MAESTRÍA EN CIENCIA DE LOS DATOS Y ANALÍTICA

Trabajo final

ANALISIS Y PREDICCIÓN DE RECONTACTOS EN UN CONTACT CENTER

AUTOR

DIANA CATALINA VELÁSQUEZ GAVIRIA

201919006228

Estudiante

TUTOR

OLGA LUCÍA QUINTERO

Aprendizaje Automático

Profesora

UNIVERSIDAD EAFIT

2020

Análisis y Predicción de recontactos en un contact center

Resumen

El propósito principal de este trabajo es resolver de manera metódica y formal, haciendo uso de modelos de aprendizaje automático, un problema real del sector productivo que permita además de agregar valor para la toma de decisiones, proveer una metodología y un modelo compacto, simple y confiable que pueda ser desplegado y puesto en producción en la plataforma tecnológica que soporta la atención de llamadas de un centro de contactos de tal forma que puedan generarse beneficios en la prestación del servicio para diferentes sectores, generando eficiencias en el uso del canal y maximizando la experiencia del cliente en la atención de sus requerimientos. Para lograr este propósito se tomó un conjunto de datos de una aerolínea que contiene el detalle de todas las llamadas históricas que han realizado los clientes a un centro de contactos durante un período de 7 mes (febrero a agosto de 2019) e información asociada al desempeño de los agentes que atienden dichas llamadas con el fin de predecir si los usuarios generarán al menos un recontacto al contact center para la atención de sus requerimientos antes de tres días con respecto a su contacto inicial. La metodología utilizada se centró en realizar una adecuada selección de características y escoger un modelo de aprendizaje automático que genere óptimos resultados y posibilite una fácil implementación permitiendo identificar en tiempo real aquellos clientes con altas probabilidades de volver a comunicarse, de tal forma que pueda desarrollarse una estrategia con ellos que mejore su experiencia. Se encontró que con 7 variables asociadas al comportamiento histórico de los clientes en el uso del canal tales como frecuencia de llamadas (monto), duración promedio, cantidad de agentes que han atendido al cliente (agentes), tiempo transcurrido entre primera y última llamada (vigencia), cantidad de días del mes en los que se realizan las llamadas (PromedioDiasEnMes), promedio de llamadas por día (Promedio-Diario) y tiempo transcurrido desde la última llamada del cliente hasta la fecha de corte del análisis (Recencia) es posible predecir con un modelo sencillo y con resultados muy buenos (AUC=88.9 %) si un cliente volverá a comunicarse al centro de contactos.

Abstract

The main purpose of this work is to solve in a methodical and formal way, making use of machine learning models, a real problem of the productive sector that allows in addition to adding value for decision making, to provide a methodology and a compact, simple and reliable model that can be deployed and put into production in the technological platform that supports the call handling of a contact center so that benefits can be generated in the provision of the service for different sectors, generating efficiencies in the use of the channel and maximizing the customer experience in the attention of their requirements. To achieve this purpose, an airline dataset was taken containing the detail of all historical calls made by customers to a contact center during a period of 7 months (February to August 2019) and information associated with the performance of the agents handling those calls in order to predict whether users will generate at least one recontact to the contact center for the attention of their requirements before three days with respect to their initial contact. The methodology used was focused on making an appropriate selection of features and choosing a machine learning model that generates optimal results and enables an easy implementation allowing to identify in real time those customers with high probabilities of recontacting so that a strategy can be developed with them to improve their experience. It was found that with 7 variables associated with the historical behavior of customers in the use of the channel such as frequency of calls (amount), average duration, number of agents who have served the customer (agents), time elapsed between first and last call (validity), number of days per month in which calls are made (AverageDaysInMonth), average number of calls per day (Average-Day) and time elapsed since the customer's last call to the cut-off date of the analysis (Recency), it is possible to predict with a simple model and with very good results (AUC=88.9 %) whether a customer will call the contact center again.

Palabras Clave Contact center, Individual Customer's Call Arrival, rellamados, eficiencia, machine learning, features selection, AUC, f1_score.

Índice general

Índice de figuras

Introducción

1.1. Centros de contacto

Los centros de contacto representan un papel crítico para las compañías prestadoras de servicios, ya que actúan como un intermediario entre ellas y sus clientes, es por esto que una forma de acercarse a ellos es ofreciendo un servicio personalizado y múltiples y estructurados canales de comunicación. La excelencia en la prestación del servicio y el entendimiento de las individualidades de sus clientes hacen que este acercamiento sea positivo. Esto permite que a través de los centros de contacto se puedan recolectar una gran cantidad de datos de las interacciones con sus clientes que luego puedan ser aprovechados para mejorar los procesos del negocio, identificar deficiencias en el servicio, y predecir las necesidades de sus clientes. (Andrade et al., 2020,).

Consecuentemente los contact center han sido incrementalmente reconocidos por su rol en el aseguramiento de la satisfacción del cliente y debido a su posición estratégica en el relacionamiento con el cliente y a los altos costos que involucran han sido ampliamente estudiados por diferentes áreas de investigación, muchas de ellas orientadas a reducir los costos operacionales y otras a entender el comportamiento de sus clientes y su relación con ellos. (Andrade et al., 2020,).

1.2. Problema de investigación

Identificar si es posible predecir utilizando técnicas de aprendizaje automático, si un cliente volverá a comunicarse a un canal telefónico en un período determinado de tiempo para la solución de un requerimiento, con base en su comportamiento histórico de llamadas o contactos

a través de dicho canal y en caso de si ser posible, identificar las variables más relevantes para tal fin.

Las variables obtenidas a partir de los registros almacenados por la plataforma son:

1. **Monto**: Cantidad de llamadas totales por cliente
2. **FrecuenciaDia:** Cantidad de días usados para realizar el total de llamadas
3. **PromedioDiario:** Llamadas promedio por día
4. **FrecuenciaMes:** Número de meses del periodo de estudio que se usaron para realizar las llamadas.
5. **PromedioDiasEnMes:** Cantidad promedio de días usados en un mes para realizar llamadas.
6. **Vigencia:** Tiempo transcurrido entre la primera y última llamada (días)
7. **Recencia:** Tiempo desde la última llamada a la fecha de corte del período analizado
8. **DuracionPromedio:** Duración promedio de las llamadas realizadas por el cliente.
9. **Consumo:** Segmentación realizada con base en el monto y la duración promedio.
10. **Reiteratividad:** Segmentación realizada con base en las llamadas realizadas en promedio por día y en un mismo mes.
11. **Afinidad:** Segmentación realizada con base en el monto y la frecuenciaMes
12. **periodos-con-recontacto:** En cuantos periodos se presentó recontacto
13. **Transacciones-recontacto:** Cantidad de transacciones que fueron recontacto.
14. **transaccion-promedio-por-recontacto:** Transacciones en promedio por cada recontacto.
15. **diferencia 1-2 Horas-Recontacto:** Tiempo transcurrido entre la primera y segunda llamada que comprende un recontacto.
16. **periodos-con-rellamado:** en cuantos períodos se presentó rellamado
17. **Transacciones-rellamado:** Cantidad total de transacciones por rellamado

18. **transaccion-promedio-por-rellamado:** Cantidad promedio de transacciones por rellamado.
19. **diferencia 1-2 Horas rellamado:** Tiempo transcurrido entre la primera y segunda llamada que comprende un rellamado.
20. **agentes:**: Cantidad promedio de agentes que atendieron a una persona.
21. **scoreic:**: Equivale al promedio del índice de calidad de los agentes que atendieron una persona.
22. **scoreaht:** Equivale al promedio de la nota obtenida en su desempeño con respecto a la duración de llamada de los agentes que atendieron una persona.
23. **scoreadh:** Equivale al promedio de la nota obtenida en su desempeño con respecto a la adherencia a su turno de los agentes que atendieron una persona.

1.3. Objetivo General

Analizar el detalle histórico de las llamadas de los clientes en un canal telefónico de servicio al cliente con el fin de entender su comportamiento e identificar si es posible predecir qué clientes volverán a comunicarse, para con esta información poder definir estrategias que permitan mejorar la experiencia, evitar o disminuir los recontactos dando solución en la primera llamada o definiendo estrategias proactivas de call-back o información adicional a través de otros medios digitales. [1].

1.4. Objetivos Específicos

- Analizar la base de datos seleccionada (ver capitulo 3)
- Identificar los recontactos históricos para construcción de la variable respuesta y del conjunto de entrenamiento (ver capitulo 3)
- Evaluar y seleccionar las variables y espacios de características del problema de aprendizaje (ver capitulo 3)

[1] Como por ejemplo envíos de mensajes de texto con información complementaria asociada al requerimiento. Ejemplo: estado de una queja, estado de un pedido, pasos a seguir en un procedimiento, confirmación de aplicación de un pago, entre otros

- Plantear el problema de aprendizaje supervisado (ver capitulo 4)
- Solucionar el problema de aprendizaje con una máquina apropiada. (ver capitulo 4)

1.5. Metodología

Para dar solución a esta necesidad se propone aplicar técnicas de aprendizaje supervisado sobre una base de datos de 421.055 registros y 23 variables (descritas en el problema de investigación), recopilada por un período de 7 meses con el detalle de las llamadas históricas de clientes de un servicio determinado del contact center. Esta información es obtenida directamente de los logs generados por la planta telefónica. Para llevar a cabo este proceso se plantean los siguientes pasos:

- Realizar un entendimiento y descripción de los datos a partir de un análisis descriptivo de los mismos y una revisión de las variables de la base de datos.
- Construir la variable respuesta utilizando los datos detallados de llamadas a partir de la variable 'Transacciones_recontacto' la cual permite identificar si un cliente generó o no recontactos en el período de análisis.
- Analizar el comportamiento de la variable respuesta con respecto a las demás variables de estudio comparando sus medidas descriptivas para cada nivel de la variable recontactos.
- Separar la base de datos en los conjuntos de entrenamiento y validación para el entrenamiento de los modelos y la optimización de los hiperparámetros y dejar un conjunto de prueba para evaluar la capacidad de generalización del modelo.
- Estandarizar los datos para realizar el entrenamiento de los modelos de aprendizaje automático.
- Seleccionar las características que mayor influencia tengan sobre la variable respuesta a partir de diferentes métodos de selección de variables, tales como métodos de filtro (selección univariada como ROC -AUC), métodos de envoltura (utilizando algoritmos predictivos de aprendizaje de máquinas para escoger el subconjunto con mejor resultado

tales como selección hacia adelante), métodos incrustados (tales como lasso, lars, elasticnet, tree importance, etc) y métodos bayesianos (como spike and slab), los cuales se describirán en el marco teórico.

- Entrenar los modelos de aprendizaje de máquinas que serán evaluados para predecir el recontacto. Se entrenan 5 modelos de aprendizaje supervisado: Regresión logística, árbol de decisión, bosque aleatorio, gradient boosting y red neuronal multicapa perceptrón, estos modelos de igual forma se describen en el marco teórico. Estos modelos se entrenan con dos particiones diferentes de los datos en entrenamiento, validación y prueba así: 60 %-20 %-20 % y 50 %-20 %-30 % con el fin de analizar el impacto en los resultados al variar dicha partición incrementando el conjunto de prueba.

- Entrenar los 5 modelos con cada uno de los conjuntos de variables seleccionados por las diferentes técnicas para analizar los modelos que presentan menor error, también se entrenan estos 5 modelos con algunos conjuntos de interés como el de las variables netamente transaccionales que se obtienen de una misma fuente de datos que corresponde a la plataforma tecnológica de llamadas, otro con las variables que mayor coincidencia en la selección presenten la mayoría de los métodos y adicionalmente entrenar los modelos sin las variables más importantes para evaluar su impacto en el error de predicción. Esta experimentación se realiza con el fin de ver el comportamiento de los errores al incluir o eliminar variables que pudieran ser de difícil obtención y poder escoger el modelo más parsimonioso posible, teniendo en cuenta que este ejercicio puede replicarse para otros servicios del centro de contactos y estos podrían tener mayores volúmenes de registros.

- Optimizar los hiperparámetros de los modelos entrenados, realizando búsqueda de los mismos por cuadrícula, utilizando la librería GridSearchCV de python con el fin de variar los hiperparámetros para escoger el mejor modelo y realizando validación cruzada para minimizar problemas de sobreajuste.

- Seleccionar el modelo de aprendizaje automático que menor error y mejor capacidad de aprendizaje presente realizando un comparativo en el resultado con diferentes métricas como AUC y el F1_score, de todos los modelos.

- Realizar ajustes en la selección de variables validando con otras técnicas como Permutation Feature Importance, seleccionar el mejor conjunto y utilizar técnicas de balanceo con submuestreo y sobremuestreo para mejorar el desempeño del modelo.

- Seleccionar el modelo definitivo y más parsimonioso posible para ser utilizado en producción, con base en el análisis de otras métricas como *Precisión* y *Recall* que pueden ser de mayor interés en este problema acorde a las características del negocio.

1.6. Justificación

Poder predecir si un cliente volverá o no a llamar a una compañía (a través de un contact center como su canal de comunicación), puede ser útil para definir diferentes estrategias en la prestación del servicio que permitan reducir costos operacionales, mejorar la experiencia del usuario o incluso realizar estrategias de retención de clientes.

En el primer caso, al evitar recontactos o rellamados innecesarios (los cuales pueden representar entre un 10 % y un 20 % de las llamadas del canal de acuerdo con datos históricos de la compañía) que pudieron resolverse en la primera llamada, o que podrían ser atendidos por otros medios virtuales e incluso que de manera proactiva se pudiera entregar la información al cliente sin que este tenga que volver a comunicarse, podría optimizar los costos en el canal o incluso aumentar su capacidad para atender nuevas necesidades.

En el segundo caso, identificar aquellos clientes con altas probabilidades de volver a comunicarse puede servir para definir estrategias de enrutamiento inteligentes, de tal forma que dichas llamadas sean atendidas por agentes de alto rendimiento y de esta manera garantizar la prestación de un servicio más personalizado y poder satisfacer mejor las necesidades de los clientes, como lo menciona (Moazeni and Andrade, 2018,).

Por último, identificar aquellos clientes que han dejado de llamar podría ser un indicio de que han cancelado el servicio o se han ido donde otro competidor, lo cuál como lo menciona (Butgereit, 2020,), es un problema costoso para las empresas porque obtener nuevos clientes es más costoso que retener a los clientes existentes por lo que definir oportunamente estrategias de retención podría ser de mucho valor para las compañías.

Estado del arte y marco teórico

2.1. Estado del arte

La literatura relacionada con la predicción de llamadas en los contact center es escasa y se centra principalmente en la estimación de la intensidad de llegadas de las llamadas al centro de contactos, tomando como base las llamadas históricas, otros modelos se centran en predecir las llamadas en intervalos de tiempo definidos (intervalos de media hora, días, meses, etc) para realizar el dimensionamiento de recursos físicos y tecnológicos para la prestación del servicio y para esto utilizan modelos de regresión lineal de series temporales de efecto fijo. (Moazeni and Andrade, 2018,).

Hay pocos estudios relacionados con la predicción de llamadas a nivel individual, algunos buscan identificar las variables que influyen en el uso de otros canales utilizando igualmente modelos de regresión lineal, y unos pocos buscan predecir la probabilidad de recibir una llamada de un cliente en los días siguientes, a partir de la información histórica de las últimas consultas telefónicas y de características como el segmento del cliente, la recencia y frecuencia de sus interacciones y sus actividades on-line, entrenando redes neuronales para dicho propósito y alcanzando en este último caso resultados en el AUC cercanos al 76 %. Se ha encontrado además evidencia de que estas variables impactan la probabilidad de que un cliente vuelva a comunicarse al centro de contactos por un motivo particular, como lo mencionan (Moazeni and Andrade, 2018,).

En este estudio se desea realizar una predicción a nivel individual de las llamadas realizadas por los clientes en los siguientes tres días, asociadas a un motivo de contacto previo, tomando como base las llamadas históricas de los clientes del canal telefónico y los resultados de desempeño de los agentes que atienden dichas llamadas, de tal forma que sea posible su

implementación en tiempo real una vez ingresa la llamada para que se pueda tomar una acción inmediata y poder así disminuir estos recontactos que representan costos operacionales para la compañía y que pueda ser lo más transversal posible para desplegarlo en los diferentes tipos de servicios que atiende el centro de contactos.

2.2. Marco teórico

2.2.1. Aprendizaje Automático

La ciencia del aprendizaje juega un papel importante en los campos de la inteligencia artificial, la estadística y la minería de datos. Cada vez se están generando más datos y el trabajo de los estadísticos es darle sentido a ellos detectando patrones, tendencias y entendiendo lo que ellos dicen. A esto se le llama aprender de los datos. El aprendizaje automático es un área de la inteligencia artificial que busca que los computadores desarrollen una capacidad de aprender, detectar patrones o generar conocimiento a partir de los datos. Ejemplos como predecir si un paciente hospitalizado por un ataque al corazón volverá a sufrir un segundo ataque, a partir de datos como la dieta, datos clínicos o demográficos, o predecir el precio de una acción en 6 meses a partir de medidas de rendimiento y datos económicos de la empresa, o identificar factores de riesgo de próstata, basándose en variables clínicas o demográficas, son ejemplos de lo que se conoce como *Aprendizaje Supervisado* que es un tipo de aprendizaje automático donde se tienen unos resultados que pueden ser numéricos o categóricos y se busca predecirlos con base a un conjunto de características. En el caso de *Aprendizaje no Supervisado* se observan solo mediciones de las características pero no se sabe cuál es el resultado y su objetivo es describir como se organizan o agrupan los datos. (Hastie et al., 2009,).

Existen diferentes técnicas de aprendizaje supervisado, si la salida o el resultado que se quiere predecir es numérico, se trata de un problema de regresión y si el resultado es categórico se trata de un problema de clasificación. En este estudio se utilizan algunas técnicas de clasificación para resolver el problema de predicción de los recontactos en el canal telefónico, ya que la respuesta que se tiene es si el cliente se ha vuelto o no a comunicar en un período no superior a 3 días. A continuación se detalla un poco sobre cada uno de los modelos evaluados:

- **Regresión logística:** El modelo de regresión logística se utiliza para predecir el resultado de una variable predictora de tipo categórico en función de unas variables independien-

tes. Es útil para modelar la probabilidad de ocurrencia de un evento con base en unos factores y usa como función de enlace la función logit ya que proviene de la familia de modelos lineales generalizados (GLM). La regresión logística analiza datos distribuidos binomialmente y los logits de las probabilidades binomiales son modelados como una función lineal de los xj. (Wikipedia contributors, 2020,)

$$logit(pi) = \ln(\frac{pi}{1-pi}) = \beta_0 + \beta x_{1,i} + ... + \beta x_{k,i} \tag{2.1}$$

Este modelo proviene de la intención de modelar las probabilidades posteriores de las clases k, mediante funciones lineales x, asumiendo que dichas probabilidades sumen uno y estén dentro del intervalo [0,1]. Este modelo normalmente se ajusta utilizando un estimador de máxima verosimilitud. [1] (Hastie et al., 2009,)

- **Árbol de clasificación:** Los árboles de clasificación son métodos de aprendizaje automático que permiten realizar predicciones a partir de los datos, dichos modelos se obtienen dividiendo recursivamente el espacio de datos y obteniendo un modelo simple dentro de cada partición. Como resultado la partición puede representarse en un árbol de decisión.(Loh, 2011,).

  Con base en lo mencionado por (Buitinck et al., 2013,), algunas ventajas de los árboles de decisión son:

  - Son fáciles de interpretar.
  - No requieren estandarización de lo datos.
  - Pueden manejar tanto datos numéricos como categóricos.
  - El costo computacional es logarítmico.
  - Es robusto, funciona bien incluso si sus supuestos son violados y es posible validarlos usando pruebas estadísticas.

  Y entre sus desventajas están:

  - Pueden presentar problemas de generalización o sobreajuste.

[1] Máxima verosimilitud: es un método analítico utilizado para ajustar un modelo y estimar sus parámetros

- Pequeñas variaciones en los datos pueden generar árboles diferentes.

- **Bosques aleatorios:** Los bosques aleatorios son una combinación de árboles, tales que cada árbol depende de los valores de un vector aleatorio muestreado independientemente y con la misma distribución para todos los árboles del bosque. El error de generalización de un bosque depende de la fuerza de los árboles individuales y su correlación. (Breiman, 2001,).

- **Gradient boosting**:

  La potenciación del gradiente construye modelos de regresión aditivos mediante la vinculación secuencial de una función paramétrica simple (base learner) a los pseudo residuales por mínimos cuadrados en cada iteración. Los pseudo residuales son el gradiente de la función de pérdida que está siendo minimizado, con respecto a los valores del modelo en cada punto de los datos de entrenamiento evaluado en el paso actual. Se ha demostrado que tanto la exactitud de la aproximación como la velocidad de ejecución del aumento del gradiente pueden mejorarse sustancialmente incorporando aleatorización en el proceso. Concretamente, en cada iteración se extrae al azar (sin sustitución) una submuestra de los datos de entrenamiento del conjunto de datos de entrenamiento completo. Esta submuestra seleccionada al azar se utiliza luego en lugar de la muestra completa para ajustar el base learner y calcular la actualización del modelo para la iteración actual. (Friedman, 2002,).

- **Red neuronal perceptrón multicapa (MLP):** La red neuronal perceptrón multicapa es un tipo de red neuronal feed fordward, la cual consiste de al menos 3 capas de nodos: una capa de entrada, una capa oculta y una capa de salida. Excepto para los nodos de entrada, cada nodo es una neurona que utiliza una función de activación no lineal. Las redes MLP utilizan una técnica de aprendizaje supervisado llamada back propagation para su entrenamiento.

2.2.2. Selección de características

Uno de los principales objetivos de este trabajo es seleccionar un subconjunto de características que permitan predecir el recontacto. El análisis de datos de alta dimensión es un reto grande para los investigadores en los campos del aprendizaje automático y la

minería de datos. La selección de características es una solución a este problema ya que permite eliminar datos irrelevantes o redundantes lo que permite reducir los tiempos de cómputo, mejorar la precisión en los modelos de aprendizaje y facilitar la interpretación o comprensión de los modelos y de los datos en general y básicamente consiste en obtener un subconjunto del conjunto de características originales de acuerdo con ciertos criterios de selección para lo cuál existen diferentes métodos. (Cai et al., 2018,).

En este estudio interesa que el modelo pueda ser implementado en tiempo real, en el momento que son recibidas las llamadas en el centro de contactos, esto exige tiempos de respuesta rápidos, ya que una vez ingresa la llamada, el sistema de audiorespuesta debe poder identificar si dicho cliente tiene probabilidad de volver a llamar y en ese mismo momento tomar una acción, cómo por ejemplo enrutar la llamada a un agente experto, por lo tanto realizar una adecuada selección de características es importante.

Existen diferentes técnicas de selección de variables tales como:

- **Información mutua**: Es un método univariado que examina de manera individual cada variable y determina la fuerza de la relación que tiene cada una con la variable respuesta. Esto lo hace midiendo la ganancia de información de una variable X sobre una variable Y. (Gottemukkula and Derakhshani, 2011,). Se basa en el concepto de entropía de una variable aleatoria. (Wikipedia contributors, 2020,).

- **ROC-AUC univariado**: En este caso se construye un árbol de decisión por característica y luego se hace una predicción con dicha característica y se calcula la métrica de AUC. Finalmente se ordenan las características de mayor a menor valor y se seleccionan las de mayor valor.

- **Selección hacia adelante**: Es un método iterativo de selección de variables, que en este caso selecciona las más importantes. Este método comienza evaluando todas las características individualmente y selecciona la que genera el algoritmo de mayor rendimiento, este criterio para clasificación es el ROC_AUC y para la regresión es el R^2. Este método tiene la desventaja de ser muy costoso computacionalmente y

puede no encontrar el mejor modelo.

- **Regresión ridge**: La regresión Ridge reduce los coeficientes de regresión imponiendo una restricción a su tamaño. Los coeficientes minimizan una suma residual penalizada de cuadrados (Hastie et al., 2009,).

$$L_{ridge}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - X_i'\hat{B})^2 + \lambda \sum_{j=1}^{m} \hat{\beta}_j^2 = \| y - X\hat{B} \|^2 + \lambda \| \hat{B} \|^2 \qquad (2.2)$$

Al resolver esto para $\hat{\beta}$ se obtienen los estimadores $\hat{B}_{ridge} = (X'X + \lambda I)^{-1}(X'Y)$ donde I corresponde a la matriz identidad.

$\lambda \geq 0$ es el parámetro de regularización que controla la cantidad de encogimiento (shrinkage) de los coeficientes de la regresión, éstos se encogen hacia cero (y entre sí).

si $\lambda \to 0, \hat{B}_{ridge} \to \hat{B}_{OLS}$;

si $\lambda \to \infty, \hat{B}_{ridge} \to 0$.

Esto significa que cuando $\lambda = 0$ es lo mismo que utilizar mínimos cuadrados ordinarios (OLS en inglés), y cuánto mayor sea λ mayor será la penalización de los coeficientes de la regresión.

- **Regresión lasso**: Lasso (least absolute shrinkage and selection operator, por sus siglas en inglés), es conceptualmente similar a la regresión ridge, pero en lugar de penalizar la suma de los coeficientes al cuadrado (la llamada penalización L2, penaliza la suma de sus valores absolutos (penalización L1). Fue introducido por Robert Tibshirani en 1996.(Wikipedia contributors, 2020,). Lasso tiene la propiedad de poder reducir algunos coeficientes a cero cuando λ toma valores altos, por lo que permite que dicha característica pueda ser eliminada del modelo.

$$L_{lasso}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - X_i'\hat{B})^2 + \lambda \sum_{j=1}^{m} |\hat{\beta}_j| \qquad (2.3)$$

- **Lasso lars**: Es un modelo lineal entrenado con un L1 previo como regularizador. (Pedregosa et al., 2011,). El algoritmo LARS puede ser usado directamente para crear toda la ruta de solución de Elasticnet de manera eficiente con los esfuerzos de cálculo de un solo ajuste OLS. (Zou and Hastie, 2005,)

- **LarsCV** (Least Angle Regression): Es un algoritmo utilizado para ajustar modelos de regresión lineal a datos de alta dimensión, desarrollado por Bradley Efron, Trevor Hastie, Iain Johnstone y Robert Tibshirani. En lugar de dar un resultado vectorial, la solución de LARS consiste en una curva que denota la solución para cada valor de la norma L1. El algoritmo es similar a la regresión por pasos hacia adelante, pero en lugar de incluir variables en cada paso, los parámetros estimados se incrementan en una dirección equiangular a las correlaciones de cada uno con el residual.(Wikipedia contributors, 2020,)

- **Regularización elasticnet**: Es un método de regresión regularizada que combina linealmente las penalizaciones L1 y L2 de los métodos de lasso y ridge y surgió como resultado de las críticas de la regresión lasso cuya selección de variables era demasiado dependiente de los datos y por ende inestable. Tiene una representación similar a lasso y además fomenta un efecto de agrupación en el que los predictores de mayor correlación tienden a estar juntos dentro o fuera del modelo. Es una técnica particularmente útil cuando el número de predictores supera el número de observaciones. (Zou and Hastie, 2005,)

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i=1}^{n}(y_i - X_i'\hat{B})^2}{2n} + \lambda(\frac{1-\alpha}{2}\sum_{j=1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j=1}^{m}|\hat{\beta}_j|) \tag{2.4}$$

donde α es el parámetro de mezcla entre ridge (α = 0) y el lasso (α = 1).

- **Random forest importance**: Los bosques aleatorios presentan un buen indicador de selección de características. Para la clasificación, la medida de la impureza es la impureza gini o la ganancia de información/entropía. Por lo tanto, cuando se entrena un árbol, es posible calcular en qué medida cada característica disminuye la impureza. Cuanto más disminuye la impureza una característica, más importante es

la característica. En los bosques aleatorios, la disminución de la impureza de cada rasgo puede promediarse entre los árboles para determinar la importancia final de la variable. En la librería de python de sklearn [2] se proporciona una variable extra con el modelo, que muestra la importancia relativa o la contribución de cada característica en la predicción. Calcula automáticamente la puntuación de relevancia de cada característica en la fase de entrenamiento. Luego reduce la relevancia de manera que la suma de todas las puntuaciones es 1.(Pedregosa et al., 2011,).

- **RFECV usando random forest importance**: (Recursive feature elimination and cross validated selection) consiste en seleccionar características considerando recursivamente conjuntos de características cada vez más pequeños. Primero, el estimador se entrena en el conjunto inicial de características y la importancia de cada característica se obtiene mediante un coeficiente o mediante un atributo de importancia de las características. A continuación, se eliminan las características menos importantes del conjunto actual de características. Ese procedimiento se repite de forma recursiva en el nuevo hasta que finalmente se alcanza el número de características que se desea seleccionar. (Pedregosa et al., 2011,)

- **Gradient Boosted trees importance**: De manera similar a la selección de características utilizando la importancia de las características derivadas de bosques aleatorios, puede seleccionar características basadas en la importancia derivada de los gradient boosted trees y puede hacerse una sola vez, o de forma recursiva, dependiendo de cuánto tiempo se tenga, cuántas características están en el conjunto de datos, y si están correlacionadas o no.

- **Spike and slab**: Es una técnica de selección de variables bayesianas que resulta particularmente útil cuando el número de posibles predictores es mayor que el número de observaciones. Inicialmente, la idea del modelo fue propuesta por Mitchell Beauchamp (1988). El enfoque fue desarrollado significativamente por Madigan Raftery (1994) y George McCulloch (1997).Los ajustes finales al modelo fueron hechos por Ishwaran Rao (2005). (Wikipedia contributors, 2020,)

[2] Scikit-learn es una librería gratuita de Machine learning para python

- **Permutation feature importance**: Esta técnica se define como la disminución en el error del modelo cuando una característica es mezclada de manera aleatoria. Es agnóstica al modelo y puede calcularse muchas veces con diferentes permutaciones de la característica y puede ser aplicada sobre cualquier estimador. Este método puede utilizarse en el conjunto de entrenamiento o en el de prueba y validación y aquellas características que son importantes en el conjunto de entrenamiento pero no en el de validación pueden ser aquellas que hacen que el modelo se sobre-ajuste. (Pedregosa et al., 2011,)

2.2.3. Optimización de hiperparámetros

La optimización de los hiperparámetros en el aprendizaje automático tiene como objetivo encontrar aquellos hiperparámetros que obtengan el mejor rendimiento de un algoritmo de aprendizaje automático medido en un conjunto de validación. Estos hiperparámetros a diferencia de los parámetros del modelo, son elegidos por el ingeniero de aprendizaje de máquinas. Algunos ejemplos de hiperparámetros son: tasa de aprendizaje en una red neuronal, número de árboles en un bosque aleatorio o el valor de k en k vecinos más cercanos.

Teniendo en cuenta que los hiperparámetros pueden tener un impacto directo en el entrenamiento de los modelos de aprendizaje automático, existen diferentes técnicas para la optimización de los mismos, sin embargo las más comúnmente usadas son:

- **Ajuste manual de hiperparámetros**: Tradicionalmente los hiperparámetros se buscaban usando esta técnica y era básicamente a prueba de ensayo y error hasta encontrar aquellos que lograran altas precisiones en los modelos de aprendizaje automático. Sin embargo ahora existen mejores opciones y más automatizadas para realizar esta búsqueda.

- **Búsqueda de cuadrícula**: es el método más básico de ajuste de hiperparámetros. Con esta técnica se construye un modelo por cada posible combinación de

todos los valores de hiperparámetros proporcionados y se selecciona la arquitectura que produce los mejores resultados.

Una representación visual de este método es la siguiente:

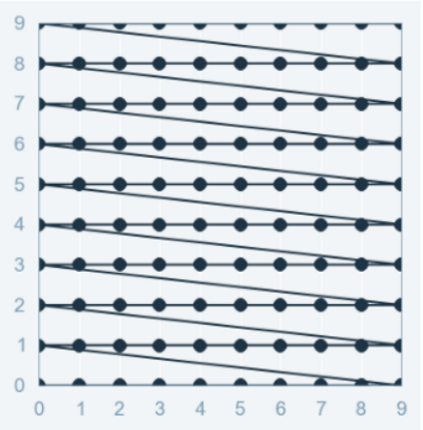

Figura 2.1: GridSearch

- **Búsqueda aleatoria**: La búsqueda aleatoria establece una cuadrícula de valores para los hiperparámetros y selecciona combinaciones aleatorias para entrenar y puntuar el modelo. Esto permite controlar explícitamente el número de combinaciones de parámetros que se intentan. El número de iteraciones de búsqueda se establece en función del tiempo o los recursos. Realizar una búsqueda aleatoria en lugar de una búsqueda por cuadrícula permite un descubrimiento mucho más preciso de los buenos valores para los hiperparámetros. Scikit Learn ofrece la función RandomizedSearchCV para este proceso. Su representación visual es como se muestra en la siguiente figura:

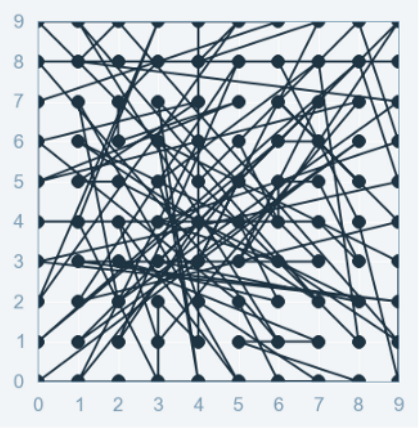

Figura 2.2: Representación Búsqueda Aleatoria

La búsqueda aleatoria funciona mejor para datos de dimensiones inferiores, ya que el tiempo necesario para encontrar el conjunto adecuado es menor con un número menor de iteraciones. La búsqueda aleatoria es la mejor técnica de búsqueda de hiperparámetros cuando hay menos número de dimensiones. (Chauhan, 2020,)

2.2.4. Selección y evaluación de modelos

La capacidad de generalización de un método de aprendizaje está relacionado con su capacidad de predecir sobre un conjunto de prueba independiente. La evaluación de ésta es muy importante ya que da una guía para la elección del modelo de aprendizaje y da una medida de su calidad.

Para resolver ambos problemas, el de seleccionar el modelo y el de evaluar el modelo, lo mejor es dividir aleatoriamente la base de datos en tres subconjuntos: El conjunto de entrenamiento que es utilizado para ajustar el modelo, el conjunto de validación es utilizado para estimar el error de predicción del modelo y el conjunto de prueba que se utiliza para evaluar el error de generalización del modelo seleccionado. Este último debe tenerse reservado y utilizarlo únicamente al final del análisis.(Hastie et al., 2009,)

No hay una regla general para calcular el número de observaciones de cada subconjunto, ya que esto depende del número de observaciones en el conjunto de entre-

namiento y del ruido que tengan los datos. Algunos autores recomiendan 50 % para el conjunto de entrenamiento y 25 % para los otros dos. Otros autores sugieren una partición de 60 %-20 %-20 % para el conjunto de entrenamiento, validación y prueba respectivamente. En este estudio se tomará esta última recomendación y se entrenarán los mismos modelos usando otra partición de 50 %-20 %-30 % para evaluar la estabilidad de los resultados. Esta última pensando en incrementar el conjunto de prueba para evaluar la capacidad de generalización, teniendo en cuenta que como la muestra es suficientemente grande no se afectaría el tamaño del conjunto de entrenamiento.

2.2.5. Métricas de evaluación de modelos

Con el fin de evaluar el rendimiento de los modelos, es necesario evaluar varias métricas de tal forma que se pueda mejorar el poder de predicción del mismo antes de ser puesto en producción y evitar malas predicciones cuando sea utilizado sobre datos nunca antes vistos. Las métricas utilizadas para la evaluación de los modelos fueron las siguientes:

- **Matriz de confusión:** Es una representación matricial de los resultados de las predicciones de un conjunto binario utilizada para medir el rendimiento de un modelo sobre datos de prueba donde el conjunto de datos reales es conocido. Permite identificar los tipos de errores que se comenten en la predicción.

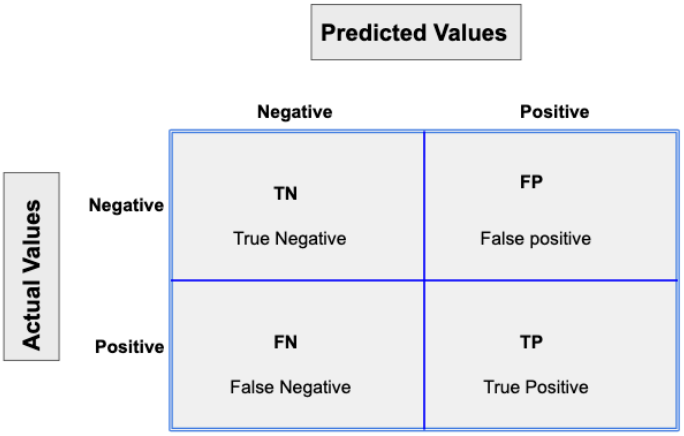

Figura 2.3: Matriz de confusión

Cada predicción puede ser uno de cuatro resultados comparando cada uno con

respecto al valor real:

- ◇ **Verdadero Positivo (TP):** Predicho verdadero y verdadero en realidad.
- ◇ **Verdadero Negativo (TN):** Predicho falso y falso en realidad.
- ◇ **Falso Positivo (FP):** Predicción de verdadero y falso en la realidad.
- ◇ **Falso Negativo (FN):** Predicción de falso y verdadero en la realidad.

- **AUC:** Es una de las métricas mas comúnmente usadas y representa el área bajo la curva ROC. (Galar et al., ,).
  Esta medida puede calcularse mediante la fórmula:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{2.5}$$

- **Precisión:** o positive predictive value (PPV): permite medir la *calidad* del modelo de machine learning en tareas de clasificación. Para el problema de los recontactos, permite identificar que porcentaje de los clientes que dijimos que iban a recontactar en realidad lo hicieron. La fórmula es la siguiente:

$$PPV = \frac{TP}{TP + FP} \tag{2.6}$$

- **Recall:** o true positive rate (TPR): también es llamada exhaustividad y permite medir la *cantidad* que el modelo es capaz de identificar. Para el caso de estudio, es la que permite medir que porcentaje de los clientes que van a recontactar en realidad lo hacen. Acorde al tipo de problema que se quiere resolver y las características del negocio, esta es la métrica de mayor interés, se calcula con la siguiente fórmula.

$$PPV = \frac{TP}{TP + FN} \tag{2.7}$$

- **F1_score:** Esta métrica permite comparar el rendimiento del modelo combinando las métricas de *precisión* y *recall* y corresponde al promedio ponderado entre ambas métricas, donde su contribución relativa es igual.(Pedregosa et al., 2011,). Su fórmula es la siguiente:

$$F1\_score = 2 * \frac{precision * recall}{precision + recall} \tag{2.8}$$

2.2.6. Balanceo de clases

El balanceo de clases es una característica de la muestra que se presenta cuando alguna de las clases (clases minoritarias) se encuentran representadas en menor medida que otras (clases mayoritarias) afectando notablemente la eficiencia de los algoritmos de clasificación, generando mejores resultados sobre la clase mayoritaria y por lo tanto generando una afectación sobre la predicción de la clase minoritaria, ya que generalmente las mejores predicciones quedan sobre la muestra del primer grupo, generando un sesgo sobre la clase mayoritaria a la hora de clasificar. Hoy en día no existe un umbral definido para determinar que una muestra se encuentra desbalanceada. (Calviño, 2017,), sin embargo, para el caso de estudio se tiene un 11.43 % de participación de la clase de interés, por lo que se usará esta técnica para identificar si se producen mejoras en el resultado de las métricas. Existen dos metodologías para solucionar los problemas de balanceo de datos:

- **Técnicas de remuestreo**: Consisten en modificar la distribución inicial de los datos para balancear las clases. Algunas de las más importantes son:
    - **Sobremuestreo**: Consiste en incrementar la clase minoritaria.
    - **Submuestreo**: Busca reducir la clase mayoritaria.
    - **Algoritmos híbridos**: Combina las técnicas de sobremuestreo y submuestreo.
- **Modificación de algoritmos**: Busca variar los algoritmos existentes para mejorar su predicción.

Análisis de recontactos en un contact center

3.1. Análisis de la base de datos

Actualmente se cuenta con gran cantidad de información de llamadas de los diferentes servicios que se atienden en el contact center. Teniendo en cuenta el problema planteado, se crearon las variables que podrían ayudar a entender el comportamiento de los clientes y adicionalmente podrían llevar a un rellamado o recontacto con base en la información generada por la plataforma (registro de cada llamada con la identificación del cliente y con su fecha y hora de ingreso) y el contexto del negocio. Esto con el fin de construir la base de datos para el estudio.

Se seleccionó un servicio particular para realizar el análisis, sin embargo este trabajo podría ser reproducible para otro servicio de cualquier sector teniendo en cuenta que la información almacenada por la plataforma y utilizada para el análisis es estándar para todos los servicios. adicionalmente se tomó un período de 7 meses para la construcción de las variables (febrero a agosto de 2019 y se definió que el periodo para analizar el recontacto es de tres días ya que en este punto es donde se observa una estabilidad del indicador (recontactos por cliente), lo que significa que se considera recontacto a al menos dos llamadas realizadas por el mismo cliente cuyo espacio de tiempo sea inferior a tres días.

La dimensión de la base de datos construida es de 421.055 registros con 23 variables (descritas anteriormente). Con estas variables es posible crear las variable respuesta para dar solución al problema de aprendizaje supervisado que se plantea (si se presentó o no un recontacto) a partir de las llamadas históricas.

A continuación se presenta un resumen preliminar de los datos con las estadísticas resumen

como la media, la mediana, la desviación estándar y los cuartiles para cada una de las variables (excepto las asociadas a los recontactos que se analizan más adelante):

	Monto	FrecuenciaDia	PromedioDiario	FrecuenciaMes	PromedioDiasEnMes	Vigencia	Recencia	Duracion_Promedio	# periodos con recontacto	#Transacciones recontacto
count	421055.00	421055.00	421055.00	421055.00	421055.00	421055.00	421055.00	421055.00	48117.00	48117.00
mean	2.14	1.67	1.19	1.28	1.21	18.41	146.32	518.75	1.14	4.03
std	10.57	3.60	0.47	0.77	0.60	49.29	88.12	475.92	0.71	27.17
min	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	1.00	2.00
25%	1.00	1.00	1.00	1.00	1.00	0.00	71.00	244.50	1.00	2.00
50%	1.00	1.00	1.00	1.00	1.00	0.00	144.00	401.50	1.00	2.00
75%	2.00	2.00	1.00	1.00	1.00	2.00	221.00	642.00	1.00	3.00
max	1129.00	266.00	11.00	10.00	26.60	304.00	304.00	21612.00	18.00	1088.00

Figura 3.1: Descriptivo general

También se realiza un análisis de correlación con las variables numéricas objeto de estudio en su escala original, tanto aquellas que corresponden al comportamiento histórico de los clientes en el uso del canal como frecuencia de llamadas (monto), recencia, promedio de llamadas diarias y mensuales, etc. (figura: 3.2) como las asociadas al desempeño de los agentes que atendieron dichas llamadas (figura: 3.3).

	Monto	FrecuenciaDia	PromedioDiario	FrecuenciaMes	PromedioDiasEnMes	Vigencia	Duracion_Promedio	Recencia
Monto	1.000000	0.931299	0.139194	0.339235	0.593766	0.213798	0.029900	-0.057101
FrecuenciaDia	0.931299	1.000000	0.083183	0.552024	0.682314	0.381804	0.034348	-0.102393
PromedioDiario	0.139194	0.083183	1.000000	0.039891	0.100509	0.027513	0.015504	-0.003384
FrecuenciaMes	0.339235	0.552024	0.039891	1.000000	0.262322	0.819612	0.024247	-0.216967
PromedioDiasEnMes	0.593766	0.682314	0.100509	0.262322	1.000000	0.197685	0.054787	-0.056179
Vigencia	0.213798	0.381804	0.027513	0.819612	0.197685	1.000000	0.016342	-0.275045
Duracion_Promedio	0.029900	0.034348	0.015504	0.024247	0.054787	0.016342	1.000000	0.055144
Recencia	-0.057101	-0.102393	-0.003384	-0.216967	-0.056179	-0.275045	0.055144	1.000000

Figura 3.2: Matriz Correlación variables de llamadas

De la figura 3.2 se concluye que las variables que mayor correlación presentan son: Monto con FrecueciaDia y con PromedioDiasEnMes, frecuenciaDia con PromedioDiasEnMes y vigencia con FrecuenciaMes.

	Monto	FrecuenciaDia	PromedioDiario	FrecuenciaMes	PromedioDiasEnMes	agentes	scoreic	scoreaht	scoreadh
Monto	1.000000	0.931299	0.139194	0.339235	0.593766	0.646213	0.006343	-0.008344	0.001854
FrecuenciaDia	0.931299	1.000000	0.083183	0.552024	0.682314	0.821439	0.009009	-0.012102	0.003246
PromedioDiario	0.139194	0.083183	1.000000	0.039891	0.100509	0.289416	0.002482	-0.009603	-0.003710
FrecuenciaMes	0.339235	0.552024	0.039891	1.000000	0.262322	0.720297	0.007727	-0.012683	0.005588
PromedioDiasEnMes	0.593766	0.682314	0.100509	0.262322	1.000000	0.662587	0.013536	-0.011288	0.002186
agentes	0.646213	0.821439	0.289416	0.720297	0.662587	1.000000	0.009700	-0.007060	0.001352
scoreic	0.006343	0.009009	0.002482	0.007727	0.013536	0.009700	1.000000	-0.139232	0.144748
scoreaht	-0.008344	-0.012102	-0.009603	-0.012683	-0.011288	-0.007060	-0.139232	1.000000	-0.022724
scoreadh	0.001854	0.003246	-0.003710	0.005588	0.002186	0.001352	0.144748	-0.022724	1.000000

Figura 3.3: Matriz Correlación variables de agentes

Sin embargo en la figura 3.3 no se observa correlación entre las variables de comportamiento de uso histórico del canal con las variables de desempeño de los agentes.

3.2. Construcción de la variable respuesta

Teniendo en cuenta que interesa conocer primero cuantos clientes vuelven a realizar una llamada, ya sea por el mismo motivo (recontacto) o por otro motivo diferente al de la llamada inicial, se crean tres nuevas variables a partir de las variables 'Transacciones-recontacto' y 'Transacciones-rellamado', que serán la base para el análisis. Estas variables son:

- **Re-contactos:** 1 si hubo re-contactos, 0 si no (llamadas del mismo motivo que el inicial)
- **Re-llamados:** 1 si hubo re-llamados, 0 si no (llamadas del otro motivo diferente al inicial)
- **Presenta-Rellam/recont:** 1 si hubo re-llamados o re-contacto, 0 si no.

3.3. Participación de rellamadas y/o recontactos en el conjunto de datos

Con las variables creadas ya podemos conocer que porcentaje de los clientes vuelven a comunicarse y cuántos lo hacen por el mismo motivo con el fin de entender que tanto ocurre esto y si en realidad su control o disminución si podría generar eficiencias operacionales o definir estrategias personalizadas que mejoren el relacionamiento con los clientes.

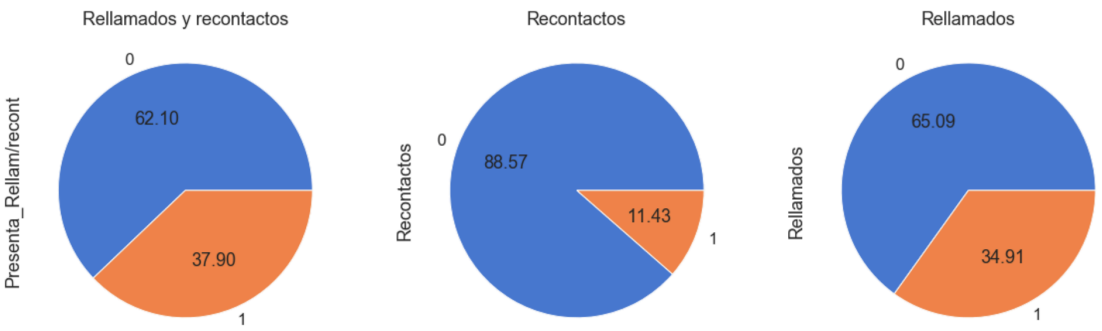

Figura 3.4: Participación recontactos y rellamados

De la figura 3.4, se observa que el 37.90 % de los clientes volvieron a comunicarse al menos una vez en el período analizado, esto equivale a 159.590 clientes. Del total de clientes el 11.43 % (48.117 clientes) lo hacen por el mismo motivo y el 34.91 % (147.007 clientes) se comunican por un nuevo motivo.

Es importante tener en cuenta que un cliente puede hacer rellamados y recontactos en el mismo período analizado, o generar un recontacto pero no tener rellamados. En este caso se desea entender el comportamiento de los recontactos y ver si es posible predecirlos a partir de las demás variables recopiladas utilizando modelos de aprendizaje automático.

3.4. Evaluación de las variables y selección de características

Evaluación de las variables

Se considera importante realizar un análisis de selección de variables, teniendo en cuenta que no todas las variables utilizadas en el estudio provienen de la misma fuente de datos (plataforma transaccional) y su obtención es más compleja (puede tomar mayor tiempo y esfuerzo) si se desea llevar el modelo a producción, pero antes de no considerarlas es importante entender su impacto. Sin embargo antes de realizar la selección de características para entrenar el modelo de aprendizaje automático, se realiza primero un análisis de la variable respuesta y su comportamiento frente a las demás variables con el fin de evaluar que tanto varían los valores de cada una para los clientes que recontactan y los que no lo vuelven a hacer.

Recontactos	Monto	FrecuenciaDia	PromedioDiario	FrecuenciaMes	PromedioDiasEnMes	Vigencia	Recencia	Duracion_Promedio
0	1.573615	1.399308	1.118646	1.208088	1.136429	14.534075	147.418536	503.356881
1	6.493880	3.790407	1.768455	1.862876	1.776847	48.410520	137.767816	607.655541

Figura 3.5: Comparativo del promedio de cada variable vs recontactos

En la figura 3.5 se observa por ejemplo que el promedio de llamadas (variable 'Monto') de las personas que generaron un recontacto es mucho mayor que aquellas que no, es decir, las personas que generan recontactos tienen en promedio 6.49 llamadas en el período analizado, las que no generan recontactos hacen en promedio 1.47 llamadas en el mismo período. De la misma manera sucede con la frecuencia diaria que para quien realiza recontactos es en promedio de 3.79, con respecto al que no que es de 1.39. También puede observarse que la duración promedio de las llamadas que corresponden a un recontacto es casi 100 segundos mayor a las que no lo son. En general en todas las variables, es mayor su valor cuando se trata de llamadas asociadas a clientes que generan recontactos.

Se analizan de manera independiente las variables relacionadas con los recontactos y se obtienen las medidas resumen para cada una de ellas (media, mediana, desviación, cuartiles) y se muestran a continuación:

	# periodos con recontacto	#Transacciones recontacto	transaccion_promedio_por_recontacto	differencia_1_2_Horas_Reconntacto
count	48117.000000	48117.000000	48117.000000	48117.000000
mean	1.144647	4.032837	2.740997	12.176936
std	0.709824	27.167684	9.399838	18.872336
min	1.000000	2.000000	2.000000	0.000000
25%	1.000000	2.000000	2.000000	0.000000
50%	1.000000	2.000000	2.000000	2.000000
75%	1.000000	3.000000	2.000000	19.000000
max	18.000000	1088.000000	1088.000000	84.000000

Figura 3.6: Variables relacionadas con los recontactos históricos

En figura 3.6 se encuentra que las personas que realizaron recontactos lo hicieron en promedio en 1.14 períodos y el mayor número de períodos con recontactos fue de 18, quienes generaron estos recontactos realizaron en promedio 2.74 transacciones por cada llamada (motivos diferentes de consulta) y el recontacto se generó en promedio a las 12 horas de la primera

llamada, sin embargo el 50 % de los clientes lo hace en 2 horas o menos, lo que indica que si un cliente tiene necesidad de volver a llamar lo hará lo antes posible y cualquier estrategia que se implemente debe ser oportuna para que si se refleje en los resultados.

A continuación se muestra la distribución para las variables 'diferencia 1-2 Horas-Recontacto' y ' periodos-con-recontacto' que corresponden al tiempo en horas transcurrido para que el cliente realice un recontacto y la cantidad de veces que históricamente ha realizado un recontacto con el fin de entender un poco las estadísticas presentadas en la figura 3.6 y entender mejor el comportamiento de los clientes que han realizado recontactos en el período analizado.

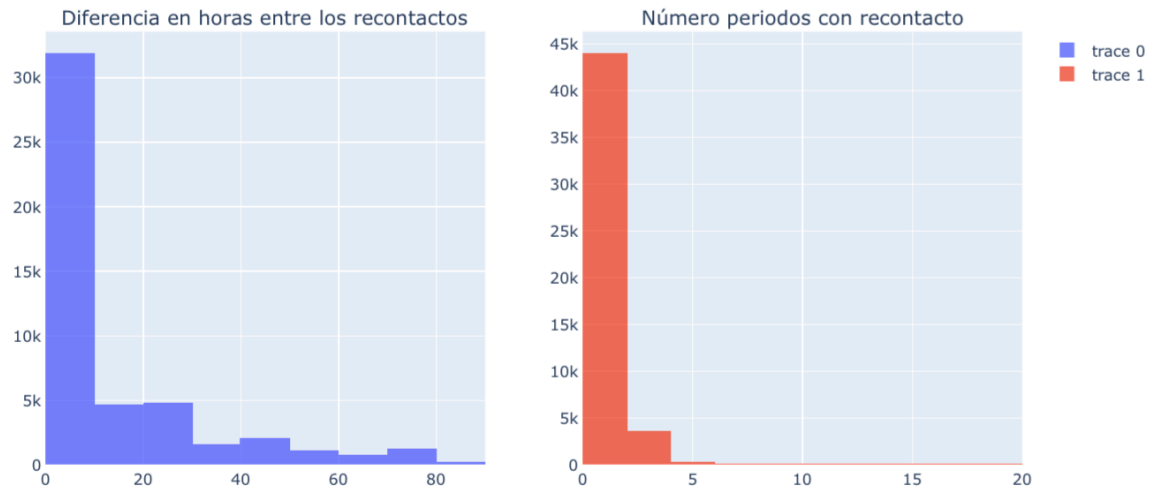

Figura 3.7: Clientes que generan recontactos

En la figura 3.7 se observa que los clientes realizan en su mayoría el recontacto antes de las 10 horas, como se mencionó antes incluso el 50 % de ellos lo hace en menos de 2 horas y que la cantidad de veces que generan un recontacto no supera dos períodos.

Ahora miremos lo que sucede al cruzar el segmento de uso del canal (consumo, afinidad y reiteratividad) con la variable recontactos en la figura 3.8:

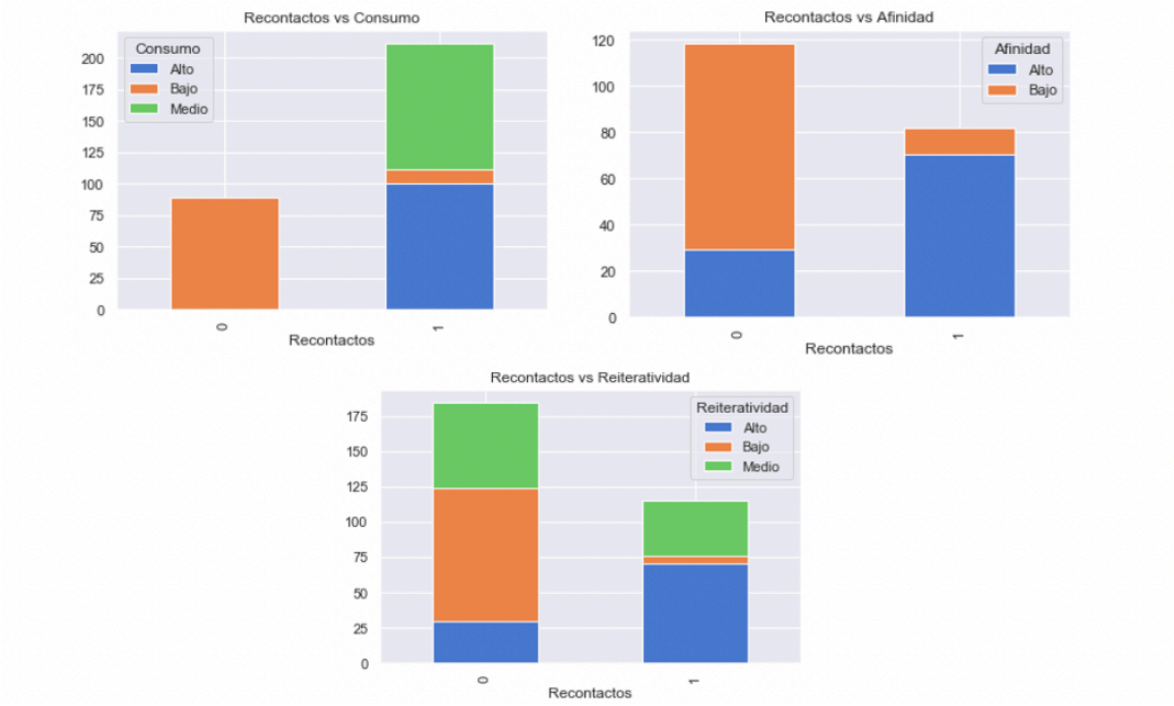

Figura 3.8: Recontactos vs variables de segmento de uso

De la figura 3.8 se concluye que las personas que generan recontactos son en su mayoría de un consumo alto y medio, igualmente de reiteratividad alta y media y además son muy afines al canal, es decir lo usan con alta frecuencia por lo que evitar recontactos en ellas, puede impactar el volumen de llamadas que ingresan al canal.

Selección de características

Uno de los objetivos de este trabajo es identificar el conjunto de variables que mayor influencia tienen sobre la probabilidad del recontacto, de tal forma que se pueda mejorar el desempeño de los modelos de aprendizaje que serán utilizados. Por lo tanto, una vez se analizó la participación de los recontactos en el conjunto de datos, se aplicaron las técnicas de selección mencionadas en el capítulo 2 y los resultados de estas técnicas se presentan en la figura 3.9.

idx variable	variable	Mutual information	Univariate roc-auc feature selection	Spike and slab	Random Forest importance	RFECV using random forests importance	Lasso Lars	LarsCV	Step forward selection	LassoCV Regularization	ElasticnetCV regularisation	Gradient Boosted trees importance	Total variables	Probabilidad Selección
1	PromedioDiario	1	1	1	1	1	1	1	1	1	1	1	11	1.000
2	agentes	1	1	1	1	1	1	0	1	1	1	1	10	0.909
3	Monto	1	1	1	1	1	1	0	1	1	0	0	8	0.727
4	PromedioDiasEnMes	1	1	1	0	0	1	1	0	1	1	0	7	0.636
5	FrecuenciaDia	1	1	1	0	0	1	0	0	1	1	0	6	0.545
6	FrecuenciaMes	0	1	1	0	0	1	1	1	1	0	0	6	0.545
7	Reiteratividad	1	1	1	0	0	0	1	0	0	1	1	6	0.545
8	Duracion_Promedio	1	1	1	1	1	0	0	0	0	0	0	5	0.455
9	scoreic	1	1	0	1	1	0	0	0	0	0	0	4	0.364
10	scoreaht	1	0	1	1	1	0	0	0	0	0	0	4	0.364
11	scoreadh	1	1	0	1	1	0	0	0	0	0	0	4	0.364
12	Recencia	0	0	1	1	1	0	0	0	0	0	0	3	0.273
13	Afinidad	0	0	1	0	0	0	1	1	0	0	0	3	0.273
14	Vigencia	1	1	0	0	0	0	0	0	0	0	0	2	0.182
15	Consumo	0	0	0	0	0	0	1	1	0	0	0	2	0.182
	Total variables	11	11	11	8	8	6	6	6	6	5	3		

Figura 3.9: Matriz de selección de variables

La columna ’variable’ contiene las variables analizadas y las demás columnas representan el método de selección de variables utilizado. En cada fila, al frente de cada variable se asigna un 1 si dicha variable fue seleccionada por el método y 0 si no lo fue. Al final se contabiliza la cantidad de veces que cada variable fue seleccionada y se calcula una probabilidad de selección que no es más que dicho total sobre la cantidad de métodos evaluados.

Las variables seleccionadas varían en un rango entre 3 y 11 y las que mayor número de veces quedaron seleccionadas (al menos la mitad de las veces) son alrededor de 8 variables y son las siguientes: ’Monto’, ’FrecuenciaDia’, ’FrecuenciaMes’, ’PromedioDiario’,’PromedioDiasEnMes’, ’Reiteratividad’,’agentes’,’Duracion_Promedio’.

Casi todas son variables transaccionales, excepto la variable ’Reiteratividad’ que es calculada con base en un modelo previo de aprendizaje no supervisado realizado al interior de la compañía que utiliza algunas de las variables transaccionales y hace referencia a clientes que realizan llamadas en cortos períodos de tiempo (varias llamadas en el mismo día o en el mismo mes) y la variable ’agentes’ que representa la cantidad de agentes que atendieron las llamadas históricas de cada cliente.

Con el fin de seleccionar el mejor subconjunto de variables, se decidió dejar todas las características seleccionadas por los diferentes métodos más otros subconjuntos de interés y con cada uno se entrenaron diferentes modelos de aprendizaje automático para identificar cual arrojaba los mejores resultados y de esta forma poder escoger el mejor conjunto de variables

y el mejor modelo de aprendizaje. Los subconjuntos adicionales de variables de interés que fueron seleccionados para el entrenamiento de modelos a partir de los resultados previos o teniendo presente la facilidad en la obtención de las variables se muestran a continuación:

idx variable	Variable	Prob Selección >45%	Todas las Variables	Solo Transaccionales	Sin las más importantes
1	PromedioDiario	1	1	1	0
2	agentes	1	1	0	0
3	Monto	1	1	1	0
4	PromedioDiasEnMes	1	1	1	1
5	FrecuenciaDia	1	1	1	1
6	FrecuenciaMes	1	1	1	1
7	Reiteratividad	1	1	0	0
8	Duracion_Promedio	1	1	1	0
9	scoreic	0	1	0	1
10	scoreaht	0	1	0	1
11	scoreadh	0	1	0	1
12	Recencia	0	1	1	1
13	Afinidad	0	1	0	0
14	Vigencia	0	1	1	1
15	Consumo	0	1	0	0
	Total variables	**8**	**15**	**8**	**8**

Figura 3.10: Otros conjuntos de variables de interés

- **Variables de mayor coincidencia en los métodos utilizados:** En este caso se tomaron las variables que fueron seleccionadas más del 45 % (casi la mitad) de las veces por los métodos aplicados partiendo de la intuición de que si tantos métodos coinciden en seleccionar una misma variable es porque esta es importante.

- **Todas las variables:** Tanto las variables relacionadas con los contactos históricos como las relacionadas al desempeño de los asesores que atendieron a cada cliente con el fin de identificar que tanto puede mejorar el ajuste de los modelos considerando todas las variables.

- **Solo variables transaccionales** Se tomaron solo las variables que podrían representar una más fácil implementación del modelo ya que se toman de una misma fuente e implican un menor tiempo de pre-procesamiento.

- **Sin variables importantes** Se eliminaron de los modelos las 3 variables más importantes, en este caso las que quedaron seleccionadas por los diferentes métodos más del

70 % de las veces con el fin de analizar el impacto en el resultado de los modelos.

En conclusión, en este capítulo se realizó una descripción de las variables que conforman la base de datos y un entendimiento general de los mismos a través de un análisis descriptivo de las variables de estudio (medidas resumen, gráficos, tablas bivariadas y análisis de correlación), también se realizó un tratamiento de valores atípicos y se construyó la variable respuesta con base en los recontactos históricos generados por los clientes, donde se definió un valor de 0 en caso de que el cliente históricamente no hubiera realizado algún recontacto y un valor de 1 en caso de si haber realizado recontactos. sobre la cual también se analizó su participación en el conjunto de datos.

Por otro lado se realizó el análisis de selección de variables y se presentaron los resultados en una matriz (figura: 3.9) que resume la selección de características de cada uno de los métodos de selección detallados en el capítulo 2 y se identificaron otros conjuntos de variables de interés (figura: 3.10) para llevar a cabo el proceso de experimentación para la selección del mejor modelo de aprendizaje automático.

En el siguiente capítulo se pretende identificar si efectivamente con la variable respuesta construida y con las variables independientes del conjunto de datos si es posible predecir el recontacto en el centro de llamadas utilizando diferentes métodos de aprendizaje supervisado y de ser así seleccionar el mejor modelo ya que es el objetivo principal inicialmente planteado.

Predicción de recontactos en un contact center

4.1. Planteamiento del problema de aprendizaje supervisado

El objetivo principal de este trabajo es identificar si es posible predecir aquellos clientes que volverán a contactarse antes de tres días al centro de contactos, utilizando técnicas de aprendizaje automático, específicamente de aprendizaje supervisado. En este caso la variable respuesta es 'Recontacto', que equivale a un 11.43 % de los clientes y las variables predictoras son las demás variables relacionadas con el comportamiento histórico de los clientes a través de dicho canal y el desempeño de los agentes que atendieron dichas llamadas.

4.2. Entrenamiento de modelos y optimización de hiperparámetros

Para llevar a cabo el proceso de modelado para la predicción de recontactos y escoger el modelo con mejor desempeño, se realizó un proceso de experimentación en el cual se entrenaron los 5 modelos de aprendizaje automático mencionados en el marco teórico (Regresión Logística, Árbol de Clasificación, Bosque Aleatorio, Gradient Boosting y una Red Neuronal Multicapa con diferentes estructuras). Estos modelos fueron entrenados con los diferentes conjuntos de características seleccionados y en dos particiones de la base de datos (60 %-20 %-20 % y 50 %-20 %-30 %).

La búsqueda de hiperparámetros se realizó con el método de búsqueda por cuadrícula explicado en el marco teórico y para llevarlo a cabo se utilizó la librería GridSearchCV de python con el hiperparámetro cv = 5 (que es el que viene por defecto), el cual corresponde al genera-

dor de validación cruzada [1]. Los hiperparámetros utilizados en cada uno de los modelos fue el siguiente:

Para el proceso de entrenamiento de la **Regresión Logística** se utilizó la librería de sklearn en python y se utilizaron los siguientes hiperparámetros:

- **Penalty:** Utilizado para especificar la norma en la penalización, en este caso se utilizaron: L1, l2, Elasticnet y None. (Buitinck et al., 2013,)
- **C:** Asociado a la fuerza de la regularización, corresponde a un valor flotante positivo y mientras más pequeño este valor más fuerte es la regularización.(Buitinck et al., 2013,)
- **l1_ratio:** Equivale a un valor flotante. Se varió en valores de 0.2, 0.5 y 0.8 y se utiliza solo cuando el hiperparámetro penalty = 'elasticnet'.(Buitinck et al., 2013,)

Para llevar a cabo el entrenamiento del **árbol de decisión** se utilizaron los siguientes hiper parámetros:

- **splitter:** Se utiliza para elegir la división de cada nodo, se tomaron los valores de 'best' y 'random' para escoger la mejor división y la mejor división aleatoria.
- **max_depth:** Corresponde a la máxima profundidad del árbol. Si no hay ninguna, entonces los nodos se expanden hasta que todas las hojas sean puras o hasta que todas las hojas contengan al menos el mínimo número de muestras para dividir un nodo interno.Se usaron los valores [None, 2,4,6]
- **min_samples_leaf:** Equivale a el número mínimo de muestras necesarias para estar en un nodo de la hoja. Se usaron los valores: [1,5,8]. (Pedregosa et al., 2011,)

Los hiperparámetros evaluados en el **bosque aleatorio** fueron:

- **max_depth**:Corresponde a la profundidad máxima de los árboles y si no hay ninguna, entonces los nodos se expanden hasta que todas las hojas sean puras o hasta que todas las hojas contengan al menos el número mínimo de muestras que divide un nodo, se evaluaron en estos valores: [None, 2,4,6].

[1]La validación cruzada es uno de los métodos más simples y ampliamente utilizados para estimar el error de predicción.

- **min_samples_leaf**: Como se mencionó anteriormente equivale al número mínimo de muestras de un nodo. Se usaron estos valores: [1,5,8].
- **n_estimators**:Corresponde al número de árboles en el bosque, se tomaron estos valores: [50, 100, 200]. (Buitinck et al., 2013,).

Los hiperparámetros utilizados en el **gradient Boosting** fueron:

- **max_depth**: Corresponde a la profundidad máxima de los árboles y si no hay ninguna, entonces los nodos se expanden hasta que todas las hojas sean puras o hasta que todas las hojas contengan al menos el número mínimo de muestras que divide un nodo, se evaluaron en estos valores: [3, 4, 5, 6, 7].
- **loss**: Se usaron 'deviance' y 'exponential' y corresponde a la función de pérdida que será optimizada donde 'deviance' se refiere a la desviación (regresión logística) para la clasificación con resultados probabilísticos y para la pérdida 'exponencial' el aumento del gradiente utiliza el algoritmo AdaBoost. (Pedregosa et al., 2011,)
- **n_estimators**: Corresponde al número de árboles en el bosque, se tomaron estos valores: [50, 100, 200]. (Buitinck et al., 2013,).

Para este problema se utilizaron varias estructuras de red con la librería de sklearn y con los siguientes hiperparámetros:

- **max_iter=100**: Corresponde al número máximo de iteraciones.
- **n_iter_no_change = 50**: Número máximo de épocas hasta que la tolerancia no mejore.
- **tol=1e-2:** Equivale a la tolerancia para la optimización.
- **hidden_layer_sizes**: Se probaron estructuras de red de una, dos y tres capas cy variando la cantidad de neuronas de la siguiente manera: [(50),(100),(50,50),(10,10), (5,5,5),(50,50,20)].
- **solver ('sgd'. 'adam'):** Es el solver utilizado para la optimización de los pesos y se usaron 'sgd' que se refiere al descenso de gradiente estocástico y 'adam' que se refiere a un optimizador basado en el gradiente estocástico propuesto por Kingma, Diederik y Jimmy Ba.

- **learning_rate_init**: Las tasas de aprendizaje para la actualización de los pesos utilizadas fueron: [0.2, 0.5, 0.9]. (Pedregosa et al., 2011,)

Con base en lo anterior, los hiperparámetros utilizados en el entrenamiento de modelos se resumen en la siguiente figura:

Modelo	Hiperparámetros
Regresión logística	Penalty: L1, l2, Elasticnet y None
	C: 1.0, 1.5, 2.0
	L1_ratio: 0.2, 0.5, 0.8
Árbol de decisión	splitter
	max_depth: None, 2, 4, 6
	Min_samples_leaf: 1, 5, 8
Bosque aleatorio	max_depth: None, 2, 4, 6
	Min_samples_leaf: 1, 5, 8
	n_estimators: 50,100,200
Gradient boosting	max_depth: 3, 4, 5, 6, 7
	loss: 'deviance' y 'exponential'
	n_estimators: 50,100,200
Red neuronal multicapa MLP	max_iter=100
	n_iter_no_change = 50
	tol=1e-2
	hidden_layer_sizes: (50),(100),(50,50),(10,10),(5,5,5),(50,50,20)
	solver : 'sgd', 'adam'
	learning_rate_init: 0.2, 0.5, 0.9

Figura 4.1: hiperparámetros

4.3. Resultados

4.3.1. Fase 1: Experimentación inicial

En esta primera fase se realizó un proceso de experimentación con el fin de encontrar el mejor modelo y el conjunto de características adecuado. Para esto se utilizaron distintas técnicas de selección de variables y cada conjunto obtenido por ellas fue probado en los diferentes modelos de aprendizaje, evaluando con qué conjunto y en qué modelo se obtienen mejores resultados. En total fueron entrenados 5 modelos en 15 conjuntos de características y con dos particiones diferentes de la base de datos (para validar la estabilidad de los resultados y el posible impacto al cambiar la partición), para un total de 150 modelos. El rendimiento de los modelos en cada conjunto de características se evaluó con las métricas de AUC y F1_score

inicialmente (en los datos de entrenamiento y validación). los resultados obtenidos para cada partición de la base se presentan en la figura 4.2 con una partición de 60 %-20 %-20 % y en la figura 4.3 con una partición de 50 %-20 %-30 %.

Train test val (60-20-20)	Validación					
AUC	**Logistic Regression**	**Gradient Boosting**	**Decision Tree**	**Random Forest**	**MLP**	**Total variables**
Totas las variables	0.698	0.725	0.73	0.722	0.712	**12**
Mutual information	0.696	0.718	0.688	0.716	0.725	**11**
Univariate roc-auc feature selection	0.696	0.715	0.689	0.716	0.71	**11**
Spike and slab	0.699	0.716	0.701	0.718	0.666	**11**
Solo Transaccionales	0.698	0.725	0.731	0.724	0.714	**9**
Random Forest importance	0.682	0.709	0.708	0.708	0.718	**8**
RFECV using random forest importance	0.682	0.712	0.708	0.708	0.71	**8**
Prob selección 45%	0.696	0.705	0.679	0.688	0.713	**8**
Sin var importantes	0.588	0.629	0.577	0.629	0.583	**8**
Step forward selection	0.695	0.694	0.694	0.691	0.681	**6**
LassoCV Regularization	0.697	0.693	0.695	0.698	0.693	**6**
Lasso Lars	0.697	0.693	0.688	0.699	0.712	**6**
LarsCV	0.692	0.675	0.673	0.664	0.648	**6**
ElasticnetCV regularisation	0.692	0.689	0.684	0.691	0.685	**5**
Gradient Boosted trees importance	0.695	0.697	0.695	0.696	0.693	**3**

Figura 4.2: Comparativo de AUC, partición 60-20-20

Train test val (50-20-30)	Validación					
AUC	**Logistic Regression**	**Gradient Boosting**	**Decision Tree**	**Random Forest**	**MLP**	**Total variables**
Totas las variables	0.698	0.729	0.727	0.725	0.703	12
Mutual information	0.697	0.716	0.691	0.719	0.739	11
Univariate roc-auc feature selection	0.697	0.715	0.693	0.719	0.715	11
Spike and slab	0.7	0.72	0.699	0.72	0.711	11
Solo Transaccionales	0.698	0.727	0.725	0.725	0.718	9
Random Forest importance	0.683	0.712	0.719	0.71	0.708	8
RFECV using random forest importance	0.683	0.713	0.719	0.711	0.687	8
Prob selección 45%	0.696	0.705	0.684	0.694	0.704	8
Sin var importantes	0.59	0.633	0.636	0.635	0.636	8
Step forward selection	0.697	0.694	0.694	0.699	0.688	6
LassoCV Regularization	0.697	0.695	0.693	0.699	0.714	6
Lasso Lars	0.697	0.695	0.693	0.699	0.689	6
LarsCV	0.693	0.704	0.674	0.666	0.663	6
ElasticnetCV regularisation	0.693	0.696	0.692	0.698	0.7	5
Gradient Boosted trees importance	0.695	0.691	0.694	0.695	0.675	3

Figura 4.3: Comparativo de AUC, partición 50-20-30

En general se observan resultados muy similares de AUC con las dos particiones de los

datos realizadas. En las figuras 4.4 y 4.5 se observa que en la experimentación se obtuvo un promedio de 0.70 en el AUC y una desviación estándar entre los diferentes modelos y conjuntos de variables entrenados de aproximadamente 0.02, lográndose resultados ligeramente superiores en los modelos con mayor número de variables (entre 9 y 12 características). El mejor resultado (auc=0.74) se logró con el conjunto de variables obtenido con la técnica de *información mutua* para una *red neuronal multicapa perceptrón* con la partición de entrenamiento, prueba y validación de 50-20-30 respectivamente y un total de 11 características y el resultado más bajo de AUC (auc=0.65) fue obtenido con el conjunto de variables obtenido con la técnica de selección de *larsCV* que seleccionó 6 variables, pero dentro de ellas no está la variable 'agentes' y 'monto' que fueron seleccionadas por la mayoría de métodos, esto también con una *red multicapa perceptrón* pero en una partición de entrenamiento, prueba y validación de 60-20-20 respectivamente. También se observa que en general los resultados más bajos de AUC generalmente se obtienen con el modelo de *regresión logística*.

	count	mean	std	min	25%	50%	75%	max
AUC	140.0	0.7	0.02	0.65	0.69	0.7	0.71	0.74

Figura 4.4: Estadísticos AUC experimentación

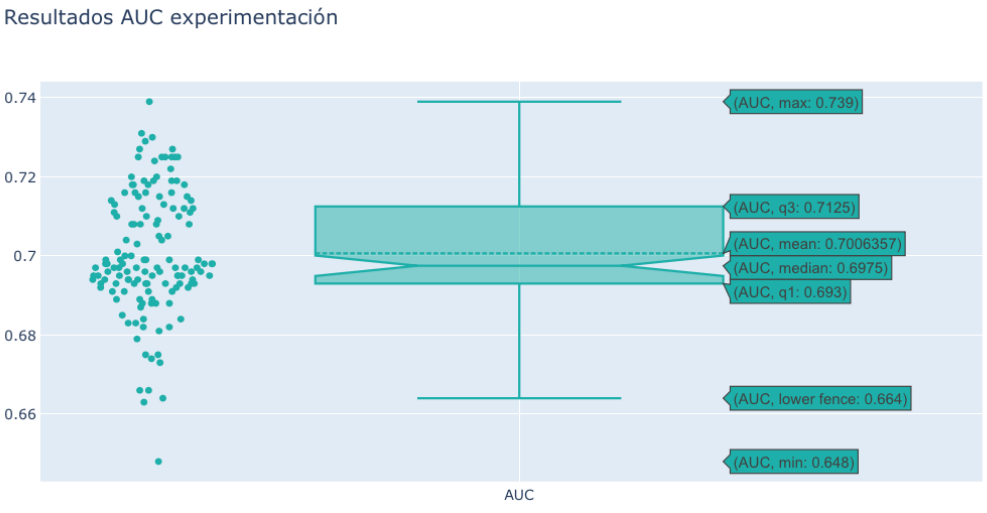

Figura 4.5: Distribución resultados AUC experimentación

Se puede observar en la figura 4.3 que corresponde a los resultados de la partición 50 %-

20 %-30 % de los datos, que se conservan los mismos patrones mencionados anteriormente y que adicionalmente los resultados del AUC no varían mucho, lo que indica que se presenta estabilidad en los modelos y que una buena selección de características puede ser la que mayor facilidad de despliegue permita, lo mismo que el modelo seleccionado, ya que al ser tan similares los resultados se pretende escoger un modelo que sea parsimonioso y eficiente computacionalmente. Por otro lado se demuestra que no se requieren muchos datos, múltiples fuentes o el modelo más complejo para obtener resultados muy similares a los mencionados en el estudio realizado por (Moazeni and Andrade, 2018,) pero buscando una implementación del mismo en tiempo real, de tal forma que pueda mejorarse la experiencia de los clientes y generarse eficiencias en costos operacionales asociados a personas y recursos tecnológicos para la atención de llamadas que no agregan valor.

Se realizó el mismo ejercicio anterior pero comparando la métrica del f1_Score y se obtuvieron los siguientes resultados

Train test val (60-20-20)	Validación					
F1 SCORE	Logistic Regression	Gradient Boosting	Decision Tree	Random Forest	MLP	Total variables
Totas las variables	0.734	0.761	0.757	0.756	0.75	12
Mutual information	0.733	0.754	0.729	0.752	0.756	11
Univariate roc-auc feature selection	0.733	0.751	0.73	0.751	0.744	11
Spike and slab	0.735	0.754	0.738	0.754	0.713	11
Solo Transaccionales	0.734	0.76	0.758	0.758	0.752	9
Random Forest importance	0.721	0.747	0.743	0.746	0.752	8
RFECV using random forest importance	0.721	0.75	0.743	0.746	0.746	8
Prob selección 45%	0.733	0.744	0.723	0.73	0.749	8
Sin var importantes	0.617	0.667	0.604	0.666	0.613	8
Permutation Importance	0.726	0.757	0.745	0.75	0.759	7
UnderSampling	0.731	0.721	0.715	0.72	0.717	7
OverSampling	0.732	0.735	0.716	0.753	0.72	7
Step forward selection	0.734	0.736	0.735	0.734	0.726	6
LassoCV Regularization	0.734	0.735	0.735	0.739	0.735	6
Lasso Lars	0.734	0.735	0.729	0.739	0.747	6
LarsCV	0.728	0.719	0.716	0.71	0.695	6
ElasticnetCV regularisation	0.73	0.732	0.727	0.733	0.729	5
Gradient Boosted trees importance	0.732	0.737	0.734	0.736	0.733	3

Figura 4.6: Comparativo de f1_score, partición 60-20-20

Al analizar la figura 4.6 se observan resultados similares (promedio de f1_score de 0.73 y desviación estándar de 0.01) para casi todos los modelos que utilizaron el conjunto de variables transaccionales y el conjunto de variables en los que casi la mitad de los métodos de selección de características coincidieron en seleccionar, siendo mejores los resultados de f1_Score de los modelos bosque aleatorio, gradient boosting y el árbol de decisión (promedio de f1_score

de 0.76 y desviación estándar de 0.001).

4.3.2. Fase 2: Afinación en la búsqueda de variables

En esta fase, teniendo en cuenta que los resultados en la primera fase fueron tan similares y no se tenía aun un criterio claro para seleccionar el mejor conjunto de características se decidió realizar una nueva evaluación utilizando una técnica de inspección de variables llamada *Permutation Feature Importance*. Al utilizar esta técnica, con 10 iteraciones, (ya que la variabilidad en los resultados era mínima y con un mayor valor de iteraciones los resultados eran casi los mismos) Los resultados fueron los siguientes:

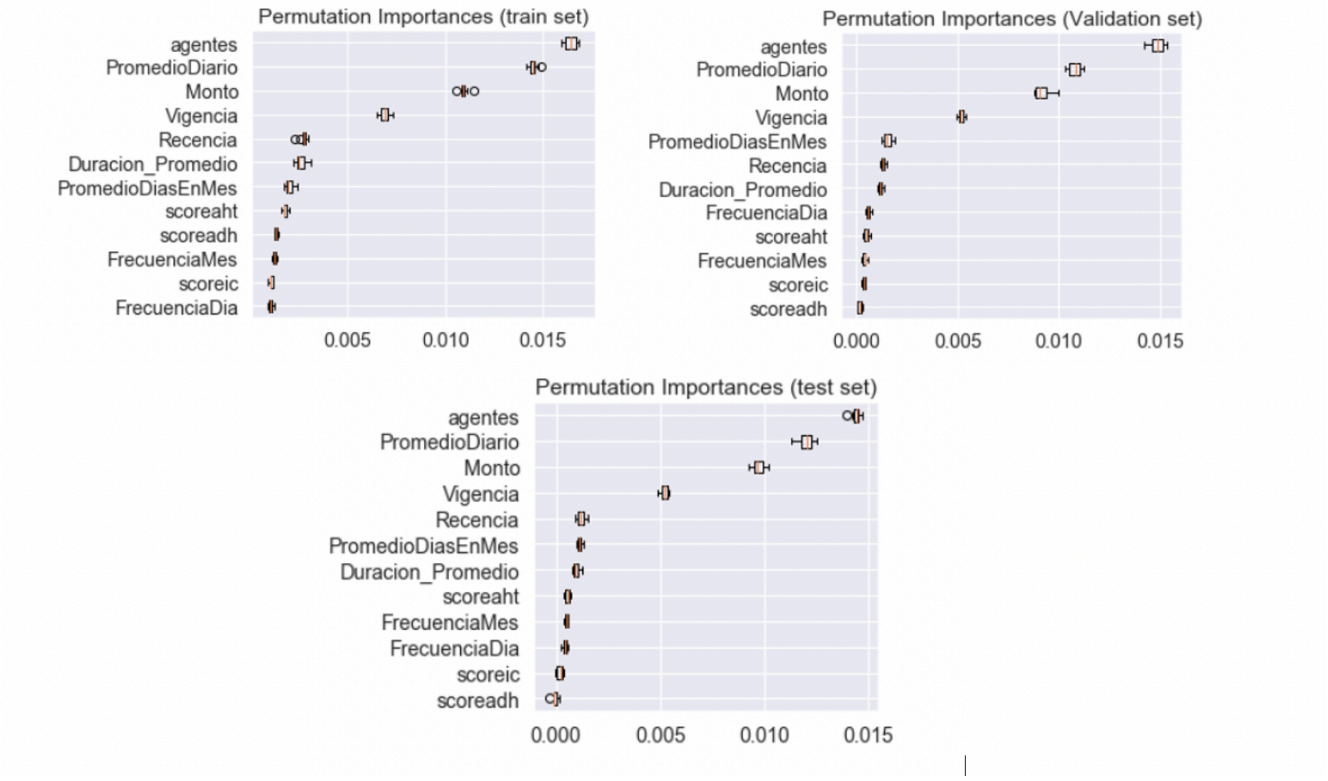

Figura 4.7: Permutation Feature Importance

De la figura 4.7 se observa que las variables más importantes son: 'agentes', 'PromedioDiario', 'Monto','Vigencia','Recencia', 'PromedioDiasEnMes' y 'Duracion_Promedio'. todas son variables transaccionales de una misma fuente de información por lo que se evaluaron los modelos anteriormente mencionados con este conjunto de características y el resultado fue el

siguiente:

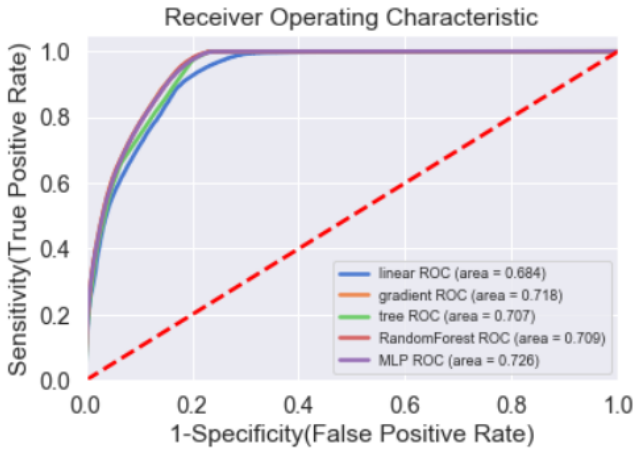

Figura 4.8: Comparativo curvas ROC-AUC

	linear	gradient	tree	RandomForest	MLP
auc	0.687	0.721	0.710	0.711	0.727
precision	0.804	0.817	0.805	0.817	0.808
recall	0.687	0.721	0.710	0.711	0.727
f1score	0.726	0.757	0.745	0.750	0.759

Figura 4.9: Comparativo métricas

En las figuras 4.8 y 4.9 se observan resultados similares a los obtenidos en la fase 1 en los conjuntos de características más grandes, para la mayoría de modelos al comparar las distintas métricas y adicionalmente es un modelo más parsimonioso, ya que obtiene este resultado con 7 variables y todas transaccionales lo que facilita su implementación. La estructura de red neuronal multicapa es de dos capas ocultas y 50 neuronas en cada capa, y con los siguientes hiperparámetros escogidos en el grid search: learning_rate_init=0.2, max_iter=100, n_iter_no_change=50, solver='sgd', tol=0.01. El bosque aleatorio se obtiene con una profundidad máxima de 9, un número mínimo de 8 muestras en un nodo y un total de 200 estimadores. El Gradient Boosting usa una profundidad máxima de 6 y de la misma manera 200 estimadores y para el árbol de decisión se usa el criterio de Gini, una profundidad máxima de 6 y un número mínimo de muestras en las hojas de 8.

Hasta este punto los modelos con mejores resultados en las distintas métricas son el gra-

dient boosting y la red neuronal. Sin embargo antes de pasar al siguiente capítulo de afinación de los modelos se analiza el resultado de las métricas de *precisión* y *recall* que son importantes para identificar el mejor modelo de predicción acorde al problema planteado.

La métrica de *precisión* hace referencia a la *calidad* del modelo y para este caso responde a la pregunta: Qué porcentaje de los clientes que dijimos que recontactarían así lo hicieron?, esto significaría que a los que no recontacten pero si esperábamos que lo hicieran, simplemente no se les aplicaría alguna estrategia y en ese caso no pasaría nada. Con base en esto para el gradient boosting y para el bosque aleatorio el resultado de precisión fue de 0.817 en ambos casos, es decir, que aplicaríamos alguna estrategia al 81.7 % de los clientes que recontacten y el modelo se estaría equivocando en un 19 % aproximadamente.

Por otro lado, teniendo en cuenta que la métrica de *recall* se enfoca en la *cantidad* de recontactos que el modelo es capaz de identificar, estaríamos respondiendo a la pregunta: Qué porcentaje de los clientes que recontactarán el modelo es capaz de identificar?. La respuesta sería 72 % de los clientes, para los modelos de gradient boosting y red neuronal por ejemplo.

Con base en lo anterior, para este problema interesa encontrar el modelo que mejores resultados tenga en su métrica de *recall*, ya que con la precisión, en caso de equivocarse al decir que alguien recontacta y no lo hace, no pasa nada, simplemente no se activaría la estrategia, sin embargo, si es mejor poder identificar el mayor número de clientes que se espera que vuelvan a llamar para darles un tratamiento especial.

Ajuste fino de los modelos

5.1. Optimización de modelos con balanceo de clases

Con el fin de mejorar los resultados en la predicción de la clase minoritaria (en este caso los clientes que generan un recontacto) y maximizar el resultado de la métrica de *recall* la cual por lo explicado anteriormente es la métrica que más sentido tiene en este problema, se decidió realizar un balanceo de clases, utilizando la librería imbalanced-learn de scikit learn de python. En este caso se evaluaron dos métodos, uno para aumento de la clase minoritaria y otra para disminución de la clase mayoritaria. Este ejercicio se realizó con el último conjunto de características seleccionado con la técnica de *permutation feature importance* estudiada anteriormente. En este caso se realizó el balanceo con ambas técnicas sobre el conjunto de entrenamiento para no sesgar los resultados del conjunto de validación y prueba.

5.1.1. Sobremuestreo

Para llevar a cabo este proceso se utilizó el método Naive random over-sampling, el cuál consiste en aumentar las clases que se encuentran subrepresentadas por medio de un muestreo aleatorio con reemplazo de éstas clases, el RandomOverSampler ofrece este esquema. (Lemaître et al., 2017,) A continuación se muestra en las siguientes figuras los resultados obtenidos a partir del OverSampling, tanto las curvas ROC-AUC como las diferentes métricas evaluadas.

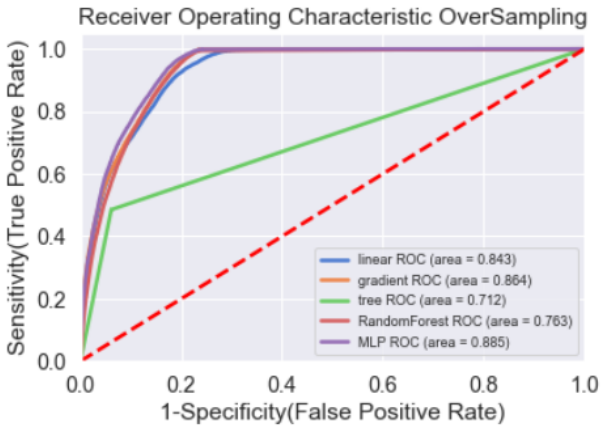

Figura 5.1: Comparativo curvas ROC-AUC Oversampling

	linear	gradient	tree	RandomForest	MLP
auc	0.843	0.864	0.712	0.763	0.885
precision	0.696	0.699	0.720	0.743	0.685
recall	0.843	0.864	0.712	0.763	0.885
f1score	0.732	0.735	0.716	0.752	0.710

Figura 5.2: Comparativo métricas Oversampling

En las figuras 5.1 y 5.2 se observa una mejora significativa tanto el *AUC* como en el *recall* en algunos de los modelos, que es nuestra métrica de interés. Las diferencias en el *recall* son las siguientes:

	Conjunto validación				
Recall	**Regresión**	**Gradient Boosting**	**Árbol de decisión**	**Bosque aleatorio**	**Red MLP**
Sin balanceo	0,697	0,722	0,731	0,707	0,676
Con Sobremuestreo	0,843	0,864	0,712	0,763	0,885
Diferencia	**0,146**	**0,142**	**-0,019**	**0,056**	**0,209**

Figura 5.3: Comparativo recall Oversampling

El modelo de regresión, el gradient boosting y la red neuronal presentan mejoras significativas, siendo esta última la de mejor resultado de *recall*. Sin embargo el árbol y el bosque no presentan mejora en esta métrica.

5.1.2. Submuestreo

Para el caso de submuestreo se utilizó el método Controlled under-sampling techniques, el cual consiste en seleccionar aleatoriamente un subconjunto de datos para las clases objetivo. (Lemaître et al., 2017,). Los resultados al aplicar esta técnica se presentan a continuación:

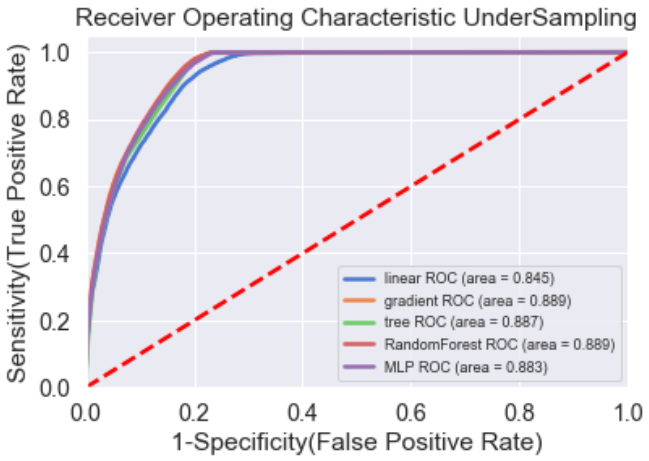

Figura 5.4: Comparativo curvas ROC-AUC Undersampling

	linear	gradient	tree	RandomForest	MLP
auc	0.844	0.889	0.888	0.889	0.884
precision	0.696	0.692	0.688	0.691	0.689
recall	0.844	0.889	0.888	0.889	0.884
f1score	0.731	0.721	0.715	0.720	0.717

Figura 5.5: Comparativo métricas Undersampling

En las figuras 5.4 y 5.5 se observa que al utilizar la técnica de submuestreo para balancear la base de datos se logra una mejora significativa tanto en el *AUC* como en el *recall*. En este caso la métrica de precisión se afecta un poco, pero como ya se mencionó interesa maximizar la métrica de *recall*. las diferencias se presentan en la siguiente figura:

	Conjunto validación				
Recall	**Regresión**	**Gradient Boosting**	**Árbol de decisión**	**Bosque aleatorio**	**Red MLP**
Sin balanceo	0,697	0,722	0,731	0,707	0,676
Con Submuestreo	0,848	0,891	0,886	0,89	0,885
Diferencia	**0,151**	**0,169**	**0,155**	**0,183**	**0,209**

Figura 5.6: Comparativo recall Undersampling

En este caso se obtienen muy buenos resultados de *recall* en todos los modelos, siendo los mejores el Gradient Boosting y el Bosque Aleatorio. A continuación se presentan los resultados de los modelos tanto con submuestreo como con sobremuestreo en el conjunto de entrenamiento y validación, con el fin de seleccionar el modelo definitivo.

UNDER	**AUC**		**PRECISIÓN**		**RECALL**	
SAMPLING	**TRAIN**	**VAL**	**TRAIN**	**VAL**	**TRAIN**	**VAL**
REG LINEAL	0.851	0.848	0.851	0.696	0.851	0.848
ÁRBOL	0.886	0.886	0.895	0.695	0.886	0.886
BOSQUE	0.896	0.889	0.910	0.690	0.896	0.889
GRADIENT	0.893	0.891	0.906	0.692	0.893	0.891
RED MLP	0.890	0.886	0.903	0.689	0.890	0.886

OVER	**AUC**		**PRECISIÓN**		**RECALL**	
SAMPLING	**TRAIN**	**VAL**	**TRAIN**	**VAL**	**TRAIN**	**VAL**
REG LINEAL	0.846	0.843	0.846	0.696	0.846	0.843
ARBOL	1.000	0.712	1.000	0.720	1.000	0.712
BOSQUE	1.000	0.763	1.000	0.742	1.000	0.763
GRADIENT	1.000	0.763	1.000	0.742	1.000	0.763
RED MLP	0.891	0.885	0.900	0.695	0.891	0.885

Figura 5.7: Comparativo modelos final

El mejor resultado de *recall* en el conjunto de validación se obtiene realizando balanceo de clases, con submuestreo en los modelos de bosque aleatorio y gradient boosting, donde se obtiene un valor en esta métrica de 0.89 %. De la misma manera se obtienen los mejores resultados en la métrica de AUC y no se observa sobreajuste, ya que los resultados en entrenamiento son muy similares en el conjunto de validación.

Para finalizar se evaluó la capacidad de generalización de los modelos en el conjunto de prueba (nunca antes visto) y teniendo en cuenta que el objetivo es poder implementar el modelo en tiempo real, se evaluó también el tiempo de entrenamiento y de inferencia de cada uno de los modelos y se obtuvo lo siguiente:

MODELO	RECALL - TEST		T.ENTRE	T. INFE
	UNDER	OVER		
REG LINEAL	0.851	0.845	0.656	0.003
ARBOL	0.885	0.706	0.042	0.009
BOSQUE	0.890	0.759	3.079	1.275
GRADIENT	0.890	0.863	1.753	0.110
RED MLP	0.886	0.885	10.317	0.209

Figura 5.8: Comparativo tiempos

Con base en la figura 5.8, se puede concluir que el bosque aleatorio y el gradient boosting, presentan una capacidad de generalización muy similar al ser aplicados sobre el conjunto de prueba y el de mejor desempeño computacional lo presentó el gradient boosting, siendo este el modelo seleccionado para la predicción de recontactos, con un valor en su métrica de *AUC* y de *recall* de 0.89, un valor superior al alcanzado por el modelo trabajado en el estudio revisado en el estado del arte (Moazeni and Andrade, 2018,).

Bosque A.	predicho	
real	No Recontacto	Recontacto
Negativo	88,994	22,889
Positivo	228	14,205

Gradient B.	predicho	
real	No Recontacto	Recontacto
Negativo	89,335	22,494
Positivo	256	14,232

Figura 5.9: Comparativo matriz de confusión

Si se comparan las matrices de confusión de ambos modelos, para comprender mejor su resultado, se observa que con el gradient boosting se obtienen 395 falsos positivos (clientes que el modelo predijo que recontactarían y no lo hicieron) menos con respecto al bosque aleatorio (22289 - 22494) y 28 falsos negativos adicionales. (clientes que el modelo predijo que no recontactarían y si lo hicieron), por lo tanto no se tuvo una estrategia prevista para ellos, es por esto que la métrica de interés para la selección de modelos fue el *recall*, que tiene en cuenta

estos falsos negativos.

Hallazgos principales

6.1. Discusión de los Resultados

Teniendo en cuenta los resultados obtenidos en toda la experimentación realizada a través de las diferentes técnicas de selección de variables y los diferentes modelos entrenados (150 modelos en total), se encontró que a nivel general los resultados obtenidos no presentaron una variación significativa en las métricas del ROC_AUC y del f1 Score.

Con base en estos resultados tan similares entre sí, se decidió utilizar la técnica de *Permutation Feature Importance* para tener mayor claridad en la selección del conjunto de características más adecuada, con la cuál se seleccionaron las 7 características más representativas y se corrieron nuevamente los 5 modelos vistos anteriormente (Regresión logística, árbol de decisión, bosque aleatorio, gradient boosting y red multicapa perceptrón), encontrando resultados promedio de auc = 0.7112 en los 5 modelos entrenados, un valor levemente superior al promedio del resultado en todos los modelos de la experimentación inicial pero con tan solo 7 variables.

En busca de optimizar los modelos se logró un incremento en métricas como el AUC (89.1 %) y el Recall (89.1 %) en el conjunto de validación, realizando un balanceo de muestras (disminuyendo la clase mayoritaria). Los resultados de la métrica de precisión no cambiaron mucho (69.2 %), sin embargo, como ya se mencionó anteriormente, para este tipo de problema puede sacrificarse un poco esta métrica ya que no se tendría tanto impacto porque en el peor de los casos se estaría haciendo un refuerzo sobre la gestión del cliente o no se estaría activando una estrategia de enrutamiento sobre los clientes que no recontactan.

Conclusiones

En este documento se propone la implementación de un modelo de aprendizaje automático para predecir la probabilidad de que un cliente vuelva a comunicarse al centro de contactos en los próximos tres días. Adicionalmente se plantea una selección rigurosa de las características que más influyen en que se presente dicho recontacto. Para esto se analizan alrededor de 421 mil registros de clientes del año 2019 de una reconocida aerolínea Colombiana y aproximadamente 15 variables relacionadas con los contactos históricos y los agentes y desempeño asociado en la atención de dichos contactos. La metodología planteada se basa en una selección rigurosa de características, utilizando alrededor de 11 técnicas diferentes y analizando en una matriz las variables sobre las cuales se presenta una mayor coincidencia de selección por los diferentes métodos. Adicionalmente se realiza un proceso de experimentación con 5 modelos de aprendizaje automático para identificar con qué conjunto de características y con qué modelo de aprendizaje se obtiene un mejor resultado en la predicción del recontacto , para esto se evalúan distintas métricas tales como ROC-AUC, Precisión, Recall y f1-Score, siendo el recall la métrica de interés en este estudio. Finalmente con el conjunto de características seleccionado se realiza un proceso de optimización de modelos realizando un balanceo de clases en busca de un mejor resultado en las métricas mencionadas. Se encontró que dentro de las variables más importantes y adicionalmente de mayor coincidencia por los distintos métodos están la de cantidad de agentes que atienden las llamadas de cada cliente, la frecuencia de los contactos históricos de cada cliente y el promedio diario de llamadas de cada cliente en un mismo día. También pudo observarse que aquellos usuarios que más usan el canal y de manera más reciente son los que presentan mayores probabilidades de generar un recontacto.

Para terminar, luego de llevar a cabo todo el proceso de experimentación, validación y ajuste de modelos, lo más recomendable y además robusto es utilizar el modelo de aprendizaje *Gradient Boosting* con un balanceo de la clase minoritaria ya que además de sus buenos resultados en las diferentes métricas, específicamente en la métrica de *recall* (0.891), fue también

el modelo que tomó menos tiempo en el proceso de entrenamiento e inferencia. Todo esto más una buena selección de características facilita mucho la implementación en tiempo real permitiendo obtener beneficios asociados a la mejora en la experiencia del cliente y disminución de costos operaciones asociados a la atención de llamadas que pueden empezar a minimizarse o atenderse de manera proactiva antes de que vuelvan a llamar.

Anexos

A continuación se describen los notebooks que contienen el código que soporta este trabajo:

- **00-Preprocesamiento y descriptivo.ipynb**: Contiene el preprocesamiento, análisis de valores atípicos y análisis descriptivo de las variables de estudio.
- **01-Proyecto recontados-sin balanceo.ipynb**: Contiene el entrenamiento de modelos asociados a los conjuntos de variables de la figura 3.10 con una partición del train test y validación de 60 %-20 %-20 %.
- **02-Proyecto recontados-cambio train-test-val.ipynb**: Contiene el entrenamiento de modelos asociados a los conjuntos de variables de la figura 3.10 con una partición del train test y validación de 50 %-20 %-30 %.
- **04-Proyecto recontados-feature selection.ipynb**: Contiene el desarrollo de las técnicas de selección de variables utilizadas en este estudio.
- **05-Proyecto recontados-model_selection.ipynb**: Contiene el desarrollo del entrenamiento de modelos con cada uno de los conjuntos de variables seleccionadas con las técnicas de selección de variables para una partición del test y validación de 60 %-20 %-20 %.
- **06-Proyecto recontados-model_selection-50-20-30.ipynb**: Contiene el desarrollo del entrenamiento de modelos con cada uno de los conjuntos de variables seleccionadas con las técnicas de selección de variables para una partición del test y validación de 50 %-20 %-30 %.
- **07-Permutation Importance.ipynb**: Desarrolla la técnica de selección de variables de permutation feature importance y se entrenan los modelos con el conjunto de variables seleccionado.

- **08-Balanceo de clases UnderSampling.ipynb**: Aplica la técnica de balanceo de clases con el método de undersampling para el conjunto de características seleccionado con el método de permutation feature importance.
- **09-Balanceo de clases OverSampling.ipynb**: Aplica la técnica de balanceo de clases con el método de oversampling para el conjunto de características seleccionado con el método de permutation feature importance.
- **10-Decision Tree con undersampling-interpretation.ipynb**: realiza la visualización de un árbol de clasificación con las variables finalmente seleccionadas para su interpretación.
- **12-distribución resultados auc experimentación.ipynb**: analiza la distribución de los resultados de AUC de los diferentes modelos entrenados en la fase de experimentación.
- **13-Matriz_Feature_Selection.xlsx**: contiene el resumen de métodos de selección de variables y los resultados de AUC y F1 score de los diferentes modelos entrenados.

Referencias

Andrade, R., Moazeni, S., and Ramirez-Marquez, J. E. (2020). A systems perspective on contact centers and customer service reliability modeling. *Systems Engineering*, 23(2):221–236.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Butgereit, L. (2020). Work towards using micro-services to build a data pipeline for machine learning applications: A case study in predicting customer churn. In *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, pages 87–91.

Cai, J., Luo, J., Wang, S., and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79.

Calviño, H. A. A. (2017). Métodos para la mejora de predicciones en clases desbalanceadas en el estudio de bajas de clientes (churn). *Master en Técnicas Estadísticas USC-Universidad da Coruña*.

Chauhan, N. S. (2020). Optimización de hiperparámetros. urlhttps://www.datasource.ai/es/data-science-articles/optimizacion-de-hiper-parametros-para-modelos-de-aprendizaje-automatico.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., and Herrera, F. Nuevas métricas de poda basadas en ordenamiento para conjuntos de clasificadores en problemas con clases no balanceadas.

Gottemukkula, V. and Derakhshani, R. (2011). Classification-guided feature selection for nirs-based bci. In *2011 5th International IEEE/EMBS Conference on Neural Engineering*, pages 72–75. IEEE.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

Moazeni, S. and Andrade, R. (2018). A data-driven approach to predict an individual customer's call arrival in multichannel customer support centers. In *2018 IEEE International Congress on Big Data (BigData Congress)*, pages 66–73.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Wikipedia contributors (2020). Mutual information — Wikipedia, the free encyclopedia. [Online; accessed 31-July-2020].

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.