Universidad de Lima
Facultad de Ingeniería y Arquitectura
Carrera de Ingeniería de Sistemas

MODELO DE APRENDIZAJE AUTOMATIZADO DEL PROCESO DE VENTA DE PRODUCTOS FINANCIEROS EN UN CALL CENTER

Trabajo de suficiencia profesional para optar el Título Profesional de Ingeniero de Sistemas

Jorge Joao Gutierrez Salas
Código 20011728

Vanessa Stephany Vigo Liñan
Código 20020884

Asesor

Manuela Linares Barbero

Lima – Perú
Febrero de 2021

UNIVERSIDAD DE LIMA
MCMLXV
SCIENTIA ET PRAXIS

AUTOMATED LEARNING MODEL OF THE PROCESS OF SELLING FINANCIAL PRODUCTS IN A CALL CENTER

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

El presente proyecto se enfocó en el diseño, construcción e implementación de un servicio que realizó la predicción del comportamiento de un potencial cliente, con el fin de concretar la venta de un producto financiero de manera anticipada, el cual está basado en machine learning. El prototipo ha sido probado con una base de datos de clientes de una entidad financiera, a los cuales se les ofreció un producto financiero como, por ejemplo, un préstamo de libre disponibilidad, tarjetas de crédito, préstamos a pymes, créditos hipotecarios, créditos vehiculares, etc. y obteniendo como resultado una venta concretada o una desestimación del ofrecimiento. Con esta información, y a través de diferentes algoritmos predictivos, se construyó un modelo adecuado que permita predecir ventas de productos financieros. Los beneficiarios de la solución implementada serán empresas que brinden servicios de outsourcing (BPO) a entidades financieras. Estas empresas obtienen utilidades por comisión de venta utilizando recursos humanos y tecnológicos para lograr concretar ventas. Bajo este esquema, el modelo predictivo implementado permitió disponibilizar un servicio el cual, al ser invocado, permita aumentar la probabilidad de venta y a su vez logró optimizar la operación a nivel de recurso humano, reduciendo la cantidad de ejecutivos de venta, y aumentando la productividad del área de back office del servicio de outsourcing, evitando tiempos muertos propios de una operación dependiente de las ventas concretadas.

El módulo de predicción desarrollado se presentó en una aplicación web que permitió ingresar los datos de entrada (registro histórico de ventas) y como resultado se mostró las predicciones basadas en los modelos de machine learning que obtuvieron mejores resultados. La predicción y su evolución en el tiempo se presentaron en un dashboard interactivo mostrando los resultados de venta mensuales, campañas de venta, cantidad de leads, modelos predictivos, efectividad de venta, venta por mes y venta total.

Palabras clave:

Machine learning, base de datos, modelos, dashboard, productos financieros.

ABSTRACT

This project focused on the design, construction and implementation of a service that predicted the behavior of a potential client, in order to finalize the sale of a financial product in advance, which is based on machine learning. The prototype has been tested with a database of clients of a financial institution, which were offered a financial product such as, for example, a freely available loan, credit cards, loans to pymes, mortgage loans, vehicle loans, etc. and obtaining as a result a final sale or a rejection of the offer. With this information, and through different predictive algorithms, an adequate model was built to predict sales of financial products. The beneficiaries of the implemented solution will be companies that provide outsourcing services (BPO) to financial entities. These companies obtain profits from sales commission using human and technological resources to achieve sales. Under this scheme, the predictive model implemented made it possible to make a service available which, when invoked, allows to increase the probability of sale and in turn managed to optimize the operation at the human resource level, reducing the number of sales executives, and increasing productivity. of the back office area of the outsourcing service, avoiding downtime typical of an operation dependent on the completed sales. The prediction module developed was presented in a web application that allowed the input data to be entered (historical sales record) and as a result the predictions based on the machine learning models that obtained the best results were shown. The prediction and its evolution over time were presented in an interactive dashboard showing the monthly sales results, sales campaigns, number of leads, predictive models, sales effectiveness, sales per month and total sales.

Keywords:

Machine learning, data base, models, dashboard, financial products.

CAPÍTULO I: INTRODUCCIÓN

El sector banca en el Perú, cuyas diversas operaciones se realizan en territorio nacional, terceriza y contrata complejos servicios de BPO (Business Process Outsourcing) para sostener, operar y hacer eficientes sus múltiples procesos de negocio, siendo uno de ellos la venta de productos financieros, como por ejemplo venta de créditos de libre disponibilidad, venta de tarjetas de créditos, venta de créditos vehiculares, entre otros.

Para ello, estas entidades financieras proveen de información necesaria (correspondiente a los datos de clientes/no clientes potenciales como nombres, apellidos, edad, lugar de residencia, números telefónicos, oferta, etc.) a sus BPO para lograr la contactabilidad, colocación y venta de sus productos, convirtiéndose en socios estratégicos.

Las empresas de BPO que brindan este servicio a los bancos realizan un despliegue de procesos, recursos humanos e infraestructura importante para garantizar una correcta contactabilidad a un posible cliente potencial. Mediante los centros de gestión de venta denominados Call Centers se realizan llamadas telefónicas directas, realizadas por personal especializado (ejecutivos de venta telefónica), a un posible comprador, y que, mediante un protocolo informativo de ofrecimiento, se tienta concretar la venta.

Las empresas BPO que utilizan Call Centers, reciben una comisión por venta concretada, por lo que buscan maximizar la venta utilizando mecanismos de optimización y eficiencia de la operación como realizar un tratamiento de datos que permita maximizar la venta y convirtiéndose en una oportunidad para maximizar la utilidad del servicio prestado.

A continuación, se explicará en los diferentes capítulos como se abordará esta solución.

Capítulo 2: Se explicó los fundamentos teóricos de machine learning, modelos, bases para elección de mejor modelo de predicción.

Capítulo 3: En este capítulo se explicó la deseabilidad, factibilidad de nuestro proyecto como los beneficios esperados como económicos.

Capítulo 4: Se definió el proyecto, se explicará el flujo del servicio, indicadores, objetivo del proyecto, los roles requeridos para el desarrollo de nuestro proyecto, cronograma y fases del proyecto basados en CRISP-DM.

Capítulo 5: En este capítulo se desarrolló el MVP basado en las diferentes fases de la metodología CRISP-DM como son: fase de comprensión del problema, fase de comprensión de los datos, fase de preparación de los datos, fase de modelado, fase de evaluación y fase de construcción de dashboard.

CAPÍTULO II: FUNDAMENTOS TEÓRICOS

2.1 Inteligencia Artificial

Una de las tendencias en el ámbito de las ciencias de la computación de los últimos años, es sin duda, el estudio y desarrollo de programas que intentan alcanzar funciones cognitivas e interpretativas, con el fin de lograr resultados "predictivos e inteligentes" en diferentes ámbitos del común quehacer humano, basados en un aprendizaje estadístico-matemático. Tukey y Naur (como fueron citados en Lope et al., 2020, p. 74) explican una evolución de ambas ciencias, definiendo por primera vez el análisis de datos como procedimiento para lograr una interpretación.

En este contexto, uno de los objetivos de la implementación de proyectos de inteligencia artificial es el descubrimiento de insights y patrones que permitan identificar necesidades ocultas, y poder utilizar esta información con el fin de proponer u ofrecer soluciones con valor agregado, e incluso implementarlos de forma automatizada, sin intervención humana.

En el enfoque empresarial se tiene la oportunidad de no solo utilizar esta tecnología para centrarse en la utilidad y ampliar su segmento de mercado, en lugar de sacarle provecho en el campo de estratégica de gestión de relaciones con los clientes (CRM), ya que, al tener la información sobre el comportamiento del cliente, se pueden desarrollar planes de marketing apropiados, incluido el desarrollo de una relación de lealtad a largo Plazo (Wassouf et al., 2020).

Existen diversas ramas claramente identificadas dentro de la Inteligencia Artificial, como Machine Learning (aprendizaje de máquina), Deep Learning (aprendizaje profundo), Natural Procesing Language (procesamiento de lenguaje natural), etc. y cada una con un campo de acción definido y niveles o capas técnicas altamente especializadas.

Ante una necesidad en el mercado de contar con científicos de datos y una oportunidad claramente identificada ya que la demanda exige estos recursos para atender las nuevas tecnologías como machine learning, su rápido desarrollo requiere de avances significativos según Mamaqui et al. (como citarón en Lope et al., 2020, p. 73).

Según White (como se citó en Lope et al., 2020, p. 73), los científicos de datos se convertirán en los profesionales que suministrarán soporte y servicios de implementación, utilizan técnicas especializadas para examinar la información y descubrir nuevas ideas.

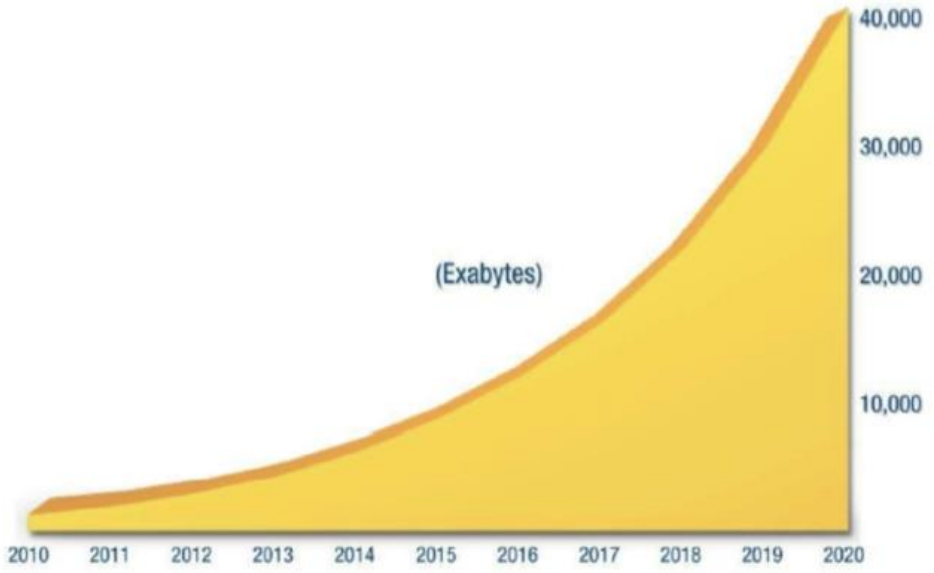
Estos perfiles, según Naur (como se citó en Lope et al., 2020, p. 73), se agrupan en competencias que provienen de diferentes áreas y especialidades de conocimiento como matemáticas, estadística, economía, programación informática, gestión de empresa y publicidad.

Es importante mencionar que el análisis y tratamiento de datos (en muchos de los casos, en grandes volúmenes) es la fuente de esta explotación descubridora, ya no solo considerando datos estructurados, que van desde registros en hojas de cálculo hasta grandes Datawarehouse, sino también datos no estructurados, como por ejemplo grabaciones de voz, datos en formato texto, data generada por dispositivos de IoT, imágenes en diversos formatos, etc. El crecimiento de estos datos se ha dado de forma exponencial desde el inicio del nuevo milenio.

Según un estudio realizado por IDC (International Data Corporation) (como se citó en Gonzáles, 2016, p. 101), teniendo como sponsor a EMC Corporation (compañía multinacional), realizado en 2012 muestra el crecimiento en exabytes de la cantidad de información generada hasta el 2020, llegando a un total de 40 900 exabytes. Podemos apreciar que prácticamente toda la información generada en los últimos 10 años equivale a la totalidad de información generada por toda la humanidad a lo largo de la historia, considerando que hasta el 2005 se habían generado solo 130 exabytes.

Figura 2.1

Tendencia de generación de información entre 2010 y 2020

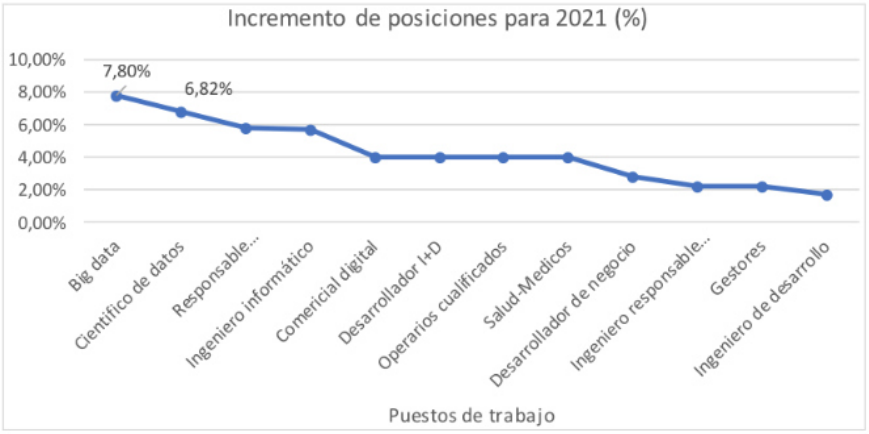


Nota. La cantidad de información de la tendencia se expresa en exabytes. De "Big data para el análisis de las necesidades traductológicas en cinco capitales de Europa, 2016", por Adela González, 2016, Skopos, N.º. 7, p. 101.

A su vez, según las proyecciones para los próximos años, la demanda en posiciones de Data Science (Ciencia de Datos) y Big Data serán altamente demandadas, confirmando de esta forma que la tendencia a implementar tecnología orientada al análisis de datos cognitivo se repuntará. Las profesiones digítales emergentes tendrán como principal representante a los Científicos de Datos y especialistas en Big Data (Lope et al., 2020).

Figura 2.2

Tendencias de los puestos de trabajo más demandados

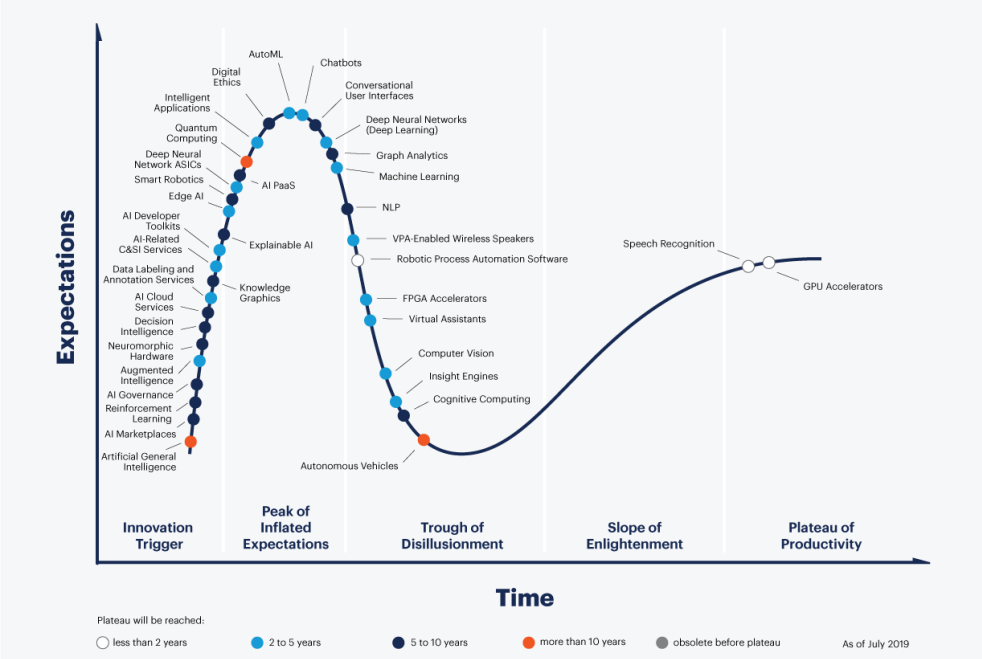


Nota. De "La Inteligencia Artificial: desafíos teóricos, formativos y comunicativos de la ratificación, 2020", por Lope Salvador, V., Mamaqi, X. y Vidal Bordes, J, 2020, Icono 14, 18 (1), p.72 (https://doi.org/10.7195/ri14.v18i1.1434)

A nivel corporativo, según Gartner, entre 2018 y 2019 las empresas que implementaron IA en sus organizaciones crecieron entre 4% y 14% según Gartner's 2019 CIO Agenda Survey, evidenciando que las implementaciones asociadas a IA forman parte de la estrategia de crecimiento de las organizaciones a nivel mundial. A su vez, el Hype Cycle de la IA para Gartner ubica al aprendizaje de máquina aún dentro de la curva de altas expectativas de crecimiento entre (2 a 5 años). (Goasduff, 2019)

Figura 2.3

Proyección de Gartner de tendencias tecnológicas de inteligencia artificial 2019



Nota. De "Top trends on the Gartner hype cycle for artificial intelligence, 2019", por Smarter with Gartner, 2019 (https://www.gartner.com/smarterwithgartner/top-trends-on-the-gartner-hype-cycle-for-artificial-intelligence-2019/)

2.2 Machine Learning

El Machine Learning es la ciencia o arte de programación de computadoras las cuales pueden aprender a partir de la data (Géron, 2019). El Machine Learning se enfoca a un aprendizaje en base a información histórica, estableciendo una función que explique el comportamiento de una variable dependiente, utilizando para ello variables independientes significativas, las cuales tengan un impacto directo e infiriendo al resultado predictor. Este aprendizaje se realiza mediante el análisis de datos muestreados aleatoriamente que genera una distribución de datos estacionaria (Lesort et al., 2020).

Los ejemplos que utiliza el sistema para aprender se llama *training set* (Géron, 2019), y su representatividad del total de la muestra dependerá de las consideraciones que realice el científico de datos.

Existen diferentes algoritmos que se utilizan dependiendo del escenario a estudiar, que van desde una regresión lineal básica, hasta análisis avanzados como el Deep Learning que implican diseño de modelos predictivos con redes neuronales convolucionales.

Machine learning no es auto programación, sino auto aprendizaje de datos y experiencia para generar patrones y resolver nuevas tareas. Este aprendizaje es la combinación de técnicas, datos, conceptualización de análisis de datos y algoritmos para generar nuevos patrones o modelos de predicción. (Manrique, 2020)

A continuación, se citan algunos ejemplos, según Manrique (2020), de la aplicación de machine learning más comunes:

- Detección de correos electrónicos no deseados (spam)
- Detección de patrones en imágenes a través de la cámara fotográfica
- Uso de Amazon para patrones de compra (Manrique, 2020)

A continuación, se citan algunos ejemplos, según Géron (2019) de aplicación de machine learning:

- Detección de tumores cerebrales.
- Análisis de imágenes de productos en una línea de producción (clasificación).
- Detección de comentarios ofensivos en forums.
- Automatización y generación de resúmenes de textos o documentos.
- Creación de chatbots o asistentes personales.
- Detección de fraudes utilizando tarjetas de crédito.
- Segmentación de clientes basadas en compras, para diseñar diferentes estrategias de marketing.
- Recomendación de productos basadas en compras pasadas.
- Clasificación automática de nuevos artículos. (Géron, 2019)

Los tipos de aprendizaje en Machine learning son:

- Aprendizaje supervisado: en donde se enseña al algoritmo a realizar su trabajo para producir una salida que ya se conoce. (Manrique, 2020)

  El aprendizaje supervisado se utiliza comúnmente en aplicaciones donde datos históricos predicen eventos futuros probables. (Manrique, 2020). Por ejemplo, puede anticipar cuándo es probable que transacciones con tarjetas de crédito sean fraudulentas o qué cliente de una aseguradora tiene la probabilidad de iniciar un reclamo. (Pallarés, 2019).

  Según Géron (2019), los más importantes algoritmos de aprendizaje supervisado son:

- K-Nearest neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines.
- Decision Trees and Random Forests
- Neural networks, otros. (Géron, 2019)

- Aprendizaje no supervisado: el sistema intenta aprender sin un profesor (Géron, 2019), pero siempre bajo el modelo predictivo entrenado, comprende los datos no clasificados o etiquetados para descubrir patrones similares. (Manrique, 2020). Según Géron (2019) los más importantes algoritmos no supervisado son:
  - Clustering (K-Means, DBSCAN, Hierarchical Cluster Analysis).
  - Anomaly Detection and novelty detection (One-class SVM, Isolation Forest)
  - Visualization and dimensionality reduction (Principal Component Analysis, Kernel PCA, Locally Linear Embedding – LLE, etc.)
  - Association rule learning, otros. (Géron, 2019)

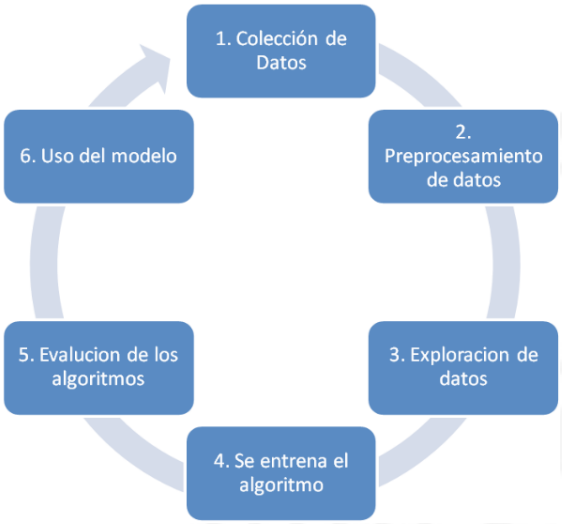
- Aprendizaje reforzado: el sistema de aprendizaje, llamado agente, puede observar el entorno, seleccionar y realizar acciones obteniendo recompensas como resultado, teniendo una política que defina la acción. (Géron, 2019)

Adicionalmente se trata de un aprendizaje automático, el sistema aprende sin tener información de la posible salida. (Manrique, 2020)

Los pasos para poder construir un modelo de machine learning son por lo general seis, pero dependerá de la metodología a aplicar en donde se definirán los pasos bajo las mejores prácticas.

Figura 2.4

Proceso de construcción de un modelo de machine learning



Nota: De "Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo", Manrique Esperanza, 2020, Revista Ibérica de Sistemas e Tecnologías de Información; Lousada N.º E28, 586-599.

Según Manrique (2020), se describen los pasos de la siguiente forma:

1. Recolectar los datos: Los datos se pueden recolectar de fuentes tal como un sitio web, utilizando una API o una base de datos. Este paso es uno de los más complicados y requiere un tiempo determinado.

2. Preprocesamiento de los datos: Con los datos disponibles, se debe asegurar que todos tengan un formato correcto para alimentar el algoritmo de aprendizaje. Por lo general se tiene que realizar varias tareas de preprocesamiento antes de poder usar los datos.

   Implica una limpieza de los datos que se utilizarán para realizar el modelo de machine learning. Para lograr un preprocesamiento de datos se deben considerar la importación de un dataset obteniendo los datos, tratamiento de

datos faltantes o desconocidos, tratamiento de datos categóricos, dividir el conjunto de datos en el conjunto de entrenamiento y el conjunto de prueba, y finalmente, el escalamiento de datos. (Gomila et al., 2020)

Según Gomila et al., (2020), Python brinda una serie de librerías para realizar un correcto preprocesamiento de datos:

- Importar Dataset: Pandas.
- Tratamiento de NAs: sklearn.preprocessing import imputer
- Codificar datos categóricos: sklearn.preprocessing import LabelEncoder, OneHotEncoder.
- Escalado de variables: sklearn.preprocessing import StandarScaler.
- Dividir el conjunto de entrenamiento y el conjunto de testing: sklearn.model_selection import train_test_split.

3. Explore los datos: Luego se realiza un análisis previo para corregir los casos de valores faltantes o tratar de encontrar a primera vista cualquier patrón en ellos que facilite la construcción del modelo. En este punto, se deben detectar valores atípicos; o encuentre las características que tienen más influencia para hacer una predicción.
4. Entrena el algoritmo: los algoritmos de aprendizaje se alimentan con los datos que se procesaron en las etapas anteriores. La idea es que los algoritmos pueden extraer información útil de los datos iniciales y luego realizar predicciones.
5. Evaluar el algoritmo. Se realizan las pruebas de la información que genera el conocimiento del entrenamiento previo que se obtuvo a través del algoritmo.

Según Gomila et al., (2020), existen los siguientes tipos de modelos en machine learning:

- Modelos de Regresión: Los modelos de regresión se utilizan para poder predecir variables numéricas o valor continuo, como precio, sueldo, kilogramos, etc. Existen muchas técnicas como la regresión lineal, regresión lineal múltiple, regresión polinómica, árboles de decisión, entre otros.

- Modelos de Clasificación: Los modelos de clasificación se caracterizan por predecir una categoría, por ejemplo, el sexo, tipo o característica, deseo de compra, entre otros. Los modelos de clasificación incluyen modelos lineales como la regresión logística, SVM, así como no lineales como el modelo KNN, Kernel SVM, árboles de decisión aleatorios, aumento de gradiente, Naive Bayes etc.

En complemento, existen diversos flujos o fases que permiten igualmente tener una hoja de ruta en cuanto a la ejecución de un proyecto de machine learning. Según Géron (2019) proponen la siguiente lista de pasos.

1. Estudiar la data.
2. Seleccionar el modelo.
3. Entrenar sobre la data de entrenamiento.
4. Aplicar el modelo para realizar predicciones con nuevos casos. (Géron, 2019)

A continuación, explicaremos los modelos de machine learning de clasificación Random Forest y Gradient Boosting Classifier, los cuales fueron los algoritmos con mayor importancia e impacto dentro del modelo de machine learning propuesto.

Random Forest:

Como indica Breiman (como se citó en Reis et al., 2019, p.2) Random Forest es un método de aprendizaje conjunto que opera mediante la construcción de una gran cantidad de árboles de decisión durante el proceso de entrenamiento.

Un árbol de decisión es un modelo no paramétrico, el cual se describe como un gráfico de árbol de arriba hacia abajo y es utilizado tanto en modelos de regresión y clasificación, cuya relación entre las variables independientes y variable dependiente está representada por una serie de condiciones conjuntas, organizadas en una estructura de árbol. El proceso de entrenamiento empieza con la totalidad del dataset de entrenamiento y un nodo de árbol simple, el cual se denomina raíz del árbol. El algoritmo busca la mejor división entre los objetos de dos clases. La definición de mejor separación es un parámetro del algoritmo, contando con opciones como la entropía o el parámetro de Gini. (Reis et al., 2019)

Por lo expuesto, el algoritmo Random Forest contiene un conjunto de predictores basados en una estructura de árbol, y que cada uno es construido usando una inyección de aleatoriedad, es por ello por lo que son denominados bosques aleatorios (los árboles individuales que componen el bosque se cultivan a máxima profundidad). Un elemento importante del algoritmo está en el hecho de que su rendimiento óptimo se logre aparentemente con un solo parámetro de ajuste, al que la sensibilidad es mínima, permitiendo que la metodología sea notable y sostenible. (Segal, 2004)

Por ejemplo, el modelo de árboles de decisión aleatorios (Random Forest), demuestra ser un clasificador de múltiples clases, eficaces y rápidos para muchas tareas, incluyendo una integración de manera eficiente con GPUs. (Shotton et al., 2011)

Este modelo fue utilizado en el Proyecto de Kinect, de Microsoft, para obtener seguimiento interactivo del cuerpo humano. El algoritmo forma un componente central de la plataforma de juegos Kinect.

Gradient Boosting Classifier:

La familia de los métodos Boosting (de aumento) se caracteriza por añadir nuevos modelos al conjunto de entrenamiento de forma secuencial, en cada iteración se entrena un nuevo modelo respecto al error. (Natekin et al., 2013), y cuyo objetivo es iterar clasificadores simples y utilizar sus resultados para obtener mejores resultados (pasando de un aprendizaje débil hacia un aprendizaje robusto).

La idea fundamental del aumento de gradiente es hacer que el aprendizaje básico no vuelva a ponderar observaciones, pero en el vector gradiente negativo de la función de pérdida evaluado en la iteración anterior (Mayr et al., 2014).

Para el desarrollo del modelo predictivo propuesto, seleccionamos el modelo de aprendizaje de clasificación, debido a que nuestra variable a predecir es categórica (Si compra el producto financiero / No compra el producto financiero), con el fin de obtener el modelo de clasificación que obtenga los mejores resultados predictivos, orientados en el problema de determinar qué tan probable es que un lead pueda adquirir un producto financiero. (Shotton et al., 2011)

Lenguajes de Programación en Machine Learning:

Existen diferentes lenguajes de programación que pueden utilizarse para desarrollar modelos de machine learning y es importante elegir el adecuado dependiendo de las necesidades.

Tener más datos llevará al modelo a tener mejores resultados, el objetivo de generar más datos se puede basar en:

- Regresión lineal o polinómica
- Árboles de decisión
- Redes neuronales
- Red bayesiana
- Cadenas de Markov (Manrique, 2020)

Según Manrique (2020), los lenguajes de Programación de Machine learning, más comunes son: Python, Julia, R y Matlab, que será utilizado dependiendo del problema que se quiera resolver.

Figura 2.5

Lenguajes de programación para Machine learning

Nota: De "Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo", Manrique Esperanza, 2020, Revista Ibérica de Sistemas e Tecnologías de Información; Lousada N.º E28, 586-599.

1.- Python

Lenguaje de programación orientado a objetos, donde el código se ejecuta en el navegador al cargar la página.

Ventajas:

- Velocidad al ejecutar el programa.

- Cuenta con diferentes bibliotecas y funciones incorporadas.
- Es gratis.
- Simple y con velocidad para la creación de los programas (menos líneas de código)
- Se puede desarrollar en diferentes plataformas: unix, windows, OS/2, mac, entre otras. (Manrique, 2020)

Características de Python:

- Se puede aplicar en machine learning para el desarrollo web y automatización de scripts y procesos.
- Muchas bibliotecas y macros ayudan la codificación y ahorran tiempo.
- Facilidad de uso con código legible y conciso.
- Permite probar algoritmos de manera ágil sin necesidad de implementarlos.
- Fácil de leer para poder colaborar en el desarrollo.
- Código abierto y con gran documentación. (Manrique, 2020)

Bibliotecas de Python para Machine learning:

Según Manrique (2020), se cuentan con diferentes bibliotecas que ayudan a ampliar las funcionalidades del lenguaje, a continuación, se listan las más importantes.

Tabla 2.1

Librerías de Python

Librería	Aplicación o Uso
Scikit-Learn	Se utiliza para clasificaciones, extracción de características, regresiones, agrupaciones, reducción de dimensiones, selección de modelos o preprocesamiento. (Pedregosa, Varoquaux, Gramfort, et. Alabama. ,2011).
Statsmodels	Es otra gran biblioteca que se centra en modelos estadísticos y se utiliza principalmente para análisis predictivos y exploratorios. (Seabold, S. y Perktold, J., 2010).
PyMC	Implementa modelos estadísticos bayesianos, incluida la cadena Markov Monte Carlo (MCMC). Ofrece funcionalidades para hacer que el análisis Baiesiano sea lo más simple posible (Patil, A., Huard, D. y Fonnesbeck, C. J. , 2010).
NTLK	Es la biblioteca líder para el procesamiento del lenguaje. Proporciona interfaces fáciles de usar para más de 50 cuerpos y recursos léxicos, como WordNet, junto con un conjunto de bibibliotecas de procesamiento de texto para clasificación, tokenización, etiquetado, análisis y razonamiento semántico (Patil, A., Huard, D ., Y Fonnesbeck, CJ, 2010).

Nota: De "Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo", Manrique Esperanza, 2020, Revista Ibérica de Sistemas e Tecnologías de Información; Lousada N.º E28, 586-599.

2.- Lenguaje R

Lenguaje de programación orientado al análisis estadístico y representación gráfica de resultados, es libre y multiplataforma, lenguaje basado en comandos para implementar un flujo de trabajo o script R.

Características:

- Permite crear gráficos en LaTex
- Contiene herramientas estadísticas: modelo lineal, no lineal, pruebas estadísticas, algoritmos de clasificación y agrupación.
- Integración con diferentes bases de datos.
- Permite visualización de gráficos, manipulación de datos y aprendizaje automático. (Manrique, 2020)

Librerías de Lenguaje R:

Las librerías más comunes son:

- Tidyverse: permite explorar, ordenar y analizar los datos que incluye bibliotecas ggplot2, tibble, tidyr, readr, purrr y dplyr
- Caret: ofrece diferentes herramientas para la construcción del modelo para las etapas de preparación de los datos, seleccionar los atributos y evaluar los modelos. (Manrique, 2020)

3.- Lenguaje Matlab

Basado en matrices para mostrar las matemáticas, contiene gráficos integrados para facilitar la visualización y obtención de la información. Contiene una amplia biblioteca de herramientas integradas (toolboxes) que permite trabajar de manera conjunta.

Características de Matlab:

- Permite ajuste de hiperparámetros y selección de funciones automáticas para optimizar los modelos.
- Permite usar el mismo código para escalar el procesamiento de big data y clúster.
- Generación de código C/C++ para aplicaciones integradas y de alto rendimiento.

- Contiene todos los algoritmos comunes de aprendizaje supervisado y no supervisado,
- Ejecución rápida para los cálculos estadísticos y de aprendizaje automático.
- Permite comparar los enfoques como regresión logística, árboles de clasificación, máquinas de vectores de soportes, métodos combinados y aprendizaje profundo.
- Utiliza flujos de trabajo integrados para el análisis de datos.
- Cuenta con un kit de herramientas para algoritmos de aprendizaje supervisado y no supervisado que incluye máquinas de vectores de soporte (MVS), árboles de decisión, vecinos k más cercanos, medios k, medoides k, agrupamiento jerárquico, modelos gaussiana y modelos Markov. (Manrique, 2020)

4.- Lenguaje Julia

Utiliza tipos dinámicos y soporte de uso interactivo. Lenguaje de alto rendimiento, se compila el código en múltiples plataformas bajo LLVM (máquina virtual de bajo nivel). Cuenta con una biblioteca estándar con entradas y salidas asíncronas, control de procesos, administrador de paquetes, entre otros.

Ventajas de lenguaje Julia

- El lenguaje Julia es más rápido que Python por el uso de declaraciones de tipo como la compilación JIT (Just in time).
- Gestión automática de la memoria no existe sobrecarga para liberar y asignar memoria.
- Sintaxis según las matemáticas, similar a las fórmulas matemáticas de fácil entendimiento.
- Uso de paralelismo para la gestión de recursos para no cargar el sistema. (Manrique, 2020)

Comparación de los cuatro lenguajes:

A continuación, se muestra un cuadro comparativo con diferentes factores para la decisión del lenguaje de programación:

Tabla 2.2

Factores comparativos para elección de lenguaje

Factores	Descripción
Velocidad	Al elegir el mejor lenguaje de programación, la velocidad es esencial. R fue desarrollado básicamente como un lenguaje estadístico, esto significa que tiene un mayor análisis de datos y soporte estadístico. Por su parte, Python depende de los paquetes, por lo tanto, cuando se trata de tareas relacionadas con estadísticas, R tiene una ventaja en comparación con Python y es un poco más rápido.
Curva de aprendizaje	Cuando se trata de la perspectiva funcional, R es el lenguaje de programación, mientras que cuando se trata de estar orientado a objetos, Python es el lenguaje. Si pertenece al grupo de programadores funcionales, entonces aprender Python será mucho más fácil en comparación con R. Cuando llegue a Matlab y Julia, ambos son similares a escribir algunas ecuaciones matemáticas, y sí, son fáciles de aprender e implementar.
Costo	El único lenguaje que se paga y necesita una licencia para su uso es Matlab. Los otros tres lenguajes son de código abierto y es completamente gratuito. Por lo tanto, cuando tiene recursos gratuitos disponibles, ¿por qué alguien elegiría pagar? Esta es la razón por la cual Matlab se retrasa un poco en comparación con otros lenguajes.
Comunidad para soporte	Todos los lenguajes de programación son muy populares en el mercado y cuentan con un gran apoyo de la comunidad. Aunque debe mencionarse que Python es la que tiene la comunidad más grande en Internet que es bastante solidaria en el momento de un problema con los desarrollos.

Nota: De "Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo", Manrique Esperanza, 2020, Revista Ibérica de Sistemas e Tecnologías de Información; Lousada N.º E28, 586-599.

En conclusión, cuando se trata de análisis estadístico el mejor es lenguaje R, para tareas relacionadas con la visión por computadora es Matlab, si se trata de bioinformática o biología Julia es lo mejor, pero si es para procesamiento de datos y resultados lo recomendable es Python.

Bajo este concepto, Gartner indica que el aprendizaje de máquina puede resolver problemas que van desde problemas comerciales hasta problemas de diagnóstico médico o lavado de activos. Por ejemplo, la empresa Volvo utiliza el machine learning para predecir fallas de piezas en sus vehículos, y de esta forma predecir que unidades necesitan servicio de mantenimiento, logrando optimizar su operación con el adicional de mantener una experiencia del cliente final con valor agregado, disminuyendo el riesgo de averías y de accidentes automovilísticos. (Goasduff, 2019)

Fase de Evaluación

El propósito de la fase de evaluación en machine learning es determinar la utilidad de los clasificadores resultantes en colecciones de conjuntos de datos.

Según Japkowicz, 2006 el propósito de la evaluación es ofrecer formas convenientes de juzgar el desempeño de un aprendizaje o una hipótesis y compararlo con otros, los métodos de evaluación pueden verse como resúmenes de rendimiento de los sistemas.

Según Japkowicz, 2006 el proceso de evaluación de un modelo debe incluir los siguientes pasos:

1.- Decidir qué propiedades de la clasificación deben medirse y elegir una métrica de evaluación.

2.- Decidir qué método de estimación de confianza debe utilizarse para validar los resultados.

3.- Verifique que las suposiciones hechas por la métrica de evaluación y el método estimación de confianza sean verificados bajo el dominio en consideración.

4.- Ejecutar el método de evaluación con el método elegido y analizar los resultados.

5.- Interprete estos resultados con respecto a dominio.

Las consideraciones sobre la evaluación del aprendizaje automático se dividen en dos partes:

- Las métricas de desempeño empleadas.
- El método de estimación de confianza.

A su vez, Japkowicz, 2006, indica que se da más peso a la creación de algoritmos sofisticados para resolver problemas que las observaciones que conducen a este algoritmo.

Según Japkowicz, 2006 las 3 métricas de evaluación más comunes en machine learning son las siguientes:

- Accuracy (error rate).
- Precision/Recall
- ROC Analysis.

Para el presente trabajo, hemos realizado análisis de evaluación utilizando las métricas Accuracy, Precision/Recall y ROC Analysis, teniendo como base del cálculo a la matriz de confusión (Confusion Matrix).

La matriz de confusión permite determinar los falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos de un modelo de aprendizaje automático de clasificación, y que cuyo análisis permite determinar la eficacia del modelo ejecutado.

Tabla 2.3

Matriz de confusión

True class ->	**Positive**	**Negative**
Hypothesized Class		
Yes	TP	FP
No	FN	TN
	P= TP + FN	N= FP + TN

Nota. De "Why Question Machine Learning Evaluation Methods?", Japkowicz, Nathalie, 2006

Donde:

TP: Recuento de verdaderos positivos.

FN: Recuento de falsos negativos.

FP: Recuento de falsos positivos.

TN: Recuento de verdaderos negativos.

P: Suma de TP y FN

N: Suma de FP y TN

Las métricas de evaluación utilizados son:

Accuracy = (TP + TN)/ (P + N)

Precision = TP/(TP+FP)

Recall = TP/P

FP Rate = FP/N

Accuracy:

El accuracy o exactitud permite medir el porcentaje de aciertos tanto de verdaderos positivos y verdaderos negativos sobre el total de mediciones del modelo. No obstante, no distingue entre los tipos de error que comete, falsos positivos y falsos negativos. (Japkowicz, 2006)**.**

En este punto es importante mencionar que al no considerar este problema estaríamos cometiendo errores de gran magnitud, dependiendo del caso de estudio. En

nuestro caso, al ser un modelo de predicción de concretización de venta, el valor de FN (Falsos negativos) indica los leads que el modelo indico no comprarían, sin embargo, en la realidad si lo hicieron, siendo este caso de menor impacto. Por ejemplo, en nuestro modelo, en caso no haya predecido una posible venta esta no se pierde, ya que esta forma parte de sus funciones de contactabilidad.

Precisión y Recall:

La precisión evalúa hasta qué punto el clasificador determinó correctamente clasificando ejemplos como positivos, mientras que recall evalúa en qué medida todos los ejemplos que debían ser clasificados como positivos fueron así, Japkowicz, 2006.

FP Rate:

Mide el porcentaje de los falsos positivos en base al total de mediciones falsas que se dieron en la realidad.

ROC analysis

La curva ROC (Receiver Operating Characteristic) es el gráfico resultante de representar, para cada valor umbral, las medidas de sensibilidad y especificidad de la prueba diagnóstica.

La metodología ROC fue desarrollada en el contexto de la detección de señales electrónicas en los inicios de la década de los 50. A mediados de los 60 se habían usado las curvas ROC en psicología y psicofísica experimental.

Las medidas que utiliza la curva ROC son el Recall (TP / P) y la tasa de falsos positivos (FP / N), donde el FP Rate en el eje x, y el Recall en el eje y. Los puntos del gráfico se pueden interpretar como refleja un mejor desempeño si son ubicados simultáneamente cerca de 0 en el eje x y cerca de 1 (o 100%) en el eje y.

Si la prueba es perfecta (sin solapamiento) existirá una región en la que cualquier punto de corte tiene sensibilidad y especificidad iguales a 1: la curva sólo tiene un punto (0,1). Si la prueba fallará, la sensibilidad (verdaderos positivos) es igual a la proporción de falsos positivos, la curva sería una línea recta de (0,0) a (1,1), por tanto, cuanto más cerca esté la línea de la esquina superior izquierda, mayor será la precisión de la prueba.

2.3 Inteligencia Artificial Explicable

Un tema muy importante a considerar es la transparencia, la explicabilidad y la interpretabilidad del modelo (Barredo et al., 2020). Dentro de lo que denominan AIX (IA explicable), la comprensibilidad mide el grado en que un humano puede entender una decisión tomada por un modelo, ya que en términos tradicionales (diseño de algoritmos), cualquier científico de datos puede sesgar de forma no intencional el resultado de la predictibilidad. Por ello se debe comprender el cómo y por qué el modelo machine learning sustenta la toma de decisiones y sus resultados, y de esta forma abrir cajas negras de los algoritmos de machine learning.

Por ejemplo, en el sector financiero se realizan muchos proyectos de IA y machine learning asociados a la venta de productos y servicios, tales como préstamos de libre disponibilidad, protección de tarjetas, créditos hipotecarios, etc. Al realizar el análisis de solvencia de los potenciales clientes, existen diversos factores que no son considerados dentro de los modelos, quizás por la complejidad de encontrar estas variables, así como de obtenerlo y/o convertirlos a formato digital. Usualmente la calificación de riesgos de un potencial cliente es un factor determinante, ya que al ser calificado de forma negativa por las centrales de riesgos es automáticamente separado de la evaluación.

La evaluación del perfil del postor/ solicitante cuenta con diferentes factores como la situación laboral, ingresos, gastos, motivo de préstamo, expectativas, etc. No obstante, existen consideraciones que no son tomadas en cuenta como, por ejemplo, la opción de aportar garantías adicionales, estar preparado para consolidar su deuda, presente económico estable conllevando a una situación favorable, inversiones que conllevan a una utilidad en el corto plazo, etc. (NDB Noticias, 2019)

Bajo este concepto, la entidad financiera BBVA ha desarrollado modelos predictivos contrafactuales, los cuales tienen como objetivo explicar porque el modelo rechaza la solicitud de préstamos de un cliente. En base a un gemelo digital, el cual hereda las características del perfil del cliente (edad, transacciones, etc.) con el fin de hacer variaciones a este setup, encontrar la configuración correcta que le permita obtener el préstamo fruto de reiteradas variaciones. Lo interesante aquí es que se logra explicar las consideraciones que ha tomado el modelo para finalmente aprobar el

préstamo, y sin la intervención del equipo que diseñó la solución, no siendo evidentes para los desarrolladores del modelo. (NDB Noticias, 2019)

2.4 Principales retos del Machine Learning

Existen diversos problemas que los proyectos de machine learning deben de tomar en cuenta para evitar predicciones deficientes. Según Géron (2019), podemos clasificar estos problemas en dos grupos, inconvenientes a nivel de la información y a nivel de los algoritmos:

A nivel de los datos:

- Insuficiente cantidad de información: Los algoritmos de machine learning toman una gran cantidad de información para que trabajen de forma apropiada, de miles de registros para problemas simples, hasta millones de muestras en escenarios complejos. El machine learning convencional necesita de grandes cantidades de información para lograr un entrenamiento que permita una buena predicción.
- Training data no representativa: la data de entrenamiento debe ser representativa para poder generalizar los patrones.
- Pobre calidad de información: Al presentar errores, características faltantes, entre otros, el tratamiento y corrección manual podrán ayudar a corregir la integridad de la información, no obstante contar con una información fidedigna reduce el esfuerzo del tratamiento de corrección y, sobre todo, asegura una mejor predicción.
- Características irrelevantes: Contar con suficientes características relevantes permiten que la predicción se realice de la mejor forma. Se deben realizar técnicas de selección y extracción de características para obtener mejores resultados, pero no contar con características relevantes significantes problemas mayores, que las técnicas anteriormente mencionadas no aplican. (Géron, 2019)

A nivel de los algoritmos:

- Sobreajuste de datos de entrenamiento: el algoritmo que realiza la predicción genera un modelo tan ajustado a los datos de entrenamiento que es imposible generalizar un patrón, cayendo en el sobreajuste en el modelo. Un nuevo dato de entrada, si cuenta con pequeñas diferencias en comparación al modelo, no será considerado dentro de una clasificación.
- Desajuste de datos de entrenamiento: es lo opuesto al sobreajuste. El modelo obtiene patrones generales, permitiendo que cualquier muestra sea considerada dentro de la clasificación, perdiendo el sentido de predicción esperado. (Géron, 2019)

2.5 Contact Center

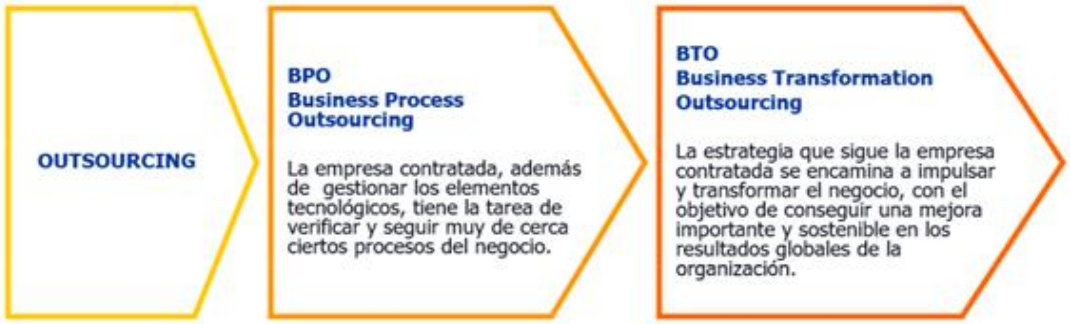
Un call center o contact center ofrece dos tipos de servicios: Servicios Inbound en donde el cliente es quien realiza la llamada para consultar sobre un producto o servicios y Servicios Outbound que es cuando el contact center llama al cliente para vender diferentes productos como; tarjetas, líneas de crédito, etc.

Adicionalmente a ello tenemos el outsourcing que es la modalidad que la mayoría de las empresas ha adoptado ya que según el instituto Internacional de Outsourcing (como se citó en Maximie, 2010, p.8) permite ahorrar hasta el 90% de los costos. Y en base a ello tenemos dos tipos de servicios que las mismas brindan:

a) Business Process Outsourcing (BPO): Uso para tareas específicas y hasta operaciones enteras de procesos de negocio.
b) Business Transformation Outsourcing (BTO): Orientación más al cliente y generación de valor. (Maximixe, 2010)

Figura 2.6

Tipos de outsourcing en un contact center

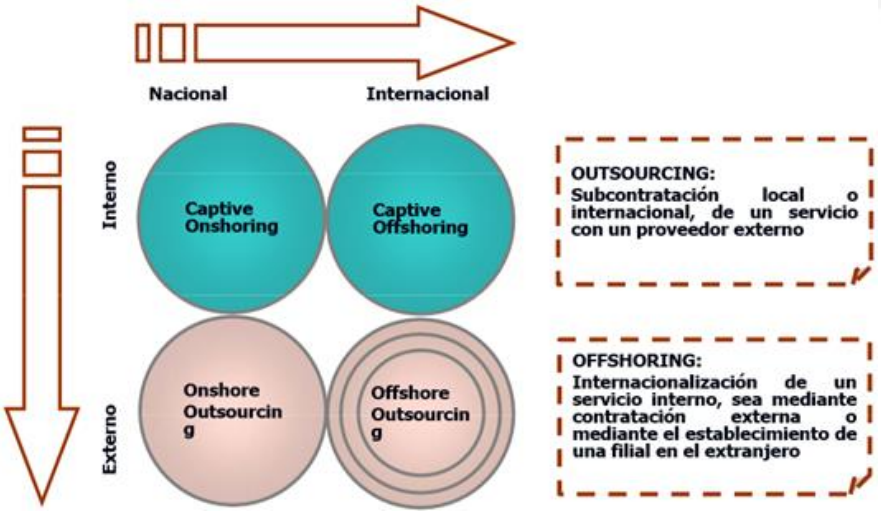


Nota. De "Plan estratégico y operativo del sector contact center en el Perú, 2010", por Maximixe, 2010, Recuperado de (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi5oZ-FiKLqAhUEGLkGHbqkDP8QFjACegQIARAB&url=http%3A%2F%2Fwww.siicex.gob.pe%2Fsiicex%2Fdocumentosportal%2F464948877radF4FC7.pdf&usg=AOvVaw0DDuRaY4cPbXTXdlvY_4Qc)

Adicionalmente hay que considerar el concepto de Offshoring que es contratar a una empresa que se encuentra en el extranjero ya sea para trasladar funciones a esta empresa o por que se piensa colocar una filial en el extranjero. (Maximixe, 2010)

Figura 2.7

Modelo de trabajo de un outsourcing y offshoring



Nota. De "Plan estratégico y operativo del sector contact center en el Perú, 2010, por Maximixe", 2010, Recuperado de (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi5oZ-FiKLqAhUEGLkGHbqkDP8QFjACegQIARAB&url=http%3A%2F%2Fwww.siicex.gob.pe%2Fsiicex%2Fdocumentosportal%2F464948877radF4FC7.pdf&usg=AOvVaw0DDuRaY4cPbXTXdlvY_4Qc)

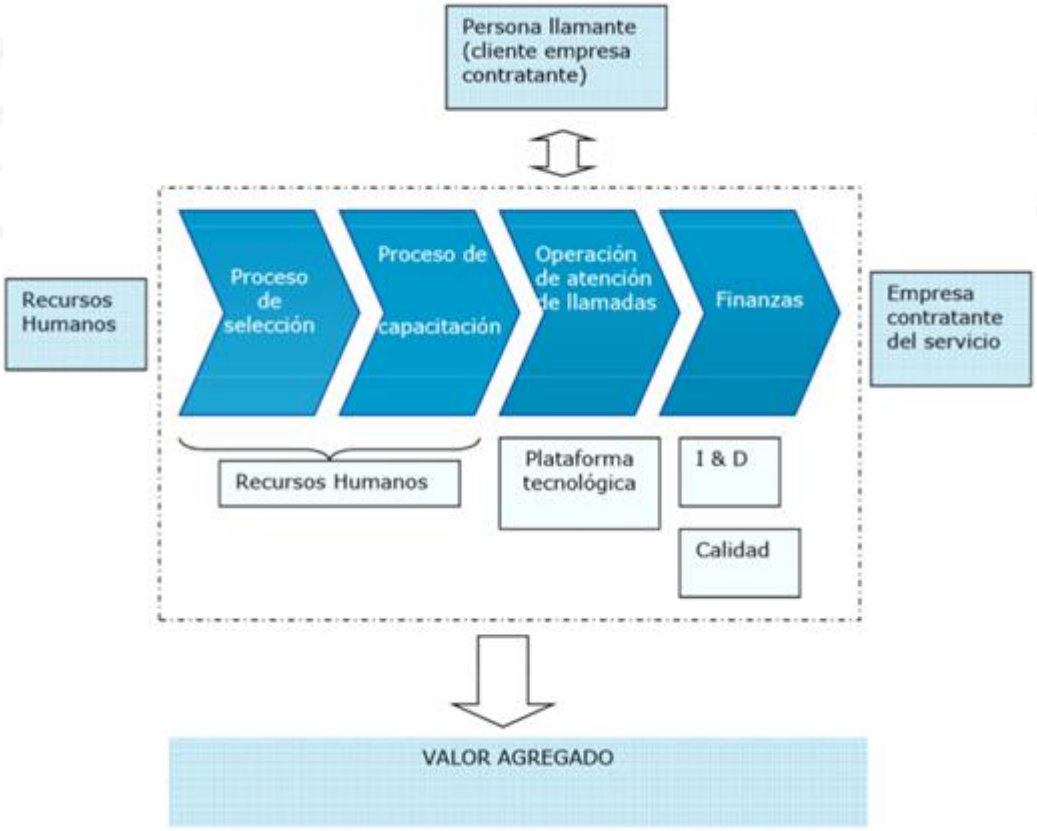
2.5.1 Cadena de Valor de un Contact Center

En esta sección hablaremos de los principales procesos dentro de un Contact Center:

a) Proceso de Selección: Cada empresa utiliza diferentes técnicas como las entrevistas para poder elegir al agente ideal.
b) Proceso de Capacitación: En este proceso se capacita al personal seleccionado para que pueda atender al cliente con calidad.
c) Proceso de Operación de Atención de Llamadas: Aquí es cuando ya se brinda el servicio al cliente y desempeña su labor el agente.
d) Finanzas: Se involucra a esta área cuando se concreta la venta, y se le da aviso para que ejecute la transacción necesaria. (Maximixe, 2010)

Figura 2.8

Cadena de valor de un contact center



Nota. Ejemplo de una cadena de valor en un contact center. De "Plan estratégico y operativo del sector contact center en el Perú, 2010", por Maximixe, 2010, (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi5oZ-FiKLqAhUEGLkGHbqkDP8QFjACegQIARAB&url=http%3A%2F%2Fwww.siicex.gob.pe%2Fsiicex%2Fdocumentosportal%2F464948877radF4FC7.pdf&usg=AOvVaw0DDuRaY4cPbXTXdlvY_4Qc)

Las empresas de Contact Center buscan equilibrar estas variables para tener mayor competitividad frente a los gastos que implica cada proceso. En esta industria la mano de obra es lo más resaltante ya que tiene un impacto social al dar empleo a las personas.

2.5.2 Divisiones dentro de una empresa Contact Center

Figura 2.9

Divisiones en un contact center

Nota. Ejemplo de las diferentes divisiones en un contact center. De "Plan estratégico y operativo del sector contact center en el Perú, 2010", por Maximixe, 2010, (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi5oZ-FiKLqAhUEGLkGHbqkDP8QFjACegQIARAB&url=http%3A%2F%2Fwww.siicex.gob.pe%2Fsiicex%2Fdocumentosportal%2F464948877radF4FC7.pdf&usg=AOvVaw0DDuRaY4cPbXTXdlvY_4Qc)

Por poner un ejemplo adicional en la empresa Atento, se cuenta dentro de la división con un administrador de campaña y un analista de Business Analytics; y cómo los principales clientes: bancos, empresas de telecomunicaciones, autos, cobranza, seguros, entre otros. (S. Herrera, comunicación personal, 10 de junio de 2020).

2.6 Sistema Financiero Peruano

El sistema financiero peruano se compone por un grupo de instituciones enfocados a la transferencia de fondos de ahorristas hacia inversionistas, bajo dos alternativas:

- Utilizando intermediarios financieros indirectos (bancos, financieras, cajas, etc.)
- Utilizando intermediarios financieros directos, con la utilización de bonos, acciones y derivados financieros.

El regulador del mercado de intermediación financiero es la Superintendencia de Banca y Seguros y, quién supervisa a las empresas relacionadas con intermediación financiera indirecta.

Según la SBS (2020), el sistema financiero contempla las siguientes empresas de operaciones múltiples, considerando su participación en % a setiembre 2020, obteniendo una posesión de activos de 565 574 (Monto en S/ Millones).

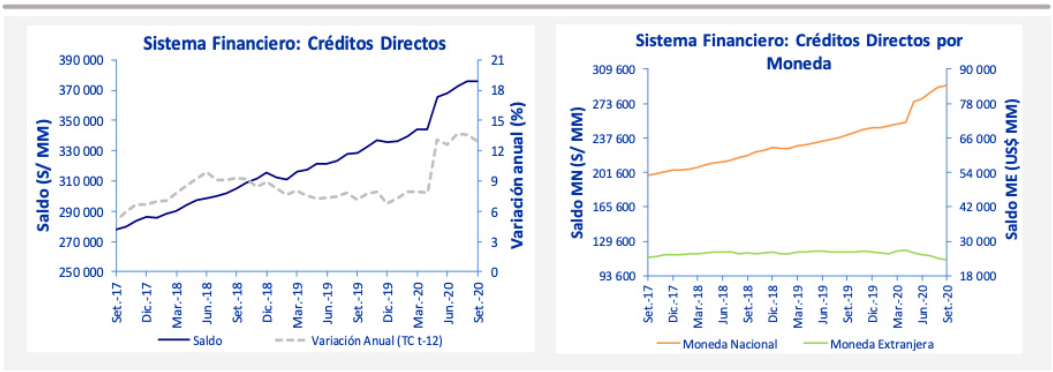
- Banca Multiple: 90.15%
- Empresas Financieras: 2.93%
- Cajas municipales (CM): 5.89%
- Cajas rurales de ahorro y crédito (CRAC): 0.52%
- Entidades de desarrollo de la pequeña y microempresas (Edpyme): 0.52%

Según la SBS (2020), en el mes de setiembre, se tienen 54 empresas que realizan operaciones múltiples y poseen activos por casi S/ 566 mil millones de soles, y confirmando que los bancos son las entidades financieras que más colocan créditos en el país (participación del 90.15%).

Un dato importante es el aumento de los créditos otorgados durante los últimos 12 meses del sistema financiero (a setiembre 2020), con un aumento de 14.4%, y de ellos el 19% representa créditos directo de consumo.

Figura 2.10

Cantidad de Créditos Directos



Tasa de Variación Anual	Set-18 / Set-17	Set-19 / Set-18	Set-20 / Set-19
Créditos Totales (expresado en S/ con TC corriente)	9.6%	7.9%	14.4%
Créditos en MN (expresado en S/)	10.6%	9.7%	21.6%
Créditos en ME (expresado en US$)	6.2%	0.7%	-10.8%
Créditos Totales (expresado en S/ con TC t-12)	9.3%	7.2%	12.9%

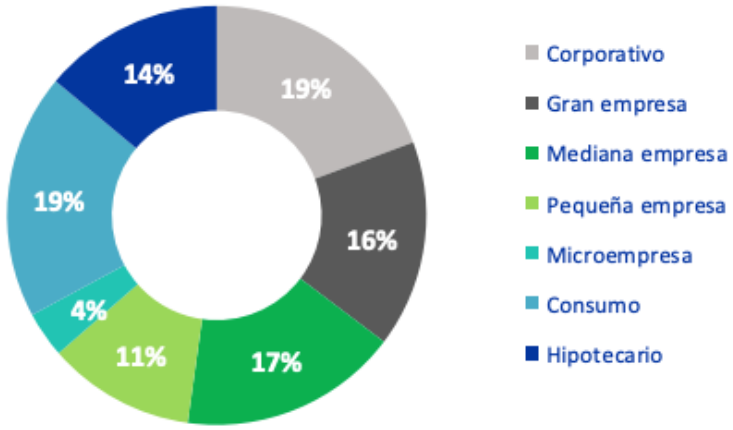
Nota: De "Reportes del Sistema Financiero – Presentación del Sistema Financiero", por Superintendencia de Banca y Seguros, 2020, (https://intranet2.sbs.gob.pe/estadistica/financiera/2020/Octubre/SF-0003-oc2020.PDF)

El sistema financiero tiene la siguiente estructura de créditos directos, y añadiendo el porcentaje de participación obtenidos a setiembre 2020:

- Corporativo: 19%
- Gran empresa: 16%
- Mediana empresa: 17%
- Pequeña empresa: 11%
- Microempresa: 4%
- Consumo: 19%
- Hipotecario: 14% (SBS, 2020)

Figura 2.11

Estructura de créditos directos septiembre 2020



Nota: De "Reportes del Sistema Financiero – Presentación del Sistema Financiero", por Superintendencia de Banca y Seguros, 2020, (https://intranet2.sbs.gob.pe/estadistica/financiera/2020/Octubre/SF-0003-oc2020.PDF)

Dentro de la banca múltiple, la lista de entidades financieras cuenta con 16 bancos: Banco BBVA Perú, Banco de Comercio, Banco de Crédito, Banco Pichincha, Banco de Interamericano de Finanzas, Scotiabank Perú, Citibank, Interbank, Mibanco, Banco GNB, Banco Falabella Perú, Santander Perú S.A., Banco Ripley, Banco Azteca, B. ICBC, Bank of China (SBS, 2020).

Los bancos mencionados comparten los tipos y modalidades de productos financieros. Debido a que en el presente trabajo nos enfocamos en la modalidad crédito de consumo, los créditos tipificados y validados por la SBS son:

- Sobregiros en cuenta corriente.
- Tarjetas de crédito.
- Préstamos Revolventes (crédito que se renueva automáticamente cada vez que se cumple con el pago)
- Préstamos No Revolventes (consume de créditos personales)
- Arrendamiento financiero y Lease-back.
- Pignoraticios.
- Otros créditos de consumo.

CAPÍTULO III: FUNDAMENTACIÓN DEL PROYECTO

3.1 Fundamentación de la deseabilidad del proyecto

Actualmente existen muchas personas que desearían poder contar con un préstamo de libre disponibilidad, crédito vehicular, crédito hipotecario u otro para poder satisfacer sus necesidades inmediatas. La coyuntura que vive el país con el COVID-19 en 2020, es un factor que puede incentivar la búsqueda de seguridad debido a la incertidumbre que atraviesa el país, contando con un soporte económico que permita reactivar negocios, invertir o poder contar con liquidez.

Adicionalmente estas afirmaciones anteriores, se basan en lo siguiente:

> Será fundamental que las empresas comiencen a enfocar sus esfuerzos en establecer nuevos modelos de venta e interacción, pues es un hecho que los consumidores habrán cambiado para siempre tras el fin de la pandemia; empezarán a buscar aquellas marcas que logren dar soluciones y acompañamiento a sus nuevas prioridades, asegura Ana Sordo, Marketing Team Manager en Latinoamérica de Hubspot (como se citó en Aldea Digital, 2020, p.3)

Durante la pandemia las formas y soluciones que adoptaron muchas empresas para poder digitalizar procesos serán un valor diferenciador en escenario Post COVID-19, considerando que el usuario final del mercado también tendrá características diferentes, propias de un aprendizaje en el uso de medios digitales.

Como se estima será un cliente post Covid-19:

- Hábitos cada vez más digitales: Se estima que el 31% de los usuarios seguirán comprando productos o servicios por medios digitales. (Aldea Digital, 2020).
- Más cercano a la marca: Los usuarios buscarán más a las empresas por los medios digitales. Existe un aumento del 51% de las comunicaciones por chat. (Aldea Digital, 2020).

Actualmente las empresas de outsourcing de call centers, brindan sus servicios a diferentes bancos para lograr esta contactabilidad, siendo el mecanismo por el cual se logra la interacción con el cliente final. Estas empresas también desean optimizar sus procesos de captación, reduciendo costos operativos y maximizando sus ingresos, dado que reciben ingresos por lograr concretar la venta del producto financiero por comisiones.

El objetivo del presente trabajo es poder construir un modelo predictivo que permita la concretización de la venta de un producto financiero, permitiéndole a los call centers incrementar sus ingresos por comisión y poder discriminar de forma efectiva la cantidad de leads que proporcionan los bancos.

Para nuestro proyecto estamos contando con tres tipos de clientes:

a.- Usuario Final: Que es la persona que compra el producto financiero, esta persona será entre 30 a 40 años, el cual es un profesional con trabajo estable que se encuentra en el sistema financiero, ser una persona con el deseo de mejorar su situación económica.

b.- Usuario Intermedio: Es el usuario que consumirá nuestro producto, que son los agentes del call center, que tendrán acceso a los dashboards online con la información de los leads de las ventas de los productos financieros.

c.- Usuario Directo: Son las empresas que comprarán nuestro producto directamente, estas empresas son las que brindan servicios de outsourcing de call centers a entidades financieras, quienes serán nuestros clientes directos, y que en base a ellos se desarrollarán los siguientes capítulos, sin dejar de lado los otros dos tipos de usuarios que tenemos.

Nuestra propuesta de valor contemplará:

- Contar con un servicio de valor agregado.
- Ejecución factible/conocida.
- Incremento de ventas de productos financieros.
- Contar con expertos en construcción de modelos predictivos.

Identificamos los siguientes retos:

- Dificultad para tangibilizar los resultados.

- Tiempo de maduración prolongado.
- Sostenibilidad en el tiempo.

Para fundamentar lo antes mencionado se realizaron diferentes entrevistas a 10 personas que se encuentran dentro del rango del usuario final, en lo cual podemos llegar a las siguientes conclusiones de lo que necesitan:

- Tiempo y rapidez en la llamada.
- Conocimiento del cliente, para poder ofrecer el producto que se necesita según sus necesidades
- Ofrecer beneficios adicionales por la compra del producto financiero
- Ofrecer intereses financieros competitivos en comparación con la competencia
- Mejorar los horarios de las llamadas y se realicen según la disponibilidad del cliente.
- Permitir tener una mejor interacción cuando se hacen las llamadas, actualmente no se entiende las explicaciones de las simulaciones de un préstamo cuando se explican por el agente.

Adicionalmente se entrevistó a una encargada de la empresa Atento la cual nos dio los siguientes alcances:

Atento brinda servicios a los bancos más importantes del país actualmente es líder en el negocio como contact center, brinda servicios de cobranza y ventas como los más importantes, en donde cada asesor tiene un supervisor y este un jefe.

La capacitación es un fuerte proceso dentro de la empresa, ya que se busca que el equipo sea altamente capacitado para que pueda responder adecuadamente a los clientes, y poder tener niveles de satisfacción altos.

Existe un protocolo de llamada en donde las pautas las brinda el mismo banco y cada agente debe seguir rigurosamente.

Atento cuenta con el área de Business Analytics que ayuda al agente a poder brindarle información del mejor horario para poder llamar a un cliente.

Y en los últimos meses la cantidad de leads se ha mantenido en el tiempo, el crecimiento no ha sido significativo, y con la coyuntura de covid-19 se ha disminuido la capacidad de atención.

3.2 **Fundamentación de la factibilidad del proyecto**

Utilizando inteligencia artificial, específicamente en la rama del machine learning, soportadas en herramientas especializadas en predicción, matemática y estadística es posible construir modelos que permitan discriminar (o en todo caso, llegar a establecer un porcentaje de éxito) si un posible cliente terminará por aceptar o desestimar la compra del producto.

Nuestro proyecto estuvo enfocado en utilizar la herramienta Anaconda (Jupiter/Spyder) basada en Python, la cual cuenta con las librerías que permiten un correcto análisis predictivo, para poder lograr diseñar y construir la solución de aprendizaje automatizado.

Se implementó la solución en base al análisis realizado, al pre-procesamiento de datos, y al algoritmo que mejor se ajuste a la distribución de los datos.

Un punto importante para lograr lo anteriormente mencionado, es contar con una data histórica que será la base de este análisis. Nos enfocamos en la información que nos proporcionó una empresa de outsourcing que le brinda servicios de Call Center a entidades financieras. Esta información se remonta al año 2016 en adelante, confirmando de esta manera que existe información histórica suficiente para ejecutar el análisis.

Nuestra propuesta comtempló los siguientes puntos:

- Consultoría de servicios de Machine Learning
- Soporte online permanente
- Implementación de soluciones de Machine Learning
- Documentación del servicio

Esto permitirá a nuestros usuarios:

- Incrementar ingresos
- Agentes con mejores comisiones

- Excelencia operativa
- Acompañamiento permanente ante dudas o problemas.

Y esto se logró:

- Con dashboard visuales
- Uso de tecnología emergente
- Servicio con experiencia

Citando el servicio de experiencia del cliente, queremos citar lo siguiente:

"La Experiencia del Cliente es un concepto más amplio que, si bien involucra a la Calidad, deberíamos pensarlo como un proceso ya que es el recorrido de las vivencias del cliente con el producto o servicio" (como se citó en Basile, 2020, párr.6).

"De este modo, podemos deducir que la CX incluye expectativas, emociones, y pretende ir más allá de la apreciación de la calidad de lo que se vende. Aunque es implícito que la misma debe ser la mejor". (Basile, 2020, párr.7)

"Hoy la tecnología nos ayuda a conocer las preferencias de cada cliente y consolidar los datos en información para establecer patrones de preferencias. Una organización centrada en el cliente debe comprender esta información y tomar acciones en función de esta" (como se citó en Basile, 2020, párr.10).

En síntesis, queda evidenciado que la calidad y la experiencia del cliente no son sinónimos. Lo cierto, es que se interrelacionan y sus resultados impactan entre sí, lo que hace sumamente necesario entender ambos procesos para mejorar la gestión y alcanzar la satisfacción de los clientes. (Basile Elsa, 2020, párr.12).

3.3 Beneficios esperados

El proyecto de aprendizaje automatizado al proceso de venta de productos financieros propuso la siguiente estructura de ingresos y costos.

3.3.1 Ingresos:

La oferta del servicio contempló dos modalidades:

a.- Por proyecto de implementación, llave en mano:

Consiste en la provisión del servicio por un tiempo determinado e implementando la metodología de minería de datos CRISP-DM. Este tiempo será establecido por los siguientes factores:

- Cantidad de fuentes de datos
- Información del área funcional (# agentes, flujos)
- Arquitectura Actual (infraestructura base que se utiliza actualmente)
- Datos de Operación (procesos de trabajo, AS IS del negocio)

Dependiendo del caso de negocio, la data a analizar y diferentes factores se estimaron los sprints necesarios de implementación, esto quiere decir que cada rol determinó la cantidad de horas necesarias a desarrollar, por ejemplo:

- Analista de Datos: Determinará la cantidad de horas que necesitará para normalizar la data (Etls. Jobs, etc)
- Científico de Datos: Modelo de Python a utilizar en base a funciones.

La tarifa para el proyecto será la siguiente (costo de servicios por hora).

Tabla 3.1

Tarifa de Recursos

Recurso	**Tarifa por hora (S/)**
Líder/Arquitecto de Proyecto	110.00
Científico de Datos	120.00
Analista de datos	100.00
FrontEnd	120.00
BackEnd	120.00

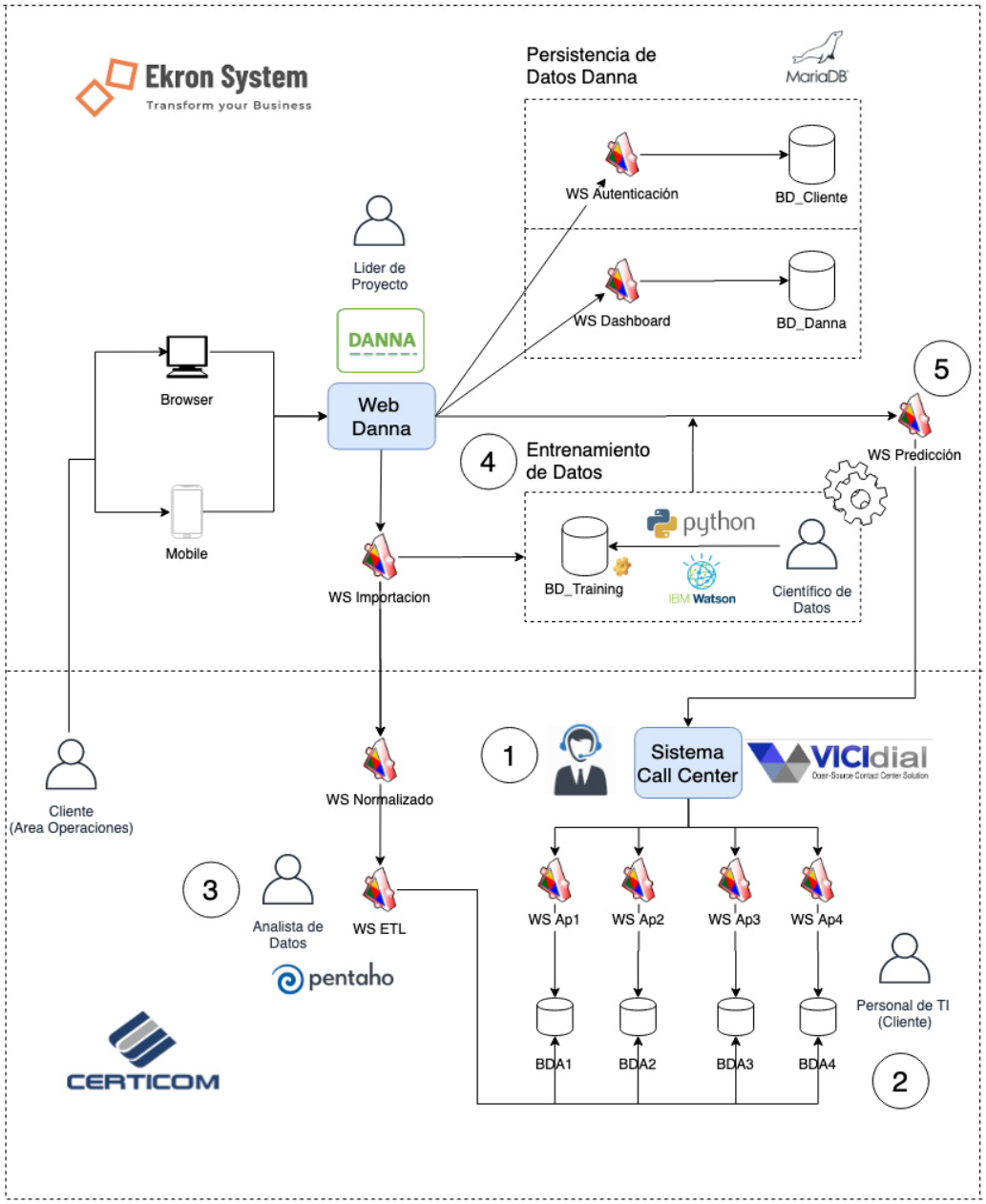
Nota. Datos referenciados en empresas de servicios de desarrollo de software (Everis, Globant, GFI, entre otros).

Adicionalmente el cliente podrá elegir si desplegar este modelo sobre su arquitectura o contratará con nosotros un servicio en Nube IBM.

En caso el cliente requiera un servicio en nube se utilizará como referencia la siguiente arquitectura:

Figura 3.1

Arquitectura de modelo de machine learning



Nota. Los números presentados representan la secuencia del flujo.

Consideraciones:

- Costos sin I.G.V.

- El día laborable cuenta con 8 horas.
- El mes laborable cuenta con 22 días.
- La función por desarrollar no será modificada en el tiempo.
- El modelo de predicción de machine learning no se actualiza.

b.- Por producto vendido:

Consiste en una comisión por cada producto vendido y que haya sido ejecutado por el modelo. Este modelo deberá contemplar un contrato fijo.

Procederemos a explicar cómo actualmente las empresas de outsourcing comisionan según cantidad vendida o monto colocado.

Tabla 3.2

Comisiones por tipo de modelo

Modelo Pyme x Comisión	
Importe Desembolsado (S/)	**Comisión**
Menor a 10000	S/.100 soles por cantidad
Entre 10000 a 30000	1.75% x monto
Entre 30001 a 50000	1.25% x monto + S/.150 soles x cantidad
Entre 50001 a 100000	S/1000 x cantidad
Mayor a 100000	S/1200 x cantidad

Modelo de Préstamo de Libre Disponibilidad x Comisión	
Importe Desembolsado (S/)	**Comisión**
Menor a 10000	S/.150 soles por cantidad
Entre 10000 a 20000	S/.220 soles por cantidad
Entre 20001 a 30000	S/.260 soles por cantidad
Entre 30001 a 125000	0.8% x monto
Mayor a 125000	S/1000 soles x cantidad

Como se puede observar existen diferentes tipos de comisión que se puede utilizar, para nuestro proyecto utilizamos la comisión por cantidad de producto colocado, y para hacer una simulación se muestra lo siguiente:

Tabla 3.3

Proyecciones de Ingresos en los siguientes 12 meses

Ingresos por comisión de venta

Primeros 3 meses

Tipo de Préstamo	Cantidad Actual	Crecimiento de 5%	Cantidad Final	Comisión Ekron	Monto de Ganancia
Pyme	46	2	48	S/.100.00	S/.200.00
PLD	136	7	143	S/.80.00	S/.560.00
				Total Ingreso	S/.760.00

Entre 3 y 6 meses

Tipo de Préstamo	Cantidad Actual	Crecimiento de 15%	Cantidad Final	Comisión Ekron	Monto de Ganancia
Pyme	46	7	53	S/.100.00	S/.690.00
PLD	136	20	156	S/.80.00	S/.1,632.00
				Total Ingreso	S/.2,322.00

A partir del 6 mes

Tipo de Préstamo	Cantidad Actual	Crecimiento de 25%	Cantidad Final	Comisión Ekron	Monto de Ganancia
Pyme	46	12	58	S/.100.00	S/.1,150.00
PLD	136	34	170	S/.80.00	S/.2,720.00
				Total Ingreso	S/.3,870.00

Nota. (Pyme: Préstamo brindado a pequeñas y medianas empresas, PLD: Préstamo de libre disponibilidad (préstamos de consumo).

En este ejemplo se puede observar que ante un crecimiento en las ventas de un 5% con nuestro modelo de machine learning, se muestran las ganancias por comisión de venta. La tarifa por producto vendido se clasificó según el tipo de producto:

- Tarifa por producto vendido será clasificado según el tipo de producto:
- Pymes: Comisión de S/100.00 soles por producto vendido
- PLD: Comisión de S/80.00 soles por producto vendido

Adicionalmente se consideró un costo fijo dentro del contrato por S/.2880 soles trimestralmente para el mantenimiento del modelo (periodo de gracia en los tres primeros meses).

c.- Mantenimiento del Servicio

- Consiste en ajustar el modelo cada cierto tiempo, corriendo nuevas predicciones y buscando mejorar el modelo con más información y variables.
- Este tipo de servicio se podrá adquirir las veces que el cliente (empresa de outsourcing) lo necesite para ir mejorando el modelo, teniendo un costo de

S/.3000 soles. Este servicio lo trabajarán los diferentes especialistas que contamos como científico de Datos, analista de datos, entre otros.

Tabla 3.4

Ingresos por implementación y mantenimiento trimestral

Proyección de Ganancias			
Ingreso por Implementación	Tarifa por hora	Horas	Precio
Sueldo de Científico de Datos	S/.120.00	30	S/.3,600.00
Sueldo de Analista de Datos	S/.100.00	30	S/.3,000.00
Sueldo de FrontEnd	S/.120.00	5	S/.600.00
Sueldo de BackEnd	S/.120.00	5	S/.600.00
Sueldo de Lider Project Manager/ Arquitecto / Scrum Master	S/.110.00	30	S/.3,300.00
		Total Ingreso	S/.11,100.00

3.3.2 Egresos:

Los costos a considerar se muestran en la siguiente tabla:

Tabla 3.5

Lista de egresos para el proyecto

EGRESOS				
Sueldos	**Costo Fijo**	**Inversión Mensual**	**Inversión Anual**	**Inversión en los primeros meses**
Sueldo de Gerente TI	S/.6,500.00	S/.6,500.00	S/.78,000.00	
Sueldo de Científico de Datos	S/.4,000.00	S/.4,000.00	S/.48,000.00	S/.800.00
Sueldo de Analista de Datos	S/.4,000.00	S/.4,000.00	S/.48,000.00	S/.800.00
Sueldo de FrontEnd	S/.3,500.00	S/.3,500.00	S/.42,000.00	S/.800.00
Sueldo de BackEnd	S/.3,500.00	S/.3,500.00	S/.42,000.00	S/.800.00
Sueldo de Contador	S/.2,000.00	S/.2,000.00	S/.24,000.00	S/.400.00
Sueldo de Vendedor	S/.4,000.00	S/.4,000.00	S/.48,000.00	S/.400.00
Sueldo de Project Manager/Scrum Master	S/.3,500.00	S/.3,500.00	S/.42,000.00	S/.500.00
Gastos Comerciales				
Publicidad	S/.2,000.00	S/.2,000.00	S/.24,000.00	S/.200.00
Merchandising	S/.1,500.00	S/.1,500.00	S/.18,000.00	
Alquiler Oficina	S/.1,500.00	S/.1,500.00	S/.18,000.00	
Mantenimiento Oficina	S/.500.00	S/.500.00	S/.6,000.00	
Servicio de Luz	S/.350.00	S/.350.00	S/.4,200.00	
Útiles de Oficina	S/.300.00	S/.300.00	S/.3,600.00	
Servicio Cloud				
Aplicación Frontend y Backend				S/.800.00
Servicio ML				S/.800.00
Activos				
Equipos Tecnológicos	S/.8,000.00			
Muebles Oficina	S/.6,000.00			
Pasivos				
Pago de Banco	S/.1,500.00	S/.1,500.00	S/.18,000.00	
Total de Egresos		**S/.38,650.00**	**S/.463,800.00**	**S/.6,300.00**

Tener en consideración que hay roles que pueden ser ejecutados por una sola persona, por ejemplo, el científico de datos y analista de datos. De la misma forma, el rol de backend y frontend como los roles de vendedor y project manager.

Por lo tanto, los montos mensuales en los primeros meses serían los siguientes:

- Científico y analista de datos monto de S/.1600.00
- Backend y Frontend monto de S/.1600.00
- Project Manager y vendedor de S/.900.00

Adicionalmente, los montos de inversión mensual y los montos de los primeros meses distan porque se está considerando una porción de tiempo del recurso, ya que solo se trabajará con un proyecto en el primer año. En el futuro al contemplar más clientes se considerará el monto fijo de pago a los recursos.

3.3.3 Flujo de Caja:

Tabla 3.6

Flujo de caja mensual y a tres años

Meses	0	1	2	3	4	5	6	7	8	9	10	11	12	Total
Conceptos														
Ingresos Implementación y Mantenimiento				S/.11,100.00	S/.11,100.00		S/.11,100.00			S/.11,100.00			S/.11,100.00	**S/.55,500.00**
Ingresos Variables		S/.760.00	S/.760.00	S/.760.00	S/.2,322.00	S/.2,322.00	S/.2,322.00	S/.3,870.00	S/.3,870.00	S/.3,870.00	S/.3,870.00	S/.3,870.00	S/.3,870.00	**S/.32,466.00**
Egresos		S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	S/.6,300.00	**S/.75,600.00**
Capital	S/.2,500.00													
Utilidad Neta	**-S/.2,500.00**	**-S/.5,540.00**	**-S/.5,540.00**	**S/.5,560.00**	**S/.7,122.00**	**-S/.3,978.00**	**S/.7,122.00**	**- S/.2,430.00**	**-S/.2,430.00**	**S/.8,670.00**	**-S/.2,430.00**	**-S/.2,430.00**	**S/.8,670.00**	**S/.12,366.00**
COK	10%													
VAN	S/.747.31													
TIR	11%													

Año	0	1	2	3	Total
Conceptos					
Ingresos		**S/.87,966.00**	S/.92,364.30	S/.96,982.52	S/.277,312.82
Egresos		**S/.75,600.00**	S/.77,868.00	S/.80,204.04	S/.233,672.04
Capital	S/.2,500.00				
Utilidad Neta	**-S/.2,500.00**	**S/.12,366.00**	**S/.14,496.30**	**S/.16,778.48**	S/.43,640.78
COK	10%				
VAN	S/.33,328.15				
TIR	508%				

Nota. Considerar que el ingreso en el mes cuarto corresponde al pago de la implementación inicial, que ha sido acordado con el cliente (Call Center Certicom).

CAPÍTULO IV: DEFINICIÓN DEL PROYECTO

4.1 Definición del proyecto

El proyecto estuvo enfocado en el diseño y la construcción de una inteligencia artificial basada en machine learning, con el fin de poder predecir el comportamiento de compra de posibles clientes, los cuales tengan gran probabilidad de adquisición de un producto financiero en un momento determinado. Los beneficiarios directos serán las empresas que brindan servicios de contactabilidad de clientes del banco (empresa BPO outsourcing).

El proyecto contempló la implementación de machine learning para poder determinar a los grupos de clientes que tienen alta probabilidad de adquirir un producto financiero, implicando la evaluación de una data histórica (información de 6 meses de antigüedad). La data histórica fue evaluada en 5 fases (explicados en el capítulo V) y se utilizaron diferentes herramientas para lograr construir un modelo confiable. Finalmente se construyó una aplicación web para ingresar datos al modelo y un dashboard interactivo para mostrar los resultados.

4.1.1 Servicio de venta por Telemarketing

El servicio de telemarketing consiste en la gestión de venta mediante promotores, utilizando el canal telefónico, para lograr la contactabilidad, colocación y venta de productos financieros (préstamos de libre disponibilidad, tarjetas de crédito entre otros), cumpliendo con los protocolos proporcionados por la entidad financiera. La contactabilidad puede realizarse a clientes que solo se encuentren en la base de datos proporcionada por los bancos.

El proveedor (BPO) utiliza sus propias instalaciones para realizar el servicio contratado, utilizando sus propios medios financieros y técnicos, inclusive debe de provisionar el recurso humano (promotores, personal de backoffice, etc). A su vez, el proveedor deberá contar con los elementos de hardware, software y comunicación necesarios, y ser compatible con la infraestructura tecnológica de la entidad financiera.

Este servicio debe garantizar el cumplimiento de los acuerdos de nivel de servicio establecidos entre la entidad financiera y el proveedor.

Para poder brindar el servicio de telemarketing, el banco provee la información e inducción necesaria al proveedor para poder comercializar el producto, tales documentos informativos como las especificaciones de los productos materia de contratación, asesorías, etc.

El proveedor es el responsable de las gestiones realizadas para la capacitación de clientes a través del otorgamiento de créditos, así como la veracidad de la documentación entregada para su respectiva evaluación, a fin de garantizar los siguientes puntos:

- El canal de procedencia o Fuerza de Ventas Externa (FUVEX)
- El promotor que gestiona la operación.
- Validar que la información entregada esté completa para su evaluación.
- La documentación aportada se encuentra debidamente actualizada.
- La información que se reciba pueda ser verificada y contrastada.
- La validación de la identidad de los clientes.

Esta información es de suma importancia para que el banco pueda validar la gestión de la operación y por ende validar el pago por comisión por ventas.

4.1.2 Indicadores de seguimiento de base de datos:

Los indicadores que el proveedor brindará de forma mensual al banco serán el embudo de ventas (resumen semanal de ventas logradas por el call center – cantidad) y mejora de la calidad de base de datos denominada Feed Back (calidad de los datos para poder identificar que los mismos sean reales y actualizados). Según lo indicado con los indicadores de seguimiento de base de datos, concluimos como una necesidad real y una oportunidad de mejora incorporar un modelo predictivo para la provisión del servicio, logrando mejorar la calidad de esta información y la probabilidad de aumentar las ventas.

4.1.3 Flujo de provisión del Servicio:

El detalle y el flujo actual del servicio es el siguiente:

1.- Las entidades bancarias subcontratan el servicio de venta vía telefónica de productos financieros de una base de datos de clientes previamente analizada y preparada. Este proceso de análisis y filtrado que realiza la entidad financiera es una caja negra y se considera como información confidencial.

2.- Esta base de datos es asignada a las diversas empresas que les brindan el servicio de call center (empresa BPO outsourcing) de forma distribuida, de tal forma que cada empresa de outsourcing de forma mensual cuenta con una base de datos única. La base de datos contiene información del cliente de la entidad financiera, tales como, documento de identidad, nombres, apellidos, edad, sexo, domicilio, monto de préstamo pre aprobado, tipo de producto, entre otros. A su vez el call center realiza un pre filtrado y segmentación de campañas.

3.- Con la base de datos asignada, deberá ser trabajada y gestionada por la fuerza de ventas del outsourcing, para brindar asignación de la contactabilidad mediante un software de call center que permita la gestión de operación, lograr la contactabilidad vía telefónica con el cliente o prospecto y realizar la gestión de venta con el fin de colocarla y confirmarla.

4.- En el caso de confirmar la venta, existen procesos de backoffice según el producto vendido. Por ejemplo, el proveedor deberá confirmar la calidad de la documentación que se adjunta a cada solicitud de préstamos y/u otro, auditando la veracidad de las copias proporcionadas por los clientes.

- Validando con SUNAT, para validar la existencia del centro laboral consignado en la solicitud.
- Con ESSALUD, para verificar la dependencia laboral del cliente solicitante.
- Con los filtros internos, para verificar sobre endeudamiento y otra información que el proveedor considere necesario.
- Con RENIEC, para validar el dígito de control impreso en el DNI presentado por el potencial cliente.
- Y otros que el Banco estime necesario.

5.- Esta base de datos considera préstamos preaprobados, los cuales deberán pasar por una contactabilidad por la parte de la empresa de outsourcing, detallar el producto, ofrecerlo, y en el mejor de los casos, cerrar con la confirmación del cliente.

6.- En cuanto al tratamiento de los datos, el proveedor se hace responsable de la recopilación, registro, organización, conservación, elaboración, modificación, almacenamiento, extracción, consulta, bloqueo, supresión, transferencia terceros y/o fuera del territorio nacional y uso de los datos,

ante el titular de estos, la autoridad nacional de protección de datos y cualquier otra autoridad competente.

4.2 Objetivos del proyecto

4.2.1 Objetivo general

Desarrollar un modelo predictivo basado en machine learning que permita evaluar diferentes modelos y poder optimizar el servicio, con ello la utilidad del servicio de outsourcing (comisión), identificando clientes con mayor probabilidad de compra, prediciendo un comportamiento acorde con un potencial cliente, en base a los datos provenientes del banco, así como de fuentes de datos adicionales.

La base de datos que se utilizará en nuestro proyecto es información proporcionada por los bancos que es entregada a las empresas que brindan outsourcing de call center, esta información se encuentra estructurada de forma totalizada (ni ordenada, ni priorizada), permitiendo lograr filtrar al grupo con mayor opción de captación y enfocar esfuerzos en ese segmento, a través de un entrenamiento previo con la base de datos histórica de prospectos que finalmente se convertirán en clientes. A su vez, se cuenta con unas bases de datos adicionales, las cuales, al ser tratadas de forma correcta, significan un input adicional para realizar un correcto análisis de predicción.

En base a la expuesto, el objetivo será construir el modelo predictivo basándose en las bases de datos antes mencionadas, que permitirá contar con un análisis de XIA, y así deducir la lógica algorítmica del modelo de machine learning, con el fin de descubrir insights de variantes que permitan mejorar el modelo y con ello en el futuro elevar las ventas.

4.2.2 Objetivos específicos

- Lograr excelencia operativa, optimizando recursos humanos correspondientes a ejecutivos de ventas. Considerando que este filtrado de posibles clientes estaría enfocado en un segmento preferente, la cantidad de ejecutivos de venta también pueden ser reducidos en un 10% (actualmente se cuentan con 30 agentes), ya que no todos los registros de la base de datos original se consideran. A su vez, el tiempo que conlleva la gestión de desembolsos de efectivo se optimizaría (actualmente el tiempo de desembolso es de 48 horas y podrá reducirse a 36 horas), al contar con el

personal BackOffice mínimo indispensable. Al contar con una priorización de los leads, la gestión de contactabilidad reduce los tiempos de barrido (actualmente el barrido de toda la base de datos se realiza en dos semanas y este se reducirá en 30%) de las bases de datos mensuales, optimizando las horas hombre dedicadas a la contactabilidad y al proceso de venta.

- Mejora de la Experiencia del Usuario: La atención a los clientes con mayor probabilidad de compra también permite no llamar a personas que no se encuentran interesadas a priori y por ende no darles prioridad, la cantidad de personas a las que no se le darán prioridad son un 60% aproximadamente (cálculo mensual). Con ello se logra una mejor experiencia con el cliente, considerando que a los que no aceptan un producto financiero no tendrán malas experiencias con la llamada que se les realizaría, eliminando el proceso de contactabilidad en el mejor de los casos. En base a un análisis del comportamiento de prospectos y clientes, es posible descubrir insights que permitan determinar qué factores podrían ser importantes, desde una base de datos desestimada a una lista con mucha mayor probabilidad de venta.

4.3 Beneficios esperados

Nuestra empresa consultora e implementadora de soluciones de TI, EKRON SYSTEM S.A.C., brindará el servicio de implementación del proyecto de machine learning para una de las empresas que brinda el servicio al banco BBVA, siendo una empresa de outsourcing que brinda actualmente el servicio, llamada CERTICOM S.A.C.

El beneficio esperado para EKRON SYSTEM S.A.C. es obtener beneficios económicos por provisión de la implementación de la solución predictiva, cuyo principal objetivo es incrementar las ventas de productos financieros y con ello elevar la utilidad de nuestro cliente directo bajo la modalidad de comisión.

Para lograr estos beneficios, los entregables del servicio de implementación son los siguientes:

4.3.1 Servicio ML predictivo de ventas

Contar con un servicio (desplegado on cloud/onpremise) de uso automático y de interoperabilidad con el sistema web del call center del cliente, que permita de manera

online indicar la predicción de venta de un lead determinado. De esta forma toda la base de datos del banco podrá ser diferenciada con data de mayor calidad, para una mejor toma de decisiones:

- Se podrá priorizar esta información con el fin de asignar la contactabilidad a un promotor correspondiente, inclusive se podrá asignar según el expertise de este último.
- El servicio podrá ser utilizado según la necesidad, siendo de activación y desactivación automática.
- El modelo de machine learning deberá ser ajustado en intervalos de tiempo definidos, con el fin de tener información reciente y evaluar nuevas tendencias.
- El modelo permitirá realizar un análisis XAI, con el fin de evaluar iteraciones de cambio de variables que permitan identificar estas variaciones y pasar de un estado desestimado a uno de potencial cliente.

4.3.2 Acceso a la aplicación web y consulta de resultados

El segmento de mercado a la cual estamos dirigiendo el servicio de aprendizaje automatizado del proceso de venta de productos financieros, se enfoca en las empresas que brindan servicios de outsourcing a entidades financieras.

El dashboard online podrá ser utilizado por el área operativa, y dar seguimiento en línea de la operación con la inyección del servicio de machine learning predictivo de ventas, incluyendo reportería, identificación de volúmenes de venta, indicadores de productividad por campaña, monto en soles de préstamos, entre otros. La aplicación se conectará con la base de datos dedicada para dicho fin, mostrada en el gráfico de arquitectura de la solución.

4.4 Segmento de Mercado

El segmento de mercado a la cual estamos dirigiendo el servicio de aprendizaje automatizado del proceso de venta de productos financieros, se enfoca en las empresas que brindan servicios de outsourcing a entidades financieras bajo la modalidad servicio de venta por Telemarketing.

Según Guy Fort, presidente de la Apexo (Asociación Peruana de experiencia al cliente), el sector de contact center mueve alrededor de US$ 500 millones al año y crece entre 10% y 12%. (Management & Empleo, 2018)

Existen diversas empresas que cuentan con experiencia brindando esta clase de servicios al sector banca, como también a sectores de retail, telecomunicaciones, entre otros.

Entre las FUVEX que brindan servicios de venta por Telemarketing a diferentes entidades bancarias tenemos:

- ✓ ATENTO
- ✓ CERTICOM
- ✓ COALITION
- ✓ MASTER CENTER
- ✓ TCONTAKTO
- ✓ KONECTA

Para el presente proyecto de investigación, estamos tomando como caso de estudio la empresa CERTICOM S.A.C., la cual brinda en la actualidad servicios de outsourcing de call center en entidades financieras peruanas.

4.5 Roles y responsabilidades del equipo del proyecto

Los principales roles que tendremos en nuestro proyecto serán los siguientes:

Los principales roles del proyecto de implementación son:

1) **Líder/Arquitecto de Proyecto:** Se requiere un perfil mixto de líder y arquitecto del proyecto, siendo responsable de:

- Controlar el proyecto en alcance, tiempo y costo.
- Identificar los riesgos del proyecto.
- Elaborar el diseño de arquitectura para el modelo de machine learning.
- Aprobar el deploy de la arquitectura en nube.
- Comunicar en todo momento a los encargados de Ekron System el estado actual del proyecto.
- Acompañar y dar soporte técnico/funcional a Ekron System en las

reuniones con posibles clientes.

- Velar por el cumplimiento del compromiso de los recursos asignados.

2) **Científico de Datos:** Se considera contar con un científico de datos, siendo responsable de:

- Analizar, normalizar, tratar múltiples fuentes de datos (BD, archivos planos, excel, etc.).
- Limpiar los datos utilizando técnicas basadas en Python.
- Procesar los datos utilizando métodos estadísticos, utilizando librerías especializadas de Python.
- Diseñar, codificar, desplegar y analizar modelos predictivos basados en Python.

3) **Analista de Datos:** Para el proyecto se considera un analista de datos que será responsable de:

- Extraer, exportar y procesar los datos.
- Analizar agrupaciones de los datos.
- Interpretar los datos.
- Generar informes en base a lo encontrado.
- Resolución de problemas técnicos o de programación.

4) **BackEnd:** Se considerará un perfil backend para el proyecto y será responsable de las siguientes actividades:

- Realizar la programación del modelo orientado en arquitecturas de microservicios.
- Administrar repositorios de GIT.
- Garantizar y aplicar mejoras prácticas en desarrollo de software, programación y funciones.
- Identificar problemas propios del framework a utilizar, brindar soluciones y generar soluciones en etapas tempranas.

5) **FrontEnd:** Se considera un perfil frontend para el proyecto y dentro de sus responsabilidades están:

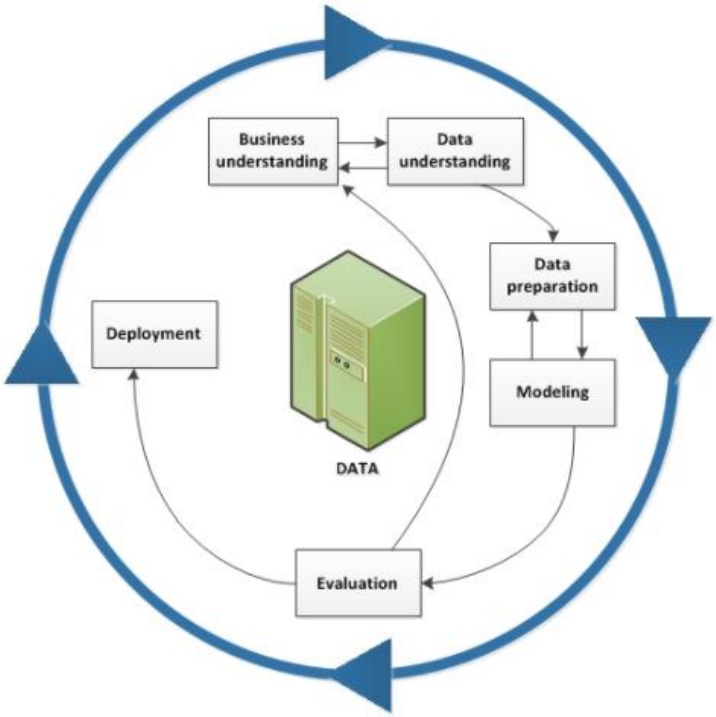
- Manejar APIs y consumo de servicios REST.
- Desarrollar componentes front end utilizando código angular JS.
- Llevar el control de plantillas, formas de diseño, tipografías.
- Realizar los dashboard de visualización de la información y posibles clientes que tendrá acceso el agente de call center.
- Llevar control de todas las APIs necesarias de integración web.

4.6 Cronograma y riesgos iniciales del proyecto

Para la ejecución del proeycto se utilizó la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) el cual es utilizado en el desarrollo de proyectos de minería de datos. CRISP-DM incluye fases propias de un proyecto, tareas para cada fase y una explicación de las relaciones de las tareas. El modelo de procesos se basa en el ciclo vital de minería de datos.

Figura 4.1

Ciclo de vida de minería de datos



Nota. De "Conceptos básicos de ayuda de CRISP-DM", IBM, 2019, (https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html)

a) Fase de Comprensión de la Necesidad: Se evaluó a los posibles beneficiarios, características, insights y problemática actual. Como resultado de esta fase obtuvimos la lista de los posibles clientes.

b) Fase de Comprensión de los Datos: Esta fase comprende obtener la data inicial. En esta fase se realizará:

- Recolección de datos iniciales
- Descripción de los datos
- Exploración de los datos
- Verificación de la calidad de los datos

c) Fase de Preparación de los Datos: La fase de preparación de datos incluye las tareas de selección y técnicas de limpieza de datos, generación de variables adicionales, integración de diferentes orígenes de datos y cambios de formato.

Esta fase se encuentra relacionada con la fase de modelado, ya que los datos serán procesados de diferente forma en función a la técnica de modelado elegida. Las principales tareas en esta fase son:

- Estructuración de los datos.
- Integración de los datos.
- Formateo de los datos

d) Fase de Modelado: En esta fase se seleccionaron las técnicas de modelado para nuestro proyecto. Considerar que previamente al modelado de los datos, se debe determinar un método de evaluación de los modelos que pueda permitir el grado de bondad de ellos. Dentro de las principales tareas de esta fase tenemos:

- Selección de la técnica de modelado.
- Generación del plan de prueba.
- Construcción del modelo.
- Evaluación del modelo.

e) Fase de Evaluación: En esta fase se evaluó el modelo, teniendo en cuenta el cumplimiento de los criterios de éxito del problema. Debe considerarse, además, que la fiabilidad calculada para el modelo se aplica únicamente para los datos sobre los que se realizó el análisis (data histórica de 6 meses). Las tareas involucradas en esta fase del proceso son las siguientes:

- Evaluación de los resultados.
- Proceso de revisión.
- Determinación de futuras fases.

f) Fase Comercial: Esta fase comprenderá la elaboración de la propuesta comercial, la presentación, la elaboración y firma de contrato con nuestro primer cliente.

A continuación, se presenta nuestro cronograma según las diferentes fases y para los siguientes 10 meses.

Tabla 4.1

Cronograma de Proyecto de Modelado de Machine Learning

CRONOGRAMA DE PROYECTO DANNA ML001									
FASES	1	2	3	4	6	7	8	9	10
Fase de Comprensión de la Necesidad									
Evaluación de posibles clientes y retos									
Entrevistas con los encargados de Business Analytics									
Revisión de propuesta de valor									
Listar posibles clientes									
Realizar las contrataciones de los recursos requeridos									
Explicación de los objetivos al equipo de trabajo									
Fase de Comprensión de los Datos									
Recolección de datos iniciales									
Descripción de los datos, documentación									
Exploración de los datos									
Verificación de la calidad de los datos									
Guardar información en base de datos									
Fase de Preparación de los datos									
Estructuración de los datos - Normalización									
Integración de los datos									
Formateo de los datos									
Preparación de versión final de base de datos									
Fase de Modelado									
Seleccionar técnica de Modelado									
Generación de plan de prueba del modelo									
Construccción de Modelo									
Evaluación del Modelo									
Documentación de los datos encontrados									
Fase de Evaluación									
Evaluación de los resultados									
Proceso de revisión de los resultados									
Determinación de futuras fases									
Documentación de los hallazgos encontrados									

Nota. El cronograma se presenta de manera mensual.

4.7 Medidas de control (indicadores)

Para la fase de construcción e implementación del proyecto, se considerará a Scrum Framework como guía del desarrollo del proyecto, tomando a los siguientes eventos como referencia:

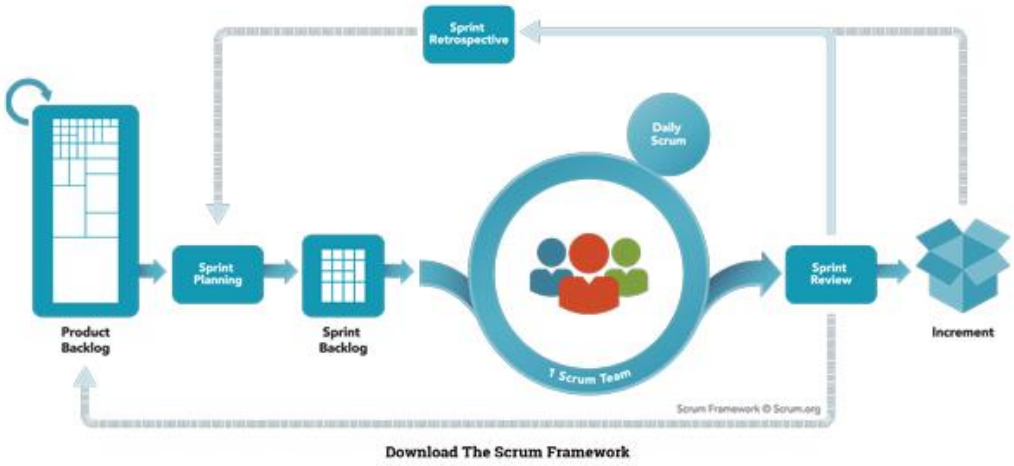
- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective.

Según el framework, los eventos se utilizan para crear regularidad y minimizar la necesidad de reuniones no definidas por Scrum. Una vez iniciado el sprint su duración es fija y no se puede acortar o alargar.

Los artefactos de scrum (Product Backlog, Sprint Backlog e incremento) representan valor para proporcionar oportunidades de inspección y adaptación durante el desarrollo del proyecto.

Figura 4.2

Modelo de Metodología Scrum



Nota. De "¿What is Scrum?", 2020, Scrum.org (https://www.scrum.org/resources/what-is-scrum)

Cada fase del proyecto será un sprint de 2 semanas. Cada inicio de sprint considerará un Sprint Planning, en base a un product backlog definido en la fase comprensión de la necesidad (USM). El Sprint Planning considerará las historias de usuario relacionadas a las tareas y desarrollos a realizar durante la duración del sprint, y contemplará el despliegue de un incremento al finalizar. Para la gestión de historias de

usuario, control del avance, monitoreo de tareas se utilizará la suite Atlalasian Software Tools, en específico Jira Software.

4.7.1 Puntos de Control según metodología Agile:

a) Durante el desarrollo del sprint:

- Daily Scrum: El Daily Scrum facilita un espacio para que el equipo del proyecto pueda expresar y comentar su avance, así como los problemas o bloqueos ocurridos durante el desarrollo de las historias de usuario.
- Burndown Chart: La gráfica de burndown Chart, la cual es generada en forma automática por Jira Software, permite visualizar el avance realizado vs. el avance esperado. La gráfica muestra de forma clara las desviaciones al desarrollo contempladas al inicio del sprint, siendo un buen indicador para tomar las acciones correspondientes y de esta forma garantizar que lo planificado y comprometido a desarrollar se cumpla.

b) Al Finalizar el sprint:

- Sprint Review: Al finalizar el sprint, el equipo realiza una presentación del incremento del producto o servicio a todos los stakeholders. El sprint review es un espacio para que el equipo de desarrollo reciba todo el feedback posible de parte de todos los involucrados, con el fin de mantenerlos informados, recibir propuestas de mejora y de esta forma generar valor. La oportunidad de tener feedback de forma temprana aumenta la probabilidad de aceptación y conformidad por parte del cliente, y eleva la moral del equipo de desarrollo.
- Sprint Retrospective: Luego del sprint review, ya obtenido el feedback por parte de los stakeholders del proyecto, el sprint retrospective es un espacio para que el equipo de desarrollo evalúe las cosas buenas y malas realizadas durante el sprint, y actualizar el sprint backlog con las mejoras o ajustes recibidos en el feedback.

4.7.2 Puntos de control de fases de machine learning:

a.- Cantidad de posibles clientes que prediga el modelo

b.- % de exactitud del modelo de machine learning

c.- Cantidad de algoritmos satisfactorios a ejecutar por modelo

d.- Resultados de matriz de confusión del modelo

4.8 Recursos y presupuesto

Recursos Humanos:

Para la implementación del servicio de machine learning predictivo de potenciales clientes de productos financieros, se necesitó los siguientes perfiles:

1.- Científico de Datos: Experto en modelado de soluciones de machine learning. Deberá tener skills utilizando herramientas estadísticas y de programación.

Debe tener una experiencia mínima de 2 años implementando soluciones de IA y ML.

Herramientas (Opensource): Python, R, SPSS, entre otros.

- Presupuesto mensual (por 66 horas ejecutadas aproximadas): s/. 800.00 (montos referenciales)

2.- Analista de Datos: Experto en tratamiento, extracción y explotación de datos. Deberá contar con skills utilizando diversas herramientas de ETL y de minería de datos. Debe tener una experiencia mínima de 2 años.

Herramientas (Opensource): Pentaho, Hadoop, Spark, entre otros.

- Presupuesto mensual (por 66 horas ejecutadas aproximadas): s/. 800.00 (montos referenciales)

3.- Analista programador frontend: Especialista en desarrollo de framework orientados a la capa front de aplicaciones web y mobile. Debe tener una experiencia mínima de 2 años implementando soluciones de frontend.

Herramientas (Opensource): Angular Js, React (Redux), CSS, entre otros.

- Presupuesto mensual (por horas 20 ejecutadas aproximadas): s/. 800.00 (montos referenciales)

4- Analista programador backend: Especialista en desarrollo con frameworks orientados a la capa back de aplicaciones web. Debe tener una experiencia mínima de 3 años, implementando servicios, APIS, etc.

Herramientas y tecnologías (Opensource): Experto en Java (springboot), desarrollo de APIS, servicios asíncronos, programación reactiva, entre otros.

- Presupuesto mensual (por 20 horas ejecutadas aproximadas): s/. 800.00 (montos referenciales)

5.- Contador: Contador público colegiado. Experiencia mínima de 5 años.

- Presupuesto mensual (por 16 horas ejecutadas aproximadas): s/. 400.00 (montos referenciales)

6.- Gestor Comercial: Profesional con skills de ventas. Experiencia mínima de 2 años.

- Presupuesto mensual (por 12 horas ejecutadas aproximadas): s/. 400.00 (montos referenciales)

7.- Líder/Arquitecto y SM del proyecto: Profesional con experiencia en liderazgo de proyectos y equipos, con sólidos conocimientos en metodologías PMI, prácticas ágiles y de preferencia certificado en SCRUM. Experiencia mínima de 5 años liderando proyectos de grande y mediana envergadura.

- Presupuesto mensual (por 12 horas ejecutadas aproximadas): s/. 500.00 (montos referenciales)

8.- CEO: Gerente de la empresa brindadora del servicio, EKRON SYSTEM S.A.C.

Durante la ejecución del proyecto durante el primer año, se emitirán recibos por honorarios profesionales a todo el equipo anteriormente mencionado, a excepción del CEO y líder del proyecto, los cuales si estarán bajo la modalidad de 5ta. Categoría.

Recursos materiales:

1.- Al estar el servicio profesional de implementación bajo modalidad de 4ta categoría, el servicio incluirá los equipos necesarios para la provisión del servicio, sean laptops, tablets, etc.

2.- Para el equipo clave se deberá contar con una laptop personal, conexión de internet, acceso a telefonía móvil e útiles de oficina.

- Presupuesto aproximado: s/. 5000.00 por única vez (laptop), s/. 500.00 mensuales por servicios de internet, acceso a telefonía móvil e útiles de oficina.

Recursos de marketing y servicios cloud:

1.- Se necesitará habilitar una página web de la empresa, la cual implicarán costos de desarrollo, hosting anual, mantenimiento y de logotipado.

- Presupuesto desarrollo página web: s/.250.00

- Presupuesto mantenimiento de página web: S/ 150.00 mensuales.

- Presupuesto hosting anual: s/. 350.00

- Presupuesto Logotipado: s/. 150.00

2.- Para los servicios que contemplen alojar los servicios de ML en ambiente cloud, se deberá asignar un monto fijo mensual en AWS, Azure o IBM CLOUD para alojar la solución. A su vez, se deberá contemplar un monto fijo mensual por la suite Web, donde se alojará el sistema de gestión de indicadores.

- Presupuesto aproximado mensual (primer año): s/. 800.00

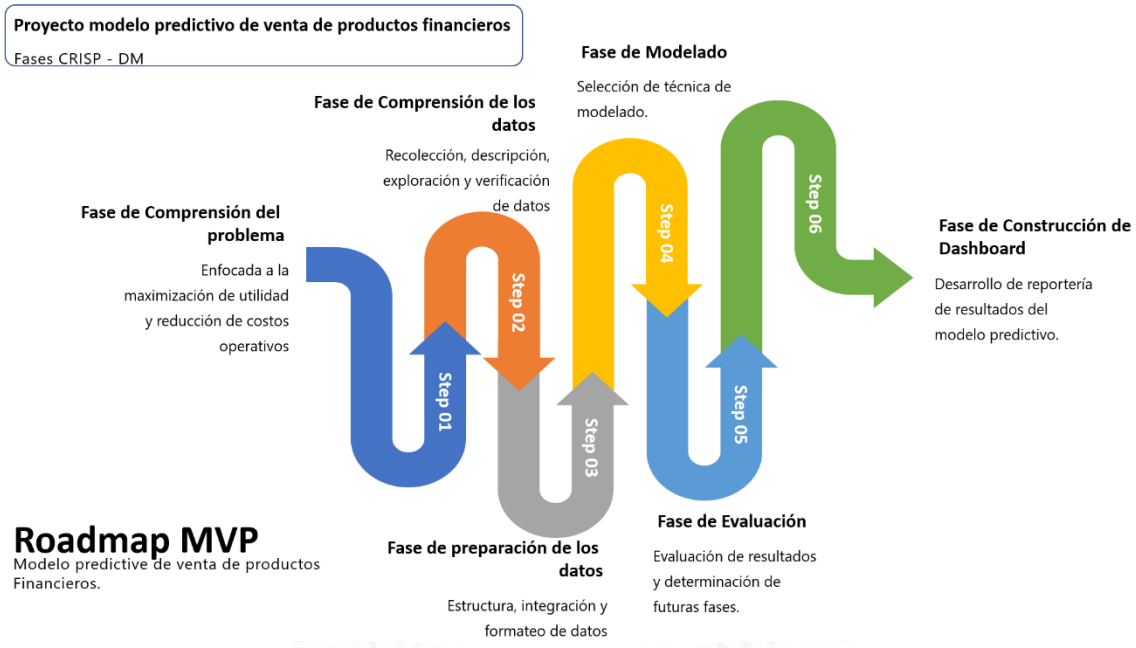
CAPÍTULO V: DESARROLLO DEL PROYECTO

En base a lo analizado en el presente trabajo, se han determinado las siguientes fases para la implementación y despliegue del MVP del proyecto de modelamiento predictivo de ventas de productos financieros.

Este MPV se basa en la metodología CRISP-DM, que nos explica cómo se desarrolló las diferentes fases, que implicó realizar diferentes actividades dentro de un proyecto para poder construir el modelo predictivo, así que basados en ello hemos logrado conseguir los resultados de compra de productos financieros.

Figura 5.1

Roadmap MVP modelo predictivo



5.1 Fase de comprensión del problema:

En el presente estudio, nos enfocamos en la empresa CERTICOM S.A.C., quién actualmente brinda el servicio de promoción y comercialización de productos financieros del banco BBVA. Dentro los servicios que realiza se encuentran los siguientes:

- **Servicio de Venta por Telemarketing:**

El proveedor, CERTICOM S.A.C., realiza la gestión de venta mediante los promotores de venta vía canal telefónico para promover y tentar la colocación de productos del banco, utilizando los protocolos de venta correspondientes. Esta labor incluye una contactabilidad del cliente o prospecto con la información proporcionada por el banco en las bases de datos asignadas.

- **Servicio de venta de agendamiento o presencial**

Consiste en que el proveedor realiza la venta con sus promotores vía visitas presenciales, así como utilizando la información proporcionada por el banco en las bases de datos asignadas.

Para los fines del presente trabajo, nos hemos enfocado en el servicio de venta por Telemarketing, ya que este tipo de venta está soportada por la infraestructura tecnológica que permitirá incorporar la solución propuesta de predicción.

5.1.1 Proceso de venta

En base a lo anteriormente mencionado, explicaremos el proceso en 5 pasos:

Paso 1: El banco proporciona de manera mensual la información de los potenciales adquirientes de productos financieros. Se adjuntan los datos del cliente, como nombres, DNI, edad, ubigeo, oferta, entre otros. A su vez, existen diversos tipos de productos:

- Producto PLD: Préstamos de libre disponibilidad, su valor asciende a más de s/. 30 000.00 por préstamo.
- Producto Renovado: Producto que permite renovar un préstamo.
- Producto Rapi-préstamo: Préstamos menores a s/. 30 000.00. No necesita acercarse a una agenda para hacerlo efectivo.
- Producto Subrogado: Permite comprar la deuda de otro banco.
- Producto Subrogado Mixto: Compra de deuda más un PLD.

Paso 2: Una vez se tiene la información (Base de Datos en formato excel) se procedió al filtrado y segmentación en campañas. La empresa proveedora (call center) utiliza el sistema Vicidial[1] (open source de call center) para gestionar la asignación de llamadas a los promotores. Esta carga de información también se realizó en base a una agrupación, utilizando listas que permiten realizar campañas de venta a la medida, no obstante, la lógica de esta segmentación no obedece a estudios estadísticos.

Paso 3: Las campañas se cargaron al sistema Vicidial, el cual se encarga de aperturar las listas configuradas y asignar de forma aleatoria los registros de la base de datos (clientes) a los promotores. Vicidial realizó la llamada al número de contacto del cliente (celular) y, si se obtuvo una respuesta entrante, asignó esta llamada a un promotor que se encontraba liberado (sin llamadas asignadas).

Paso 4: Una vez que se asignó la llamada, empezó el protocolo de venta realizado por el promotor. Aquí se ofrece el producto financiero al posible cliente, y considerando muchos factores durante la contactabilidad, se desestimó o se concretó la venta.

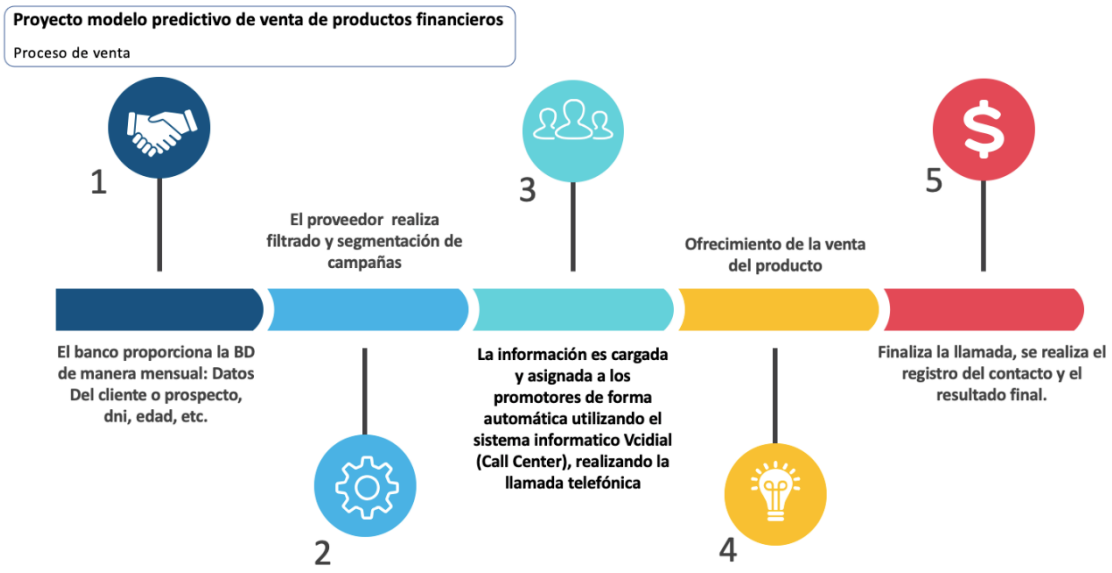
Paso 5: Dependiendo del resultado del proceso de venta, la trazabilidad de la contactabilidad se registra en el sistema Vicidial, y en el caso que la venta haya sido aceptada, y dependiendo del producto vendido, empezó el proceso backoffice, el cual se realizó en el área especializada correspondiente, realizando validaciones de la venta (documentación, gestiona firmas de aceptación, gestión del depósito, entre otros).

En el siguiente gráfico se muestran los 5 pasos del proceso de venta:

[1] Vicidial con url http://www.vicidial.com

Figura 5.2

Modelo predictivo de venta de productos financieros



Según lo indicado en la figura 5.2 (Escenario "AS IS") la ejecución de la predicción utilizando el modelo predictivo se realizó luego del paso 1, contando con la información proporcionada por el banco (leads). Previamente se realizó el entrenamiento del modelo (base de datos histórica), y con el modelo ya construido (disponibilizando mediante servicios web) se invocó y ejecutó en cada inicio de mes.

5.1.2 Comprensión del problema:

Durante el proceso de venta que se realiza de manera mensual en el call center, la cantidad de promotores asignados deberán realizar todas las llamadas posibles para aumentar la probabilidad de conseguir ventas durante el periodo en mención, con el fin de aumentar la rentabilidad del negocio (ingreso por comisión). No obstante, el esfuerzo requerido para completar todas las llamadas implica actualmente una cantidad de 30 agentes, y dependiendo de la base de datos entregada por el banco, cuya cantidad de registros varía.

Por lo expuesto, la eficiencia operativa del call center está sujeta a un esfuerzo desmedido o subutilizado, dependiendo de la cantidad de registros y calidad de la información proporcionada por el banco. Al contar con un staff fijo por mes "promotores", el dimensionamiento óptimo de la operación, actualmente se basa en nociones y en la intuición de la jefatura de operaciones, no considerando estimaciones

probabilísticas ni estadísticas que permitan tener un dimensionamiento basado en un análisis numérico.

Es por ello que, optimizando la operación, se logra reducir costos, y a su vez, teniendo una estrategia de carga inicial, enfocada a aumentar la probabilidad de venta realizando un análisis previo de la información, logrando una segmentación de alta calidad.

Según lo expuesto en el paso 2 del proceso de venta, durante la carga de la información, se segmentó las listas de registros en base a consideraciones subjetivas, determinadas por el área de operaciones, a su vez, estos grupos no contemplarón consideraciones predictivas.

5.2 Fase de comprensión de datos:

Esta fase se realizó la recolección inicial de datos, con el objetivo de establecer un primer contacto con el problema, identificar su calidad y establecer las relaciones más evidentes que permitieron definir las primeras hipótesis.

Las principales tareas que se desarrolló en esta fase son:

- Recolección de datos iniciales: Implicó la extracción de los datos de la BD en producción de la información histórica. Para ello se utilizó herramientas de extracción de Data Integration (ETL) para obtener datamart específicos por: tipo de producto y por segmentación de oferta. La herramienta que utilizamos fue Pentaho.
- Descripción de los datos: Una vez obtenidos los datamarts, se realizó una descripción de los datos en base a un análisis funcional con los dueños del servicio, con el fin de comprenderlos al 100%. Esta labor implicó reuniones específicas con los usuarios del sistema, así como con la parte técnica (personal especialista DBA).
- Exploración de los datos: Una vez comprendidos todos los atributos de las tablas del datamart, se realizó un análisis exploratorio a alto nivel de la información, realizando cruces y/o cubos. Para ello se utilizó herramientas como Powerbi o herramientas propias del módulo data integration.
- Verificación de la calidad de los datos: Finalmente se realizó una verificación de los resultados y análisis de los atributos a considerar, todo

ello con el fin de tener una data de alta calidad, objetivo principal de la siguiente fase de preparación de datos.

5.3 Fase de preparación de datos:

La información proporcionada por el banco, la cual se tiene en formato hoja de cálculo, contuvo diversos campos, atributos de los clientes o prospectos durante el mes de ejecución. Se obtuvieron los siguientes datos:

- Nombre 1
- Nombre 2
- Apellido 1
- Apellido 2
- Dirección
- Ubigeo
- Fecha de nacimiento (edad)
- Ámbito (Lima, provincia)
- Clúster (Independiente, dependiente)
- Oferta (Monto en soles)
- Teléfono de contacto 1
- Teléfono de contacto 2
- Teléfono de contacto 3
- Teléfono de contacto 4
- Código de producto
- Descripción del producto, etc.

Para realizar una predicción de venta, se utilizaron los datos históricos almacenados en la BD del sistema de call center, ya que contiene el dato final del resultado de la venta, sea positiva o negativa. Para realizar un correcto entrenamiento de los datos, se realizó la preparación de estos, respetando una estructura, integración y formateo de datos.

5.3.1 Estructura

En base a los datos originales, tanto a nivel histórico como los datos actuales, se debió realizar un análisis previo de la selección de estos, considerando que toda la base de datos evoluciona, contando con nuevos atributos a la información, actualizándolos,

referenciando otros datos e incluso eliminándolos. Esta fase también incorporó un análisis exploratorio de la información contenida en la base de datos del banco.

5.3.2 Integración

Se realizó una limpieza de datos, considerando que existe información con nulos, formatos de fecha/hora, datos numéricos y categóricos faltantes, etc. Este proceso se le denomina "Pre Procesado de datos", para tal fin se utilizó la herramienta Anaconda basada en Python la cual contiene las librerías especializadas en limpieza y seteo inicial de datos, ejecutando lo siguiente:

- Obtención del conjunto de datos.
- Importación de librerías
- Importación datasets
- Ingresó datos faltantes o desconocidos (datos numéricos y categóricos).
- Tratamiento de NA

5.3.3 Formateo de datos

Se realizó un formateo de los datos con el fin de que se encuentre a una misma escala (escalado de variables), y de esta forma realizar un entrenamiento del modelo de forma correcta. A su vez se realizó una división del dataset, un conjunto de datos de entrenamiento y un conjunto de datos de prueba. Según la necesidad y del tipo de algoritmo a modelar los tamaños de conjunto de datos de entrenamiento y de prueba variarón.

5.3.4 Resultados de la Fase Exploración de Datos:

Para la exploración preliminar de datos, se utilizó la herramienta OLAP Pentaho Server. El objetivo de la exploración consiste en realizar un primer análisis cualitativo de la información proporcionada por CERTICOM. La información corresponde a las ventas de los primeros 7 meses del año 2020 (de enero 2020 a julio 2020). Para el presente análisis se consideró los datos correspondientes a la venta de productos financieros dentro de la coyuntura de la pandemia del coronavirus COVID-19, no considerando información de meses anteriores, ya que la información del año 2019 podría sesgar en

los resultados y predicciones, ya que contiene variables externas que impactarían al modelo.

El cubo considera lo siguiente:

Medidas:

- Monto soles: cantidad en soles colocados por cada venta del producto financiero.
- Oferta pp: cantidad original de la oferta del producto financiero.
- Id negocio: código del producto financiero, en este caso el producto es el préstamo de libre disponibilidad.

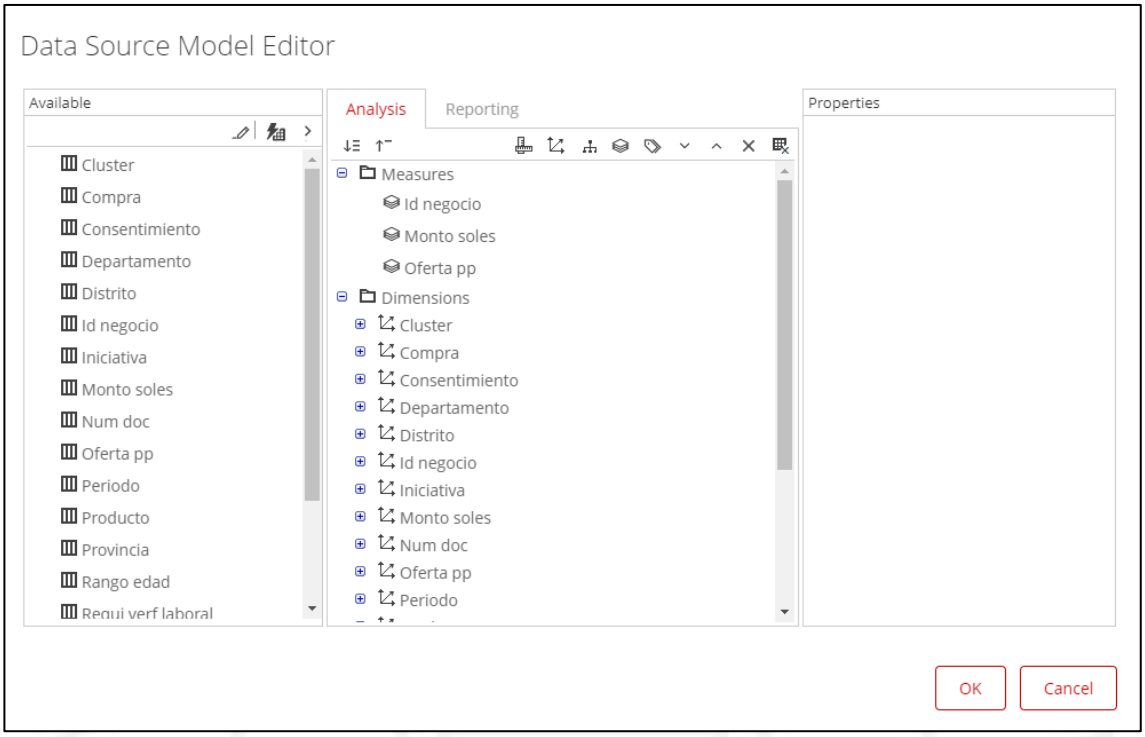
Dimensiones:

- Clúster: Campo que idéntica al cliente en un grupo específico
- Compra: Determina si se logró realizar la venta, o no.
- Consentimiento
- Departamento: Ubigeo
- Distrito: Ubigeo
- Iniciativa
- Num doc: DNI del cliente
- Periodo: Mes
- Producto: Tipo de producto
- Provincia: Ubigeo
- Rango edad: Especifica el rango de edad donde se encuentra el cliente.
- Requi verf laboral: Valida si el cliente requiere verificación laboral.
- Requi verf domici: Valida si el cliente requiere verificación domiciliaria.
- Segmento riesgos
- Tea pp: Tasa efectiva anual del préstamo.
- Tipo cliente: Ejemplo cliente antiguo, cliente nuevo, etc.

A continuación, se muestra el cubo OLAP generado en Pentaho Server:

Figura 5.3

Cubo Olap en Pentaho



Nota. Generado de los datos de empresa Certicom

Para la visualización del cubo, utilizamos el módulo Saiku Analytics, el cual se encuentra configurado en nuestra solución de Pentaho Server.

5.3.5 Resultados del Análisis Exploratorio:

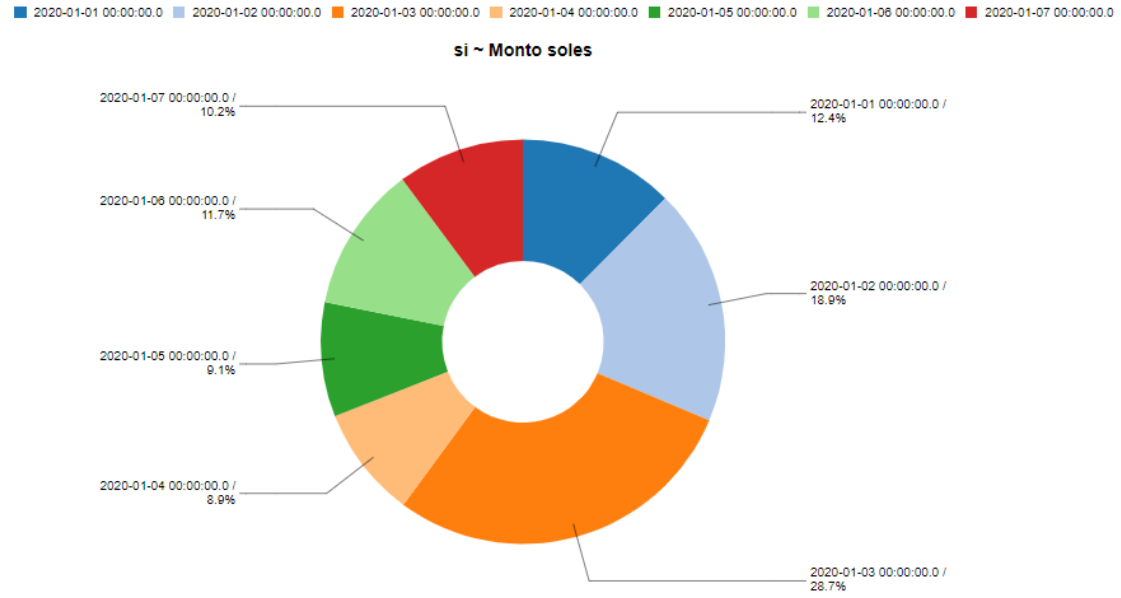
En base al cubo configurado se realizaron los cruces de dimensiones y medidas de la información proporcionada por el banco en el periodo de enero y julio 2020, obteniendo los siguientes resultados:

a) Dimensión volumen total (en soles) de préstamos concedidos

Considera el total en efectivo prestado a los clientes. El siguiente gráfico muestra las cantidades de ventas colocadas (en % y en soles) realizadas de enero a julio 2020.

Tabla 5.1

Tabla de préstamos en soles (PLD)



Periodo	Monto Soles
Enero 2020	6.844.732
Febrero 2020	10.397.536
Marzo 2020	15.807.861
Abril 2020	4.914.729
Mayo 2020	5.031.450
Junio 2020	6.433.388
Julio 2020	5.590.871
Grand Total	**55,020,567**

Nota. Generado de los datos de empresa Certicom

Observaciones:

- Se evidencia que en abril 2020 ocurre una caída significativa en colocación en soles de los productos.
- En enero, febrero y marzo existe un alza de ventas, colocando a 15.8M de soles. En abril se produce una caída llegando solo a 4.9M, manteniéndose constante hasta julio 2020.
- De enero a julio 2020 se colocaron préstamos por el valor de 55M.

b) Cantidad total de ofertas proporcionados por el banco (mensual) y efectividad de venta

Permite conocer la cantidad total mensual de las operaciones de venta, así como el resultado final de la misma (si compro, no compro).

Tabla 5.2

Cantidad Mensual de Venta

Periodo/Venta	No	Si	% Efectividad	Grand Total
Enero 2020	37.917	633	2%	**38,550**
Febrero 2020	47.363	941	2%	**48,304**
Marzo 2020	72.298	974	1%	**73,272**
Abril 2020	41.831	341	1%	**42,172**
Mayo 2020	34.776	226	1%	**35,002**
Junio 2020	37.82	373	1%	**38,193**
Julio 2020	33.431	345	1%	**33,776**
Grand Total	**305,436**	**3,833**		

Nota. Generado de los datos de empresa Certicom

Observaciones:

- Podemos apreciar que durante el periodo de enero a julio 2020, el banco proporcionó 309, 269 ofertas, y de las cuales se concretaron 3, 833 representando el 1.24% del total.
- El banco proporcionó una cantidad mensual de opciones de compra (registro de contactabilidad) notándose que a partir de abril 2020 las cantidades se reducen, obteniendo su valor más bajo en julio 2020 (33 776 registros).

c) Dimensión Rango de Edad vs. Resultado de venta

Una de las variables importantes que se consideró dentro del presente análisis, es el rango de edad. A continuación, se muestra la información segmentada por Rango de edad y resultado de la venta.

Tabla 5.3

Resultado de venta según rango de edad

Rango Edad	No	Si	% Efectividad	Grand Total
01. [20 - 30]	36.346	730	2%	**37,076**
02. [31 - 40]	84.994	1449	2%	**86,443**
03. [41 - 55]	148.2	1408	1%	**149,607**
04. [56 - 65]	1.039	6	1%	**1,045**
05. [66 a más>	34.82	240	1%	**35,060**
06. SIN INFO	38	-		**38**
Grand Total	**305,436**	**3,833**		

Nota. Generado de los datos de empresa Certicom

A continuación, se muestran los resultados por rango de edad desde los meses de enero a julio 2020, con valores de venta de quienes compraron y quienes no y el % de efectividad de la compra.

Tabla 5.4

Resultados de ventas por mes y rango de edad 2020

	Ene-20			Feb-20			Mar-20			Abr-20		
Rango edad	**No**	**Si**	**% Efectividad**	**No**	**Si**	**% Efectividad**	**No**	**Si**	**% Efectividad**	**No**	**Si**	**% Efectividad**
01. [20 - 30]	4.47	127	0.3%	5.399	176	0.4%	9.475	204	0.3%	3.714	52	0.1%
02. [31 - 40]	10.32	233	0.6%	13.05	349	0.7%	20.66	361	0.5%	11.055	121	0.3%
03. [41 - 55]	19.16	238	0.6%	24.11	365	0.8%	34.53	353	0.5%	21.679	141	0.3%
04. [56 - 65]	-	-		-	-		298	1	0.001%	741	5	0.01%
05. [66 a más>	3.966	35	0.1%	4.801	51	0.1%	7.325	55	0.1%	4.612	22	0.1%
06. SIN INFO	-	-		-	-		8	-		30	-	
Total	**37,917**	**633**		**47,363**	**941**		**72,298**	**974**		**41,831**	**341**	

	May-20			Jun-20			Jul-20			
Rango edad	**No**	**Si**	**% Efectividad**	**No**	**Si**	**% Efectividad**	**No**	**Si**	**% Efectividad**	**Total**
01. [20 - 30]	2.933	25	0.1%	4.394	73	0.2%	5.961	73	0.2%	**37,076**
02. [31 - 40]	8.895	83	0.2%	10.83	151	0.4%	10.17	151	0.4%	**86,443**
03. [41 - 55]	17.39	87	0.2%	17.65	128	0.3%	13.69	96	0.3%	**149,607**
04. [56 - 65]	-	-		-	-		-	-		**1,045**
05. [66 a más>	5.563	31	0.1%	4.948	21	0.1%	3.605	25	0.1%	**35,060**
06. SIN INFO	-	-		-	-		-	-		**38**
Total	**34,776**	**226**		**37,820**	**373**		**33,431**	**345**		

Nota. Generado de los datos de empresa Certicom

Realizando el cruce entre el rango de edad, la variable de compra (si compra/no compra) y el periodo (mes) obtenemos los siguientes resultados:

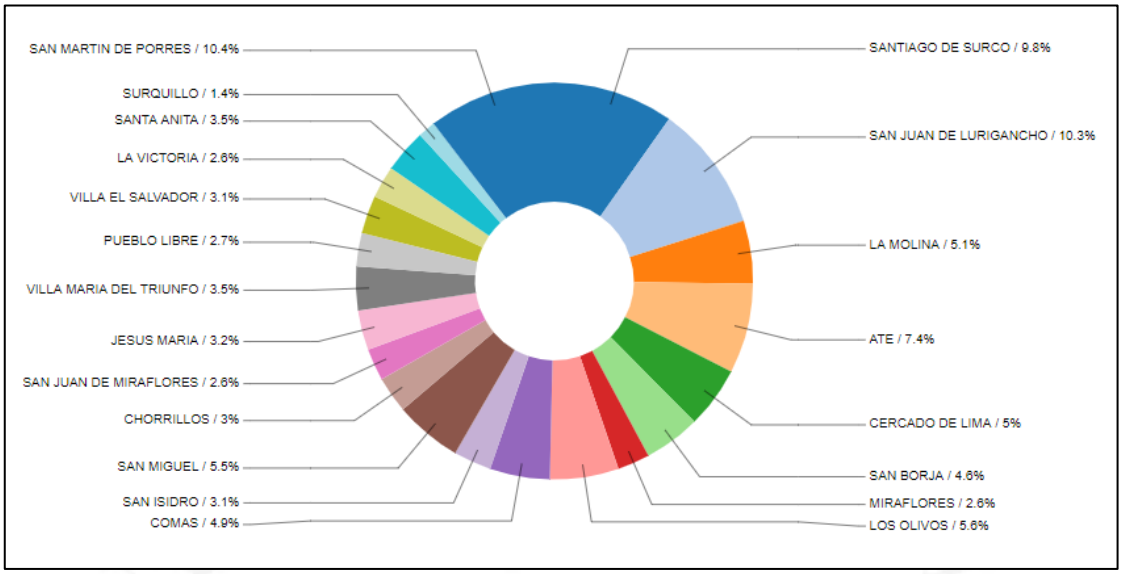
- Podemos apreciar que en base al rango de edad, el comportamiento de compra varía significativamente. Por ejemplo, entre las edades de 31 a 40 y de 41 a 55 la cantidad de registros representa el 76% del total de registros.
- La mayor cantidad de registros (ofertas de productos) se ubica en el rango de edades de 41 a 50 años (149,607).
- La mayor cantidad de ventas se ubica en el rango de edad de entre 31 y 40 años (1449)
- El mes con mayor cantidad de ventas es marzo 2020 con 974 colocaciones, y mayo obtiene la cantidad menor (226).
- El rango de edad de 56 a 65 años de prospectos cuenta con información solo de marzo y abril 2020.
- La cantidad de información correspondiente al rango de edad de 66 años a más permanece relativamente constante durante los 7 meses de análisis.
- La categoría 06. SIN INFO, no se consideró dentro del modelo predictivo, debido a que sólo existen 38 registros entre enero y julio 2020.

d) Dimensión Distrito vs. Resultado de venta

La variable distrito está asociada al lugar de residencia del cliente. A continuación, se muestran la cantidad total de préstamos concedidos (en soles) por distrito, así como la cantidad total de oferta (los 21 distritos más representativos) y cantidad total de ventas (los 7 distritos más representativos) durante el periodo enero a julio 2020.

Figura 5.4

Resultados de venta por distrito en %



Nota. Generado de los datos de empresa Certicom

En la siguiente tabla se muestra la cantidad de leads por distrito según la base de datos en donde los distritos con mayor cantidad de posibles clientes son Santiago de Surco, San Juan de Lurigancho y San Martin de Porres.

Tabla 5.5

Cantidad total de posibles clientes por distrito según la base de datos

Distrito	Leads
SANTIAGO DE SURCO	27.317
SAN JUAN DE LURIGANCHO	20.731
SAN MARTIN DE PORRES	19.206
LA MOLINA	14.240
ATE	14.120
CERCADO DE LIMA	13.470
SAN BORJA	12.926
MIRAFLORES	12.579
LOS OLIVOS	12.332
COMAS	11.489
SAN ISIDRO	10.478
SAN MIGUEL	9.696
CHORRILLOS	9.619
SAN JUAN DE MIRAFLORES	9.502
JESUS MARIA	8.037

Distrito	Leads
VILLA MARIA DEL TRIUNFO	7.592
PUEBLO LIBRE	7.578
VILLA EL SALVADOR	7.298
LA VICTORIA	6.627
SANTA ANITA	5.812
SURQUILLO	5.608

Nota. Generado de los datos de empresa Certicom

Finalmente, en esta dimensión de distrito, se muestra en la siguiente tabla los distritos que consiguieron mayores ventas.

Tabla 5.6

Ventas en soles por distrito

Distrito	Monto soles
SAN MARTIN DE PORRES	4.467.012
SAN JUAN DE LURIGANCHO	4.419.675
SANTIAGO DE SURCO	4.201.306
ATE	3.154.570
LOS OLIVOS	2.400.378
SAN MIGUEL	2.357.600
LA MOLINA	2.195.142
CERCADO DE LIMA	2.139.305
COMAS	2.094.899
SAN BORJA	1.978.600
VILLA MARIA DEL TRIUNFO	1.522.440
SANTA ANITA	1.511.600
EL AGUSTINO	1.396.800
JESUS MARIA	1.390.000
VILLA EL SALVADOR	1.334.100
SAN ISIDRO	1.318.100
CHORRILLOS	1.271.550
PUEBLO LIBRE	1.175.784
MIRAFLORES	1.128.000
LA VICTORIA	1.122.200
SAN JUAN DE MIRAFLORES	1.121.300

Nota. Generado de los datos de empresa Certicom

Adicionalmente se presenta la siguiente tabla mostrando la cantidad de ventas entre los leads por distrito, en donde se puede determinar el monto promedio por distrito.

Tabla 5.7

Monto promedio por distrito

Distrito	Leads	Desembolso/No Desembolso	Cantidad	Monto desembolsado (S/)	Desembolso promedio por Lead (S/)	% Efectividad
SANTIAGO DE SURCO	27.317	No	27.091	0	0	0.83%
		Si	226	4.201.306	18.590	
SAN JUAN DE LURIGANCHO	20.731	No	20.371	0	0	1.74%
		Si	360	4.419.675	12.277	
SAN MARTIN DE PORRES	19.206	No	18.882	0	0	1.69%
		Si	324	4.467.012	13.787	
LA MOLINA	14.240	No	14.126	0	0	0.80%
		Si	114	2.195.142	19.256	
ATE	14.120	No	13.889	0	0	1.64%
		Si	231	3.154.570	13.656	
CERCADO DE LIMA	13.470	No	13.305	0	0	1.22%
		Si	165	2.139.305	12.965	
SAN BORJA	12.926	No	12,824	0	0	0.79%
		Si	102	1.978.600	19.398	
MIRAFLORES	12.579	No	12.514	0	0	0.52%
		Si	65	1.128.000	17.354	
LOS OLIVOS	12.332	No	12,163	0	0	1.37%
		Si	169	2.400.378	14.203	
COMAS	11.489	No	11.333	0	0	1.36%
		Si	156	2.094.899	13.429	
SAN ISIDRO	10.478	No	10.421	0	0	0.54%
		Si	57	1.318.100	23.125	
SAN MIGUEL	9.696	No	9,600	0	0	0.99%
		Si	96	2.357.600	24.558	
CHORRILLOS	9.619	No	9.505	0	0	1.19%
		Si	114	1.271.550	11.154	
SAN JUAN DE MIRAFLORES	9.502	No	9.392	0	0	1.16%
		Si	110	1.121.300	10.194	
JESUS MARIA	8.037	No	7.973	0	0	0.80%
		Si	64	1.390.000	21.719	

(continúa)

(continuación)

Distrito	Leads	Desembolso/No Desembolso	Cantidad	Monto desembolsado (S/)	Desembolso promedio por Lead (S/)	% Efectividad
VILLA MARIA DEL TRIUNFO	7.592	No	7.467	0	0	1.65%
		Si	125	1.522.440	12.180	
PUEBLO LIBRE	7.578	No	7.491	0	0	1.15%
		Si	87	1.175.784	13.515	
VILLA EL SALVADOR	7.298	No	7,175	0	0	1.69%
		Si	123	1.334.100	10.846	
LA VICTORIA	6.627	No	6.548	0	0	1.19%
		Si	79	1.122.200	14.205	
SANTA ANITA	5.812	No	5.719	0	0	1.60%
		Si	93	1.511.600	16.254	
SURQUILLO	5.608	No	5.555	0	0	0.95%
		Si	53	613.850	11.582	

Nota. Generado de los datos de empresa Certicom

Realizando el cruce entre el distrito, la variable de compra (si compra/no compra) y el periodo (mes) obtenemos los siguientes resultados:

- Las mayores ofertas se realizaron en los distritos de Santiago de surco, San Juan de Lurigancho, San Martín de Porres, La Molina, Ate, Cercado de Lima y San Borja.
- El distrito que ha recibido productos (Monto total soles) es San Martin de Porres (4.467.012), seguido por San Juan de Lurigancho (4.419.675) y Santiago de Surco (4.201.306).
- El distrito de San Juan de Lurigancho obtiene la mayor cantidad de ventas con el 9.4% de las ventas totales en el periodo de enero a julio 2020, seguido por San Martín de Porres (8.5%), Ate (6%), Santiago de Surco (5.9%), los olivos (4.4%), cercado de lima (4.3%) y comas (4.1%).
- El distrito que tiene mayor porcentaje de efectividad es San Juan de Lurigancho con 1.74%.

e) Dimensión Tipo de cliente vs. Resultado de venta

El banco proporciona una variable llamada Tipo de Cliente, la cual segmenta la información considerando a un cliente antiguo, ex cliente, no cliente y cliente nuevo.

Tabla 5.8

Ventas por tipo de cliente

Tipo cliente	No	Si	Grand Total
CLIENTE ANTIGUO	163.820	3.134	**166,954**
EX CLIENTE	68.751	276	**69,027**
NO CLIENTE	71.567	391	**71,958**
CLIENTE NUEVO	1.298	32	**1,330**
Grand Total	**305,436**	**3,833**	

Nota. Los datos representan la cantidad de número de clientes. Generado de los datos de empresa Certicom

Observaciones:

- La base de datos contiene en un 53.98% de clientes antiguos.
- La mayor proporción (3134 clientes) de ventas se realizó en el tipo de cliente antiguo.
- La menor proporción de venta se realizó en el tipo de cliente nuevo, contando solo con 32 ventas.

f) Dimensión Producto vs. Resultado de ventas (número de clientes)

En esta dimensión se explican los productos que se ofrecen con la cantidad de clientes que compró y que no compró en el periodo de 6 meses.

Tabla 5.9

Ventas por tipo de producto

Producto	No	Si	Grand Total
BASE ADICIONAL	69.542	860	**70,402**
PLD PRESTAMOS RENOVADOS	5.414	793	**6,207**
RAPIPRESTAMO	159.088	1.416	**160,504**
PLD SUBROGADO	26.678	473	**27,151**
PRESTAMO BP	34.259	232	**34,491**
PLD SUBROGADO PLD	10.455	59	**10,514**
Grand Total	**305,436**	**3,833**	

Nota. Generado de los datos de empresa Certicom

Observaciones:

- La base de datos contiene 160 504 registros del producto Rapi-préstamo, siendo el de mayor concentración representando el 51.89%, y a su vez representa la mayor cantidad de ventas, con 1416 operaciones satisfactorias.
- El producto PLD préstamo renovado obtiene la menor cantidad de registros durante el periodo enero – julio 2020, no obstante, no es el de menor venta.

g) **Dimensión Segmento de Riesgo vs. Resultado de ventas**

En la siguiente tabla se mostrará los segmentos de riesgos y los leads que se cuentan dependiendo de la compra.

Tabla 5.10

Ventas por segmento de riesgo

Compra	No	Si	
Segmento riesgos	**Leads**	**Leads**	**Total**
RIESGO BAJO	174.565	2.133	**176,698**
RIESGO MEDIO	46.472	450	**46,922**
RIESGO MEDIO ALTO	46.482	617	**47,099**
SIN INFORMACION	37.917	633	**38,550**
Grand Total	**305,436**	**3,833**	

Nota. Los datos representan la cantidad de número de clientes. Generado de los datos de empresa Certicom

Observaciones:

- El segmento de riesgo bajo es el de mayor porcentaje del total de registros del periodo enero-julio 2020, representando más del 50% del total de registros.
- La cantidad de personas que aceptaron el producto financiero se mantiene en valores constantes en los segmentos riesgo medio, riesgo medio alto y sin información.

h) Dimensión Clúster vs. Resultado de ventas

La dimensión clúster segmenta la información en base a una tipificación relacionada a los valores de alto valor, consumo, independiente, joven.

En la siguiente tabla se muestra los valores en leads según el tipo de clúster, que significa como se segmenta al cliente.

Tabla 5.11

Cantidad de ventas por clúster o tipo de cliente

Clúster	No	Si	Grand Total
ALTO VALOR	50 633	590	**51,223**
CONSUMO	159 877	1 457	**161,334**
INDEPENDIENTE	38 726	679	**39,405**
JOVEN	56 200	1.107	**57,307**
Grand Total	**305,436**	**3,833**	

Nota. Generado de los datos de empresa Certicom

Observaciones:

- El segmento consumo representa el 38% de los productos vendidos, pero también representan la mayor cantidad de registros durante todo el periodo enero – julio.
- El clúster Alto Valor, representa la menor cantidad de ventas.
- El clúster Joven representa la segunda mayor cantidad de ventas, incluso obtiene solo la tercera parte de registro en comparación del clúster consumo.

i) Dimensión Requiere verificación laboral/domiciliaria vs. Resultado de ventas

La dimensión de verificación laboral y domiciliaria indica la necesidad de validar ambos escenarios para realizar la venta. A continuación, se muestra la cantidad de leads según si compra o no compra de acuerdo a la dimensión.

Tabla 5.12

Cantidad de leads según dimensión de requerimiento de verificación

Requiere verificación laboral	No	Si	Grand Total
NO	233.761	3.025	**236,786**
SI	71.675	808	**72,483**
Grand Total	**305,436**	**3,833**	

Requiere verificación domiciliaria	No	Si	Grand Total
NO	241.308	2.997	**244,305**
SI	64.128	836	**64,964**
Grand Total	**305,436**	**3,833**	

Nota. Generado de los datos de Certicom

Observaciones:

- En el caso de verificación laboral, la mayor parte de ventas no requiere verificación laboral (79.62%)
- En el caso de verificación domiciliaria, la mayor parte de ventas no requiere de esta verificación (78.18%)

j) Dimensión Iniciativa vs. Resultado de ventas

En esta dimensión se cuenta con una categorización del cliente por cuatro tipos, según la data analizada.

Tabla 5.13

Cantidad de leads según dimensión de iniciativa

Iniciativa	No	Si	Grand Total
CAPTA	156.473	767	**157,240**
RECONQSTA	10.956	244	**11,200**
VINCULA1	88.208	1.230	**89,438**
VINCULA2	49.799	1.592	**51,391**
Grand Total	**305,436**	**3,833**	

Nota. Generado de los datos de Certicom

Observaciones:

- Las iniciativas de VINCULA1 y VINCULA2 representan el 73.62 % del total de ventas.
- La iniciativa Captación y Reconquista solo representan el 26.37% del total de ventas.

k) Dimensión consentimiento vs. Resultado de ventas

La dimensión consentimiento indica si el cliente ha dado su consentimiento para que sea contactado y puedan realizarle un ofrecimiento de algún producto financiero.

Tabla 5.14

Cantidad de leads según consentimiento

Consentimiento	No	Si	Grand Total
PENDIENTE	171 254	1 278	**172,532**
SI	134 182	2 555	**136,737**
Grand Total	**305,436**	**3,833**	

Nota. Generado de los datos de Certicom

Observaciones:

- La mayor cantidad de ventas la representa el grupo de clientes que si ha dado su consentimiento de contactabilidad (66.65%).

5.3.6 Integración y formateo de datos:

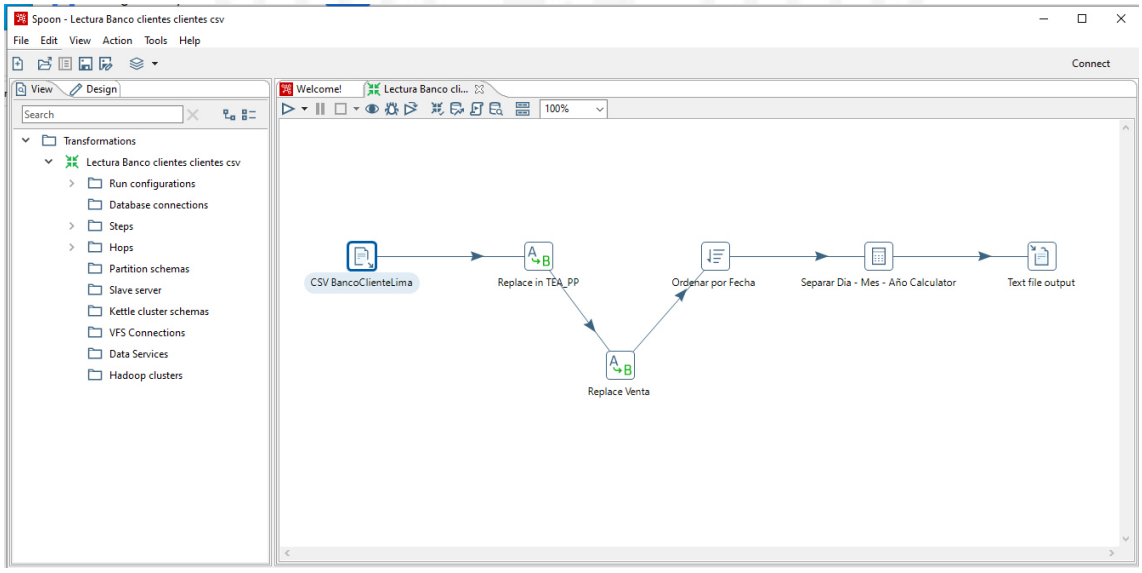
Para realizar la integración de los datos (limpieza de nulos, formatos de fecha, datos numéricos, faltantes) se utilizó la herramienta Pentaho Data Integration (PDI), la cual permitió realizar una ETL transformando los datos, llegando a la integración de los mismos.

En base al archivo original proporcionado por CERTICOM y su análisis previo, se realizó lo siguiente:

- Carga inicial de archivo (PDI)
- Formateo del campo TEA (convertir en numérico)
- Formateo del campo “compra”, convirtiéndolo en binario (0=no compra, 1=si compra)
- Separación del campo “periodo” en los campos MesCliente, AñoCliente y DiaCliente.
- Obtención del DatamartLima, el cual fue input para el modelado predictivo.

Figura 5.5

Flujo ETL de transformación de la información



Nota. Generado de los datos de empresa Certicom

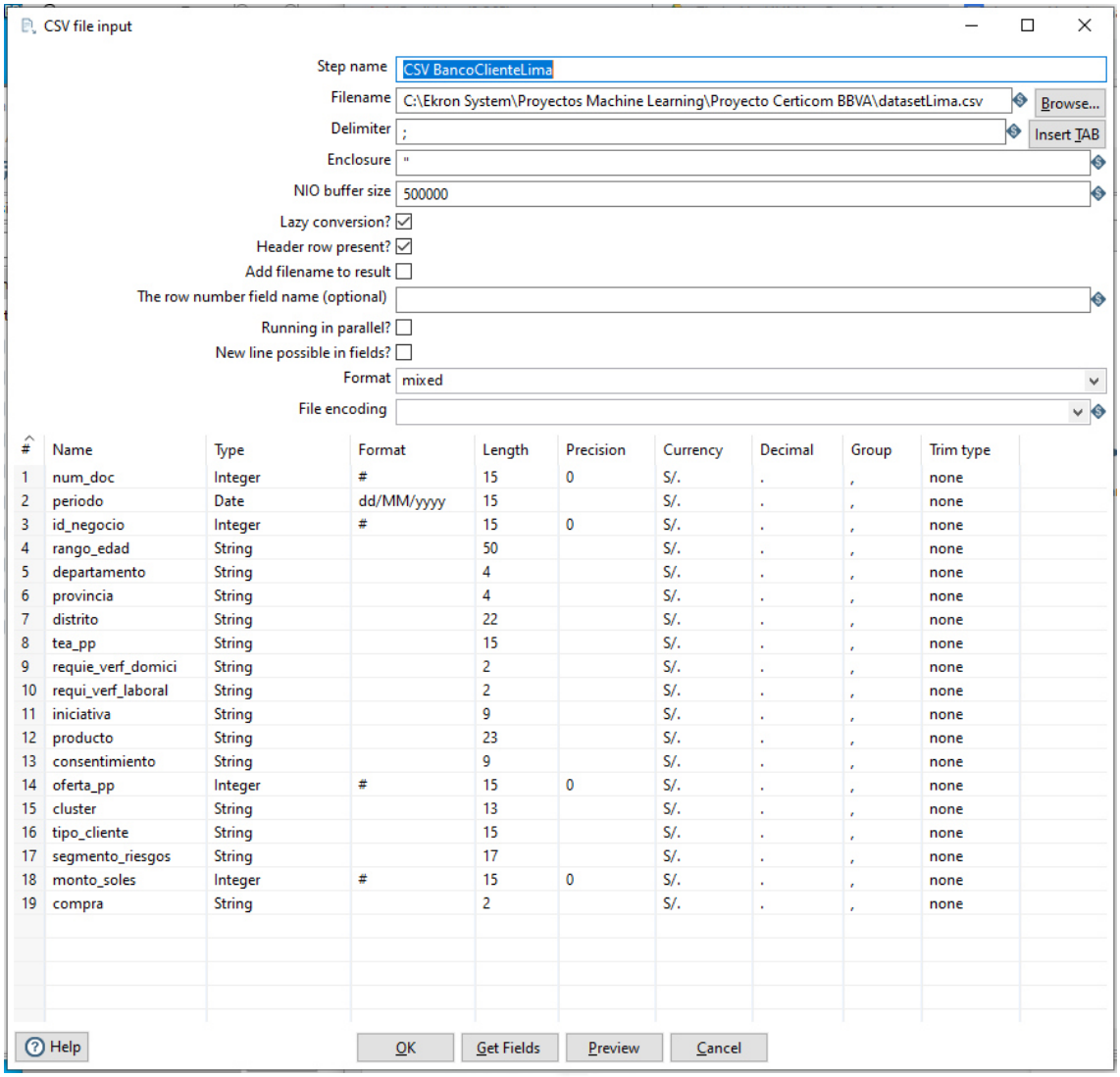
5.3.7 Procedimiento de transformación de datos

Paso 1: Carga Inicial de los datos a PDI (Pentaho Data Integration)

Aquí se muestra la carga del dataset original proporcionada por la empresa Certicom, en donde se cargan los datos tal cual fueron recibidos.

Figura 5.6

Carga Inicial de datos



#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	num_doc	Integer	#	15	0	S/.	.	,	none
2	periodo	Date	dd/MM/yyyy	15		S/.	.	,	none
3	id_negocio	Integer	#	15	0	S/.	.	,	none
4	rango_edad	String		50		S/.	.	,	none
5	departamento	String		4		S/.	.	,	none
6	provincia	String		4		S/.	.	,	none
7	distrito	String		22		S/.	.	,	none
8	tea_pp	String		15		S/.	.	,	none
9	requie_verf_domici	String		2		S/.	.	,	none
10	requi_verf_laboral	String		2		S/.	.	,	none
11	iniciativa	String		9		S/.	.	,	none
12	producto	String		23		S/.	.	,	none
13	consentimiento	String		9		S/.	.	,	none
14	oferta_pp	Integer	#	15	0	S/.	.	,	none
15	cluster	String		13		S/.	.	,	none
16	tipo_cliente	String		15		S/.	.	,	none
17	segmento_riesgos	String		17		S/.	.	,	none
18	monto_soles	Integer	#	15	0	S/.	.	,	none
19	compra	String		2		S/.	.	,	none

Nota. Generado de los datos de empresa Certicom

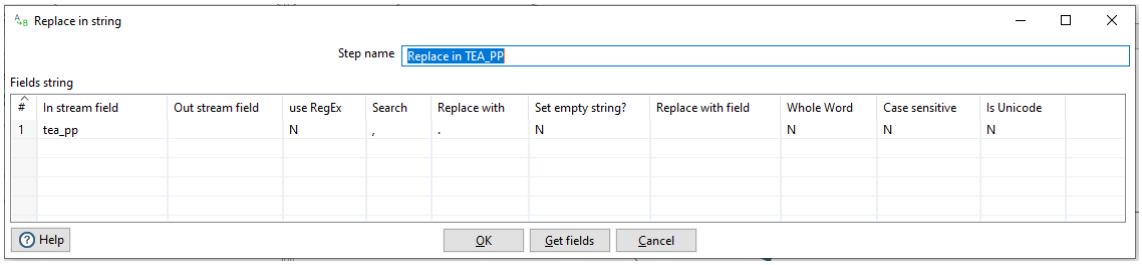
Paso 2: Formatear el campo TEA

Los valores originales llegaron el formato string se convirtieron en formato numérico de 4 decimales, el cual se muestra evidencia a continuación.

Inicialmente la data llegó no homogeneizada, el campo TEA- PP cuenta con datos con el símbolo % en algunos casos y en otros en formato numérico.

Figura 5.7

Evidencia de formateo de TEA



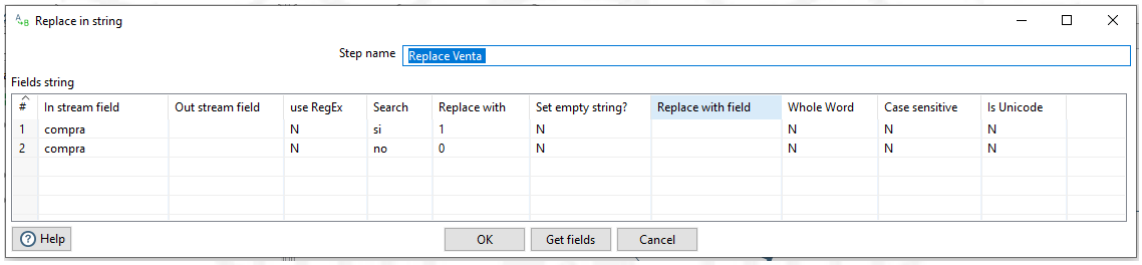
Nota. Generado de los datos de empresa Certicom

Paso 3: Formateo del campo compra

El valor original contenía el valor de si o no, el cuál fue modificado a valor binario 0 o 1, como se muestra a continuación.

Figura 5.8

Evidencia de formateo de campo compra



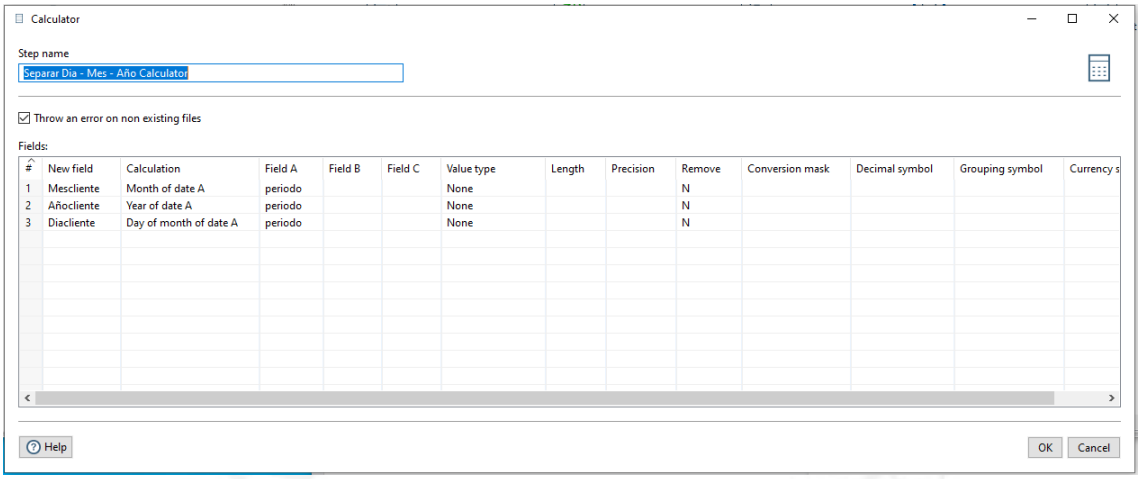
Nota. Generado de los datos de empresa Certicom

Paso 4: Separar del campo "período" en los campos MesCliente, AñoCliente y DiaCliente.

El campo original llego en formato date con nomenclatura día/mes/año, este campo fue separado por tres campos para poder separar meses y tener el modelo mensual. A continuación, se muestra evidencia de esta separación.

Figura 5.9

Evidencia de separación campo período



Nota. Generado de los datos de empresa Certicom

Paso 5: Obtener el DatamartLima

A la finalización de todos estos pasos se obtuvo un formato csv que luego se utilizó en la siguiente fase de modelado.

5.4 Fase de Modelado

En esta fase se seleccionará la técnica de modelado, lo cual involucra poder decidir los algoritmos que se utilizarán para esta sección y que se explican a continuación**.**

5.4.1 Algoritmos de Clasificación:

Para el aprendizaje supervisado se colocó una etiqueta, que según los datos de entrada se pueda predecir la salida para ello se utilizó los algoritmos de clasificación. Y basándose en información histórica el modelo va aprendiendo y prediciendo con mayor exactitud. Los algoritmos de clasificación se utilizan, por ejemplo: para predecir una compra, para determinar ejecutar o no una actividad, básicamente predecir comportamientos con probabilidades de un “Si” o un “No”.

Mediante los algoritmos de clasificación se logró predecir variables categóricas para determinar si compra o no un producto financiero, se realizó el análisis de diferentes escenarios de entrenamiento. Nuestra función objetivo logró predecir si la compra se produce o no, por tanto, este tipo de aprendizaje automatizado considerará los siguientes algoritmos:

- KNN (K Nearest Neighbors)
- SVM Lineal
- SVM Kernel
- Teorema de Bayes
- Arboles de Decisión
- Radom Forest
- Gradient Boosting Classifier

Para mostrar los resultados de cada algoritmo se utilizará la "**Matriz de confusión/Exactitud",** para efecto de evaluación del modelo, la cual permite determinar en la data de prueba (X_test, y_test) cuántos aciertos tuvo el modelo.

Así mismo para medir la exactitud y eficacia del modelo, considerando variables independientes que impacten en el resultado de la venta (correcta asignación de variables), se utilizó la variable "Accuracy".

Los pasos previos al ajuste del clasificador y de la predicción, pasaron por las siguientes fases obligatorias, desde la importación de librerías hasta el escalado de variables:

1.- Se importaron las librerías matplotlib.pyplot (generación de gráficos), pandas (carga de dataset) y desde sklearn.metrics importamos parámetros para la curva ROC.

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import roc_curve, roc_auc_score
```

2- Importamos el dataset DatamartLima2020.csv

```
dataset = pd.read_csv('DatamartLima2020.csv')
```

3.- Codificamos los datos categóricos.

```
def createDummies(df, var_name):
    dummy = pd.get_dummies(df[var_name], prefix=var_name)
    df= df.drop(var_name, axis = 1)
    df= pd.concat([df, dummy], axis =1)
    return df

dataset = createDummies(dataset,'rango_edad')
dataset = createDummies(dataset,'Region')
dataset = createDummies(dataset,'requie_verf_domici')
dataset = createDummies(dataset,'requi_verf_laboral')
dataset = createDummies(dataset,'iniciativa')
dataset = createDummies(dataset,'producto')
dataset = createDummies(dataset,'consentimiento')
dataset = createDummies(dataset,'cluster')
dataset = createDummies(dataset,'tipo_cliente')
dataset = createDummies(dataset,'segmento_riesgos')
```

4.- Se separó el dataset en dos, "X" con los datos de las características e "y" con los datos a predecir (Compra)

```
X = dataset.drop('compra', axis=1)
y = dataset['compra'].values
```

5.- Se dividió el dataset en el conjunto de entrenamiento y conjunto de testing. A su vez se realizó un escalado de variables con el fin de homogenizar la información del dataset de características (X).

```
# Dividir el data set en conjunto de entrenamiento y conjunto de testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,
random_state = 0)


# Escalado de variables
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

1. K Nearest Neighbors (KNN):

El algoritmo K Nearest Neighbors (K vecinos más cercanos) permitió clasificar un conjunto de datos donde los K vecinos deciden a que grupo pertenece un nuevo dato, la cual no tiene clasificación previa, basadas en proximidades (categorías vs nuevo dato).

KNN Pasos

Los pasos que se realizó para aplicar KNN son los siguientes:

1.- Ajuste del clasificador KNN: Se utilizó la librería sklearn.neighbors para importar el clasificador y se ajustaron los datos de entrenamiento (X_train, y_train).

```
# Ajustar el clasificador en el Conjunto de Entrenamiento
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = "minkowski", p
= 2)
classifier.fit(X_train, y_train)
```

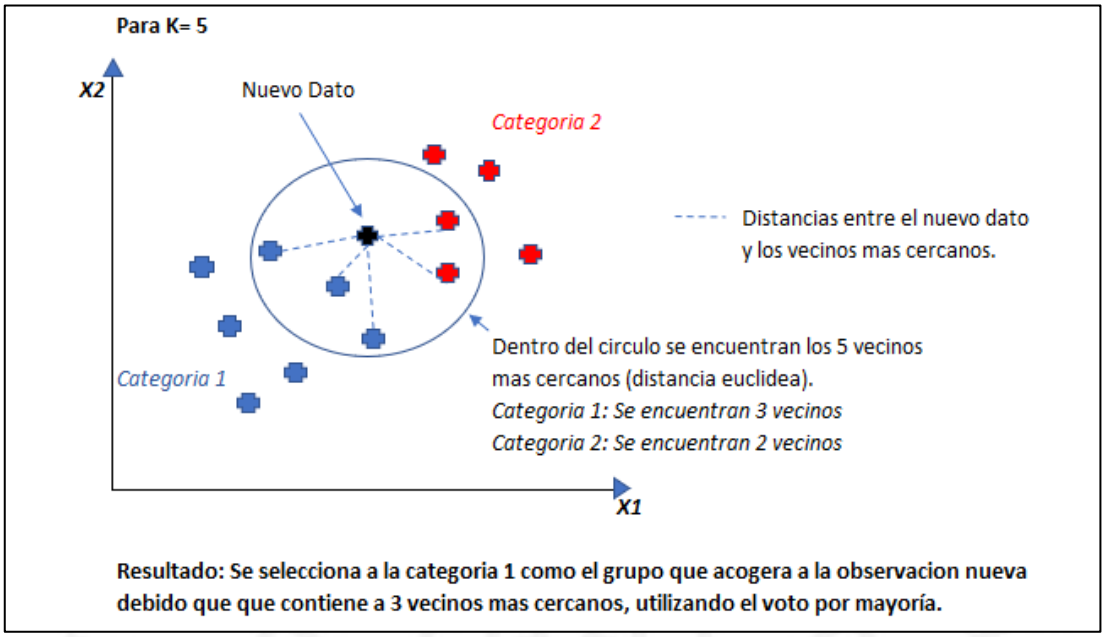
P

2.- Predicción de los resultados: Se realizó la predicción (y_pred) utilizando el clasificador ajustado y utilizando los datos de testing (X_test). A su vez, se realizó una comparación (accuracy) de los datos de la predicción (y_pred) vs. los datos reales (y_test).

```
# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)
```

Figura 5.10

KNN Descripción gráfica



Nota. Adaptado de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Librería y parámetros:

Para el modelo utilizando KNN, se utilizó la clase KNeighborsClassifier de Python, con los siguientes parámetros básicamente que ayudan a mejorar el modelo, por ejemplo al momento de verificar los vecinos, del dato que se busca predecir:

- n_neighbors = 5 (cantidad de vecinos)
- metric = "minkowski"
- p = 2 (distancia euclídea)

Resultados - Matriz de confusión/Exactitud

A continuación, se muestran los resultados de la matriz de confusión y del valor que se tuvo como resultado de la variable accuracy.

Tabla 5.15

Resultado de matriz de confusión Knn

		Predicción	
		0	**1**
Valor Real	**0**	76333	26
	1	951	8

El modelo ha acertado 76333 de los casos de No compra, no obstante, acertó en 8 ventas concretadas.

Exactitud

Resultado: 0.9873638738715435

Accuracy: El modelo tiene un 98.73% de exactitud de predecir el resultado.

2. Support Vector Machine - Máquinas de soporte vectorial (SVM)

El algoritmo de SVM busca resolver el problema de la separación entre grupos, que utiliza el concepto de margen máximo (corredor o pasillo), detectando los dos puntos más cercanos a una línea separadora, siendo el margen máximo de error el espacio entre los planos determinados por los dos puntos de soporte, esperando que las distancias sean máximas, y de esta forma diferenciar a los grupos o segmentos utilizando esta división. Para ello se seleccionan dos puntos, vectores de soporte vectorial del modelo, equidistantes a una línea separadora, hiperplano separador en escenarios de diversas dimensiones, que es capaz de separar lo más posible las categorías a analizar.

SVM Pasos:

1.- Ajuste del clasificador SVM: Se utilizó la librería sklearn.svm para importar el clasificador y se ajustaron los datos de entrenamiento (X_train, y_train).

```
# Ajustar el SVM en el Conjunto de Entrenamiento
from sklearn.svm import SVC
classifier = SVC(kernel = "linear", random_state = 0)
classifier.fit(X_train, y_train)
```

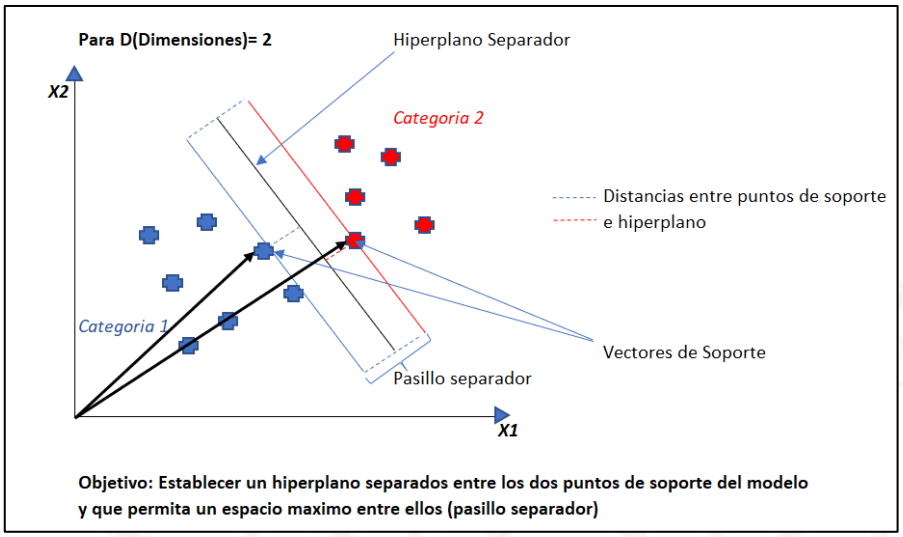
2.- Predicción de los resultados: Se realizó la predicción (y_pred) utilizando el clasificador ajustado y utilizando los datos de testing (X_test). A su vez, se realizó una comparación (accuracy) de los datos de la predicción (y_pred) vs. los datos reales (y_test).

```
# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)
```

A continuación, se presenta la descripción gráfica del algoritmo que se tomará de referencia para la construcción de nuestro modelo.

Figura 5.11

SVM Descripción gráfica



Nota. Adaptado de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Librería y parámetros

Para el modelo utilizando SVM, utilizamos la clase SVC de Python, utilizando los siguientes parámetros:

- C = Parámetro de penalización.
- Kernel = Kernel lineal (Maquinas de soporte vectorial con núcleos lineales)
- Random_state = semilla, valor 0.

Resultados - Matriz de confusión/Exactitud

A continuación, se muestran los resultados de la matriz de confusión y del valor que se tuvo como resultado de la variable accuracy.

Tabla 5.16

Resultado de matriz de confusión SVM

		Predicción	
		0	**1**
Valor Real	**0**	76359	0
	1	959	0

El modelo ha acertado 76359 de los casos de No compra, no obstante, no acertó con las ventas concretadas.

Exactitud

Resultado: 0.9875966786518017

Accuracy: El modelo tiene un 98.75% de exactitud de predecir el resultado.

3.- Support Vector Machine - Máquinas de soporte vectorial con núcleos no lineales (SVM Kernel).

El algoritmo de SVM Kernel igualmente busca resolver el problema de la separación entre grupos, pero considerando que realizará la clasificación con núcleos no lineales, considerando la separación de grupos y no utilizando una recta lineal (ya que no es factible dentro de una nube de puntos la separación lineal), buscando una transformación a una dimensión superior. El SVM Kernel se abre paso cuando no hay forma de encontrar un hiperplano que permita separar dos clases.

SVM Kernel Pasos:

1.- Ajuste del clasificador SVM: Se utilizó la librería sklearn.svm para importar el clasificador y se ajustaron los datos de entrenamiento (X_train, y_train).

```
# Ajustar el SVM en el Conjunto de Entrenamiento
from sklearn.svm import SVC
classifier = SVC(kernel = "rbf", random_state = 0)
classifier.fit(X_train, y_train)
```

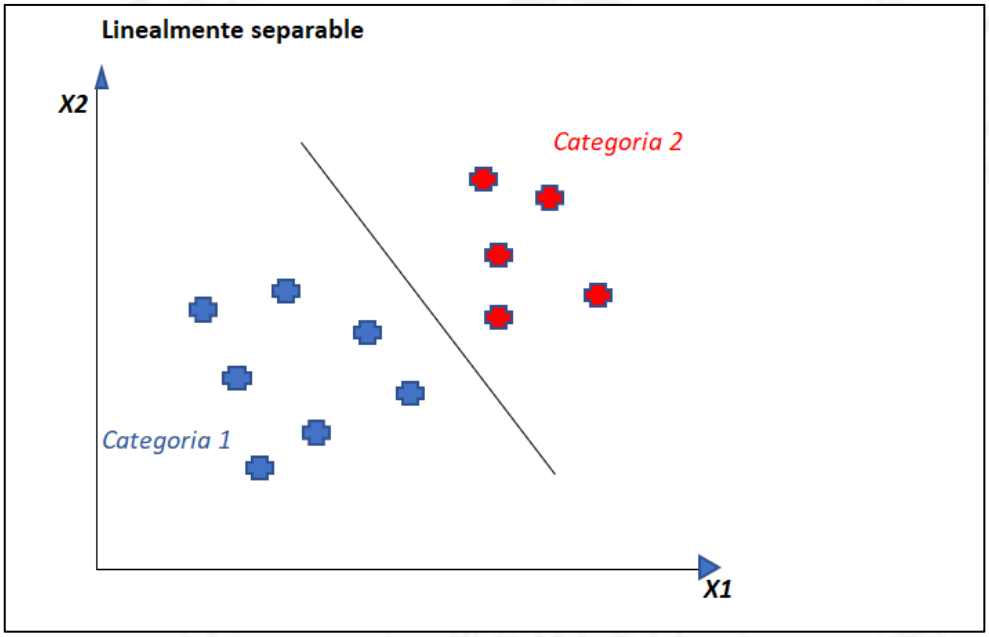
2.- Predicción de los resultados: Se realizó la predicción (y_pred) utilizando el clasificador ajustado y utilizando los datos de testing (X_test). A su vez, se realizó una comparación (accuracy) de los datos de la predicción (y_pred) vs. los datos reales (y_test).

```
# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)
```

A continuación, se presenta el gráfico que sirvió de base para nuestro modelo.

Figura 5.12

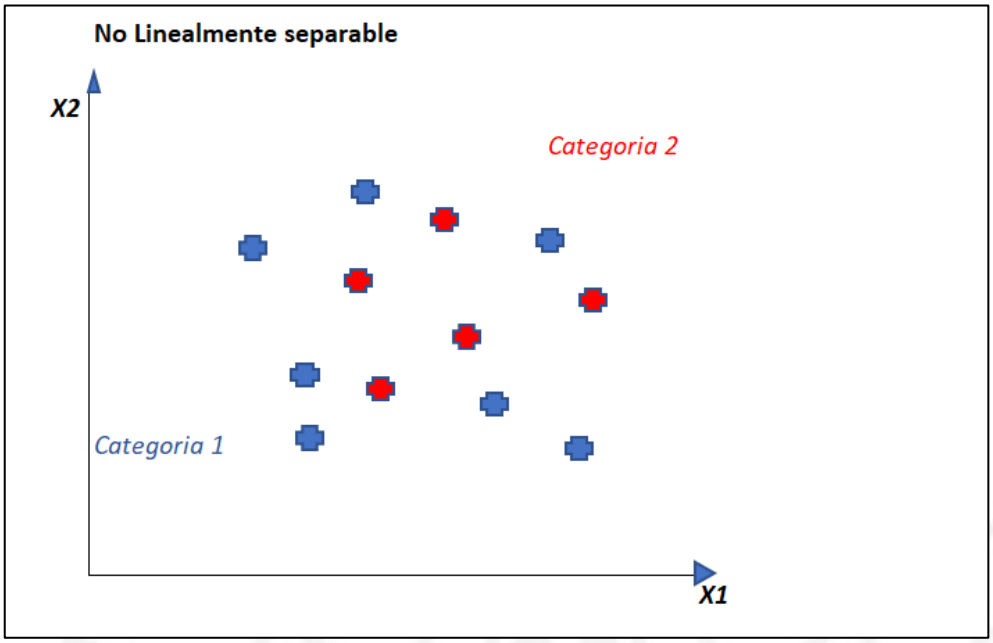
Descripción gráfica SVM Kernel linealmente separable



Nota. Adaptado de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Figura 5.13

Descripción gráfica SVM Kernel no linealmente separable



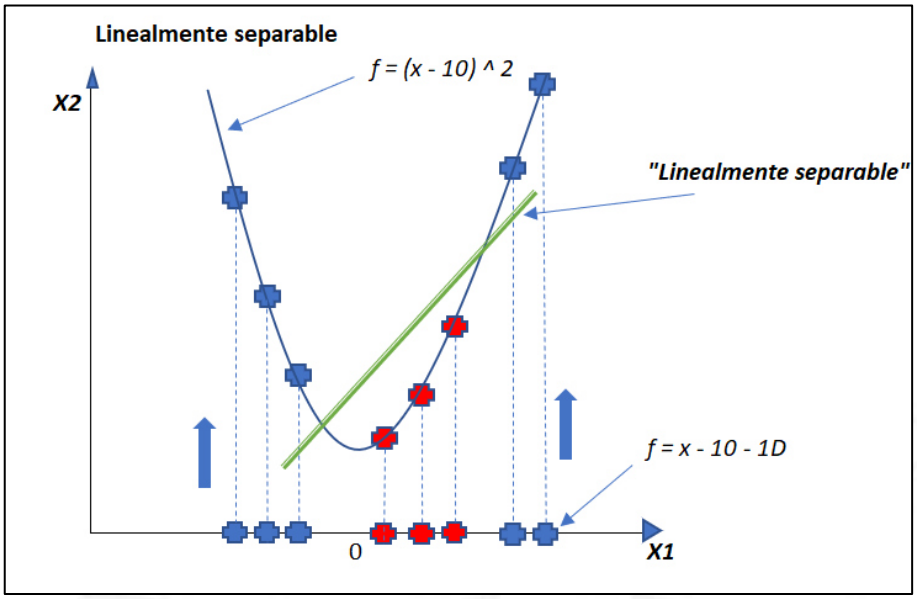
Nota. Adaptado de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

El límite de decisión debe determinar la forma de la separación óptima (esferas, elipses, etc.) estableciendo dimensiones superiores (Dimension1, Dimension2, Dimension3, etc.), para ello se realizó transformaciones a espacios de dimensiones superiores.

En una situación de un conjunto de datos en un plano, se debe elevar a una dimensión superior para lograr una separación óptima, construyendo una función que es capaz de separar los puntos y luego elevar, por ejemplo, al cuadrado la función original con el fin elevar a otra dimensión, pasando de una dimensión 1D, a una Dimensión 2D.

Figura 5.14

SVM Kernel transformación a una dimensión superior

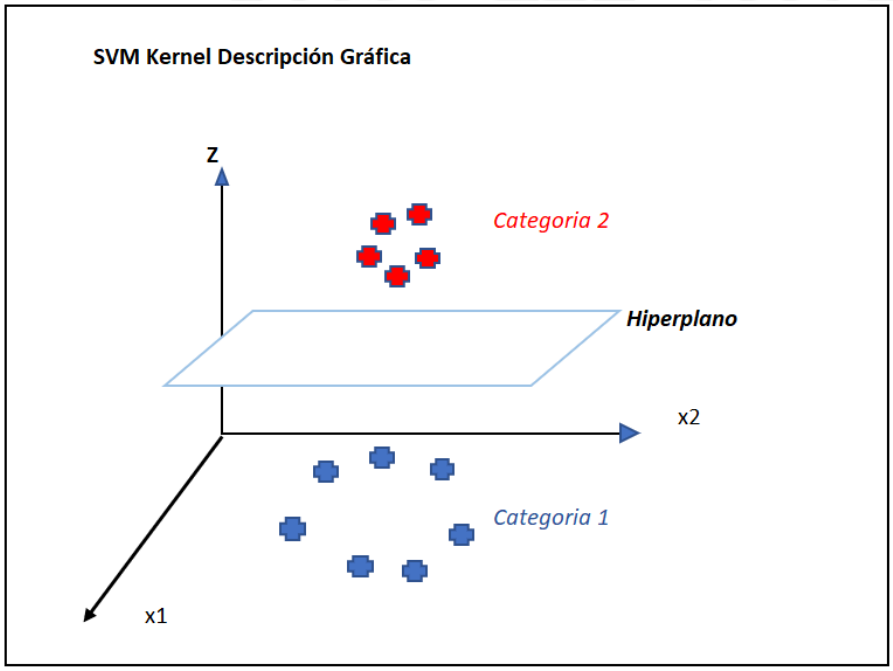


Nota. Adaptado de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Así como podemos transformar a una dimensión 2D, se realizará también una separación 3D para poder separar y transformar en dimensiones superiores.

Figura 5.15

Descripción gráfica de SVM Kernel



Nota. Adaptado de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

El problema de realizar la transformación a una variable en un espacio de dimensión superior es el alto costo computacional.

Funciones de Kernel:

Kernel Gaussiano RBF:

El Kernel Gaussiano permite separar mediante una función K (función de base radial), una campana de Gaus superior vs. la región aplanada del gráfico, dándole una forma separadora a ambas categorías.

Figura 5.16

Fórmula de kernel normal y simplificada

Kernel= 1

$$K\left(\vec{x}, \vec{l}^{\,i}\right) = e^{-\frac{\left\|\vec{x} - \vec{l}^{\,i}\right\|^2}{2\sigma^2}}$$

Kernel= 2

$$K\left(\vec{x}, \vec{l}^{\,1}\right) + K\left(\vec{x}, \vec{l}^{\,2}\right)$$

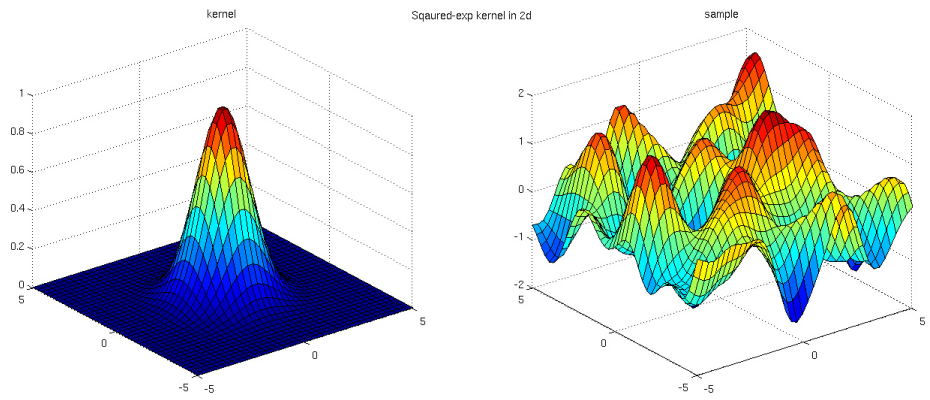
(Fórmula Simplificada)

Nota. De "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Así mismo, se muestran los diferentes diagramas de Kernel que se aplicaron a nuestro modelo.

Figura 5.17

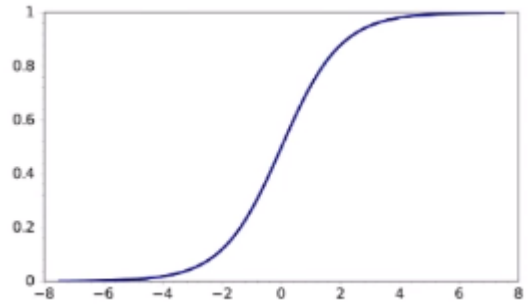
Muestra Kernel



Nota. De The Kernel Cookbook: Advice on Covariance functions, Duvenaud, David, 2014, (http://www.cs.toronto.edu/~duvenaud/cookbook/index.html)

Figura 5.18

Kernel sigmoide

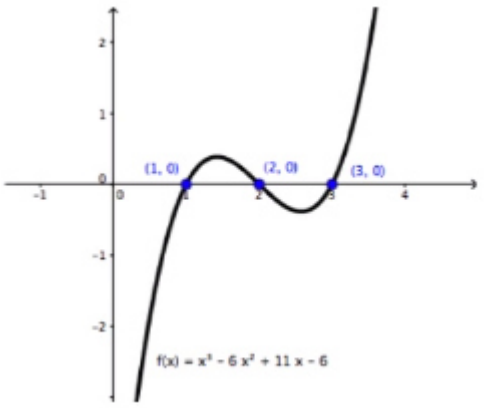


$$K(X,Y) = \tanh\left(\gamma \bullet X^T Y + r\right)$$

Nota. De The Kernel Cookbook: Advice on Covariance functions, Duvenaud, David, 2014, (http://www.cs.toronto.edu/~duvenaud/cookbook/index.html)

Figura 5.19

Kernel polinómico



$$K(X,Y) = \left(\gamma \bullet X^T Y + r\right)^d,\ \gamma > 0$$

Nota. De The Kernel Cookbook: Advice on Covariance functions, Duvenaud, David, 2014, (http://www.cs.toronto.edu/~duvenaud/cookbook/index.html)

Librería y parámetros

Para el modelo utilizando SVM, se utilizó la clase SVC de Python, con los siguientes parámetros:

- C = Parámetro de penalización.
- Kernel = Kernel rbf (Máquinas de soporte vectorial con núcleos no lineales)
- Random_state = semilla, valor 0.

Resultados - Matriz de confusión/Exactitud

A continuación, se muestran los resultados de la matriz de confusión y del valor que se tuvo como resultado de la variable accuracy.

Tabla 5.17

Resultado de matriz de confusión SVM Kernel

		Predicción	
		0	**1**
Valor Real	**0**	76359	0
	1	959	0

El modelo ha acertado 76359 de los casos de No compra, no obstante, no acertó con las ventas concretadas.

Exactitud

Resultado: 0.9875966786518017

Accuracy: El modelo tiene un 98.75% de exactitud de predecir el resultado.

4.- Naive Bayes - Teorema de Bayes

La idea de usar el teorema de Bayes en un problema de aprendizaje automático, es que podemos estimar las probabilidades a posteriori de cualquier hipótesis consistente con el conjunto de datos de entrenamiento para así escoger la hipótesis más probable (Malagón, 2003). El clasificador Naive Bayes utiliza al teorema de bayes como base para realizar el entrenamiento de datos, la cual expresa la probabilidad condicional de un evento aleatorio, vinculando la probabilidad de un evento A dado un evento B, con la probabilidad de un evento B dado A.

Figura 5.20

Fórmula teorema de Bayes

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)}$$

Nota. De "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Pasos para aplicar el Teorema de Bayes:

Paso 1: A dado B

- Donde P(A) es la probabilidad a priori: La probabilidad de que un cliente compre un producto financiero dentro del total de observaciones.
- P(B) es la probabilidad marginal: Número de observaciones similares (dentro del radio) entre total de observaciones.
- P(B|A) es la probabilidad condicionada: Número de observaciones similares entre los que compraron un producto financiero entre el total de observaciones que compraron el producto financiero.
- P(A|B) es la probabilidad de comprar un producto financiero sabiendo que cumple las características.

Paso 2: B dado A

- Donde P(A) es la probabilidad a priori: La probabilidad de que un cliente no compre un producto financiero dentro del total de observaciones.
- P(B) es la probabilidad marginal: Número de observaciones similares (dentro del radio) entre total de observaciones.
- P(B|A) es la probabilidad condicionada: Número de observaciones similares entre los que no compraron un producto financiero entre el total de observaciones que no compraron el producto financiero.
- P(A|B) es la probabilidad de no comprar un producto financiero sabiendo que cumple las características.

Finalmente se comparó ambos resultados, siendo la mayor probabilidad, la elegida para la asignación.

Naive Bayes Pasos:

1.- Ajuste del clasificador Naive Bayes: Se utilizó la librería sklearn.naive para importar el clasificador y se ajustaron los datos de entrenamiento (X_train, y_train).

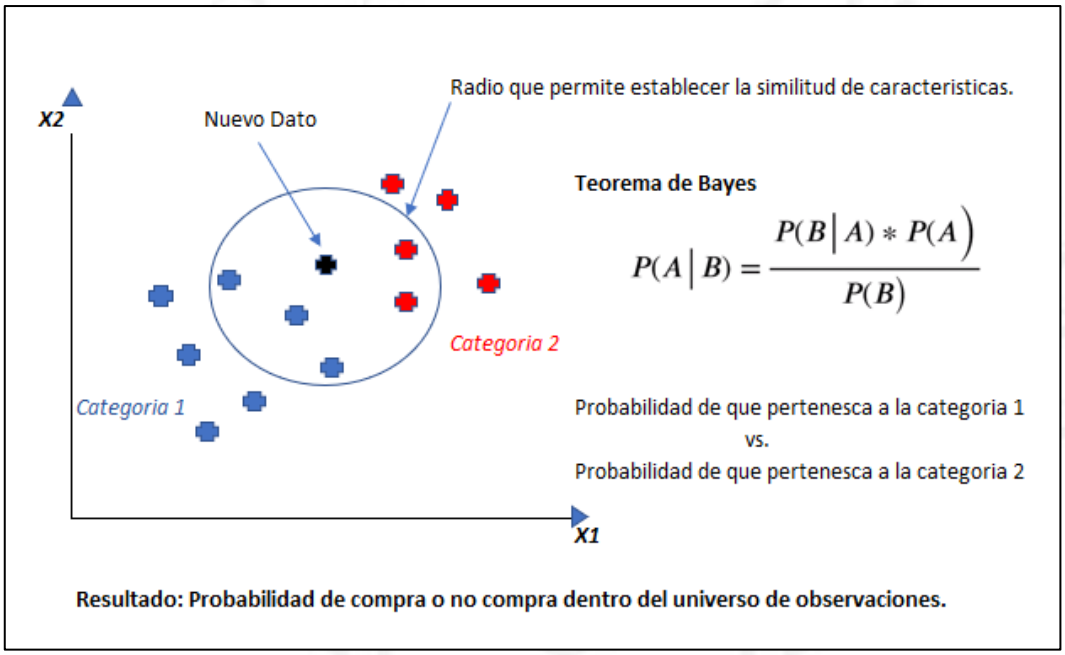
```
# Ajustar el clasificador en el Conjunto de Entrenamiento
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

2.- Predicción de los resultados: Se realizó la predicción (y_pred) utilizando el clasificador ajustado y utilizando los datos de testing (X_test). A su vez, se realizó una comparación (accuracy) de los datos de la predicción (y_pred) vs. los datos reales (y_test).

A continuación, se presenta el gráfico del modelo que sirvió de base para nuestro trabajo.

Figura 5.21

Descripción Naive Bayes



Nota. Adaptación de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Librería y parámetros

Para el modelo utilizando Naive Bayes, utilizamos la clase GaussianNB de Python.

Resultados - Matriz de confusión/Exactitud

A continuación, se muestran los resultados de la matriz de confusión y del valor que se tuvo como resultado de la variable accuracy.

Tabla 5.18

Resultado de matriz de confusión Naive Bayes

		Predicción	
		0	**1**
Valor Real	**0**	76283	76
	1	956	3

El modelo ha acertado 76283 de los casos de No compra, no obstante, acertó con 3 ventas concretadas.

Exactitud

Resultado: 0.9866525259318658

Accuracy: El modelo tiene un 98.66% de exactitud de predecir el resultado.

5.- Árboles de decisión.

El algoritmo de árboles de decisión realiza separaciones (splits) con el fin de agrupar observaciones dependiendo de la variable categórica (Y). Introduce el concepto de entropía cuyo cálculo permite obtener una medida de desorden y/o incertidumbre, siendo el objetivo minimizarla (medida de dispersidad de la información), buscando la calidad de las divisiones, escogiendo la mejor, siendo los nodos finales del árbol, los más homogéneos posibles.

Debido a que el algoritmo no utiliza distancias (euclídeas), durante la ejecución del algoritmo en Python, no se considerará el escalamiento de variables.

Árboles de decisión Pasos:

1.- Ajuste del clasificador árboles de decisión: Se utilizó la librería sklearn.tree para importar el clasificador y se ajustaron los datos de entrenamiento (X_train, y_train).

```
# Ajustar el clasificador de Árbol de Decisión en el Conjunto de
Entrenamiento
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = "entropy", random_state = 0)
classifier.fit(X_train, y_train)
```

2.- Predicción de los resultados: Se realizó la predicción (y_pred) utilizando el clasificador ajustado y utilizando los datos de testing (X_test). A su vez, se

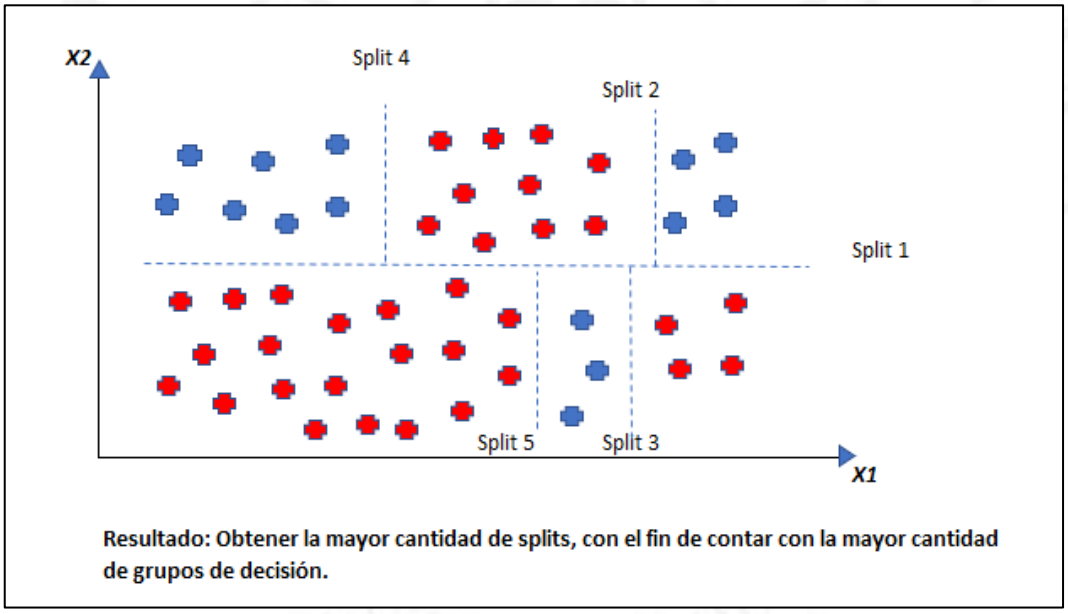
realizó una comparación (accuracy) de los datos de la predicción (y_pred) vs. los datos reales (y_test).

```
# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)
```

A continuación, se presenta el gráfico de árboles de decisiones.

Figura 5.22

Descripción árboles de decisión

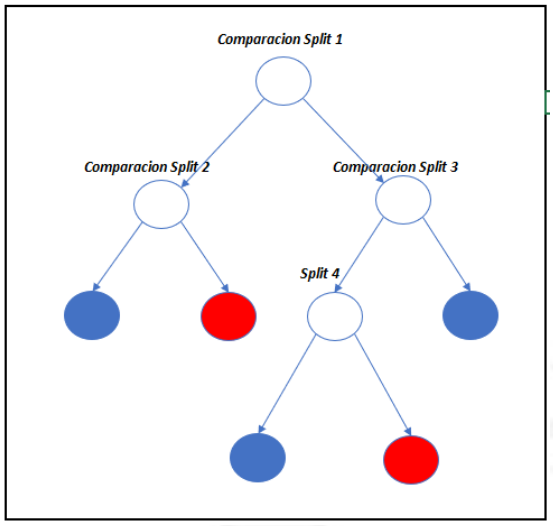


Nota. Adaptación de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Para lograrlo, el árbol de decisión va generando ramas según el valor de decisión, hasta encontrar un resultado (valores Y, compra o no compra).

Figura 5.23

Resultados del modelo de árboles de decisión



Nota. Adaptación de "Machine Learning de A a la Z: R y Python para Data Science", Gomila, Juan, et al., 2020 (https://www.udemy.com/course/machinelearning-es/learn/lecture/14060457#overview)

Librería y parámetros

Para el modelo utilizando árboles de decisión, se utilizó la clase DecisionTreeClassifier de Python, utilizando los siguientes parámetros:

- criterion = "entropy"
- random_state = 0

Resultados - Matriz de confusión/Exactitud

A continuación, se muestran los resultados de la matriz de confusión y del valor que se tuvo como resultado de la variable accuracy.

Tabla 5.19

Resultado de matriz de confusión resultado árboles de decisión

		Predicción	
		0	**1**
Valor Real	**0**	75359	1000
	1	847	112

El modelo ha acertado 75359 de los casos de No compra, no obstante, acertó con 112 de las ventas concretadas.

Exactitud

Resultado: 0.9761116428257327

Accuracy: El modelo tiene un 97.61% de exactitud de predecir el resultado.

6.- Random Forest

El algoritmo Random Forest, junta la potencia de diferentes algoritmos de machine learning (aprendizaje en conjunto), y realiza el voto por mayoría, para la selección.

Random Forest Pasos:

1.- Ajuste del clasificador Random Forest: Se utilizó la librería sklearn.ensemble para importar el clasificador y se ajustaron los datos de entrenamiento (X_train, y_train).

```
# Ajustar el clasificador  Random Forest en el Conjunto de Entrenamiento
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 1000, criterion =
"entropy", random_state = 0)
classifier.fit(X_train, y_train)
```

2.- Predicción de los resultados: Se realizó la predicción (y_pred) utilizando el clasificador ajustado y utilizando los datos de testing (X_test). A su vez, se realizó una comparación (accuracy) de los datos de la predicción (y_pred) vs. los datos reales (y_test).

```
# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)
```

Librería y parámetros

Para el modelo utilizando árboles de decisión, se utilizó la clase RandomForestClassifier de Python, utilizando los siguientes parámetros que según

nuestro modelo son los mas recomendables ya que por ejemplo n_estimators indicando un buen valor puede llevar a tener mejores resultados:

- n_estimators = 1000
- criterio= “entropy”
- random_state = 0

Resultados - Matriz de confusión/Exactitud

A continuación, se muestran los resultados de la matriz de confusión y del valor que se tuvo como resultado de la variable accuracy.

Tabla 5.20

Resultado de matriz de confusión Random Forest

		Predicción	
		0	**1**
Valor Real	**0**	76321	38
	1	906	53

El modelo ha acertado 76321 de los casos de No compra, no obstante, acertó con 53 de las ventas concretadas.

Exactitud

Resultado: 0.9877906826353501

Accuracy: El modelo tiene un 98.77% de exactitud de predecir el resultado.

7. Gradient Boosting Classifier

El algoritmo de clasificación Gradient Boosting (aumento de gradiente), parte de un modelo inicial el cual se convertirá en un conjunto aditivo, y cuyos errores se corrigen en diversas iteraciones a través de árboles de regresión que se encargan de corregir los errores de la etapa anterior (construyendo árboles en base a los errores del árbol previo). (StatQuest with Josh Starmer, 2019).

Gradient Boosting Classifier Pasos:

1.- Ajuste del clasificador Gradient Boosting Classifer: Se utilizó la librería sklearn.ensemble para importar el clasificador y se ajustaron los datos de entrenamiento (X_train, y_train).

```
# Ajustar el clasificador  Gradient Boosting en el Conjunto de Entrenamiento
from sklearn.ensemble import GradientBoostingClassifier
classifier = GradientBoostingClassifier(n_estimators=2000,random_state = 0)
classifier.fit(X_train, y_train)
```

2.- Predicción de los resultados: Se realizó la predicción (y_pred) utilizando el clasificador ajustado y utilizando los datos de testing (X_test). A su vez, se realizó una comparación (accuracy) de los datos de la predicción (y_pred) vs. los datos reales (y_test).

```
# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)
```

Librería y parámetros

Para el modelo utilizando el algoritmo de aumento de gradiente, se utilizó la clase GradientBoostingClassifier de Python, usando el siguiente parámetro:

- n_estimators = 2000

Resultados - Matriz de confusión/Exactitud

A continuación, se muestran los resultados de la matriz de confusión y del valor que se tuvo como resultado de la variable accuracy.

Tabla 5.21

Resultado de matriz de confusión Gradient Boosting Classifier

		Predicción	
		0	**1**
Valor Real	**0**	76332	27
	1	952	7

El modelo ha acertado 76332 de los casos de No compra, no obstante, acertó con 7 de las ventas concretadas.

Exactitud

Resultado: 0.9873380066737371

Accuracy: El modelo tiene un 98.73% de exactitud de predecir el resultado.

5.5 Fase de Evaluación

Luego de la ejecución de los modelos anteriormente descritos, obtuvimos los siguientes resultados:

Variables independientes:

- Cantidad de registros (Leads): 309,269
- Monto en soles: No se consideró, debido a que toda compra implica un monto depositado al cliente, y con ello se convierte en variable dependiente.
- Departamento: Lima
- Provincia: Lima
- Tea: Valor en porcentaje, ajustado en 4 decimales.
- Año/mes: Año 2020, de enero a julio.
- Oferta: Monto de la oferta del producto financiero en S/.
- Distritos: Todos.
- Productos: Todos
- Tipo de cliente: Todos.
- Rango de Edad: Todos.
- Consentimiento: Todos.
- Clúster: Todos.
- Iniciativa: Todos.

- Requerimiento de verificación domiciliaria y laboral: Todos.

Variable dependiente:

- Compra: Si o No.

Resultados:

Las métricas que se consideraron para el análisis son los siguientes:

1.- Matrix de Confusión: Permite la visualización del desempeño de algoritmos dentro de un esquema de aprendizaje supervisado.

2.- Accuracy: Nivel de precisión entre las predicciones correctas sobre el total de predicciones.

Accuracy: (Predicciones correctas) / (Número total de Predicciones).

3.- Curva ROC (Característica Operativa del Receptor): Permite establecer que tan probable es distinguir entre un resultado de la clasificación (Si/No).

4.- AUROC: Área total debajo de la curva ROC. Cuanto más cercano a 1 se encuentre, mejor serán los resultados.

A continuación, se mostrará los resultados de los diferentes algoritmos analizados en el punto 5.4 en donde la predicción de quien compra y quien no y el % de accuracy (nivel de precisión) del modelo revisado, así como de la gráfica de la curva Roc de cada modelo.

Para iniciar se mostrará los resultados del algoritmo KNN y su curva ROC.

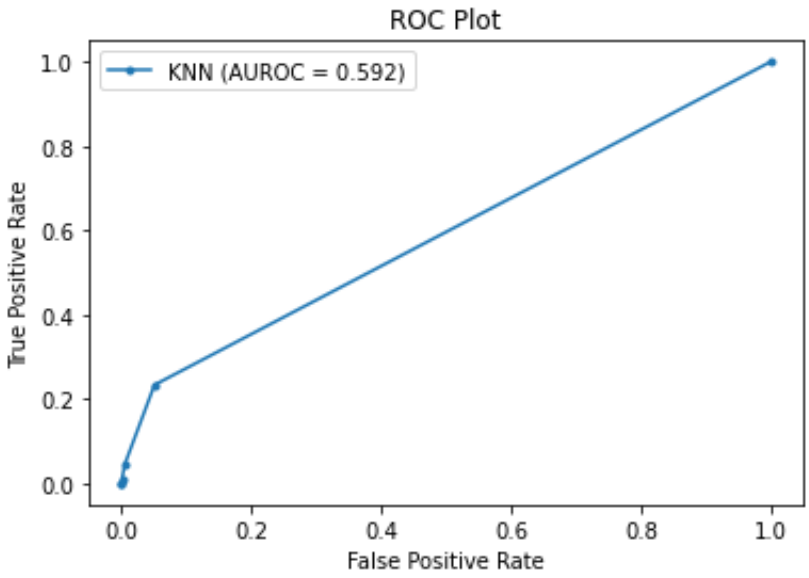
Tabla 5.22

Resultados del algoritmo KNN

Modelo 1	**KNN**			
		0	**1**	**Porcentaje**
	0	76333	26	99.93%
	1	951	8	0.83%
Accuracy	98.74%			

Figura 5.24

Curva ROC/AUROC del algoritmo KNN



A continuación, se presentan los resultados de los algoritmos SVM y SVM kernel, con los resultados de compra y no compra de los productos financieros.

Tabla 5.23

Resultados del algoritmo SVM

Modelo 2	**SVM**			
		0	**1**	**Porcentaje**
	0	76359	0	100.00%
	1	959	0	0.00%
Accuracy	98.76%			

Tabla 5.24

Resultados del algoritmo SVM Kernel

Modelo 3	**SVM Kernel**			
		0	**1**	**Porcentaje**
	0	76359	0	100.00%
	1	959	0	0.00%
Accuracy	98.76%			

A continuación, se presentan los resultados del algoritmo Naive Bayes, así como la gráfica de la curva Roc.

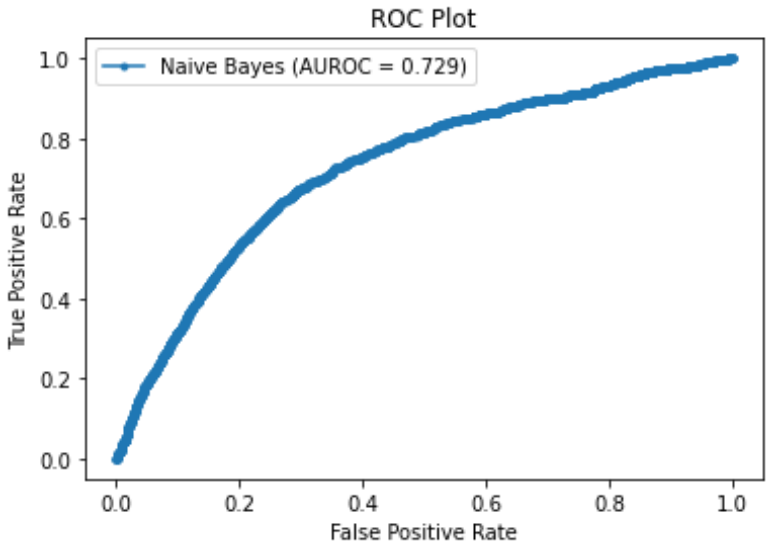
Tabla 5.25

Resultados del algoritmo Naive Bayes

Modelo 4	**Naive Bayes**			
		0	**1**	**Porcentaje**
	0	76283	76	99.80%
	1	956	3	0.31%
Accuracy	98.67%			

Figura 5.25

Curva ROC/AUROC del algoritmo Naive Bayes



A continuación, se presenta los resultados del algoritmo Tree Clasification, así como la gráfica de la curva Roc.

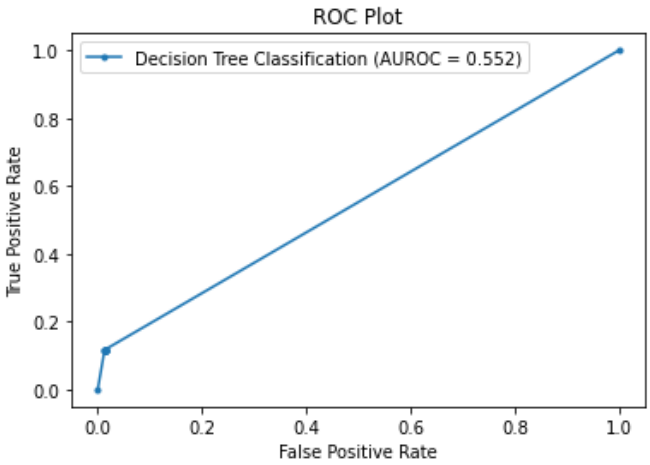
Tabla 5.26

Resultado de algoritmo Tree Clasification

Modelo 5	**Tree Clasification**			
		0	**1**	**Porcentaje**
	0	75359	1000	97.38%
	1	847	112	11.68%
Accuracy	97.61%			

Figura 5.26

Curva ROC/AUROC de algoritmo Tree Clasification



A continuación, se presentan los resultados del algoritmo Random Forest, así como la gráfica de la Curva Roc.

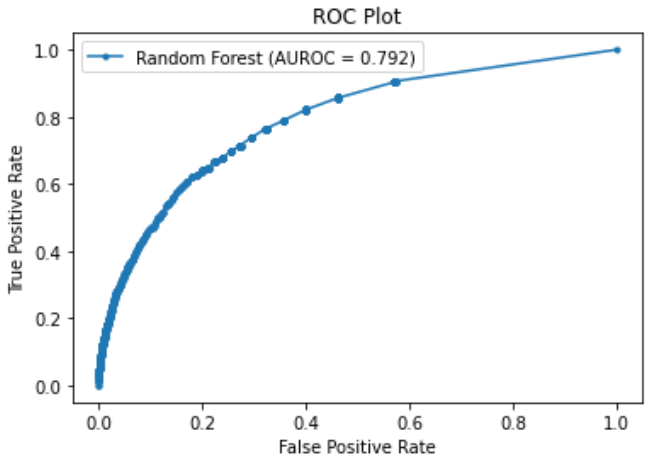
Tabla 5.27

Resultados de algoritmo Random Forest

Modelo 6	**Random Forest**			
		0	1	**Porcentaje**
	0	76321	38	99.90%
	1	906	53	5.53%
Accuracy	98.78%			

Figura 5.27

Curva ROC/AUROC de algoritmo Random Forest



A continuación, se presentan los resultados del algoritmo Random Forest, así como la gráfica de la Curva Roc.

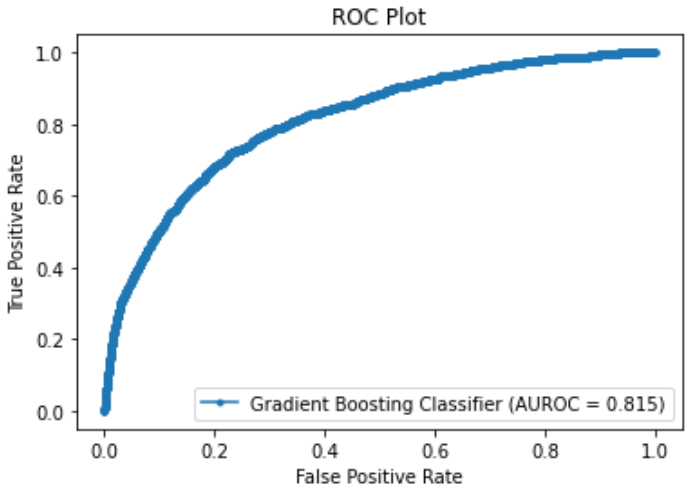
Tabla 5.28

Resultados de algoritmo Gradient Boosting Classifier

Modelo 7	Gradient Boosting Classifier			
		0	1	**Porcentaje**
	0	76332	27	99.96%
	1	952	7	0.73%
Accuracy	98.73%			

Figura 5.28

Curva ROC/AUROC de algoritmo Gradient Boosting Classifier



Según los resultados obtenidos en los datos de testing (representando 25% del total) presentamos las siguientes conclusiones:

1. Los modelos que obtuvieron valores de en la sección Real:1, Preditivo:1 fueron:

- KNN, con 8 aciertos y 951 falsos negativos.
- Naive Bayes, con 3 aciertos y 956 falsos negativos.
- Tree Classification, con 112 aciertos y 847 falsos negativos.
- Random Forest, con 53 aciertos y 906 falsos negativos.
- Gradient Boosting Classifier, con 7 aciertos y 952 falsos negativos.

2. Al realizar pruebas de entrenamiento con diferentes datasets, se decidió realizar el entrenamiento de los datos considerando todos los meses de estudios y con todos los distritos, todos los resultados obtenidos superan el 98% de Accuracy, lo cual confirma que los datos independientes tienen un alto grado de exactitud dentro de los modelos propuestos.

3. Los falsos negativos representan un nivel alto de incidencia, no obstante, existen aciertos con alto nivel de Accuracy, como en los casos de Naive Bayes (98.67%), Random Forest (98.78%) y Gradient Boosting Classifier (98.73%).
4. Según los resultados, seleccionamos 3 modelos para realizar la predicción para los siguientes meses del año 2020, no considerando datos repetitivos. Los modelos seleccionados son Naive Bayes, Random Forest y Gradient Boosting Classifier, debido a que cuentan con la mayor cantidad de aciertos (1:1) y que presentan un alto nivel de Accuracy, no obstante, debido a que el algoritmo Gradient Boosting Classifier obtuvo muy buenos resultados en la curva ROC, será el modelo para aplicar con prioridad. Con ello se proyectaron escenarios futuros de manera mensual en lo restante del año 2020.
5. La tendencia de registros (leads) por mes se consideró en 35,000 para los meses de setiembre, octubre, noviembre y diciembre 2020 (cantidad aproximada promedio ejecutada durante el inicio de la pandemia COVID-19).
6. La proyección de venta para el mes de setiembre 2020 (escenario propuesto), se estimó en base a un porcentaje de compra pesimista de 1.24% (promedio mensual actual) añadiendo la cantidad de compras proyectadas por los modelos (Ventas proyectadas Setiembre 2020) en base al porcentaje de efectividad de compra del escenario evaluado (escenario test enero – julio 2020), el cual representa el 25% de la información total.

Tabla 5.29

Predicción según algoritmo de árbol de decisiones

Tree Classification	***Escenario de Test (enero 2020 - julio 2020)***	***Porcentaje de aciertos***	***Escenario setiembre 2020***
Total, Leads	77,318		35,000
No compra	76,359		34,566
Si compra	959		434
Venta No predictiva	847		383
Aciertos Tree Classification	112	**11.68%**	**51**
Porcentaje de compra	1.24%		

Tabla 5.30

Predicción según algoritmo de árbol de Random forest

Random Forest	***Escenario de Test (enero 2020 - julio 2020)***	***Porcentaje de aciertos***	***Escenario setiembre 2020***
Total, Leads	77,318		35,000
No compra	76,359		34,566
Si compra	959		434
Venta No predictiva	906		410
Aciertos Random Forest	53	**5.53%**	**24**
Porcentaje de compra	1.24%		

7. Según los resultados del análisis de machine learning, en el mejor de los escenarios para el mes de setiembre 2020 se obtendrían 51 ventas predictivas sobre la ejecución del algoritmo de Tree Classification y 24 ventas predictivas de Random Forest, representando el 17% del total de ventas.
8. Dado que el escenario propuesto es recomendar campañas adicionales dentro del proceso de contactabilidad del call center, al tener un valor de entrada, los registros/leads de setiembre 2020 serán ingresados en ambos modelos seleccionados. Como Output obtendremos dos "Campañas IA", las cuales se evaluarán durante el mes de operación, utilizando el dashboard propuesto a nivel de frontend.

5.5.1 IBM Watson Studio

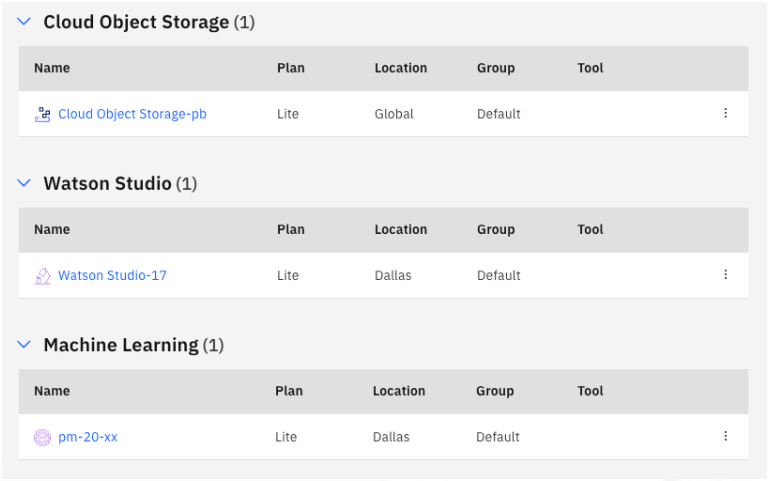
Adicionalmente al análisis realizado utilizando herramientas de Python (Anaconda IDE: Spyder) también utilizamos la herramienta IBM Watson Studio para validar nuestra hipótesis de selección de los algoritmos. Para este escenario utilizamos la mima información de entrada (Dataset con información de Leads de enero a julio 2020) dentro del entorno de Watson Studio.

Consideraciones:

Para realizar este análisis utilizamos 3 servicios de IBM Watson Studio (Plan Lite a setiembre 2020) los cuales nos permitió realizar el entrenamiento del modelo:

Figura 5.29

Servicios utilizados en IBM Watson Studio



Nota. De IBM Cloud, 2020 (https://cloud.ibm.com/)

A continuación, se procede a detallar cada uno de los servicios:

1. Watson Studio:
- Suite que proporciona un conjunto de herramientas y un entorno colaborativo para científicos de datos, desarrolladores y afines.
- 1 usuario autorizado.
- Capacidad de 50 unidades por hora mensuales.
- Entorno: Capacidad en unidades requeridas por hora.
2. Cloud Object Storage (COS):
- 1 instancia de servicio COS
- Almacenamiento de hasta 25GB/mes
- Hasta 2000 solicitudes GET/mes
- Hasta 2000 solicitudes PUT/mes
- Hasta 10GB de recuperación de datos/mes
- Hasta 5GB de salida pública.
3. Machine learning:
- Capacidad de 20 unidades por hora mensuales, en las cuales los modelos pueden entrenar, evaluar, implementar y calificar
- AutoAI: 8vCPU y 32GB RAM.
- Optimización de la decisión: 2 vCPu y 8 GB RAM.
- Integrable con Watson Studio.

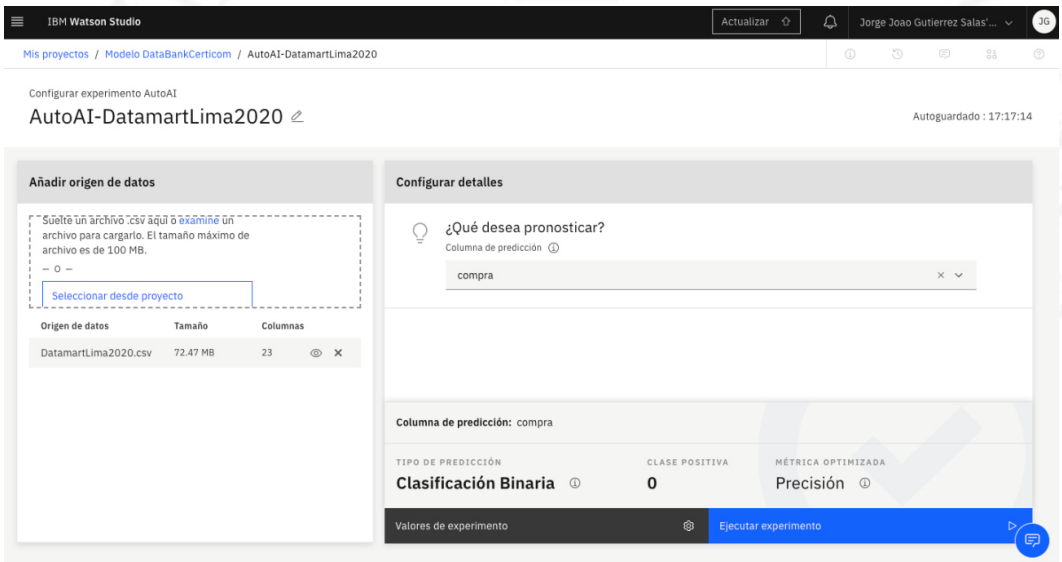
Utilizando Watson Studio creamos el proyecto Modelo DataBankCerticom, enfocado en el aprendizaje automatizado para la venta de productos financieros, de enero a julio 2020.

Configuración de Watson Studio

Para lograr el entrenamiento y modelo predictivo, ejecutamos el experimento AutoAI (método automatizado para crear un modelo de clasificación o regresión) llamado DatamartLima2020, y utilizando el archivo DatamartLima2020.csv de 72.47 MB como valor de entrada. El modelo obtiene la denominación de tipo de predicción de clasificación binaria, siendo el campo “Compra” la variable a predecir.

Figura 5.30

Selección de origen de datos en IBM Watson



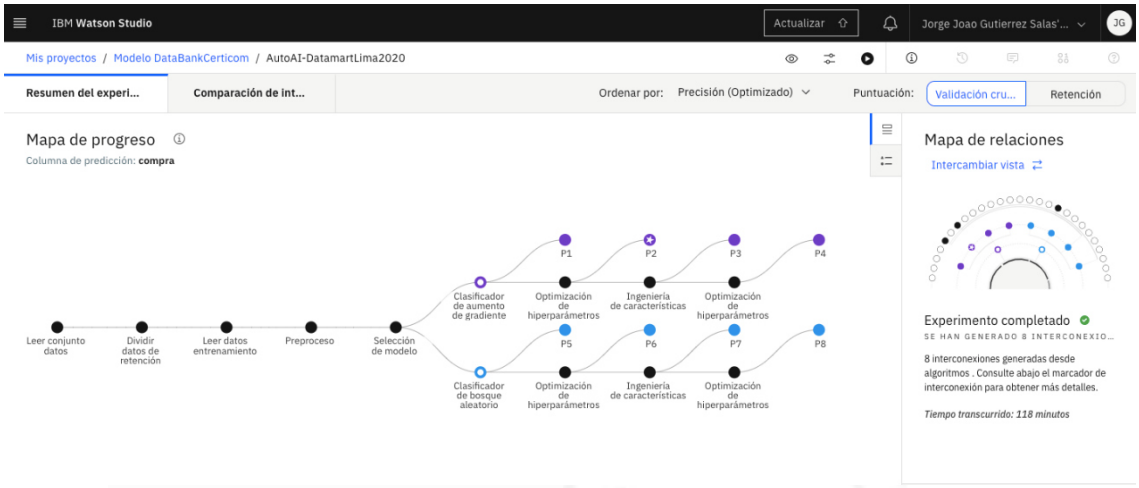
Nota. De IBM Cloud, 2020 (https://cloud.ibm.com/)

Luego de ingresar los valores de entrada, al ejecutar el experimento se visualizó el mapa de progreso del modelo, considerando los siguientes pasos:

- Leer un conjunto de datos.
- Dividir datos de retención (datos de entrenamiento y validación).
- Leer datos de entrenamiento.
- Pre-proceso.
- Escoger el modelo

Figura 5.31

Mapa de proceso en IBM Watson



Nota. De IBM Cloud, 2020 (https://cloud.ibm.com/)

Luego de la ejecución del entrenamiento, se obtuvieron 2 algoritmos con buenos resultados:

- Clasificador de aumento de gradiente.
- Clasificador de bosque aleatorio.

En ambos algoritmos, Watson realiza mejoras al modelo, utilizando optimización de hiper parámetros e ingeniería de características.

Figura 5.32

Resultados de los algoritmos desde IBM Watson

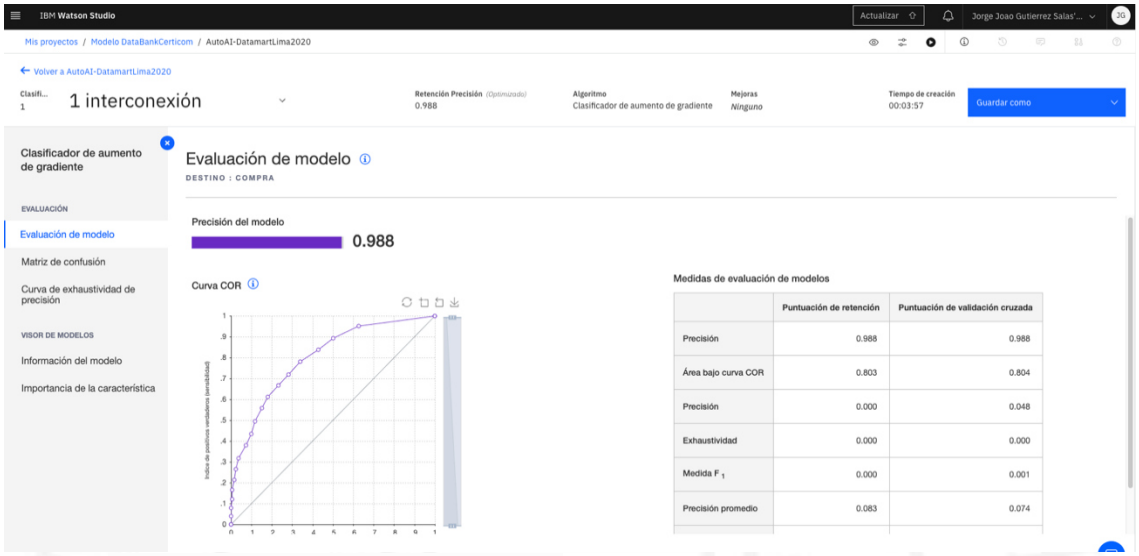
Marcador de interconexión

Clasificac...	Nombre	Algoritmo	ROC AUC	Mejoras	Tiempo de creación
★ 1	1 interconexión	Clasificador de aumento de gradiente	0.804	*Ninguno*	00:03:57
2	2 interconexión	Clasificador de aumento de gradiente	0.777	HPO-1	00:07:10
3	3 interconexión	Clasificador de aumento de gradiente	0.777	HPO-1 FE	00:27:28
4	4 interconexión	Clasificador de aumento de gradiente	0.777	HPO-1 FE HPO-2	00:13:51
5	7 interconexión	Clasificador de bosque aleatorio	0.626	HPO-1 FE	00:21:35
6	8 interconexión	Clasificador de bosque aleatorio	0.626	HPO-1 FE HPO-2	00:25:48
7	5 interconexión	Clasificador de bosque aleatorio	0.622	*Ninguno*	00:00:38
8	6 interconexión	Clasificador de bosque aleatorio	0.622	HPO-1	00:09:18

Nota. De IBM Cloud, 2020 (https://cloud.ibm.com/)

Como se puede apreciar los resultados de la curva ROC en el caso del clasificador de aumento de gradiente sobrepasa el valor de 0.7. IBM Studio también realiza un análisis de validación cruzada para obtener valores adicionales que contribuyen a una adecuada toma de decisiones.

Figura 5.33
Resultado de la curva ROC



Nota. De IBM Cloud, 2020 (https://cloud.ibm.com/)

En conclusión, y en base a los resultados de IBM Watson Studio, los algoritmos seleccionados para la muestra de registros de leads de la empresa CERTICOM (enero a julio 2020) son un clasificador de bosque aleatorio (Random forest classifier) y un clasificador de aumento de gradiente (Gradient boosting classifier), obteniendo el mismo resultado del modelo predictivo basado en Python (con todos los componentes y librerías utilizadas).

5.6 Fase de Construcción de Dashboard

En base a lo analizado en los modelos propuestos y con el fin de conseguir resultados que puedan predecir un comportamiento de compra, el escenario utilizando machine learning no será el único que se ejecutará en ambientes productivos. Actualmente el call center utiliza diversas campañas de venta para poder agrupar los leads, y con ello generar ventas.

El área de operaciones del call center agrupa a ciertos leads según criterios establecidos, como por ejemplo segmentación por edad, tasa de préstamo y

combinándolo con un análisis histórico. No obstante, no utiliza definiciones estructuradas para realizar esta segmentación, y se basan en las experiencias ganadas durante el paso del tiempo (aproximadamente el servicio préstamo de call centers a entidades financieras inició en el año 2010) del equipo de operaciones del servicio Call Center, incluso utilizando intuición.

Ejemplo:

Campaña X1: Priorizar por rango de edad y tasa del préstamo.

Campaña X2: Priorizar según el tipo de cliente (cliente nuevo, ex cliente, etc.)

Campaña X3: Priorizar por segmento de riesgo.

Campaña X4: Priorizar distritos, entre otros.

A las campañas utilizadas actualmente, se añadirán las campañas que fueron segmentadas en base a los modelos predictivos construidos, con el fin de medir su desempeño en comparación con las campañas convencionales. De esta forma el área de operaciones del call center y los stakeholders de la empresa CERTICOM podrán realizar un análisis de tendencias y de resultados de las ventas incorporando modelos de machine learning a la operación de contactabilidad y venta de productos financieros.

Indicadores

Se construyó los siguientes indicadores para lograr un monitoreo y análisis de los resultados de manera online, así como histórica.

Gráfico Venta por campaña

Permite comparar las campañas con respecto a la cantidad de ventas concretadas en un periodo determinado.

Figura 5.34

Prototipo de Venta por Campaña

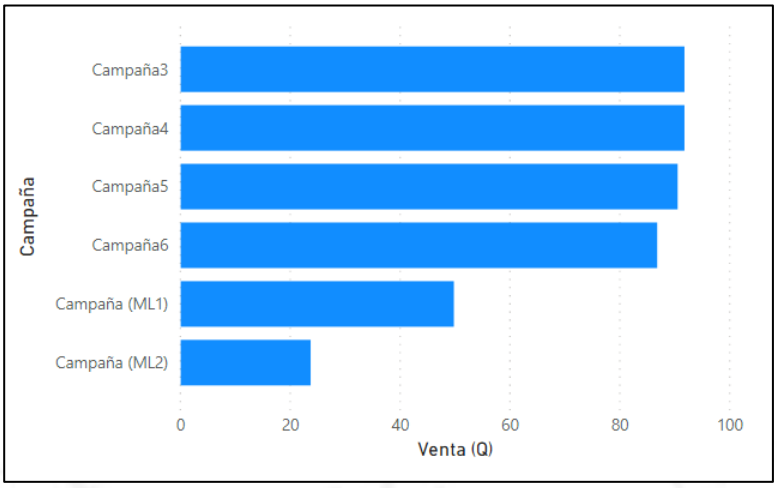


Gráfico Venta por Machine Learning

Permite visualizar la cantidad de ventas en porcentaje, realizada por las campañas basadas en machine learning vs el total de las campañas. Dado que se visualiza por fechas (meses) muestra la evolución del impacto de los modelos de machine learning en las ventas totales.

Figura 5.35

Prototipo de efectividad de venta por machine learning

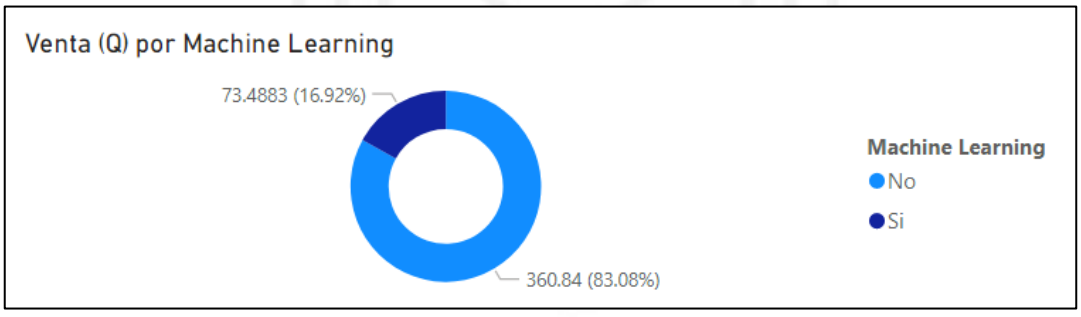


Gráfico Venta por Mes

Permite visualizar el aumento o la disminución de las ventas totales durante los meses seleccionados. Considera datos de todas las campañas.

Figura 5.36

Prototipo de venta por mes

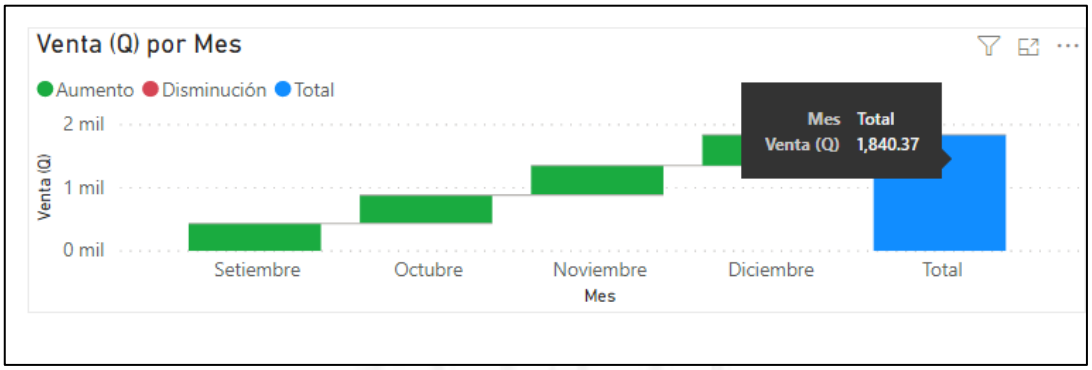


Tabla comparativa de leads/ventas por campaña

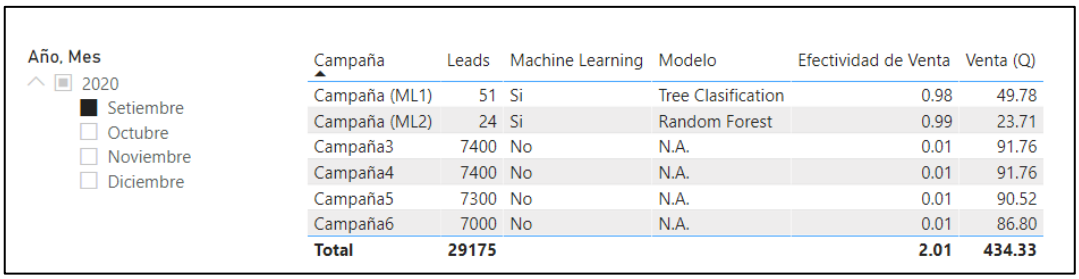
Permite comparar los resultados de todas las campañas activas, incluyendo las campañas basadas en machine learning.

Campos para considerar:

- Campaña: Nombre identificador de la campaña de venta.
- Leads: Cantidad de registros de oportunidad de venta, corresponde a la totalidad de contactos asignados a cada una de las campañas.
- Machine Learning: Indica si la campaña se basa en un modelo de machine learning o no.
- Modelo: Nombre del algoritmo de clasificación. Para las campañas convencionales esto no aplica.
- Efectividad de Venta: Indica la efectividad en porcentaje de las ventas alcanzadas en un periodo determinado, basadas en los leads asignados a cada campaña.
- Venta (Q): Cantidad de ventas concretadas en un periodo determinado.

Figura 5.37

Dashboard de resultado machine learning



Campaña	Leads	Machine Learning	Modelo	Efectividad de Venta	Venta (Q)
Campaña (ML1)	51	Si	Tree Clasification	0.98	49.78
Campaña (ML2)	24	Si	Random Forest	0.99	23.71
Campaña3	7400	No	N.A.	0.01	91.76
Campaña4	7400	No	N.A.	0.01	91.76
Campaña5	7300	No	N.A.	0.01	90.52
Campaña6	7000	No	N.A.	0.01	86.80
Total	**29175**			**2.01**	**434.33**

Not

Indicador de ventas totales

Permite visualizar el total de ventas en un periodo determinado, así como la contribución del modelo de machine learning.

Figura 5.38

Indicadores de ventas

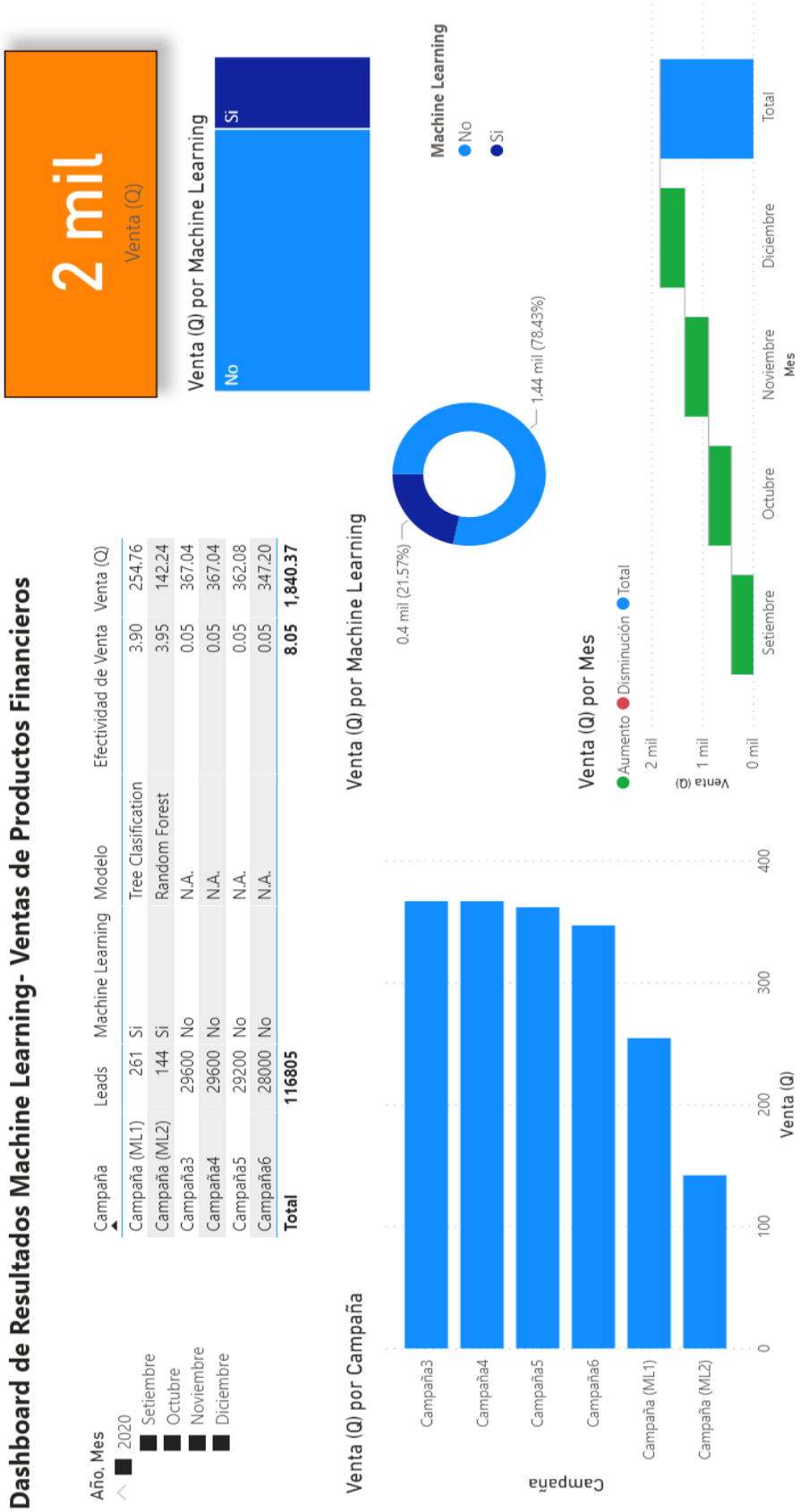
Dashboard de indicadores

El dashboard de resultados de Machine Learning vs. Ventas de productos financieros agrupa a todos los indicadores previamente mencionados. Esta interfaz podrá ser visualizada también en ordenadores (permitiendo su visualización en tableros de control) y dispositivos móviles.

La siguiente gráfica muestra el dashboard de resultados desde un ordenador.

Figura 5.39

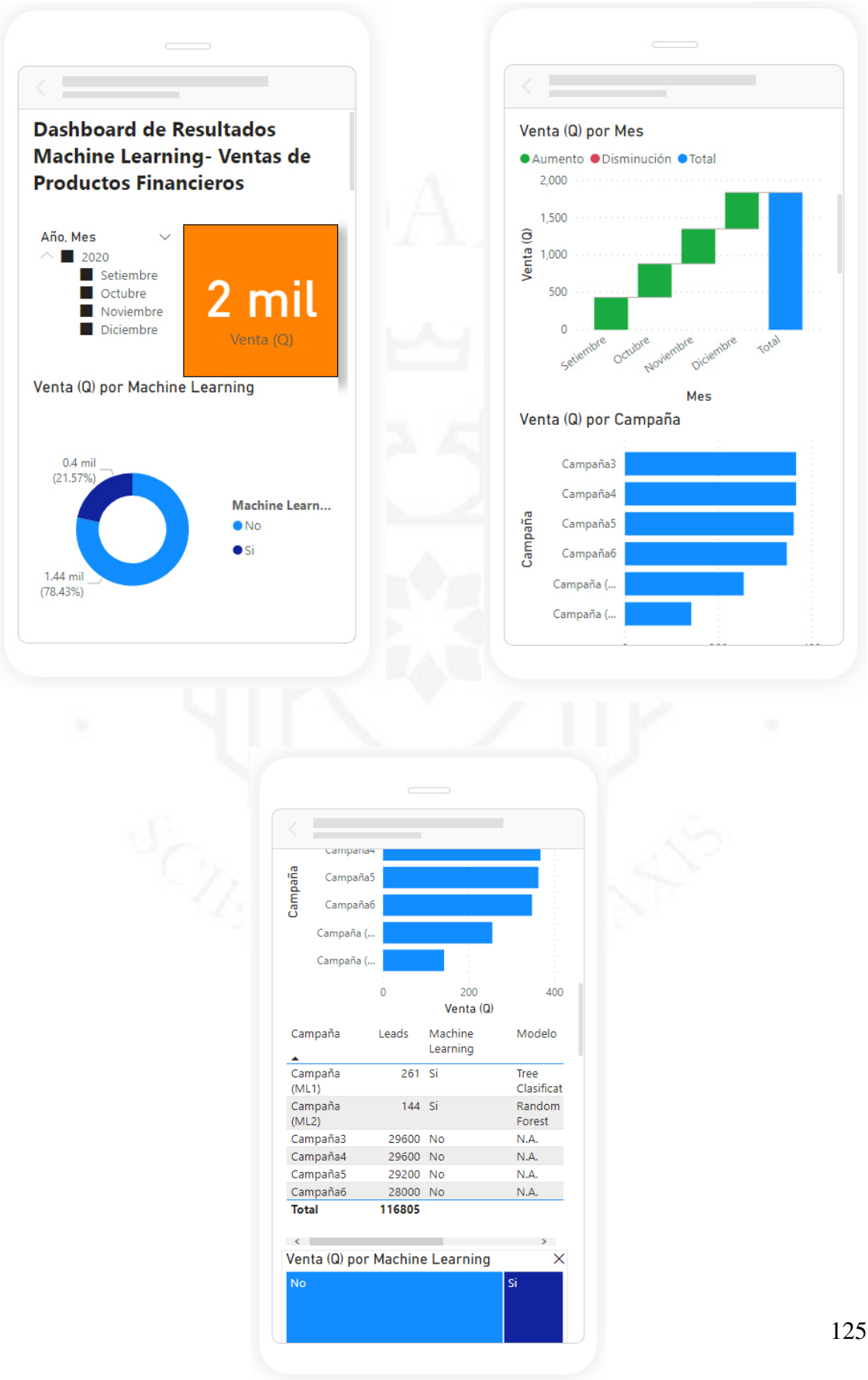
Dashboard de resultados de ventas de productos financieros



La siguiente gráfica muestra el dashboard de resultados desde un web en un dispositivo móvil.

Figura 5.40

Dashboard de resultados en versión Mobile



CONCLUSIONES

- Se afirma que los resultados esperados han sido alcanzados, ya que se cuenta con dos modelos de aprendizaje automatizado con niveles óptimos para realizar una predicción del comportamiento de compra del caso expuesto. Con estos modelos se podrá predecir un comportamiento de compra afirmativa (El cliente X, Si comprará) o negativa (El cliente Y, No comprará).
- Los resultados del aprendizaje automatizado utilizando las herramientas de machine learning provistas por el lenguaje de programación Python, y el dataset de clientes potenciales proporcionados por la entidad financiera del caso expuesto, dan como resultado dos algoritmos óptimos (Random forest classifier y Gradient boosting classifier) que dan sustento a una tentativa propuesta de implementación en producción de un modelo predictor de comportamiento de compra de productos financieros, según las métricas de evaluación realizadas.
- En cuanto a la evaluación de resultados en Python, los algoritmos de Random forest classifier y Gradient boosting classifier obtienen valores de Accuracy elevados, teniendo como input (potenciales clientes / leads) datos de los meses de enero a julio 2020. Accuracy Random Forest con 98.78%, y Gradient Boosting Classifier con 98.73%.
- En cuanto a la Curva ROC, métrica de evaluación de resultados con alto nivel de significancia y de toma de decisiones, el algoritmo Random forest alcanza un valor de 0.792 y Curva ROC de Gradient boosting classifier con un valor de 0.815
- Los resultados obtenidos de la herramienta IBM Watson Studio, (utilizando el mismo dataset de entrada) dan como resultado a los dos mejores modelos predictivos (según el motor predictivo de Watson) según la información proporcionada por la entidad financiera, a los algoritmos de Random forest y Gradient boosting.
- Los resultados obtenidos en IBM Watson Studio y en Python obtuvieron resultados muy similares (tanto Random forest y Gradient boosting)

confirmando que los dos modelos propuestos en el presente trabajo tienen fundamento sólido.

RECOMENDACIONES

- El aprendizaje automatizado realizado en el presente trabajo de suficiencia profesional, ha tomado como referencia la información proporcionada por la entidad financiera del caso expuesto, la cual cuenta con información necesaria para lograr la venta, sin embargo, pueden existir otros datos que pueden incorporarse dentro del modelo, con el fin de obtener mayor impacto en la variable de decisión de compra, como por ejemplo, situación actual en las centrales de riesgo, nivel de instrucción, sexo, entre otros. Por tanto, se recomienda realizar las gestiones correspondientes para que la empresa BPO pueda contar con información adicional con el fin de mejorar los resultados, implicando una negociación con la entidad financiera, con el fin de que esta última pueda proporcionar la información solicitada con nuevos campos, con el objetivo de incrementar las ventas.

- En cuanto a la cantidad información (registros) proporcionada y utilizada para la ejecución del modelo, se tienen datos de los meses de enero a julio 2020 en Lima Metropolitana. Se recomienda obtener más información histórica para realizar entrenamientos que conduzcan a mejores resultados. Igualmente se recomienda entablar una negociación entre la BPO Call Center y la entidad financiera.

- La información utilizada está sujeta a realizar segmentación de la información según la necesidad, por ejemplo, se puede realizar la ejecución del aprendizaje automatizado utilizando datamarts segmentando por distrito, por departamento, entre otros. De esta forma se podrían realizar modelos con segmentación.

- El presente modelo está sujeto a variables externas que pueden influenciar en la toma de decisiones de los potenciales clientes al aceptar o no aceptar un crédito, y con ello afectar las predicciones. La pandemia del COVID-19 del año 2020, es un factor externo muy importante para considerar. La recomendación es poder incorporar variables adicionales que permitan considerar variables externas, utilizando dataset de redes sociales, de bases de datos externas (SBS, centrales

de riesgo, etc.) e información relevante que puedan mejorar la calidad de los resultados predictivos.

- Los modelos de predicción resultantes cuentan con resultados diversos en la matriz de confusión, siendo los más destacados los algoritmos de Random forest classifier y Gradient boosting classifier, sin embargo, existen diversos algoritmos que puedan igualmente proporcionar modelos con altos niveles de accuracy, como las redes neuronales artificiales de clasificación (RNA) las cuales están en el campo del aprendizaje profundo (Deep Learning) o como el algoritmo XGBoost (algoritmo de aprendizaje supervisado el cual es muy veloz a nivel de procesamiento y sigue el principio del aumento de gradiente). Para futuras evaluaciones, se recomienda utilizar ambos algoritmos enriqueciendo las opciones de modelamiento predictivo.

- Debido a que la información proporcionada por el banco evidencia un alto porcentaje de leads de compra fallidos, en comparación con los registros que si obtuvieron resultados positivos, se recomienda realizar estrategias de re-muestreo para conjuntos de datos desequilibrados, por ejemplo utilizando técnicas de Undersampling (tomando muestras de la clase mayoritaria, igualando a la clase minoritaria) y técnicas de Oversampling (copiando muestras de la clase minoritaria para igualar a la clase mayoritaria), y poder utilizarlas siempre y cuando se evite el sobreajuste o la pérdida de datos.

- Un beneficio esperado es poder ofrecer otros servicios orientados a la inteligencia artificial, considerando que el negocio de Call Center tiene una amplia gama de procesos, que brindan una gran oportunidad de incorporar tecnologías emergentes, tales como el procesamiento de lenguaje natural (NPL), el cual puede ser utilizado para gestionar las grabaciones de voz almacenadas, y evaluar posibles patrones de comportamiento tanto por parte del cliente final como en la atención de los promotores.

GLOSARIO DE TÉRMINOS

Accuracy: Medida que se usa para determinar qué modelo de machine learning es el mejor para identificar relaciones y patrones entre variables en un conjunto de datos en función de los datos de entrada o de entrenamiento.

Algoritmo: Serie de pasos ordenados para efectuar una tarea, con ellos podemos obtener la información que necesitamos para tomar decisiones o predecir el comportamiento de los datos.

Backoffice: Conjunto de actividades que se realizan como apoyo al negocio y que no tiene contacto directo con el cliente.

Big Data: Es un gran volumen de datos que se puede analizar para tomar decisiones y movimientos estratégicos en las empresas.

Business Analytics: Conjunto de herramientas capaces de analizar las interacciones, provee de información a la empresa sobre tendencias y comportamientos de consumo, para poder alcanzar las metas del negocio.

Business Process Outsourcing: Subcontratación de funciones de un proceso de negocio en proveedores de servicios, ya sea internos o externos a la empresa, que son menos costosos o más eficientes y eficaces.

Call Center: Centro de operación donde se realizan o reciben llamadas para ofrecer o brindar un servicio al consumidor.

Científicos de Datos: Son expertos en datos analíticos que tienen habilidades técnicas para resolver problemas complejos.

Crisp-DM: Metodología que describe enfoques comunes en los expertos de minería de datos.

Curva ROC: Método estadístico que permite determinar la exactitud de las pruebas, determina el punto de corte de una escala continua en el que se alcanza la sensibilidad y especificidad más alta, evaluar la capacidad discriminativa de la prueba de diagnóstico, y comparar la capacidad discriminativa de dos o más pruebas.

Dashboard: Interfaz gráfica de usuario que presenta vistas de los indicadores clave de rendimiento para un objetivo particular o proceso de negocio.

Datamart: Versión especial de almacén de datos (data warehouse), son subconjuntos de datos con el propósito de ayudar a que un área específica.

Dataset: Representación de datos residente en memoria que proporciona un modelo de programación relacional coherente independientemente del origen de datos.

Datawarehouse: Almacén electrónico donde una empresa mantiene una gran cantidad de información de forma segura, fiable, fácil de recuperar y de administrar**.**

Deep Learning: Conjunto de algoritmos de machine learning que modela abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos.

ETL: Proceso de compilación de datos a partir de un número ilimitado de fuentes, posterior organización y centralización en un único repositorio.

Insigths: Visión interna o percepción, son aspectos que están ocultos en la mente del consumidor, hacen referencia a la motivación profunda hacia una marca o producto.

Matriz de Confusión: Herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado.

Pymes: Empresa pequeña o mediana en cuanto a volumen de ingresos, valor del patrimonio y número trabajadores.

Opensource: Software cuyo código fuente y otros derechos que normalmente son exclusivos para quienes poseen los derechos de autor, son publicados bajo una licencia de código abierto o forman parte del dominio público.

Servicio de Outsourcing: Gestión o la ejecución permanente de una función empresarial por un proveedor externo de servicios.

Sprint: Ciclos o iteraciones que se tiene dentro de un proyecto Scrum, van a permitir tener un ritmo de trabajo con un tiempo prefijado.

Vicidial: Software de centro de contacto de código abierto con capacidades de marcador predictivo. Maneja llamadas entrantes, salientes y combinadas.

REFERENCIAS

Aldea Digital. (2020). 5 puntos para entender a los nuevos consumidores, *Call Center News*. https://www.callcenternews.com.ar/aldea-digital/1597-5pnc

Barredo, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI [Inteligencia artificial explicable (XAI): conceptos, taxonomías, oportunidades y desafíos hacia una IA responsable]. *Information Fusion*, 58, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Basile, Elsa, (2020). Tendencias en Customer Experience, la evolución de calidad a CX, *Call Center News*. https://www.callcenternews.com.ar/management/1493-tcsx

Duvenaud, David. (2014) *Automatic model construction with Gaussian processes* [Construcción automática de modelos con procesos gaussianos], [Tesis de doctorado] Universidad de Cambridge. https://doi.org/10.17863/CAM.14087

Géron, Aurélien (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* [Aprendizaje automático práctico con Scikit-Learn, Keras y TensorFlow: conceptos, herramientas y técnicas para construir sistemas inteligentes], Sebastopol: O'Reilly Media, 2nd Edition.

Gestión, Management & Empleo. (2018). Centros de contacto: Informalidad afecta a 15,000 empleados, ¿qué plantea Apexo? *Noticias Gestión.* https://gestion.pe/economia/management-empleo/centros-contacto-informalidad-afecta-15-000-empleados-plantea-apexo-242758-noticia/

Goasduff, Laurence. (2019). *Top Trends on the Gartner Hype Cycle for Artificial Intelligence, 2019* [Principales tendencias en el ciclo de Gartner Hype para la inteligencia artificial, 2019], Smarter With Gartner. https://www.gartner.com/smarterwithgartner/top-trends-on-the-gartner-hype-cycle-for-artificial-intelligence-2019

Gomila, J., De Ponteves, H., Eremenko, K., & Super Data Science Team (2020). *Aprende a crear algoritmos de Machine Learning en Python y R con expertos en Data Science* [Video] https://www.udemy.com/course/machinelearning-es/

Gonzáles Fernandéz, A. (2016). Big data para el análisis de las necesidades traductológicas en cinco capitales de Europa, *Skopos: revista internacional de traducción e interpretación*, Nº. 7, 99-128

IBM Cloud. (2020). https://cloud.ibm.com/

Japkowicz, Nathalie. (2006). *Why Question Machine Learning Evaluation Methods?* [¿Por qué cuestionar los métodos de evaluación de Machine Learning?], School of Information Technology and Engineering University of Ottawa

Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., & Díaz-Rodríguez, N. (2020). Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges [Aprendizaje continuo para robótica: definición, marco, estrategias de aprendizaje, oportunidades y desafíos]. *Information Fusion,* 58, 52-68. https://doi.org/10.1016/j.inffus.2019.12.004

Lope, V., Mamaqi, X., & Vidal, J. (2020). La Inteligencia Artificial: desafíos teóricos, formativos y comunicativos de la datificación, *Icono 14*, 18 (1), 58-88. https://doi.org/10.7195/ri14.v18i1.1434

Malagón, Constantino. (2003). Clasificadores Bayesianos. El algoritmo Naive Bayes.

Manrique, Esperanza. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para Desarrollo, *Revista Ibérica de Sistemas e Tecnologías de Información*; Lousada N.º E28, 586-599

Maximixe, PromPerú. (2010*). Plan Estratégico y Operativo del Sector Contact Center en el Perú, 2010*
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi5oZ-FiKLqAhUEGLkGHbqkDP8QFjACegQIARAB&url=http%3A%2F%2Fwww.siicex.gob.pe%2Fsiicex%2Fdocumentosportal%2F464948877radF4FC7.pdf&usg=AOvVaw0DDuRaY4cPbXTXdlvY_4Qc

Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014) The Evolution of Boosting algorithms from Machine Learning to Statistical Modelling [La evolución de Boosting algoritmos desde Machine Learning hasta el modelado estadístico]. *Methods of Information in Medicine,* 53(6): 419–427, http://dx.doi.org/10.3414/ME13-01-0122

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial [Máquinas de Gradient boosting, un tutorial]. *Front. Neurorobot. 7:2,* https://doi.org/10.3389/fnbot.2013.00021

NBD Noticias, BBVA. (2019). ¿Qué es la IA explicable (XAI) y por qué es más necesaria que nunca? *New Digital Businesses*
https://www.bbva.com/ndb/es/articulo/que-es-la-ia-explicable-xai-y-por-que-es-mas-necesaria-que-nunca/#

Pallarés, Alvaro. (2019). *Aplicación y comparación de modelos de machine learning destinados a la puntuación del riesgo de crédito* [Trabajo Fin de Máster]. Universidad Complutense, Madrid.

Reis, I., Baron, D., & Shahaf, S. (2019). Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Data Sets [Probabilístico Radom Forest: Un algorítmo de Machine Learning para conjunto de datos ruidosos]. *The Astronomical Journal,* 157:16*,* https://doi.org/10.3847/1538-3881/aaf101

Scrum.org. (2020). *What is scrum?* https://www.scrum.org/resources/what-is-scrum

Segal, Mark. (2004). Machine Learning Benchmarks and Random Forest Regression [Puntos de Referencia en Machine Learning y Regresion Random Forest]. *UCSF: Center for Bioinformatics and Molecular Biostatistics,* https://escholarship.org/uc/item/35x3v9t4

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011). Real-Time Human Pose Recognition in Parts from Single Depth Images [Reconocimiento de pose humana en tiempo real en partes a partir de imágenes únicas de profundidad], *Best Paper Award*, Published by IEEE, CVPR*,* Providence, 1297-1304, https://doi.org/10.1109/CVPR.2011.5995316

StatQuest with Josh Starmer. (2019). *Gradient Boost Part 4: Classification Details* [Video] https://www.youtube.com/watch?v=StWY5QWMXCw

Superintendencia de Banca y Seguros. (2020). "*Reportes del Sistema Financiero – Presentación del Sistema Financiero*". https://intranet2.sbs.gob.pe/estadistica/financiera/2020/Octubre/SF-0003-oc2020.PDF

Wassouf, W., Alkhatib, R., Salloum, K., & Balloul, S. (2020). Predictive analytics using big data for increased customer loyalty [Análisis predictivo con big data para aumentar la fidelidad de los clientes]. *Syriatel Telecom Company case study. J Big Data* **7,** 29. https://doi.org/10.1186/s40537-020-00290-0

BIBLIOGRAFÍA

Carnegie, Dale (2012) *Maestría en Liderazgo.* Argentina: Penguin Random House.

Davenport, Thomas H., & Harris, Jeanne G. (2007). Computing on Analytics. Massachusettts: Harvard Business School.

Edwards, Benjamin (2020). *Marketing en movimiento: Como huir del equilibrio y el reposo creando valor*. Lima: Penguin Random House.

Fischman, David (2015). *El líder Tranformador I 1ra Edición*. Lima: Planeta.

Fischman, David (2015). *El líder Tranformador II 1ra Edición*. Lima: Planeta.

Fischman, David (2017). *El camino del líder*. Lima: Planeta.

Fischman, David (2019). *Motivación 360 1ra Edición*. Lima: Planeta.

Font Barrot, Alfred (2013). *Las 12 leyes de la negociación: O eres estratega o eres ingenuo.* España: Penguin Random House.

Grus, Joel (2019). *Data Science from Scratch: First Principles with Python 2nd Edition.* Sebastopol: O'Reilly Media.

Hastie, T., Tibshirani R., & Friedman, J (2016). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer; 2nd Edición

Hill, Charles, & Jones, Gareth R. (2011). *Administración estratégica: Un enfoque integral.* Cuauhtémoc: Editorial Progreso.

Levine, David M., Krehbiel, Timothy C. & Berenson, Mark L. (2014). *Estadística para administració*n *6ta Edición*. México: Pearson Educación.

Project Management Institute (2013). Guía de los fundamentos para la dirección de proyectos (guía del PMBOK) 5ta Edición. Pensilvania:PMI

Quezada Lucio, Neil (2014), *Estadística con SPSS 22*. Lima: Macro.

Quiñones, Cristina (2019). *Estrategias con calle: Insights y tendencias del consumo para la transformación Cultural.* Lima: Editorial Planeta Perú.

Shalev-Shwartz, Shai & Ben-David Shai (2014). *Understanding Machine Learning: From theory to algorithms.* Cambridge University Press; 1st Edición

VanderPlas, Jake (2017). *Python Data Science Handbook: Essential Tools for working with Data.* Sebastopol: O'Reilly Media 1st Edition.

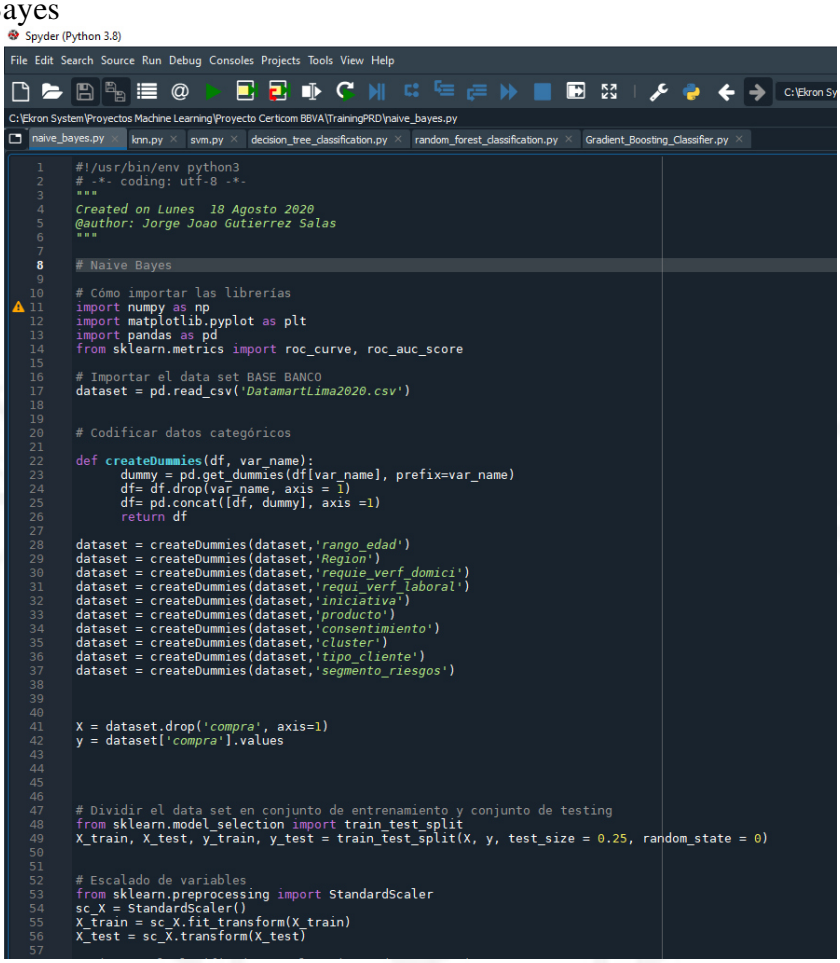
Varela V., Rodrigo (2008*). Innovación Empresarial: Arte y ciencia en la creación de empresas 3ra Edición.* Colombia: Pearson Educación.

ANEXOS

Anexo 1: Pantallas de Python del modelo de Machine Learning

A continuación, se mostrarán las pantallas de Python que evidencian los modelos y algoritmos desarrollados:

1.- Naive Bayes



```
# Ajustar el clasificador en el Conjunto de Entrenamiento
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)

# Prediccion de probabilidad - Elaborar Curva ROC
r_probs = [0 for _ in range(len(y_test))]
nb_probs = classifier.predict_proba(X_test)
nb_probs = nb_probs[:, 1]

r_auc = roc_auc_score(y_test, r_probs)
nb_auc = roc_auc_score(y_test, nb_probs)

print('Naive Bayes: AUROC = %.3f' % (nb_auc))

nb_fpr, nb_tpr, _ = roc_curve(y_test, nb_probs)


plt.plot(nb_fpr, nb_tpr, marker='.', label='Naive Bayes (AUROC = %0.3f)' % nb_auc)

# Title
plt.title('ROC Plot')
# Axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Show legend
plt.legend() #
# Show plot
plt.show()
```

2.- KNN

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Ekron System\Proyectos Machine Learning\Proyecto Certicom BBVA\TrainingPRD\knn.py

naive_bayes.py | knn.py | svm.py | decision_tree_classification.py | random_forest_classification.py | Gradient_Boosting_Classifier.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Lunes  18 Agosto 2020
@author: Jorge Joao Gutierrez Salas
"""

# K - Nearest Neighbors (K-NN)


# Cómo importar las librerias
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import roc_curve, roc_auc_score

# Importar el data set BASE BANCO
dataset = pd.read_csv('DatamartLima2020.csv')



# Codificar datos categóricos

def createDummies(df, var_name):
      dummy = pd.get_dummies(df[var_name], prefix=var_name)
      df= df.drop(var_name, axis = 1)
      df= pd.concat([df, dummy], axis =1)
      return df

dataset = createDummies(dataset,'rango_edad')
dataset = createDummies(dataset,'Region')
dataset = createDummies(dataset,'requie_verf_domici')
dataset = createDummies(dataset,'requi_verf_laboral')
dataset = createDummies(dataset,'iniciativa')
dataset = createDummies(dataset,'producto')
dataset = createDummies(dataset,'consentimiento')
dataset = createDummies(dataset,'cluster')
dataset = createDummies(dataset,'tipo_cliente')
dataset = createDummies(dataset,'segmento_riesgos')



X = dataset.drop('compra', axis=1)
y = dataset['compra'].values



# Dividir el data set en conjunto de entrenamiento y conjunto de testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)


# Escalado de variables
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

```
58
59
60  # Ajustar el clasificador en el Conjunto de Entrenamiento
61  from sklearn.neighbors import KNeighborsClassifier
62  classifier = KNeighborsClassifier(n_neighbors = 5, metric = "minkowski", p = 2)
63  classifier.fit(X_train, y_train)
64
65  # Predicción de los resultados con el Conjunto de Testing
66  y_pred  = classifier.predict(X_test)
67
68  # Elaborar una matriz de confusión
69  from sklearn.metrics import confusion_matrix
70  cm = confusion_matrix(y_test, y_pred)
71
72  accuracy= classifier.score(X_test, y_test)
73
74
75
76  # Prediccion de probabilidad - Elaborar Curva ROC
77  r_probs = [0 for _ in range(len(y_test))]
78  nb_probs = classifier.predict_proba(X_test)
79  nb_probs = nb_probs[:, 1]
80
81  r_auc = roc_auc_score(y_test, r_probs)
82  nb_auc = roc_auc_score(y_test, nb_probs)
83
84  print('KNN: AUROC = %.3f' % (nb_auc))
85
86  nb_fpr, nb_tpr, _ = roc_curve(y_test, nb_probs)
87
88
89  plt.plot(nb_fpr, nb_tpr, marker='.', label='KNN (AUROC = %0.3f)' % nb_auc)
90
91  # Title
92  plt.title('ROC Plot')
93  # Axis labels
94  plt.xlabel('False Positive Rate')
95  plt.ylabel('True Positive Rate')
96  # Show legend
97  plt.legend() #
98  # Show plot
99  plt.show()
100
101
```

3.- SVM

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

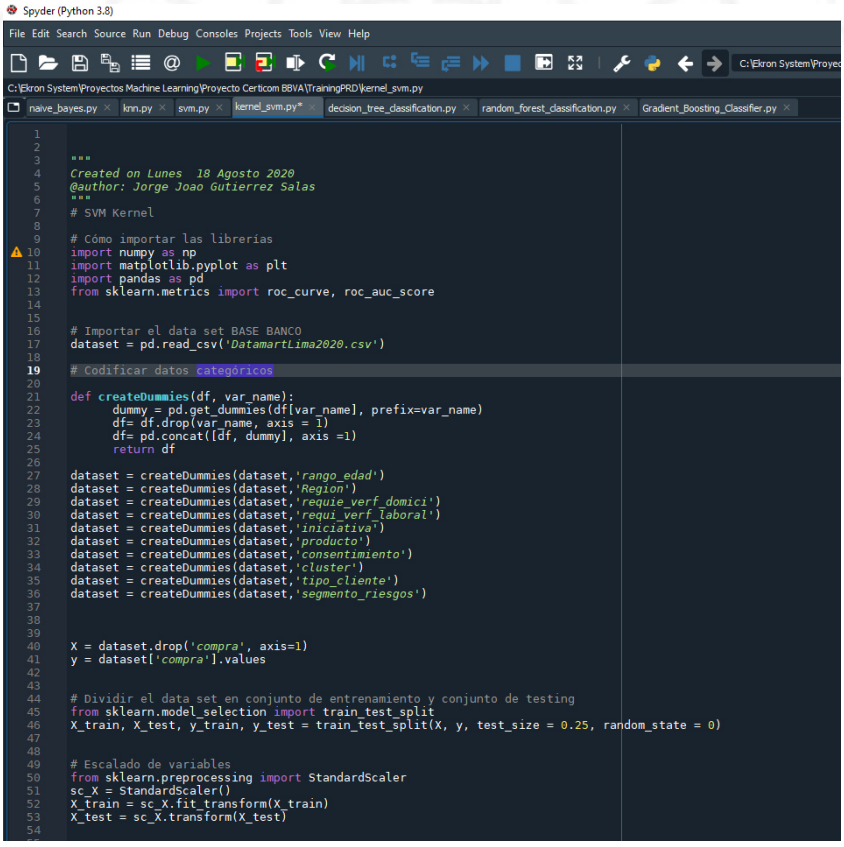
C:\Ekron System\Proyectos Machine Learning\Proyecto Certicom BBVA\TrainingPRD\svm.py

naive_bayes.py | knn.py | svm.py | decision_tree_classification.py | random_forest_classification.py | Gradient_Boosting_Classifier.py

```
1   """
2   Created on Lunes  18 Agosto 2020
3   @author: Jorge Joao Gutierrez Salas
4   """
5
6   # SVM
7
8
9   # Cómo importar las librerías
10  import numpy as np
11  import matplotlib.pyplot as plt
12  import pandas as pd
13  from sklearn.metrics import roc_curve, roc_auc_score
14
15
16  # Importar el data set BASE BANCO
17  dataset = pd.read_csv('DatamartLima2020.csv')
18
19  # Codificar datos categóricos
20
21  def createDummies(df, var_name):
22        dummy = pd.get_dummies(df[var_name], prefix=var_name)
23        df= df.drop(var_name, axis = 1)
24        df= pd.concat([df, dummy], axis =1)
25        return df
26
27  dataset = createDummies(dataset,'rango_edad')
28  dataset = createDummies(dataset,'Region')
29  dataset = createDummies(dataset,'requie_verf_domici')
30  dataset = createDummies(dataset,'requi_verf_laboral')
31  dataset = createDummies(dataset,'iniciativa')
32  dataset = createDummies(dataset,'producto')
33  dataset = createDummies(dataset,'consentimiento')
34  dataset = createDummies(dataset,'cluster')
35  dataset = createDummies(dataset,'tipo_cliente')
36  dataset = createDummies(dataset,'segmento_riesgos')
37
38
39
40  X = dataset.drop('compra', axis=1)
41  y = dataset['compra'].values
42
43
44
45  # Dividir el data set en conjunto de entrenamiento y conjunto de testing
46  from sklearn.model_selection import train_test_split
47  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
48
49
50  # Escalado de variables
51  from sklearn.preprocessing import StandardScaler
52  sc_X = StandardScaler()
53  X_train = sc_X.fit_transform(X_train)
54  X_test = sc_X.transform(X_test)
55
56
```

```
56
57  # Ajustar el SVM en el Conjunto de Entrenamiento
58  from sklearn.svm import SVC
59  classifier = SVC(kernel = "linear", random_state = 0)
60  classifier.fit(X_train, y_train)
61
62
63  # Predicción de los resultados con el Conjunto de Testing
64  y_pred  = classifier.predict(X_test)
65
66  # Elaborar una matriz de confusión
67  from sklearn.metrics import confusion_matrix
68  cm = confusion_matrix(y_test, y_pred)
69
70  accuracy= classifier.score(X_test, y_test)
71
72
73
74  # Prediccion de probabilidad - Elaborar Curva ROC
75  r_probs = [0 for _ in range(len(y_test))]
76  nb_probs = classifier.predict_proba(X_test)
77  nb_probs = nb_probs[:, 1]
78
79  r_auc = roc_auc_score(y_test, r_probs)
80  nb_auc = roc_auc_score(y_test, nb_probs)
81
82  print('SVM: AUROC = %.3f' % (nb_auc))
83
84  nb_fpr, nb_tpr, _ = roc_curve(y_test, nb_probs)
85
86
87  plt.plot(nb_fpr, nb_tpr, marker='.', label='SVM (AUROC = %0.3f)' % nb_auc)
88
89  # Title
90  plt.title('ROC Plot')
91  # Axis labels
92  plt.xlabel('False Positive Rate')
93  plt.ylabel('True Positive Rate')
94  # Show legend
95  plt.legend() #
96  # Show plot
97  plt.show()
```

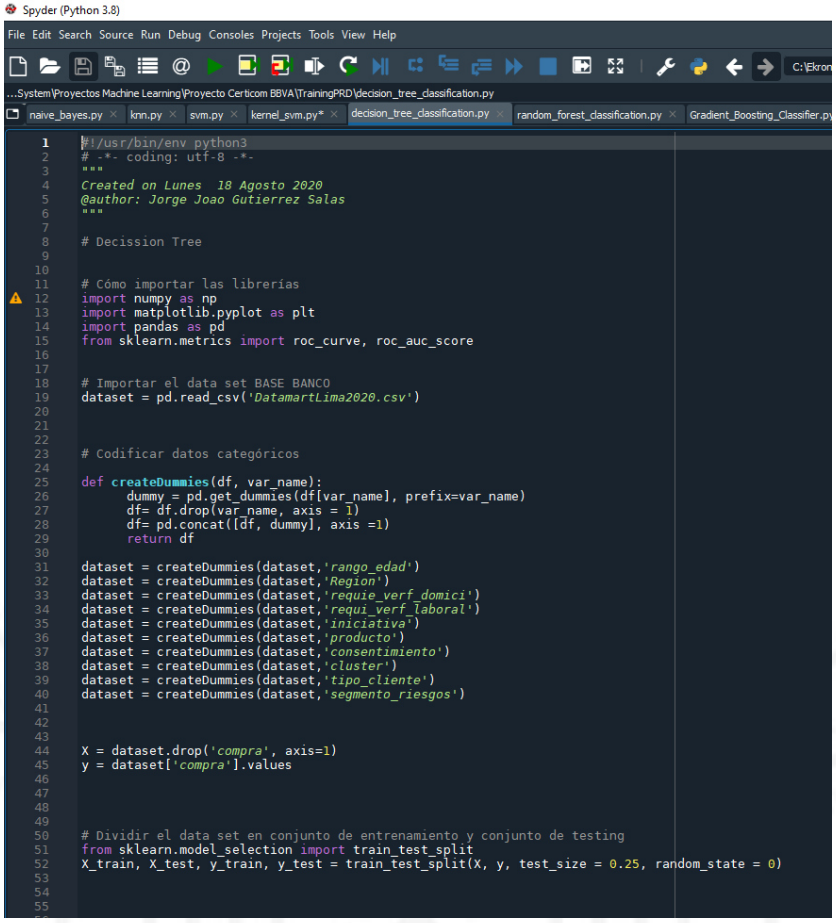
4.- SVM Kernel

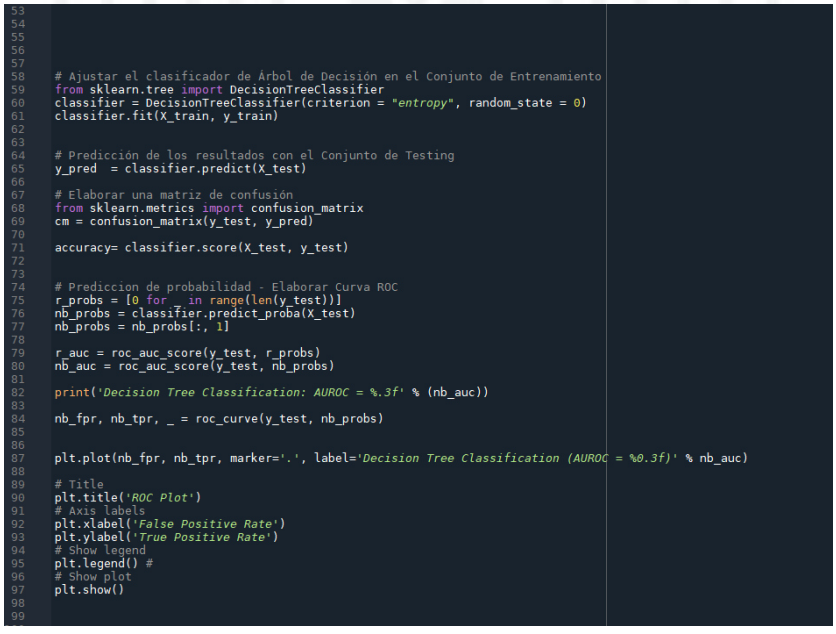



```
1
2
3   """
4   Created on Lunes  18 Agosto 2020
5   @author: Jorge Joao Gutierrez Salas
6   """
7   # SVM Kernel
8
9   # Cómo importar las librerías
10  import numpy as np
11  import matplotlib.pyplot as plt
12  import pandas as pd
13  from sklearn.metrics import roc_curve, roc_auc_score
14
15
16  # Importar el data set BASE BANCO
17  dataset = pd.read_csv('DatamartLima2020.csv')
18
19  # Codificar datos categóricos
20
21  def createDummies(df, var_name):
22        dummy = pd.get_dummies(df[var_name], prefix=var_name)
23        df= df.drop(var_name, axis = 1)
24        df= pd.concat([df, dummy], axis =1)
25        return df
26
27  dataset = createDummies(dataset,'rango_edad')
28  dataset = createDummies(dataset,'Region')
29  dataset = createDummies(dataset,'requie_verf_domici')
30  dataset = createDummies(dataset,'requi_verf_laboral')
31  dataset = createDummies(dataset,'iniciativa')
32  dataset = createDummies(dataset,'producto')
33  dataset = createDummies(dataset,'consentimiento')
34  dataset = createDummies(dataset,'cluster')
35  dataset = createDummies(dataset,'tipo_cliente')
36  dataset = createDummies(dataset,'segmento_riesgos')
37
38
39
40  X = dataset.drop('compra', axis=1)
41  y = dataset['compra'].values
42
43
44  # Dividir el data set en conjunto de entrenamiento y conjunto de testing
45  from sklearn.model_selection import train_test_split
46  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
47
48
49  # Escalado de variables
50  from sklearn.preprocessing import StandardScaler
51  sc_X = StandardScaler()
52  X_train = sc_X.fit_transform(X_train)
53  X_test = sc_X.transform(X_test)
54
```

```
# Ajustar el SVM en el Conjunto de Entrenamiento - Kernel Gausinano
from sklearn.svm import SVC
classifier = SVC(kernel = "rbf", random_state = 0)
classifier.fit(X_train, y_train)


# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

accuracy= classifier.score(X_test, y_test)



# Prediccion de probabilidad - Elaborar Curva ROC
r_probs = [0 for _ in range(len(y_test))]
nb_probs = classifier.predict_proba(X_test)
nb_probs = nb_probs[:, 1]

r_auc = roc_auc_score(y_test, r_probs)
nb_auc = roc_auc_score(y_test, nb_probs)

print('SVM Kernel: AUROC = %.3f' % (nb_auc))

nb_fpr, nb_tpr, _ = roc_curve(y_test, nb_probs)


plt.plot(nb_fpr, nb_tpr, marker='.', label='SVM Kernel (AUROC = %0.3f)' % nb_auc)

# Title
plt.title('ROC Plot')
# Axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Show legend
plt.legend() #
# Show plot
plt.show()

```

5.- Decision Tree Classification


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...System\Proyectos Machine Learning\Proyecto Certicom BBVA\TrainingPRD\decision_tree_classification.py
naive_bayes.py
knn.py
svm.py
kernel_svm.py*
decision_tree_classification.py
random_forest_classification.py
Gradient_Boosting_Classifier.py
#!/usr/bin/env python3
-*- coding: utf-8 -*-
"""
Created on Lunes 18 Agosto 2020
@author: Jorge Joao Gutierrez Salas
"""
# Decission Tree
# Cómo importar las librerías
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import roc_curve, roc_auc_score
# Importar el data set BASE BANCO
dataset = pd.read_csv('DatamartLima2020.csv')
# Codificar datos categóricos
def createDummies(df, var_name):
dummy = pd.get_dummies(df[var_name], prefix=var_name)
df= df.drop(var_name, axis = 1)
df= pd.concat([df, dummy], axis =1)
return df
dataset = createDummies(dataset,'rango_edad')
dataset = createDummies(dataset,'Region')
dataset = createDummies(dataset,'requie_verf_domici')
dataset = createDummies(dataset,'requi_verf_laboral')
dataset = createDummies(dataset,'iniciativa')
dataset = createDummies(dataset,'producto')
dataset = createDummies(dataset,'consentimiento')
dataset = createDummies(dataset,'cluster')
dataset = createDummies(dataset,'tipo_cliente')
dataset = createDummies(dataset,'segmento_riesgos')
X = dataset.drop('compra', axis=1)
y = dataset['compra'].values
# Dividir el data set en conjunto de entrenamiento y conjunto de testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)


# Ajustar el clasificador de Árbol de Decisión en el Conjunto de Entrenamiento
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = "entropy", random_state = 0)
classifier.fit(X_train, y_train)
# Predicción de los resultados con el Conjunto de Testing
y_pred = classifier.predict(X_test)
# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
accuracy= classifier.score(X_test, y_test)
# Prediccion de probabilidad - Elaborar Curva ROC
r_probs = [0 for _ in range(len(y_test))]
nb_probs = classifier.predict_proba(X_test)
nb_probs = nb_probs[:, 1]
r_auc = roc_auc_score(y_test, r_probs)
nb_auc = roc_auc_score(y_test, nb_probs)
print('Decision Tree Classification: AUROC = %.3f' % (nb_auc))
nb_fpr, nb_tpr, _ = roc_curve(y_test, nb_probs)
plt.plot(nb_fpr, nb_tpr, marker='.', label='Decision Tree Classification (AUROC = %0.3f)' % nb_auc)
# Title
plt.title('ROC Plot')
# Axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Show legend
plt.legend() #
# Show plot
plt.show()

6.- Random Forest Classification

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

...System\Proyectos Machine Learning\Proyecto Certicom BBVA\TrainingPRD\random_forest_classification.py

naive_bayes.py | knn.py | svm.py | kernel_svm.py* | decision_tree_classification.py | random_forest_classification.py | Gradient_Boosting_Classifier.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Lunes  18 Agosto 2020
@author: Jorge Joao Gutierrez Salas
"""

# Random Forest

# Cómo importar las librerías
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
from sklearn.metrics import roc_curve, roc_auc_score


# Importar el data set BASE BANCO
dataset = pd.read_csv('DatamartLima2020.csv')


# Codificar datos categóricos

def createDummies(df, var_name):
      dummy = pd.get_dummies(df[var_name], prefix=var_name)
      df= df.drop(var_name, axis = 1)
      df= pd.concat([df, dummy], axis =1)
      return df

dataset = createDummies(dataset,'rango_edad')
dataset = createDummies(dataset,'Region')
dataset = createDummies(dataset,'requie_verf_domici')
dataset = createDummies(dataset,'requi_verf_laboral')
dataset = createDummies(dataset,'iniciativa')
dataset = createDummies(dataset,'producto')
dataset = createDummies(dataset,'consentimiento')
dataset = createDummies(dataset,'cluster')
dataset = createDummies(dataset,'tipo_cliente')
dataset = createDummies(dataset,'segmento_riesgos')



X = dataset.drop('compra', axis=1)
y = dataset['compra'].values




# Dividir el data set en conjunto de entrenamiento y conjunto de testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

```

```
# Escalado de variables
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)


# Ajustar el clasificador  Random Forest en el Conjunto de Entrenamiento
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 1000, criterion = "entropy", random_state = 0)
classifier.fit(X_train, y_train)

# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)


accuracy= classifier.score(X_test, y_test)


# Prediccion de probabilidad - Elaborar Curva ROC
r_probs = [0 for _ in range(len(y_test))]
nb_probs = classifier.predict_proba(X_test)
nb_probs = nb_probs[:, 1]

r_auc = roc_auc_score(y_test, r_probs)
nb_auc = roc_auc_score(y_test, nb_probs)

print('Random Forest: AUROC = %.3f' % (nb_auc))

nb_fpr, nb_tpr, _ = roc_curve(y_test, nb_probs)


plt.plot(nb_fpr, nb_tpr, marker='.', label='Random Forest (AUROC = %0.3f)' % nb_auc)

# Title
plt.title('ROC Plot')
# Axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Show legend
plt.legend() #
# Show plot
plt.show()


# Saving model to disk
pickle.dump(classifier, open('classifierRandomForest.pkl','wb'))

```

7.- Gradient Boosting Classifier

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

...System\Proyectos Machine Learning\Proyecto Certicom BBVA\TrainingPRD\Gradient_Boosting_Classifier.py

naive_bayes.py | knn.py | svm.py | kernel_svm.py* | decision_tree_classification.py | random_forest_classification.py | Gradient_Boosting_Classifier.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Lunes  18 Agosto 2020
@author: Jorge Joao Gutierrez Salas
"""

# Gradient Boosting Classifier


# Cómo importar las librerias
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
from sklearn.metrics import roc_curve, roc_auc_score


# Importar el data set BASE BANCO
dataset = pd.read_csv('DatamartLima2020.csv')


# Codificar datos categóricos

def createDummies(df, var_name):
      dummy = pd.get_dummies(df[var_name], prefix=var_name)
      df= df.drop(var_name, axis = 1)
      df= pd.concat([df, dummy], axis =1)
      return df

dataset = createDummies(dataset,'rango_edad')
dataset = createDummies(dataset,'Region')
dataset = createDummies(dataset,'requie_verf_domici')
dataset = createDummies(dataset,'requi_verf_laboral')
dataset = createDummies(dataset,'iniciativa')
dataset = createDummies(dataset,'producto')
dataset = createDummies(dataset,'consentimiento')
dataset = createDummies(dataset,'cluster')
dataset = createDummies(dataset,'tipo_cliente')
dataset = createDummies(dataset,'segmento_riesgos')



X = dataset.drop('compra', axis=1)
y = dataset['compra'].values




# Dividir el data set en conjunto de entrenamiento y conjunto de testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

```

```
# Escalado de variables
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)


# Ajustar el clasificador  Gradient Boosting en el Conjunto de Entrenamiento
from sklearn.ensemble import GradientBoostingClassifier
classifier = GradientBoostingClassifier(n_estimators=2000,random_state = 0)
classifier.fit(X_train, y_train)

# Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

# Elaborar una matriz de confusión
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)


accuracy= classifier.score(X_test, y_test)


# Prediccion de probabilidad - Elaborar Curva ROC
r_probs = [0 for _ in range(len(y_test))]
nb_probs = classifier.predict_proba(X_test)
nb_probs = nb_probs[:, 1]

r_auc = roc_auc_score(y_test, r_probs)
nb_auc = roc_auc_score(y_test, nb_probs)

print('Gradient Boosting Classifier: AUROC = %.3f' % (nb_auc))

nb_fpr, nb_tpr, _ = roc_curve(y_test, nb_probs)


plt.plot(nb_fpr, nb_tpr, marker='.', label='Gradient Boosting Classifier (AUROC = %0.3f)' % nb_auc)

# Title
plt.title('ROC Plot')
# Axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Show legend
plt.legend() #
# Show plot
plt.show()


# Saving model to disk
pickle.dump(classifier, open('classifierGradientBoosting.pkl','wb'))

```

Anexos 2: Pantallas de Front-End aplicando Machine Learning

Aplicación Web de ingreso de características

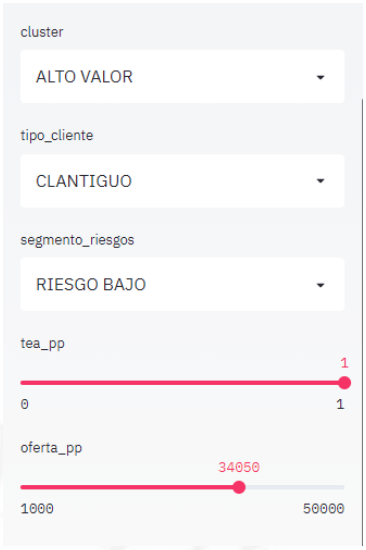
Se utilizó la librería Streamlit de Python (Streamlit v0.66.0) que permite realizar Web Apps asociadas a un archivo *.pkl* (archivo que guarda el modelo de Machine Learning previamente entrenado) y el ingreso de las características de entrada (variables independientes) para la ejecución de la predicción.

La aplicación permite el ingreso individual de las características de entrada, mediante un SideBar que contiene combos e interfaces que permiten seleccionar los valores de las entradas, así como una carga de archivos en formato .csv

Input individual:

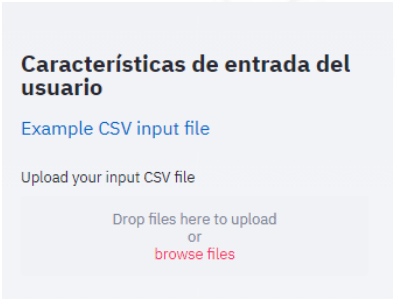
Permite ingresar un registro (lead) con las variables de entrada:

- Mes (input selectbox)
- Año (input selectbox)
- Rango de edad (input selectbox)
- Región (input selectbox)
- Verificación domiciliaria (input selectbox)
- Verificación laboral (input selectbox)
- Iniciativa (input selectbox)
- Producto (input selectbox)
- Consentimiento (input selectbox)
- Clúster (input selectbox)
- Tipo_cliente (input selectbox)
- Segmento_riesgos (input selectbox)
- Tea_pp (slider)
- Oferta_pp (slider)

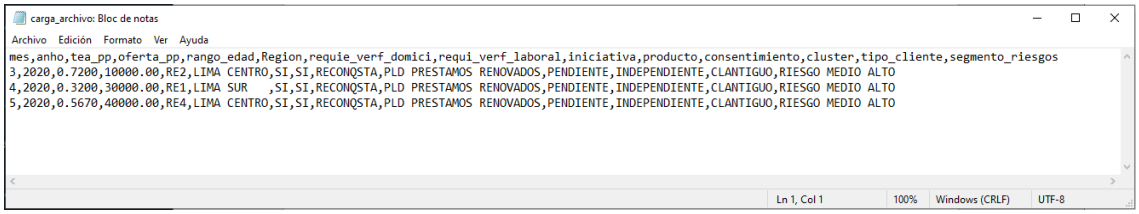


Carga de archivos CSV:

A su vez, la aplicación permite cargar archivos en formato .csv y que contengan la estructura de los parámetros de entrada, y con una cantidad de registros mayor a 1, como se muestra a continuación:



Archivo modelo en formato .csv

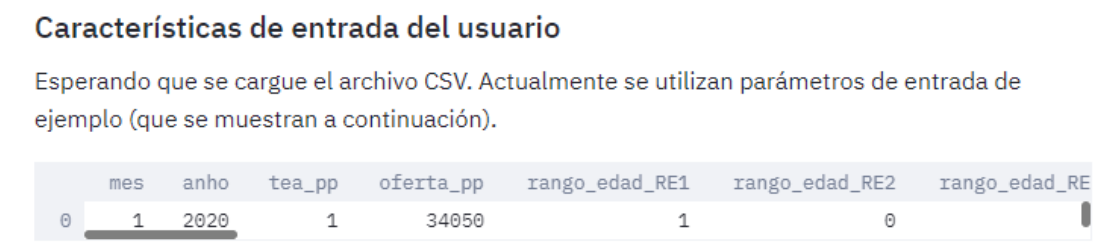


Muestra de características:

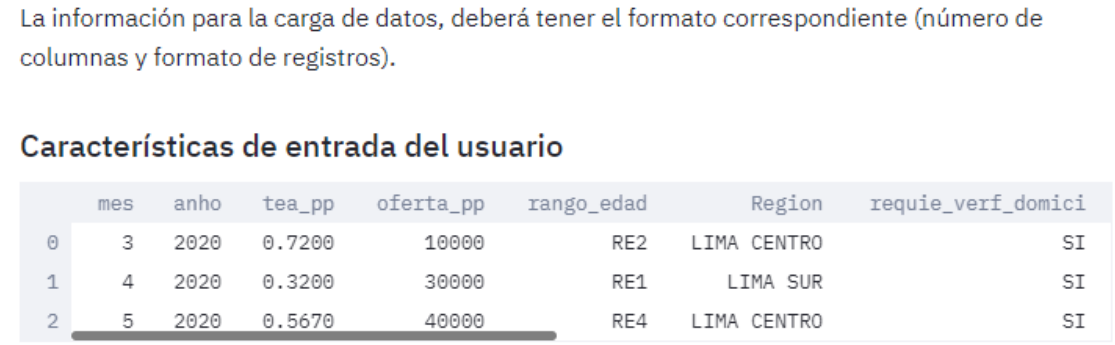
La aplicación cuenta con un display que permite visualizar los datos ingresados.

En el caso de tener un ingreso individual, se mostrará solo un registro en el display, caso contrario se podrá ingresar un archivo .csv, que mostrará en el display la cabecera y la cantidad total de registros a predecir.

Ingreso individual:

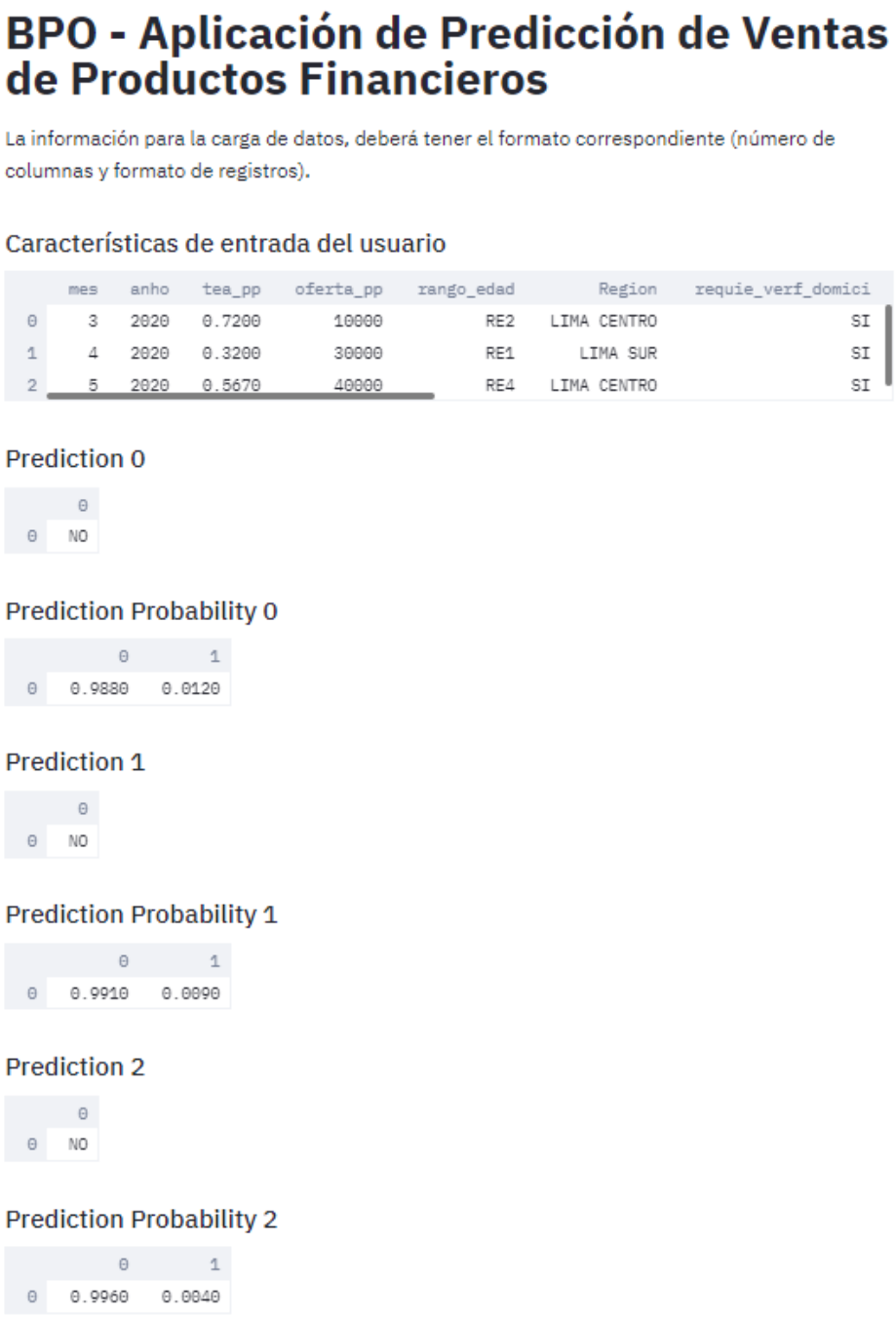


Carga de archivo:



Predicción

Una vez se contaron con los datos cargados, la aplicación en automático realizó la invocación al servicio de Machine Learning seleccionado y que previamente fue entrenado, con el fin de obtener como output el resultado y probabilidad de la predicción.



BPO - Aplicación de Predicción de Ventas de Productos Financieros

La información para la carga de datos, deberá tener el formato correspondiente (número de columnas y formato de registros).

Características de entrada del usuario

	mes	anho	tea_pp	oferta_pp	rango_edad	Region	requie_verf_domici
0	3	2020	0.7200	10000	RE2	LIMA CENTRO	SI
1	4	2020	0.3200	30000	RE1	LIMA SUR	SI
2	5	2020	0.5670	40000	RE4	LIMA CENTRO	SI

Prediction 0

	0
0	NO

Prediction Probability 0

	0	1
0	0.9880	0.0120

Prediction 1

	0
0	NO

Prediction Probability 1

	0	1
0	0.9910	0.0090

Prediction 2

	0
0	NO

Prediction Probability 2

	0	1
0	0.9960	0.0040

Según la imagen, la aplicación a procesado 3 registros (leads) los cuales han sido cargados en formato .csv, obteniendo los siguientes resultados:

Prediction 0: Correspondiente al registro 0, Resultado: NO

Prediction 1: Correspondiente al registro 1, Resultado: NO

Prediction 2: Correspondiente al registro 2, Resultado: NO

Finalmente se obtuvo un Web App que permite realizar ingresos de características del modelo de forma intuitiva, a un nivel de usuario final. La aplicación puede ser accedida vía internet, intranet o según el cliente necesite (a nivel de accesibilidad).