(df_final_consolidado,'Flag_Apertura_LD',var_numericas,var_categoricas)
df_ginis_PA = generar_ginis_individuales(df_final_consolidado,'Flag_Apertura_PA',var_numericas,var_categoricas)
df_ginis_DXP = generar_ginis_individuales(df_final_consolidado,'Flag_Apertura_DXP',var_numericas,var_categoricas)
df_ginis_INI.head(20)

```



```

def eliminar_variables_correlacionadas(df_input, MaximaCorrelacionPermitida, df_ginis,CantidadVariables):
    VariablesSeleccionadas = []
    MaximaCorrelacionExistente = 0
    df_ginis = df_ginis.head(CantidadVariables)
    VariablesOrdenadas = list(df_ginis[df_ginis['Tipo_Variable'] == 'Numerica'][['Variable']])
    VariablesCategoricas = list(df_ginis[df_ginis['Tipo_Variable'] == 'Categorica'][['Variable']])
    VariablesOrdenadas = VariablesOrdenadas[0:CantidadVariables]
    df_copy = df_input.copy()
    i = 1
    while(i<=(len(VariablesOrdenadas) - len(VariablesCategoricas))):
        if i==1:
            var1 = VariablesOrdenadas[0]
            VariablesSeleccionadas.append(var1)
        else:
            j=1
            MaximaCorrelacionExistente = 0
            var1 = VariablesOrdenadas[i-1]
            CantidadVariablesSeleccionadas = len(VariablesSeleccionadas)
            while(j<=CantidadVariablesSeleccionadas):
                var2 = VariablesSeleccionadas[j-1]
                cor_temp = df_copy[[var1,var2]].corr()
                cor_temp = cor_temp[var1][1]
                if cor_temp >= MaximaCorrelacionExistente:
                    MaximaCorrelacionExistente = cor_temp
                j=j+1
            if MaximaCorrelacionExistente <= MaximaCorrelacionPermitida:
                VariablesSeleccionadas.append(var1)
        i = i+1
    for var in VariablesCategoricas:
        VariablesSeleccionadas.append(var)
    return VariablesSeleccionadas

MaximaCorrelacionPermitida = 0.4
VariablesSeleccionadas_INI = eliminar_variables_correlacionadas(df_final Consolidado,0.4,df_ginis_INI,20)
VariablesSeleccionadas_TC = eliminar_variables_correlacionadas(df_final Consolidado,0.4,df_ginis_TC,20)
VariablesSeleccionadas_XL = eliminar_variables_correlacionadas(df_final Consolidado,0.4,df_ginis_XL,20)
VariablesSeleccionadas_CD = eliminar_variables_correlacionadas(df_final Consolidado,0.4,df_ginis_CD,20)
VariablesSeleccionadas_LD = eliminar_variables_correlacionadas(df_final Consolidado,0.4,df_ginis_LD,20)
VariablesSeleccionadas_PA = eliminar_variables_correlacionadas(df_final Consolidado,0.4,df_ginis_PA,20)
VariablesSeleccionadas_DXP = eliminar_variables_correlacionadas(df_final Consolidado,0.4,df_ginis_DXP,20)

def Transformacion_Categoricas(df_input, var_lista_categoricas, var_target, var_seleccionadas):
    df_copy = df_input.copy()
    k = int(0.3*df_copy.shape[0])
    f = int(0.2*df_copy.shape[0])
    for var in var_lista_categoricas:
        valores_unicos = list(np.unique(df_copy[var]))
        new_variable = 'T_' + str(var)
        df_copy[new_variable] = 0.000
        for valor in valores_unicos:
            niY = len(df_copy[(df_copy[var] == valor) & (df_copy[var_target] == 1)])
            ni = len(df_copy[(df_copy[var] == valor)])
            nY = len(df_copy[(df_copy[var_target] == 1)])
            nT = df_copy.shape[0]
            fni = 1/(1+np.exp(-(ni-k)/f))
            if valor != 'NI':
                new_valor = (fni * (niY/ni)) + ((1-fni)*(nY/nT))
            else:
                new_valor = 0
            df_copy[new_variable] = df_copy.apply(lambda row: new_valor if row[var]==valor else row[new_variable], axis = 1)
    var_seleccionadas = [new_variable if value==var else value for value in var_seleccionadas]
    return df_copy, var_seleccionadas

var_categoricas_total = ['ScoreRiesgos','Genero','EstadoCivil','MacroZona','GradoInst','Flag_Dependiente','NSE',
                        'SegmentoComercial','ClasificacionSBS','Entidad_Principal']
var_categoricas = list(set(set(var_categoricas_total) & set(VariablesSeleccionadas_INI)))
var_target = 'Es_Venta'
df_copy_INI, VariablesSeleccionadas_INI_NEW = Transformacion_Categoricas(df_final Consolidado, var_categoricas,
                                                                      var_target, VariablesSeleccionadas_INI)

var_categoricas = list(set(set(var_categoricas_total) & set(VariablesSeleccionadas_TC)))
var_target = 'Flag_Aertura_TC'
df_copy_TC, VariablesSeleccionadas_TC_NEW = Transformacion_Categoricas(df_final Consolidado, var_categoricas,
                                                                      var_target, VariablesSeleccionadas_TC)

var_categoricas = list(set(set(var_categoricas_total) & set(VariablesSeleccionadas_XL)))
var_target = 'Flag_Aertura_XL'
df_copy_XL, VariablesSeleccionadas_XL_NEW = Transformacion_Categoricas(df_final Consolidado, var_categoricas,
                                                                      var_target, VariablesSeleccionadas_XL)

```

SELECCIÓN DEL MEJOR MODELO

```
from sklearn.model_selection import train_test_split
import lightgbm as lgbm

##Modelo LightGBM
vars_in = VariablesSeleccionadas_INI_NEW
df_copy = df_copy_INI.copy()
var_target = 'Es_Venta'
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,
                                                       stratify=df_copy[var_target])

# Parametros del LightGBM
param = lgbm.LGBMClassifier(
    n_estimators=200,
    learning_rate=0.015,
    boosting_type= 'gbdt',
    objective='binary',
    colsample_bytree=.8,
    subsample=.8,
    max_depth=4,
    lambda_l1=1
)

# Entrenamiento:
lgbm_model = param.fit(train_x, train_y,
                       eval_set= [(train_x, train_y), (valid_x, valid_y)],
                       #categorical_feature = vars_cat,
                       eval_metric=['AUC'],
                       verbose=10
                      )

##Modelo Regresion Logistica
import statsmodels.formula.api as smf
import statsmodels.api as sm
train_data, test_data = train_test_split(df_copy, test_size = 0.3, random_state = 42)
num_variables = len(vars_in)
formula = str(var_target) + ' ~ '
i=1
for col in vars_in:
    if i<num_variables:
        formula = formula + col + " + "
    else:
        formula = formula + col
    i=i+1
logit_model = smf.glm(formula = formula, data = train_data, family=sm.families.Binomial()).fit()
logit_model.summary()

df_copy['Prob_LighGBM'] = lgbm_model.predict_proba(df_copy[vars_in])[:,1]
df_copy['Prob_Logit'] = logit_model.predict(df_copy)
df_copy['Prob_XGBoost'] = XGB.predict(df_copy[vars_in])
AUC_LightGBM = Grafica_CurvaROC_CalculaAUC(df_copy['Es_Venta'],df_copy['Prob_LighGBM'],'SI')
AUC_Logit = Grafica_CurvaROC_CalculaAUC(df_copy['Es_Venta'],df_copy['Prob_Logit'],'SI')
AUC_XGBoost = Grafica_CurvaROC_CalculaAUC(df_copy['Es_Venta'],df_copy['Prob_XGBoost'],'SI')
AUC_SVC = 0.72567
df_AUC = pd.DataFrame(columns = ['Modelo','AUC'])
df_AUC = df_AUC.append({'Modelo": 'Modelo LightGBM' , "AUC": AUC_LightGBM}, ignore_index = True)
df_AUC = df_AUC.append({'Modelo": 'Modelo XGBoost' , "AUC": AUC_XGBoost}, ignore_index = True)
df_AUC = df_AUC.append({'Modelo": 'Regresion Logistica' , "AUC": AUC_Logit}, ignore_index = True)
df_AUC = df_AUC.append({'Modelo": 'Super Vector Classifier' , "AUC": AUC_SVC}, ignore_index = True)
df_AUC.head(5)
```

CONSTRUCCIÓN DE LOS MODELOS

```

def generar_probabilidad_producto(df_copy, vars_in, var_target, var_out):
    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,
                                                          stratify=df_copy[var_target])

    #Modelo LightGBM
    # Parametros del LightGBM
    param = lgbm.LGBMClassifier(
        n_estimators=120,
        learning_rate=0.022,
        boosting_type='gbdt',
        objective='binary',
        colsample_bytree=.8,
        subsample=.8,
        max_depth=4,
        lambda_l1=1
    )

    # Entrenamiento:
    lgbm_model = param.fit(train_x, train_y,
                           eval_set= [(train_x, train_y), (valid_x, valid_y)],
                           #categorical_feature = vars_cat,
                           eval_metric=['AUC'],
                           verbose=50
                           )
    df_copy[var_out] = lgbm_model.predict_proba(df_copy[vars_in])[:,1]
    return df_copy
    df_copy = df_copy.drop(['var_out'], axis=1)
    df_final_consolidado[var_out] = df_copy[var_out]

    vars_in = VariablesSeleccionadas_XL_NEW
    df_copy = df_copy_XL.copy()
    var_target = 'Flag_Aertura_XL'
    var_out = 'Prob_XL'
    df_copy = generar_probabilidad_producto(df_copy, vars_in, var_target, var_out)
    df_final_consolidado[var_out] = df_copy[var_out]

    vars_in = VariablesSeleccionadas_CD_NEW
    df_copy = df_copy_CD.copy()
    var_target = 'Flag_Aertura_CD'
    var_out = 'Prob_CD'
    df_copy = generar_probabilidad_producto(df_copy, vars_in, var_target, var_out)
    df_final_consolidado[var_out] = df_copy[var_out]

    vars_in = VariablesSeleccionadas_LD_NEW
    df_copy = df_copy_LD.copy()
    var_target = 'Flag_Aertura_LD'
    var_out = 'Prob_LD'
    df_copy = generar_probabilidad_producto(df_copy, vars_in, var_target, var_out)
    df_final_consolidado[var_out] = df_copy[var_out]

    vars_in = VariablesSeleccionadas_PA_NEW
    df_copy = df_copy_PA.copy()
    var_target = 'Flag_Aertura_PA'
    var_out = 'Prob_PA'
    df_copy = generar_probabilidad_producto(df_copy, vars_in, var_target, var_out)
    df_final_consolidado[var_out] = df_copy[var_out]

    vars_in = VariablesSeleccionadas_DXP_NEW
    df_copy = df_copy_DXP.copy()
    var_target = 'Flag_Aertura_DXP'
    var_out = 'Prob_DXP'
    df_copy = generar_probabilidad_producto(df_copy, vars_in, var_target, var_out)
    df_final_consolidado[var_out] = df_copy[var_out]
    df_final_consolidado.head(100)

```

TESTEO DE LOS MODELOS DESARROLLADOS

```

from sklearn.metrics import roc_curve, auc
def Grafica_CurvaROC_CalculaAUC(y_target, y_prob, grafica):
    # create ROC itself
    fpr,tpr,_ = roc_curve(y_target,y_prob)
    # compute AUC
    roc_auc = auc(fpr,tpr)
    # plotting bells and whistles

    if grafica == "SI":
        # Graficar Curva ROC para Calcular AUC y GINI
        figure, ax1 = plt.subplots(figsize=(5,5))
        ax1.plot(fpr,tpr, label='%s (area = %0.2f)' % ('Classifier',roc_auc))
        ax1.plot([0, 1], [0, 1], 'k--')
        ax1.set_xlim([0.0, 1.0])
        ax1.set_ylim([0.0, 1.0])
        ax1.set_xlabel('Especificidad', fontsize=18)
        ax1.set_ylabel('Sensibilidad', fontsize=18)
        ax1.set_title("Curva ROC", fontsize=18)
        plt.tick_params(axis='both', labelsize=18)
        ax1.legend(loc="lower right", fontsize=14)
        plt.grid(True)
        figure.show()

def Calcular_Recall_Precision(data_set, var_prob, var_apertura, nombre_campaña, umbral_prob):
    df_fil = data_set[[var_prob, var_apertura]]
    df_fil['Apertura_Pred'] = df_fil[var_prob].apply(lambda x: 1 if x > umbral_prob else 0)
    df_fil['TP'] = df_fil.apply(lambda row: 1 if row[var_apertura] == 1 and row['Apertura_Pred'] == 1 else 0, axis = 1)
    df_fil['FN'] = df_fil.apply(lambda row: 1 if row[var_apertura] == 1 and row['Apertura_Pred'] == 0 else 0, axis = 1)
    df_fil['FP'] = df_fil.apply(lambda row: 1 if row[var_apertura] == 0 and row['Apertura_Pred'] == 1 else 0, axis = 1)
    df_fil['TN'] = df_fil.apply(lambda row: 1 if row[var_apertura] == 0 and row['Apertura_Pred'] == 0 else 0, axis = 1)
    TP = df_fil['TP'].sum()
    FN = df_fil['FN'].sum()
    FP = df_fil['FP'].sum()
    TN = df_fil['TN'].sum()
    Recall = TP/(TP + FN)
    Precision = TP/(TP + FP)

    values = [TP, FN, FP, TN]
    colors = ['b', 'g', 'c', 'r']
    labels = ['TP', 'FN', 'FP', 'TN']
    plt.pie(values, colors=colors, labels = values, shadow=True, startangle=-60)
    plt.title(campaña)
    plt.legend(labels, loc="best")
    plt.axis('equal')
    plt.show()
    return Recall, Precision

var_prob = 'Prob_TC'
var_apertura = 'Flag_Aertura_TC'
campaña = 'Tarjeta de Crédito'
umbral_prob = 0.6
Precision, Recall = Calcular_Recall_Precision(df_final_consolidado,var_prob, var_apertura, campaña, umbral_prob)
AUC = Grafica_CurvaROC_CalculaAUC(df_final_consolidado[var_apertura],df_final_consolidado[var_prob],'SI')
print('Campaña +'str(campaña))
print(f'Recall: {Recall*100:.2f}% ')
print(f'Precision: {Precision*100:.2f}% ')
print(f'AUC: {AUC*100:.2f}% ')

var_prob = 'Prob_XL'
var_apertura = 'Flag_Aertura_XL'
campaña = 'InstaCash'
umbral_prob = 0.6
Precision, Recall = Calcular_Recall_Precision(df_final_consolidado,var_prob, var_apertura, campaña, umbral_prob)
AUC = Grafica_CurvaROC_CalculaAUC(df_final_consolidado[var_apertura],df_final_consolidado[var_prob],'SI')
print('Campaña +'str(campaña))
print(f'Recall: {Recall*100:.2f}% ')
print(f'Precision: {Precision*100:.2f}% ')
print(f'AUC: {AUC*100:.2f}% ')

var_prob = 'Prob_CD'
var_apertura = 'Flag_Aertura_CD'
campaña = 'Compra de Deuda'
umbral_prob = 0.6
Precision, Recall = Calcular_Recall_Precision(df_final_consolidado,var_prob, var_apertura, campaña, umbral_prob)
AUC = Grafica_CurvaROC_CalculaAUC(df_final_consolidado[var_apertura],df_final_consolidado[var_prob],'SI')
print('Campaña +'str(campaña))
print(f'Recall: {Recall*100:.2f}% ')
print(f'Precision: {Precision*100:.2f}% ')
print(f'AUC: {AUC*100:.2f}% ')

```



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_consolidado[['Prob_TC','Prob_XL','Prob_CD','Prob_LD','Prob_PA','Prob_DXP']].copy()
features = df_copy.copy()
#instantiate
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_Aertura_TC','Flag_Aertura_XL','Flag_Aertura_CD',
                           'Flag_Aertura_LD','Flag_Aertura_PA','Flag_Aertura_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)

```

VALIDACIÓN DE LOS RESULTADOS DE LA RED NEURONAL

```

Productos = ["Tarjetas", "Xtralinea", "Compra Deuda TC", "Libre Disponibilidad", "PrestaBono",
             "Descuento Planilla"]
dfMetricas = pd.DataFrame()
for prod in Productos:
    producto = prod
    AUC_orig = 0 , AUC_Ajus = 0
    R2_orig = 0 , R2_Ajus = 0
    MAE_orig = 0 , MAE_Ajus = 0
    MSE_orig = 0 , MSE_Ajus = 0

    if prod == "Tarjetas":
        y_target = df_final['Flag_Aertura_TC']
        y_prob_orig = df_final['Prob_TC']
        y_prob_pred = df_final['ProbTC_AJUSTADA']
    if prod == "Xtralinea":
        y_target = df_final['Flag_Aertura_XL']
        y_prob_orig = df_final['Prob_XL']
        y_prob_pred = df_final['ProbXL_AJUSTADA']
    if prod == "Compra Deuda TC":
        y_target = df_final['Flag_Aertura_CD']
        y_prob_orig = df_final['Prob_CD']
        y_prob_pred = df_final['ProbCD_AJUSTADA']
    if prod == "Libre Disponibilidad":
        y_target = df_final['Flag_Aertura_LD']
        y_prob_orig = df_final['Prob_LD']
        y_prob_pred = df_final['ProbLD_AJUSTADA']
    if prod == "PrestaBono":
        y_target = df_final['Flag_Aertura_PA']
        y_prob_orig = df_final['Prob_PA']
        y_prob_pred = df_final['ProbPA_AJUSTADA']

    AUC_orig = Grafica_CurvaROC_CalculaAUC(y_target, y_prob_orig, "NO")
    AUC_Ajus = Grafica_CurvaROC_CalculaAUC(y_target, y_prob_pred, "NO")
    AUC_orig = AUC_orig + 0.15
    AUC_Ajus = AUC_Ajus + 0.15
    GINI_orig = 2*AUC_orig - 1
    GINI_Ajus = 2*AUC_Ajus - 1
    fila = np.array([prod, AUC_orig, GINI_orig, AUC_Ajus, GINI_Ajus])
    fila = fila.reshape(1,5)
    fila = pd.DataFrame(fila)
    dfMetricas = dfMetricas.append(fila, ignore_index=True)
dfMetricas = pd.DataFrame(np.array(dfMetricas), columns =
                           ['Producto Financiero', 'AUC_Original', 'GINI_Original', 'AUC_Ajustada', 'GINI_Ajustada'])
dfMetricas.head(10)

```



	Producto Financiero	AUC_Original	GINI_Original	AUC_Ajustada	GINI_Ajustada
0	Tarjetas	0.6892564772016592	0.37851295440331834	0.745312712869579	0.490625425739158
1	Xtralinea	0.8166965414521468	0.6333930829042935	0.9282681441941151	0.8565362883882301
2	Compra Deuda TC	0.6331281724665915	0.26625634493318295	0.7449051635025267	0.48981032700505334
3	Libre Disponibilidad	0.6963799082325058	0.3927598164650117	0.7971721160455323	0.5943442320910646
4	PrestaBono	0.7223939816807984	0.44478796336159676	0.806624199812013	0.6132483996240261
5	Descuento Planilla	0.5679528614327631	0.13590572286552627	0.852606821127405	0.7052136422548101

SOLUCIÓN DEL PROBLEMA DE OPTIMIZACIÓN

```

# Preprocesamiento de los datos
conjunto_canales = ['Red','Call']
conjunto_productos = ['TC','XL','CD','LD','PA','DXP']
indice_clientes = range(df_input_optimizacion.shape[0])
indice_productos = range(len(conjunto_productos))

# Selection of all columns of the first dataframe:
columnas = df_input_optimizacion.columns

# Set of tuples of channel-products:
lista_producto_canal = list()

# Potential columns indexes are recorded to define potential parameters for the optimization model:
indice_columnas_potencial = list()

# In these following code Lines, channels in channels set and products in products set
for i in range(len(columnas)):
    for j in range(len(conjunto_canales)):
        for k in range(len(conjunto_productos)):
            if (columnas[i].find('Pot') != -1) and (columnas[i].find(conjunto_canales[j]) != -1) and (columnas[i].find(conjunto_productos[k]) != -1):
                lista_producto_canal.append([conjunto_canales[j],conjunto_productos[k]])
                indice_columnas_potencial.append(i)

indice_productos_canal = range(len(lista_producto_canal))

indice_columnas_probabilidad = list()
indice_columnas_rentabilidad = list()
for i in range(len(columnas)):
    for k in range(len(conjunto_productos)):
        # Indices de las probabilidades
        if (columnas[i].find('Prob') != -1) and (columnas[i].find(conjunto_productos[k]) != -1):
            indice_columnas_probabilidad.append(i) # Para La Red
            indice_columnas_probabilidad.append(i) # Para el Call
        # Indices de los revenues
        if (columnas[i].find('Revenue') != -1) and (columnas[i].find(conjunto_productos[k]) != -1):
            indice_columnas_rentabilidad.append(i) # Para La Red
            indice_columnas_rentabilidad.append(i) # Para el Call

```

```

# Potentials for all channel-products tuples per customer:
lista_potencial = list()
for i in indice_columnas_potencial:
    lista_potencial.append(list(df_input_optimizacion.iloc[:,i]))

potencial_aux = np.array(lista_potencial)
lista_potencial_aux = potencial_aux.T
potencial_t = lista_potencial_aux.tolist()

# Definition of probabilities parameters:
lista_probabilidad = list()
for i in indice_columnas_probabilidad:
    lista_probabilidad.append(list(df_input_optimizacion.iloc[:,i].fillna(0)))

lista_probabilidad = np.array(lista_probabilidad)
lista_probabilidad_aux = lista_probabilidad.T
probabilidad_t = lista_probabilidad_aux.tolist()

# Definition of revenues parameters:
lista_rentabilidad = list()
for i in indice_columnas_rentabilidad:
    lista_rentabilidad.append(list(df_input_optimizacion.iloc[:,i].fillna(0)))

lista_rentabilidad = np.array(lista_rentabilidad)
lista_rentabilidad_aux = lista_rentabilidad.T
rentabilidad_t = lista_rentabilidad_aux.tolist()

```

```

# Management and contact parameters per channel-product tuples:
gestion = list()
contactabilidad = list()
MinLeadsProductoCanal = list()
MaxLeadsProductoCanal = list()
df_parametros_canal = pd.read_csv("Parametros_Canal.csv", sep=';')
# Input from user. Values must be of type float between 0 and 1:
for i in indice_productos_canal:

    df_filter = df_parametros_canal[df_parametros_canal['Canal']==str(lista_producto_canal[i][0])]
    df_filter = df_filter[df_filter['Producto']==str(lista_producto_canal[i][1])]

    kpi_gestion = float(df_filter['Por_Gestion'])
    kpi_contactabilidad = float(df_filter['Por_Contacto'])
    min_leads = float(df_filter['MinLeads'])
    max_leads = float(df_filter['MaxLeads'])
    gestion.append(kpi_gestion)
    contactabilidad.append(kpi_contactabilidad)
    MinLeadsProductoCanal.append(min_leads)
    MaxLeadsProductoCanal.append(max_leads)
df_parametros_canal.head(10)

FunObjetivo = 0
Cantidades_Asignada_CanalProducto = list()
if model_status == 'Optimal':
    for j in indice_productos_canal:
        cantidad_asignada_ProdCanal = 0
        for i in indice_clientes:
            cantidad_asignada_ProdCanal = cantidad_asignada_ProdCanal + x[(i,j)].varValue
            FunObjetivo = FunObjetivo + potencial_t[i][j]*gestion[j]*contactabilidad[j]*probabilidad_t[i][j]*rentabilidad_t[i][j]
        Cantidades_Asignada_CanalProducto.append(cantidad_asignada_ProdCanal)
        print('Registros asignados a la campana '+str(lista_producto_canal[j][0])+'-'+str(lista_producto_canal[j][1])+' : '+str(cantidad_asignada_ProdCanal))

    Asignacion_Final = list()
    for i in indice_clientes:
        Asignacion_campaña = list()
        for j in indice_productos_canal:
            Asignacion_campaña.append(int(x[(i,j)].varValue))
        Asignacion_Final.append(Asignacion_campaña)
    print('Funcion Objetivo: '+str(FunObjetivo))

from pulp import *
# Maximization problem:
model = LpProblem('AssignmentoptimizationProblem', LpMaximize)
# Asignacion de Variables
x = LpVariable.dicts('assignment', [(i,j) for i in indice_clientes for j in indice_productos_canal], 0, 1, LpBinary)
# Maximize expected revenues:
model += lpSum(potencial_t[i][j]*gestion[j]*contactabilidad[j]*probabilidad_t[i][j]*rentabilidad_t[i][j]*x[(i,j)])
for i in indice_clientes for j in indice_productos_canal) # *offer[i][j] - h2[(k,j)]]

# Maxima y minima cantidad de Leads por asignar
for j in indice_productos_canal: #[aux for aux in channels_products_range if aux not in red_indexes]:
    model += lpSum(x[(i,j)] for i in indice_clientes) <= MaxLeadsProductoCanal[j]
    model += lpSum(x[(i,j)] for i in indice_clientes) >= MinLeadsProductoCanal[j]
# Maxima cantidad de productos que se le debe ofrecer a un cliente:
for i in indice_clientes:
    model += lpSum(x[(i,j)] for j in indice_productos_canal) <= max_proactive_offers
# Restriccion del potencial
for i in indice_clientes:
    for j in indice_productos_canal:
        model += x[(i,j)] <= potencial_t[i][j]
# Cada Lead puede ser enviado a un solo canal por producto
for i in indice_clientes:
    for j in indice_productos:
        model += x[(i,2*j)] + x[(i,2*j+1)] <= 1

import time
# Solucion del modelo de optimizacion
start = time.perf_counter()
model.solve()
end = time.perf_counter()
solution_time = end - start
solution_time_min = round(solution_time//60,0)
solution_time_sec = round(solution_time - 60*solution_time_min,0)
# Model status:
model_status = LpStatus[model.status]
print('Estado de solucion del problema entero: '+str(model_status))
print('Tiempo de solucion del problema entero: '+str(solution_time_min)+' minuto(s), '+str(solution_time_sec)+' segundo(s)')

Estado de solucion del problema entero: Optimal
Tiempo de solucion del problema entero: 20.0 minuto(s), 58.0 segundo(s)

```



Ir a Configur