UNIVERSIDAD NACIONAL AGRARIA
LA MOLINA

FACULTAD DE ECONOMÍA Y PLANIFICACIÓN

"CLASIFICACIÓN DE CLIENTES POTENCIALES DE UN OPERADOR TELEFÓNICO CONTACTADOS POR UN CALL CENTER UTILIZANDO REGRESIÓN LOGÍSTICA Y ADABOOSTING"

TRABAJO DE SUFICIENCIA PROFESIONAL
PARA OPTAR TÍTULO DE
INGENIERA ESTADÍSTICA INFORMÁTICA

ROSARIO DEL PILAR LUQUE CARBAJAL

LIMA – PERÚ

2024

TSP_Rosario Luque Carbajal.docx

INFORME DE ORIGINALIDAD

15%	13%	3%	8%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	dspace.unitru.edu.pe Fuente de Internet	1%
2	repositorio.usmp.edu.pe Fuente de Internet	1%
3	upc.aws.openrepository.com Fuente de Internet	1%
4	www.lamolina.edu.pe Fuente de Internet	1%
5	www.coursehero.com Fuente de Internet	1%
6	Submitted to Universitat Politècnica de València Trabajo del estudiante	1%
7	Submitted to Pontificia Universidad Catolica del Peru Trabajo del estudiante	1%
8	repositorio.unsaac.edu.pe Fuente de Internet	1%

UNIVERSIDAD NACIONAL AGRARIA
LA MOLINA
FACULTAD DE ECONOMÍA Y PLANIFICACIÓN

“CLASIFICACIÓN DE CLIENTES POTENCIALES DE UN OPERADOR TELEFÓNICO CONTACTADOS POR UN CALL CENTER UTILIZANDO REGRESIÓN LOGÍSTICA Y ADABOOSTING”

PRESENTADO POR:
ROSARIO DEL PILAR LUQUE CARBAJAL

TRABAJO DE SUFICIENCIA PROFESIONAL PARA OPTAR EL TÍTULO DE INGENIERA ESTADÍSTICA INFORMÁTICA

SUSTENTADO Y APROBADO ANTE EL SIGUIENTE JURADO:

……	……..
M. A. Fernando René Rosas Villena	Dr. Jorge Chue Gallardo
PRESIDENTE	**ASESOR**
……	……..
MS. Grimaldo José Febres Huamán	Mg. Ana Cecilia Vargas Paredes
MIEMBRO	**MIEMBRO**

Lima – Perú

2024

DEDICATORIA

Todo el esfuerzo, tiempo y dedicación al tan soñado título es para mis padres y hermano, sobre todo para mi mamá quien siempre ha sido, es y será mi mejor y mayor inspiración. Cada vez que hablo de ella o escribo sobre lo que significa para mí, es inevitable no quebrarme, gracias infinitas a tan acendrado amor que me hace reflexionar a cada paso y me brinda su hombro consolador cada vez que me equivoco, te amo madre.

AGRADECIMIENTO

Sin duda alguna, agradezco a mi tan amada Agraria que me ha dado todo lo que una mujer puede desear: conocimiento, empoderamiento, fuerza y una hermosa familia compuesta de grandes amigos; a través de sus docentes, personal administrativo e incluso a nuestras alverjitas que me acompañaron en toda mi etapa de pregrado. En especial agradezco a mis profesores Jorge Chue Gallardo y Jesús Salinas Flores por sus constantes motivaciones y apoyo sin el cual este logro no sería posible.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo monográfico busca demostrar que, para los datos empleados en la realización de la investigación, el algoritmo adaboosting presenta mejores resultados en la clasificación de clientes potenciales de un call center a diferencia de la regresión logística. Sin embargo, no se busca determinar que un algoritmo sea mejor que el otro, sino comprobar que, dada las características de las variables independientes, un algoritmo puede presentar mejores resultados y viceversa. Se utiliza una solución analítica avanzada que parte desde el análisis descriptivo de las variables, selección de variables, imputación de los datos y modelamiento predictivo, validación de los resultados en el tiempo; hasta la puesta en marcha a partir de los grupos de ejecución que permiten el despliegue y acción sobre los resultados obtenidos dado el trabajo desarrollado, exactitud del adaboosting 86% y exactitud de la regresión logística 78%.

Palabras clave: algoritmo adaboosting, regresión logística, call center, selección de variables, balanceo de la información y modelamiento predictivo.

ABSTRACT

This monographic work aims to demostrate that, data used for this research gave positive results when applying adaboosting algorithm instead of logistic regression but it doesn't mean that one of this techniques is better than the other for classification of potential customers of a call center. On the other hand, the objective is verify that, given the characteristics of the independent variables, an algorithm can present better results and viceversa. An advanced analytical solution is used and followed by the next steps: descriptive analysis of variables, feature selection, data imputation, predictive modelling and validation of the results over time. Until the start-up is deployed, we have to make the execution groups that allow the deployment and actions in order to have the results expected, adaboosting accuracy 86% and logist regression accuracy 78%.

Keywords: adaboosting algorithm, logistic regression, call center, feature selection, balance data and predictive modelling.

I. INTRODUCCIÓN

En el Perú, existen diversos operadores telefónicos (OT) como Movistar, Claro, Entel, Virgin, Bitel, entre otros; que atienden la creciente demanda de equipos y servicios de comunicación en todo el país. La participación en el mercado de estos OT en los últimos años ha experimentado fuertes variaciones, según un informe del Organismo Supervisor de Investigación Privada en Telecomunicaciones (OSIPTEL) se señala que, al primer trimestre del año 2017, Entel subió su participación de 12.84% a 13.52% y Bitel avanzó de 9.84% a 10.8%. Mientras que, Movistar, retrocedió de 44.50% a 43.03% y Claro se mantuvo en 32.64%. Este mismo informe menciona que para el año 2017 la industria de telefonía duplique los ingresos totales con servicios más especializados de BPO como el uso de call centers (CC) para la externalización de sus procesos, alcanzando los U$S 1.000 millones (GESTIÓN - Economía - Empresas, 2017).

Uno de estos operadores telefónicos será considerado en esta investigación, que por razones de confidencialidad no se mencionará su razón social. Actualmente, el operador telefónico en estudio se encuentra utilizando los servicios de un CC para optimizar la atención de sus clientes y minimizar las altas tasas de insatisfacción en la resolución de dudas o inquietudes. Asimismo, busca mejorar la contactabilidad con sus clientes con la finalidad de realizar diferentes tipos de ofrecimiento según las necesidades y características particulares de cada potencial comprador. Para ello, se hace necesario clasificar a los clientes potenciales a través de un conjunto de variables relevantes que permitan identificar a los clientes en función a la posible realización o no de una venta de sus productos.

1.1. Problemática

El motivo de realización de la presente monografía se debe a que constantemente diversos clientes del OT objeto de estudio, se muestran insatisfechos al ser contactados por el CC que busca realizar la venta de un plan de servicio telefónico del OT con algunas mejoras para el cliente. Esta insatisfacción es un problema para el OT y el CC que ha motivado la

formulación de diversos proyectos con el objetivo de identificar y fidelizar a los clientes que sí están dispuestos a aceptar el plan de venta con las mejoras respectivas.

En uno de estos proyectos se participará haciendo uso del algoritmo adaboosting, ya que este tipo de algoritmo a diferencia de la regresión logística, se alimenta de la información del fracaso del evento para predecir el éxito del mismo. Esto último viene ocurriendo con la disminución de los indicadores de los servicios del OT. Una limitación en la aplicación del adaboosting fue el desconocimiento inicial del algoritmo y su respectiva interpretación. Por este motivo, sugiero que esta técnica y otras que en el futuro aparezcan sean difundidas entre el alumnado del Departamento de Estadística e Informática.

Por tal motivo, en el presente trabajo monográfico se investigará si el algoritmo adaboosting predice y explica mejor los datos que la técnica de regresión logística utilizando como medida de eficiencia el indicador AUC (area under the curve). El resultado de este análisis generará una mejor clasificación de los clientes potenciales para que los recursos empleados en la obtención de la venta se centren de manera correcta en el potencial comprador.

Los datos a utilizarse serán desde octubre del 2016 hasta marzo del 2017, los cuales son proporcionados por un CC. El número total de datos es de 2,104,842. Las variables son cualitativas y cuantitativas; para su procesamiento se utilizará el software estadístico R (R Core Team, 2016).

1.2. Objetivos

1.2.1. Objetivo general

Comparar las técnicas de regresión logística y adaboosting para la mejor clasificación de los clientes potenciales de un operador telefónico contactados por un call center.

1.2.2. Objetivos específicos

- Contrastar el modelo de regresión logística y del adaboosting para la correcta asignación de probabilidades a los clientes y clasificarlos por cortes de probabilidad según su efectividad.

- Considerar la respuesta comparativa del área bajo la curva AUC para determinar si adaboosting brinda una mejor clasificación respecto a la regresión logística.

II. REVISIÓN DE LITERATURA

En esta sección se procederá a revisar todos los términos teóricos generales, particulares y propios utilizados para llevar a cabo el presente trabajo monográfico, se buscará relacionar investigaciones pasadas y actuales que ayuden a la comprensión de los términos empleados, así como las técnicas desarrolladas:

2.1. Rubro Operadores

Actualmente, el Perú cuenta con 06 operadores telefónicos que comparten un mercado donde dos de ellos tienen más del 90% de participación (Movistar y Claro) seguidos por Entel, Bitel, Tuenti y Virgin. Existen tratos para el ingreso de dos nuevos operadores lo cual no es nada extraño comparado con el mercado europeo en que existen en promedio por país más de 20 operadores telefónicos.

Algunos OT cuentan con su propia red para ofrecer sus servicios y otros no. En el mundo existen dos tipos de operadores móviles: los que tienen su propia red (montan sus antenas) y los OMV (Operador Móvil Virtual) que carece de su propia antena, es decir, aquellos que alquilan la red que otros tienen y solo se dedican a ofrecer el servicio. Entre los operadores con su propia red se tiene a: Movistar, Claro, Entel y Bitel; y los OMV son: Tuenti y Virgin.

2.2. Rubro Call Center

La Asociación Peruana de Centros de Contacto reconoce a 20 centros de contactos, entre ellos: Dynamin, ECOM DATA, OLVA, Konecta, IBR, AST, Kobranzas, Teleavanca, Arvato, HDC, Fortel, Contact Center, AEGIS, tlMark, MDY, Digitex, IATEC, Atento, SCC y GSS.

Los distintos CC se encargan de realizar llamadas (llamadas salientes) o recibir llamadas (llamadas entrantes) con el fin de brindar un servicio de venta, comunicación, cobranza, etc.

Dependiendo de la empresa que la haya contratado; con motivo de generar una fidelización de los clientes a la empresa contratante. Dentro de las características que existe en un CC están la recepción de llamadas por personas o máquinas que brindan información objetiva y exacta sobre los productos a ofrecer, de manera clara y educada para generar un entorno comunicativo y seguro, y la absolución de dudas o consultas por parte de los clientes.

2.3. Regresión Logística

La regresión logística modela la probabilidad de que una variable respuesta (Y) que se encuentra en el intervalo [0, 1] (Gareth, Witten, Hastie & Tibshirani, 2013), ayude, por corte de probabilidad a distinguir la clasificación de un evento "Sí" o "No" (por ejemplo) con el fin de llegar a predecir con la mejor aproximación al éxito una clase dentro de un conjunto de datos. A diferencia de una regresión lineal que busca predecir una variable respuesta (Y) de carácter cuantitativo que puede escapar del intervalo previamente mencionado.

En la regresión lineal suele suceder que al momento de predecir un valor para $p(X)$ se obtienen resultados menores a cero o mayores a uno. Para resolver este problema se debe modelar $p(X)$ tal que proporcione valores dentro del intervalo [0, 1]. En la regresión logística, se usa la función logística:

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

$$log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 X \quad (2)$$

Para estimar los coeficientes del modelo se deberán encontrar los coeficientes que minimizan la función de cross – entropy o también conocida como log – loss (pérdida de registro) (IArtificial.net, 2020), definida como:

$$J = -\frac{1}{m}\sum_{i}^{m} \quad y_i \, log \, log \, (\hat{p}_i) \, + (1 - y_i) log \, (1 - \hat{p}_i) \quad (3)$$

Esta función se puede minimizar:

- Transformando el problema de clasificación a uno de regresión.

- Por descenso de gradiente (Martinez Heras, 2020).
- Por máxima verosimilitud (maximun likehood).

Como parte del presente trabajo monográfico se trabajó utilizando el método conocido como máxima verosimilitud (Gareth *et al*., 2013). Los coeficientes β_0 y β_1 en el modelo logístico de la ecuación (1) son desconocidos, y deben ser estimados basados en datos de entrenamiento. El método de máxima verosimilitud es preferido para realizar dicha estimación por sus propiedades estadísticas. La intuición básica para usar el método de máxima verosimilitud es la siguiente: encontrar los valores de β_0 y β_1 que son escogidos para maximizar la función de verosimilitud, de tal manera, que se consideren estos valores como estimaciones para el modelo $p(X)$. Esta intuición se puede formalizar usando una ecuación matemática llamada función de verosimilitud:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_i'=0} (1 - p(x_i')) \qquad (4)$$

El coeficiente β_1 indica el cambio en el $log\left(\frac{p(x)}{1-p(x)}\right)$ (log – odds), cambio que puede ser positivo o negativo dependiente del valor del coeficiente, cuando · cambia en una unidad (Gareth *et al*., 2013).

2.4. Tabla de Clasificación

La tabla de clasificación es una forma sencilla de evaluar el ajuste del modelo de regresión logística, no es tan objetiva, pero se usa como indicador de bondad de ajuste. La tabla de clasificación es conocida también como matriz de confusión. Es una tabla sencilla de 2x2, en la cual se presenta la distribución de los valores cuando $y = 0$ y cuando $y = 1$, conjuntamente con la clasificación a cualquiera de las dos categorías a la probabilidad estimada. La interpretación se realiza mediante el porcentaje de valores correctamente clasificados; es decir, aquellos que mediante la probabilidad estimada permanecen a su respectiva categoría. También se interpreta mediante el porcentaje de valores mal clasificados; es decir, aquellos que fueron asignados a categorías que no les corresponden. En la Tabla 1, se observa el número de los valores correcta e incorrectamente clasificados.

Tabla 1

Matriz de Confusión

Grupo Actual	Grupo estimado		Total Marginal
	0	1	
0	n_{11}	n_{12}	$n_{11} + n_{12}$
1	n_{21}	n_{22}	$n_{21} + n_{22}$
Total Marginal	$n_{11} + n_{21}$	$n_{12} + n_{22}$	

El porcentaje estimado de observaciones correctamente clasificadas mediante un modelo de regresión logística es presentado en la ecuación (3).

$$\frac{n_{11}+n_{22}}{n} x100\% \quad (5)$$

Siendo n el total de observaciones, es decir, ($n_{11} + n_{12} + n_{21} + n_{22}$). Por lo tanto, lo que espera es un porcentaje lo más alto posible, a fin de concluir que el modelo obtenido clasifica bien a las observaciones (Salcedo Poma, 2002).

De la matriz de confusión se desprenden a su vez otras métricas de vital importancia dependiendo del enfoque de solución que tiene el problema, como:

- Precisión: $\frac{n_{22}}{n_{12}+n_{22}} x100$, porcentaje de predicciones positivas correctas.
- Accuracy o Exactitud: $\frac{n_{11}+n_{22}}{n_{11}+n_{12}+n_{21}+n_{22}} x100$, porcentaje de predicciones correctas.
- Especificidad: $\frac{n_{11}}{n_{11}+n_{12}} x100$, porcentaje de casos negativos detectados.
- Sensibilidad o Recall: $\frac{n_{22}}{n_{21}+n_{22}} x100$, porcentaje de casos positivos detectados.
- F1 - score: indica que tan bien están separados los 1's de los 0's y los 0's de los 1's. Se enfoca en que los 1's y 0's estén bien predichos.

2.5. AUC (área bajo la curva)

El área bajo la curva es una métrica de rendimiento que ayuda a determinar el desempeño del clasificador, ya que es independiente del criterio de decisión tomado y las probabilidades previas, la AUC puede establecer una relación de dominio entre los clasificadores. Por la tanto, el considerar esta métrica como una comparativa entre dos clasificadores es más justo

e informativo que comparar sus tasas de incorrecta clasificación (Rokach & Maimon, 2014).

Ubicando los puntos de la curva ROC y teniendo en cuenta que en el eje Y se encuentran los valores asociados a la sensibilidad para cada observación y en el eje · se encuentran los valores asociados a 1-especificidad (Cerda y Cifuentes, 2012). Se puede determinar manualmente el valor del AUC hallando el área del trapecio y aproximándola con una integral (Kapoor, 2019).

El valor del AUC se puede interpretar como la correcta discriminación de un individuo que padezca una enfermedad, por ejemplo, versus un individuo que se cataloga como paciente enfermo pero que en realidad es un paciente sano. Por ello, para determinar un valor óptimo de AUC se debe tener en cuenta la finalidad del estudio y considerar ya sea una mayor sensibilidad o mayor especificidad, pues generalmente el punto de corte que determina la mayor sensibilidad y especificidad en conjunto solo tiene a uno de ellos como el mayor.

2.6. Árboles de Decisión

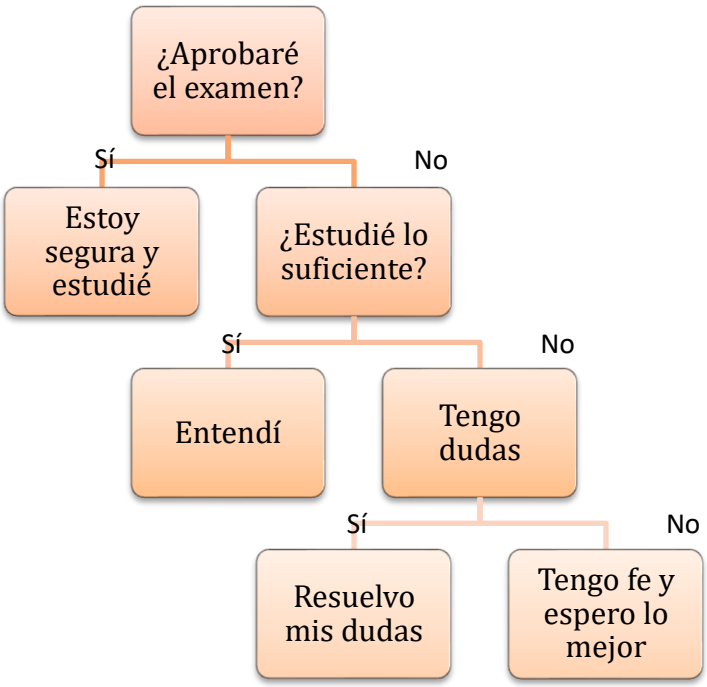
Los árboles de decisión son técnicas simples pero exitosas para predecir y explicar la relación entre algunas mediciones sobre un ítem y su objeto de estudio. A parte de su uso en minería de datos, los árboles de decisión, que originalmente derivan de la lógica, administración y estadística, actualmente son herramientas altamente efectivas en áreas como minería de textos, extracción de información, máquina de aprendizaje y reconocimiento de patrones (Rokach & Maimon, 2014).

Los árboles de clasificación son usados para clasificar un objeto o una instancia dentro de un conjunto predefinido de clases (como riesgoso o no riesgoso) basado en los valores de sus atributos (como edad o sexo). Los árboles de clasificación son frecuentemente usados en campos como finanzas, marketing, ingeniería y medicina. Es usada como una técnica exploratoria. A pesar de esto, no atenta en reemplazar los métodos de estadística tradicionales y hay otras muchas técnicas que pueden ser usadas para clasificar o predecir las afiliaciones de instancias con un conjunto de clases predefinido, como redes neuronales artificiales o máquina de soporte vectorial (Rokach & Maimon, 2014).

A continuación, se observa un ejemplo de cómo se puede llegar a una decisión a través de un árbol de clasificación:

Figura 1

Ejemplo de Árbol de Clasificación para una decisión



Un árbol de decisión puede presentar muchos subnodos dentro de los cuales se puede encontrar aquel corte que ajuste mejor el modelo y permita una interpretabilidad más estable y resultados constantes, tomar la decisión correcta de dónde realizar este corte de subnodos puede afectar altamente la precisión del árbol, lo que se busca con la creación de subnodos es incrementar la homogeneidad de los subnodos resultantes (Orellana Alvear, Bookdown, 2018). Se pueden ubicar distintos criterios de parada (corte) como: Índice de gini, Chi cuadrado, Ganancia de la información (de donde se desprende la Entropía) y Reducción en la varianza.

Para el desarrollo del presente trabajo monográfico se tomó en consideración el criterio de la Ganancia de información a través del menor valor de la Entropía.

La Entropía parte de la teoría de la información y es la medida de impureza en los datos que

busca el grado de desorganización en un sistema; si una muestra es completamente homogénea el valor de le entropía será cero, y si la muestra está igualmente dividida se tiene una entropía de uno. La entropía puede ser calculada de la siguiente manera:

$$Entropy = -p_1 * log_2(p_1) - \cdots - p_n * log_2(p_n) \quad (6)$$

Donde n es el número de clases. La entropía es máxima en el medio donde toma el valor de 1 y es mínima en los extremos donde toma el valor de 0. Se busca el menor valor de la entropía de tal manera que permita segregar mejor las clases (Dangeti, 2017). En este trabajo monográfico nos referimos a 2 clases ya que el objetivo es mejorar la predicción de la clasificación de clientes potenciales.

2.7. Algoritmo Adaboosting

Antes de que el algoritmo adaboosting sea introducido al mundo de la máquina de aprendizaje por Yoav Freund y Robert Schapire, se tomaba de manera convencional que, para realizar una clasificación de patrones, las características elegidas deberían ser lo más discriminatorias posibles. Con el algoritmo en mención, es posible usar características débiles para crear un clasificador de patrones, asumiendo que se tiene un número suficiente de características y que solo se pueden realizar clasificaciones binarias.

Por lo tanto, se podría decir que el enfocarse en los datos de entrenamiento mal clasificados por el clasificador débil anterior, busca la manera de mejorar la tasa de clasificación general; quiere decir que el enfoque adaboost apunta a que conforme se incorporen más y más clasificadores débiles en la secuencia de modelos creados, la tasa final de clasificación errónea de los datos de entrenamiento se reduzca (Kak, 2016)

El algoritmo adaboosting, perteneciente a la familia de los árboles de clasificación, es una metodología que permite asignar probabilidades a los individuos de tal manera que permite categorizarlos según el potencial que presenten a partir de la probabilidad asignada respecto al aprendizaje que indica que se alimenta de información del fracaso del evento para predecir el éxito del mismo (Pérez Tatamués, 2019).

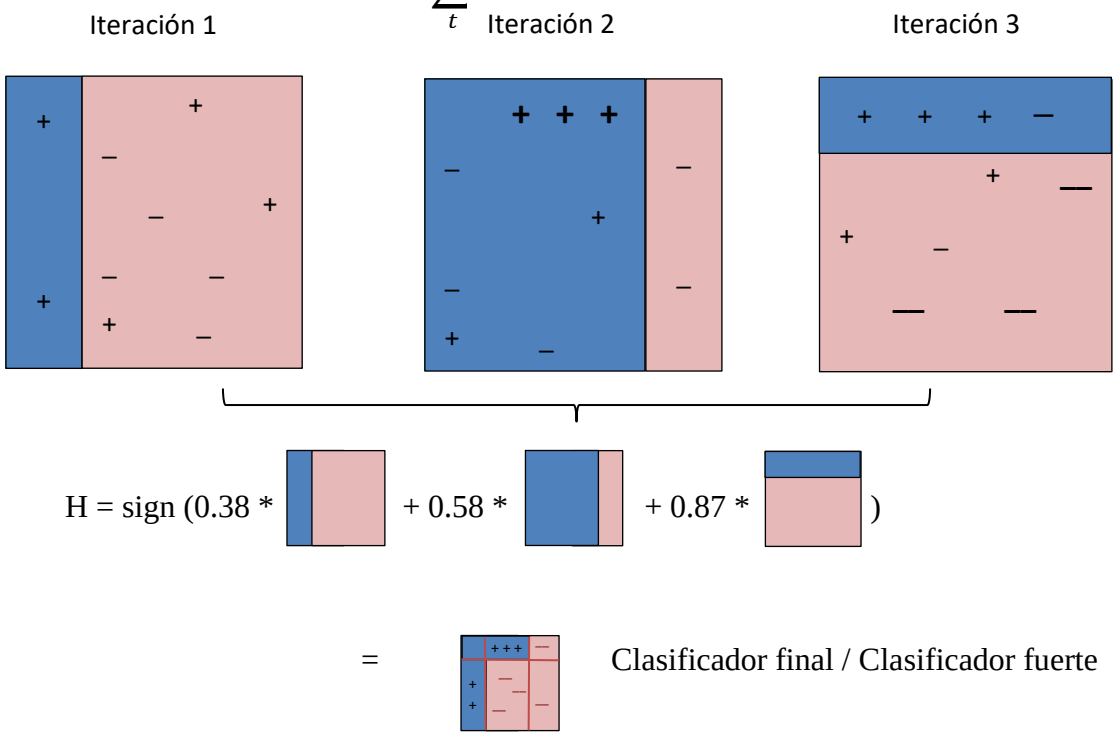
El clasificador de adaboost perteneciente a la familia de los boosting (enfoque general que

se puede aplicar a varios modelos estadísticos) trabaja de forma secuencial y no involucra un muestro boostrap, su performance se basa en la creación de árboles secuenciales cada uno de ellos ajustados con su versión anterior para obtener un clasificador robusto.

Figura 2

Ejemplificación del principio de funcionamiento del clasificador adaboost:

$$H(x) = \sum_{t} \rho_t h_t(x) \qquad (7)$$



Inicialmente (iteración 1) un clasificador simple se ajusta a la data y divide la data en 2 regiones (azul y rosa), la clase bien clasificada recibirá menor ponderación y la clase mal clasificada recibirá mayor ponderación en la siguiente iteración (iteración 2), y de nuevo se ajustará otro clasificador débil y cambiarán los pesos para la próxima iteración (iteración 3). Finalizadas las iteraciones, los pesos se calculan automáticamente para cada clasificador en cada iteración basado en la tasa de error para obtener un fuerte clasificador, que ayuda a predecir las clases con mucha precisión (Dangeti, 2017).

El algoritmo adaboosting presenta los siguientes pasos:

1) Inicializar los pesos de las observaciones $w_i = \frac{1}{N}, i = 1, 2, 3, \dots, N$.

   Donde N = número de observaciones

2) Para m = 1 a M:

   a. Ajustar un clasificador $Gm(x)$ a la data de entrenamiento usando los pesos w_i

   b. Calcular:

$$err_m = \frac{\sum_{i=1}^{N} \quad w_i I(y_i \neq \ G_m(x_i))}{\sum_{i=1}^{N} \quad w_i} \qquad (8)$$

   c. Calcular:

$$\alpha_m = log\ (\frac{1 - err_m}{err_m}) \qquad (9)$$

   d. Colocar:

$$w_i < \ - w_i * exp\ exp\left[\alpha_m * I\left(y_i \neq G_m(x_i)\right)\right]\ , i = 1, 2, \ldots, N \qquad (10)$$

3) Salida:

$$G(x) = sign[\sum_{m=1}^{M} \quad \alpha_m G_m(x)] \qquad (11)$$

Todas las observaciones toman un mismo peso; en boosting se ajustan los pesos para cada observación y no se eligen algunas columnas a diferencia de los algortimos de bagging y random forest donde se lidia con las columnas de los datos.

El error utilizado para el cálculo del peso debe darse para ese clasificador en el modelo aditivo final α. La forma en la que trabaja el algoritmo es dar mayor peso al modelo con menos errores; para que luego se actualicen los pesos de cada observación. Una vez hecho esto se aumentará el peso de las observaciones clasificadas incorrectamente para darle más foco a las siguientes iteraciones, y los pesos serán reducidos para las observaciones clasificadas correctamente. Todos los clasificadores débiles se encuentran combinados con sus respectivos pesos para formar un clasificador fuerte.

III. DESARROLLO DEL TRABAJO

A continuación, se muestra el trabajo monográfico de tipo no experimental y transversal realizado a una población en la cual se hizo uso de 2 tipos de modelos estadísticos: una regresión logística y un algoritmo adaboosting. Asimismo, se detalla el proceso de cómo se realizó el modelamiento y los análisis para poner en ejecución la solución analítica:

3.1. Tipo de investigación

El presente trabajo monográfico es no experimental y transversal porque se ubica en un momento del tiempo (Hernández Sampieri, 2010). La técnica que se desarrollará es causal porque en la investigación se cuenta con una variable independiente (y = venta) siendo esta categórica y asume dos valores 0 (fracaso) y 1 (éxito); y, con seis variables independientes o regresoras, de las cuales cinco son cualitativas y una cuantitativa.

3.2. Hipótesis de la investigación

La regresión logística y el algoritmo adaboosting tienen aproximadamente las mismas métricas de rendimiento.

Para la regresión logística se obtuvo un AUC de 69% y una tasa de mala clasificación de 22%; mientras que para el algoritmo adaboosting se obtuvo un AUC de 89% y una tasa de mala clasificación de 14%.

3.3. Variables

Variable respuesta

- VENTA: tipo de resultado de la gestión.

Variables predictoras

- RPM: realiza llamadas RPM.
- VOZ: total de minutos de las llamadas entrantes y salientes.
- DATOS: realiza tráfico de datos.
- LIMA_PROV: localidad del cliente.
- ANTIGÜEDAD: tipo de línea.
- PROM_RECARGA: promedio de recarga de los últimos 3 meses de la línea.

3.4. Operacionalización de las variables

- Y=VENTA, dicotómica (0 NO , 1 SI)
- X1= RPM, dicotómica (0 NO , 1 SI)
- X2= VOZ, categórica (BAJO, MEDIO_BAJO, MEDIO_ALTO, ALTO)
- X3= DATOS, dicotómica (0 NO , 1 SI)
- X4=LIMA_PROV, dicotómica (0 Provincia, 1 Lima)
- X5=ANTIGÜEDAD, dicotómica (0 Nueva, 1 Antigua)
- X6=PROM_RECARGA, promedio de recarga de los últimos 3 meses de la línea.

3.5. Población

Se cuenta con la información de 2,104,842 datos a partir de octubre 2016 a marzo 2017. Todos estos datos fueron considerados la población en estudio.

3.6. Muestra

En este proyecto no se dispone de una muestra aleatoria porque se utilizarán todos los datos de la población por decisión de la empresa.

3.7. Alcance

El presente proyecto se desarrollará en una empresa de Call Center con los datos de octubre 2016 a marzo 2017. Los resultados del proyecto son válidos únicamente para esta empresa y no se pueden generalizar porque no se utiliza una muestra aleatoria ni técnicas de inferencia estadística.

3.8. Regresión logística

La ecuación de la regresión logística es,

$$log\ log\ \left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + \beta_6 X_{6i} \qquad i = 1, \dots, 2', 104{,}182$$

La interpretación de sus coeficientes es,

π: probabilidad de que se realice la venta. En este caso π=p(x) de la ecuación 1 del capítulo II.

®$_0$: representa el cambio en el logaritmo de la razón $\pi/(1-\pi)$ cuando todas las variables predictoras son iguales a cero.

®$_i$: representa el cambio en el logaritmo de la razón $\pi/(1-\pi)$ cuando la variable X_i cambia en una unidad

La variable X_i representa el valor de la variable regresora correspondiente que se considera fija.

La estimación puntual y la construcción de los intervalos de confianza serán calculados con el software estadístico R (R Core Team, 2016).

Teniendo en consideración los datos previamente mencionados, así como las respectivas variables y su operacionalización, se procederá a explicar la obtención de los resultados.

3.9. Pasos en un modelamiento predictivo

Para llegar a la solución final del trabajo monográfico se realizó una serie de pasos que pueden variar según el criterio del investigador para obtener los mejores resultados, a continuación, se detallarán los pasos y acciones realizadas:

3.9.1. Necesidad de mejora y sinergia

El call center en mención ya contaba con un algoritmo (regresión logística) que permitía clasificar los clientes potenciales para realizarles acciones de colocación de planes; sin embargo, la respuesta de dichos clientes contactados no era la esperada y conforme avanzaba la gestión del CC los clientes del operador telefónico mostraban mayor rechazo a la propuesta de venta. Asimismo, las diferentes áreas involucradas en dicha gestión, ya sea desde dentro el CC como desde el OT no mostraban un conocimiento certero del cliente para llegar a él con la mejor oferta de acuerdo a sus necesidades. Es por ello que, en pro de mejorar

la contactabilidad y por ende la colocación de planes se planteó un algoritmo de máquina de aprendizaje (adaboosting) que por su naturaleza se pensó podría mostrar mejores resultados a nivel de modelamiento predictivo. Involucrando las diferentes áreas de gestión y alineando necesidades no solo se logró implementar un modelo con mejores resultados sino conocer al cliente y llegar con el correcto discurso a él.

3.9.2. Fuentes de información

Una vez realizada la identificación de la necesidad y la sinergia entre las áreas involucradas se pasó a traducir dichas necesidades en variables que permitan explicar el comportamiento de los clientes potenciales, se estructuró la información, se recolectó diversas fuentes de datos (OT, CC, feedback de gestión, gestión en tiempo real, entre otras) y se realizó el tratamiento de la información hasta llevarla al nivel deseado. Finalmente se obtuvo un tablón de datos con diferentes variables dentro del cual se encontraron aquellas que permitan discriminar mejor al momento del modelamiento. En esta parte se puede incluir un análisis exploratorio de los datos vs la variable objetivo, de tal manera que permita ir generando perspectivas útiles al negocio y al investigador.

3.9.3. Selección de variables

Una vez realizados los pasos previos y contando con un número de variables menor al de los datos (+2 millones) se procedió a la selección de variables, este paso se puede realizar con diferentes acciones: poblamiento de variables (si la variable supera el 10% de datos entra como parte de la selección), criterio de información (que permite saber la contribución de la variable independiente hacia la variable dependiente), selección de % de aporte a través de random forest, criterio de experto, entre otros. Para el presente trabajo monográfico se realizó el uso del poblamiento de variables y el criterio de información.

3.9.4. Modelamiento e imputación de datos

Se consideraron un total de 2,104,842 datos para realizar el presente trabajo monográfico. Este total de datos se partió en 3 grupos:

- Data de entrenamiento: constó de 1,441,120 registros, que se dividieron en dos subconjuntos:
  - Data train (D1): donde se realizaron las acciones de poblamiento de variables (verificando que las variables tengan al menos más del 10% de datos para

mantenerse), criterio de información e imputación (ya sea por moda si es una variable del tipo factor, media si es una variable del tipo entera o numérica). Con una semilla (set.seed (1234)) para otorgar un 70% del total de datos.

  - Data test (D2): donde se realización las acciones de poblamiento de variables (verificando que las variables tengan al menos más del 10% de datos para mantenerse), criterio de información e imputación (ya sea por moda si es una variable del tipo factor, media si es una variable del tipo entera o numérica). Con una semilla (set.seet(1234)) para otorgar un 30% del total de datos.

- Data de prueba (D3): constó de 348,642 registros, las acciones sobre esta data de prueba son iguales a la data de entrenamiento con la diferencia que en ésta no se realizaron particiones.
- Data de validación (D4): constó de 315,000 registros, esta data se tomó pura (sin realizarle ningún tipo de acción excepto las de imputación) para mediar el real impacto de los modelos utilizados y su estabilidad en el tiempo.

Los modelos de regresión logística y adaboosting aprendieron del comportamiento de la data train (D1) y se evalúo en una data test (D2), para que, posteriormente se puedan validar en una data de prueba (D3) sin particiones y con acciones de poblamiento, criterio de información e imputación; y, finalmente, en una data de validación (D4) para encontrar la estabilidad de los modelos probados.

La regresión logística se trabajó con la función glm de R y con los parámetros de la familia binomial (link=logit), el valor del área bajo la curva AUC que se obtuvo fue de 0.6873.
El algoritmo de adaboosting se trabajó bajo el modelo 1 y el valor obtenido para el área bajo la curva fue de 0.8885.

3.9.5. Balanceo de datos

Debido a la naturaleza de los datos y al comparar ambos algoritmos en igual de condiciones, los datos trabajados en el presenta trabajo monográfico no fueron balanceados. Sin embargo, a criterio del investigador se recomienda utilizar un balanceo por undersampling, de tal manera que esta técnica le permita mantener la naturaleza de la clase de éxito dentro de la población o muestra y llevar a la categoría del fracaso al mismo volumen respetando el comportamiento de dicha categoría.

3.9.6. Validación

Con los 5 pasos previos ejecutados (en caso corresponda) se validó el resultado obtenido para el algoritmo de adaboosting, midiendo su estabilidad y predictibilidad en el tiempo, en el caso del presente trabajo monográfico se realizó la validación cerrada en 1 mes; sin embargo, se recomiendo realizar la validación hasta en 3 meses de gestión para probar que los resultados obtenidos mantengan un comportamiento correcto y/o detectar patrones extraños de comportamiento. Se sugiere también, agregar técnicas de auto aprendizaje de tal manera que permita al algoritmo auto recalibrarse cuando los datos cambien su comportamiento y no se tenga que recurrir a nuevo proceso de modelamiento, sino que el investigador sea parte del mismo a través de la supervisión e inclusión de nuevas ideas.

3.9.7. Implementación

Finalmente, se procedió a la puesta en marcha de la solución analítica desarrollada que puede ir desde ejecutar el proceso cada vez que se requiera hasta la automatización del mismo para evitar carga operativa en el área de trabajo y que el investigador se pueda enfocar en seguir retando modelos pre existentes, crear nuevos modelos y/o soluciones analíticas que no necesariamente impliquen la participación de una variable dependiente.

IV. RESULTADOS Y DISCUSIÓN

A continuación, se mostrarán los resultados obtenidos para la aplicación de cada algoritmo en mención y se procederá a discutir los resultados: Los resultados obtenidos de la aplicación de la regresión logística son los siguientes:

Tabla 2

Tabla de clasificación en base a los resultados de la Regresión Logística

Grupo Real	Grupo estimado	
	0	1
0	480,373	240,187
1	80,062	640,498

De la tabla de clasificación presentada se obtienen los siguientes resultados:

- Precisión: 73%
- Accuracy o exactitud: 78%
- Sensibilidad o recall: 89%
- Especificidad: 67%
- AUC: 0.68731
- Tasa de mala clasificación: 22%

Los resultados obtenidos de la aplicación del algoritmo adaboosting son los siguientes:

Tabla 3

Tabla de clasificación en base a los resultados del Algoritmo Adaboosting

Grupo Real	Grupo estimado	
	0	1
0	600,467	80,062
1	120,093	640,498

De la tabla de clasificación presentada se obtienen los siguientes resultados:

- Precisión: 89%
- Accuracy o exactitud: 86%
- Sensibilidad o recall: 84%
- Especificidad: 88%
- AUC: 0.88855
- Tasa de mala clasificación: 14%

Lo que se puede observar es una fuerte mejoría en la clasificación de cliente no contactables en el algoritmo adaboosting vs la regresión logística lo cual permite generar grupos de ejecución más limpios y puros para que cuando se procedan a gestionar el primer contacto sea exitoso, el barrido disminuya y el cliente que no desea el producto no se encuentre entre los primero deciles de contacto.

V. CONCLUSIONES

Los resultados obtenidos gracias a la aplicación de ambos algoritmos dejan en evidencia que:

- De acuerdo al objetivo general del presente trabajo monográfico, el algoritmo de adaboosting muestra resultados más favorables para la clasificación de cliente potenciales de un operador telefónico contactados por un call center; adicionalmente a lo comentado, se busca dejar en claro que ningún algoritmo es mejor que otro, sino que por la naturaleza y comportamiento de los datos se pensó que un algoritmo adaboost podría presentar mejores resultados que una regresión logística lo cual ha quedado en evidencia.
- Sobre los objetivos específicos, el algoritmo adaboosting ha logrado ubicar en sus primeros 5 deciles de gestión +45% de clientes potenciales a recibir una gestión y aceptar el plan ofrecido versus la regresión logística que logró ubicar en sus primeros 5 deciles +20% de clientes potenciales a recibir una gestión de aceptar el plan ofrecido.
- Bajo la comparativa del AUC, el algoritmo adaboosting presenta un valor de 89% mientras que la regresión logística presentó un valor de 69%, lo cual determina que el algoritmo de adaboosting brinda una mejor clasificación respeto a la regresión logística.

VI. RECOMENDACIONES

Finalmente, se recomienda a los investigadores tener en consideración la metodología empleada como base de un modelamiento predictivo e ir aportando con nuevas formas de selección de variables, algoritmos nuevos que tengan una interpretabilidad fácil de negocio, mantener la curiosidad por descubrir nuevas metodologías y pensar siempre en entorno de productivización de tal manera que permita ir siempre añadiendo nuevos modelos, segmentaciones, perfilamientos, entre otras soluciones analíticas para el desempeño día a día.

VII. REFERENCIAS BIBLIOGRÁFICAS

Cerda, J. y Cifuentes, L. (04 de 2012). *Using ROC curves in clinical investigation. Theoretical and practical issues.* Obtenido de https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0716-10182012000200003#img02

Dangeti, P. (2017). *Statistics for Marchine Learning.* Birmingham - Mumbai: Packt.

Gareth, J., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R.* New York: Springer.

GESTIÓN - Economía - Empresas. (14 de 06 de 2017). *GESTIÓN - EMPRESAS*. https://gestion.pe/economia/empresas/telefonia-movil-peru-movistar-pierde-participacion-entel-bitel-suben-137304-noticia/?ref=signwall

Hernández Sampieri, R. (2010). *Metodología de la Investigación-Quinta Edición.* Mexico: McGraw Hill.

IArtificial.net. (21 de 09 de 2020). Obtenido de https://www.iartificial.net/regresion-logistica-para-clasificacion/

Kak, A. (2016). *AdaBoost for Learning Binary and Multiclass Discriminations.* Estados Unidos: Purdue University.

Kapoor, A. (01 de 07 de 2019). *https://github.com/akshaykapoor347/Compute-AUC-ROC-from-scratch-python/blob/master/AUCROCPython.ipynb*. https://github.com/akshaykapoor347/Compute-AUC-ROC-from-scratch-python/blob/master/AUCROCPython.ipynb

Martinez Heras, J. (20 de 09 de 2020). *IArticial.net*. https://www.iartificial.net/gradiente-descendiente-para-aprendizaje-automatico/

Orellana Alvear, J. (12-16 de noviembre de 2018). *Bookdown.* https://bookdown.org/content/2031/arboles-de-decision-parte-i.html

Pérez Tatamués, E. (11 de 12 de 2019). *Algoritmo de random forest aplicado a la detección de fraude en el sistema bancario ecuatoriano.* Ecuador: Quito,2019.

R Core Team. (December de 2016). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*. Obtenido de https://www.R-project.org/

Rokach, L. & Maimon, O. (2014). *Data Mining with Decision Trees (Theory and Applications) 2nd Edition.* Singapore: World Scientific.

Salcedo Poma, C.M. (2002). *UNMSM. Biblioteca de la Facultad de Ciencias Matemáticas.* Obtenido de https://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Basic/Salcedo_pc/enPDF/Cap2.PDF

VIII. ANEXOS

En esta sección se mostrarán algunos de los códigos empleados para el desarrollo del presente trabajo monográfico:

Se instalan algunas librerías que ayudaron no solo al modelamiento sino al entendimiento de las variables.

```
library(ParamHelpers)
library(party)
library(pillar)
library(ggplot2)
library(sqldf)
library(ggvis)
library(party)
library(Boruta)
library(pROC)
library(randomForest)
library(e1071)
library(caret)
library(glmnet)
library(mboost)
library(adabag)
library(xgboost)
library(ROCR)
library(C50)
library(mlr)
library(lattice)
library(gmodels)
library(gplots)
library(DMwR)
library(rminer)
library(polycor)
library(class)
library(neuralnet)
```

Se procede a cargar los datos:

```
## Cargamos los datos
data1<-read.delim("clipboard",h=T)
dim(data1)
head(data1)

data2<-read.delim("clipboard",h=T)
dim(data2)
head(data2)

data3<-read.delim("clipboard",h=T)
dim(data3)
head(data3)

## Juntamos todos los datos cargados
dat_com<-rbind(data1,data2,data3)
dim(dat_com)

str(dat_com)
```

Se toman los datos de la base de entrenamiento (como ejemplo) y se seleccionan los campos:

```
##############################
######### BASE TRAIN #########
##############################

## Seleccionamos la base train

library(tidyverse)

base_train <- dat_com %>% filter(BASES=="PARTE 1"|BASES=="PARTE 2"|BASES=="PARTE 3"|BASES=="PARTE 4")
dim(base_train)
str(base_train)
head(base_train)

##Seleccionando solo los campos necesarios para modelar TRAIN

head(base_train)
base_train1<-base_train[,c(2:11)]

head(base_train1)
```

Se realiza una imputación en la base de entrenamiento (como ejemplo):

```
# IMPUTACION DE LA DATA DEPENDIENDO DE LA INFORMACION Y TIPO DE VARIABLES (TRAIN)
library(mlr)
#install.packages("BBmisc")
library(BBmisc)
train_parametrica1<- impute(base_train1, classes = list(factor = imputeMode(),
                                                        integer = imputeMean(),
                                                        numeric = imputeMean()),
                            dummy.classes = c("integer","factor"), dummy.type = "numeric")


train_parametrica1=train_parametrica1$data[,1:min(dim(base_train1))]
# REVISANDO LA IMPUTACION
resumen_train=data.frame(summarizeColumns(train_parametrica1))
write.csv(resumen_train,"Resumen_Variables1_Train.csv")
```

Poblamiento de las variables en base de entrenamiento:

```
# ##########################################################################################################
# ################################## POBLAMIENTO DE LAS VARIABLES (TRAIN) ##################################
# ##########################################################################################################
#
FPPV<-function(base,dnum,dcat) {
  PPV<-data.frame()
  #crear listas de variables segun tipo
  var_num<-sapply(base, is.numeric)
  var_cat<-sapply(base, is.factor)
  #Variables Numéricas
  data_num<-base[,var_num]
  ctd_num<-nrow(data.frame(colnames(data_num)))
  list<-data.frame(colnames(data_num))
  PPVN<-data.frame()
  for (i in 1:ctd_num)
  {
    var<-data.frame(data_num[,i])
    var_name<-list[i,]
    a<-nrow(data.frame(var[var!=dnum,]))/nrow(data.frame(var[]))
    ff<-data.frame(pp=a, var_name=var_name)
    PPVN<-rbind(PPVN,ff)
  }
  #Variables Categóricas
  data_cat<-base[,var_cat]
  ctd_cat<-nrow(data.frame(colnames(data_cat)))
  list<-data.frame(colnames(data_cat))
  PPVC<-data.frame()
```

```
  for (i in 1:ctd_cat)
  {
    var<-data.frame(data_cat[,i])
    var_name<-list[i,]
    a<-nrow(data.frame(var[var!=dcat,]))/nrow(data.frame(var[]))
    ff<-data.frame(pp=a, var_name=var_name)
    PPVC<-rbind(PPVC,ff)
  }
  write.csv(PPVN,"Porcentaje_poblamiento_NUM_Train.csv")
  write.csv(PPVC,"Porcentaje_poblamiento_CAT_Train.csv")

  PPV<-list(PPVN,PPVC)
  return(PPV)
  rm(data_num, data_cat)
}

PPV<-FPPV(train_parametrica1,dnum=2,dcat=" ")

str(train_parametrica1)
```

Criterio de información en base de entrenamiento:

```
## MEDIANTE EL CRITERIO DE GANANCIA DE INFORMACION , SELECCIONAMOS LAS VARIABLES MAS IMPORTANTES ##


#Análisis de las variables para encontrar patrones de predicción
#install.packages("woe")
#install.packages("stringr")
#install.packages("Information")
#install.packages("gridExtra")
library(woe)
library(stringr)
library(Information)
library(gridExtra)

## selección de variables   - ojo la variable target debe estar en formato numero
train_parametrica1$VENTA <- as.numeric(train_parametrica1$VENTA)

class(train_parametrica1$VENTA)
#ya es NUMERIC

IV <- create_infotables(data=train_parametrica1, y="VENTA", ncore=2)
IV<-as.data.frame(IV$Summary)
write.csv(IV,'Modelo_Atento.csv')

## CONSIDERAMOS SÓLO LAS VARIABLES CON MÁS DEL 1% CRITERIO DE INFORMACION ## (TRAIN)
## POBLAMIENTO DE VARIABLES ##
train_parametrica2=subset(train_parametrica1,select=c('NRO_LINEA_INCLUIDA',
                                                      'VOZ','RPM','PROM_RECARGA',
                                                      'TECNOLOGIA','DATOS','LIMA_PROV',
                                                      'VENTA'))
```

Prueba en la regresión logística en base train:

```
#############################################################################
########################## MODELADO DE LA DATA ##############################
#############################################################################

## REGRESIÓN LOGÍSTICA
############### MODELO DE ENTRENAMIENTO ############################

mod_train01<-glm(VENTA~RPM+VOZ+DATOS+TECNOLOGIA+LIMA_PROV+PROM_RECARGA,data=data.train,family=binomial(link=logit))
summary(mod_train01)

## Clasificación

# Probabilidades
predicted.mod1<-predict(mod_train01,type="response",newdata=data.test)

# Tabla de clasificación
mod_predic1<-ifelse(predicted.mod1>=0.5,"Yes","No")
table(mod_predic1,data.test$VENTA)

##probabilidades
proba1=ifelse(predicted.mod1>=0.5,1,0)
head(proba1)
```

Principales indicadores hallados:

```
# curva ROC
library(pROC)
AUC1 <- roc(data.test$VENTA, proba1)
auc_modelo1=AUC1$auc

table(proba1)
table(data.test$VENTA)

# Gini
gini1 <- 2*(AUC1$auc) -1

# Calcular los valores predichos
PRED <-predict(mod_train01,type="response",newdata=data.test)

# Calcular la matriz de confusion
library(caret)
library(e1071)
table(data.train$PRUEBA)
table(data.test$PRUEBA)

data.train$VENTA <- as.factor(data.train$VENTA)
data.test$VENTA <- as.factor(data.test$VENTA)

tabla=confusionMatrix(PRED,data.test$VENTA,positive = "1")

# sensibilidad
Sensitivity5=as.numeric(tabla$byClass[1])

# Precision
Accuracy5=tabla$overall[1]

# Calcular el error de mala clasificaci?n
error5=mean(PRED!=data.prueba$TARGET)
```

Prueba bajo el algoritmo adaboosting:

```
##############################################################################
########################### MODELADO DE LA DATA ##############################
##############################################################################

## ADABOOSTING
############### MODELO DE ENTRENAMIENTO ##############################

library(adabag)
adaboost <- bagging(VENTA~RPM+VOZ+DATOS+TECNOLOGIA+LIMA_PROV+PROM_RECARGA,
                    coeflearn='Freund',
                    data=data.train, mfinal=20)
yhat_adaboost <- predict(adaboost, newdata=data.train)$class
tabla_adaboost <- table(yhat_adaboost, data.train$VENTA)
```

Extracción de la información para validación y puesta en marcha:

```
##############################################################################################
###### Para juntar la data con los campos que reconocen a los clientes

Score_01 <- data.frame(cbind(CODMES = base_val$CODMES, telefono = base_val$telefono, PROB = yhat_adaboost))
write.csv(Score_01,"Score_01.csv",row.names=FALSE, quote=T)
```