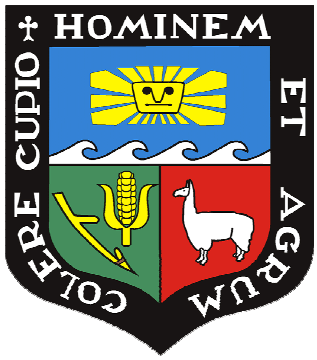
UNIVERSIDAD NACIONAL AGRARIA
LA MOLINA

ESCUELA DE POSGRADO

MAESTRÍA EN ESTADÍSTICA APLICADA



"PREDICCIÓN DE FUGA DE CLIENTES EN UNA EMPRESA DE TELEFONÍA UTILIZANDO EL ALGORITMO ADABOOST DESBALANCEADO Y LA REGRESIÓN LOGÍSTICA ASIMÉTRICA"

Presentada por:

ALDO RICHARD MEZA RODRÍGUEZ

TESIS PARA OBTENER EL GRADO DE MAESTRO
MAGISTER SCIENTIAE EN ESTADÍSTICA APLICADA

Lima-Perú
2018

RESUMEN

La presente investigación tiene como propósito aplicar y comparar el modelo de regresión logística y el algoritmo Adaboost en datos desbalanceados, esto a efecto de predecir la fuga de clientes en una empresa del sector de telefonía móvil. El algoritmo Adaboost se sustenta en el aprendizaje adaptativo al entrenar clasificadores débiles combinándolos en conjunto para obtener un clasificador cuyo rendimiento sea fuerte. En cuanto a la regresión logística su modelamiento se realizó estrictamente desde una perspectiva de minería de datos, donde la clasificación es el objetivo y el rendimiento se evaluó en un conjunto de validación. Ambas técnicas se compararon mediante dos procedimientos, el primero mediante métodos de muestreo (sub-muestreo, sobre-muestreo y SMOTE) y el segundo modificando y/o ajustando el algoritmo o función. Al trabajar con datos desbalanceados la tasa de error de clasificación es ineficiente, por lo que las medidas de desempeño para elegir al mejor modelo fueron la precisión, el recall (sensibilidad), el F-measure, y como medida principal el AUC a través de curvas ROC. Al formar modelos logísticos con los métodos de muestreo, las medidas de desempeño arrojaron resultados similares, lo mismo pasó al formar modelos con el algoritmo Adaboost, sin embargo al comparar la regresión logística (AUC=0.86) con el algoritmo Adaboost (AUC =0.93), este último tuvo el mejor desempeño. En cuanto al ajuste a nivel de algoritmo o función, en la regresión logística se trabajó de dos maneras, el primero (Logit Asym) incluyendo en la FDA un valor Kappa (k) y el segundo (Power Logit) un valor Lambda (λ), en ambos modelos se identificaron los valores óptimos de k (0.02) y λ (2.5), en cuanto al algoritmo Adaboost (Adaboost Asym) se ajustó el peso de la clase minoritaria cuyo costo de clasificación fue errónea. La comparación de estos tres modelos ajustados dio como mayor rendimiento al algoritmo Adaboost. Finalmente se realizó la validación cruzada con 10 iteraciones para todos los modelos dando resultados similares al método de retención. Realizada todas las comparaciones y las medidas de desempeño se concluye que el modelo óptimo para la predicción de fuga de clientes en la empresa de telefonía es el algoritmo Adaboost.

Palabras clave: Adaboost, datos desbalanceados, regresión logística, clasificación, Power Logit, validación cruzada.

ABSTRACT

The purpose of this research is to apply and to compare the logistic regression model and the Adaboost algorithm in unbalanced data, the purposes of predict the customer churn in a company in the mobile telephony sector. The Adaboost algorithm is based on adaptive learning when training weak classifiers, combining them together to obtain a classifier whose performance is strong. In terms of logistic regression, its modeling was done strictly from a data mining perspective, where the classification is the objective and the performance was evaluated in a validation set. Both techniques were compared using two methods, the first using sampling methods (sub-sampling, oversampling and SMOTE) and the second modifying and / or adjusting the algorithm or function. When working with unbalanced data the classification error rate is inefficient, so the performance measures to choose the best model were accuracy, recall (sensitivity), F-measure, and as a main measure the AUC through ROC curves. When forming logistic models with the sampling methods, the performance measures yielded similar results, the same happened when forming models with the Adaboost algorithm, however when comparing the logistic regression (AUC = 0.86) with the Adaboost algorithm (AUC = 0.93), the latter had the best performance. Regarding the adjustment at the level of algorithm or function, the logistic regression was worked in two ways, the first (Logit Asym) including in the FDA a Kappa value (k) and the second (Power Logit) a Lambda value (λ), in both models the optimal values of k (0.02) and λ (2.5) were identified, in terms of the Adaboost algorithm (Adaboost Asym) the weight of the minority class whose cost of classification was erroneous was adjusted. The comparison of these three adjusted models gave the Adaboost algorithm a higher performance. Finally, cross-validation was carried out with 10 iterations for all the models, giving similar results to the retention method. Once all the comparisons and measures of performance are concluded, it is concluded that the optimal model for the prediction of customer leakage in the telephone company is the Adaboost algorithm.

Keywords: Adaboost, unbalanced data, logistic regression, classification, Power Logit, cross validation.

DEDICATORIA

A mi hijo amado Jhonatan Jahir, que a pesar de que perdió a su ser más querido, me demostró que es un niño muy fuerte y valiente, a él, quien es mi motor, por quien lucho cada día y quien fue el empuje principal para concluir este trabajo de investigación.

AGRADECIMIENTOS

Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mis padres y hermanos que a pesar de no estar cerca siempre me ha dado las fuerzas y la motivación para seguir adelante.

A la Universidad Nacional Agraria la Molina quien ahora es mi alma Mater y a todos mis colegas del departamento de Estadística e Informática.

Al Mg. Sc. Jorge Chue, por el soporte y dirección en la elaboración de esta investigación.

A mis grandes amigos y colegas, Joao Rado, Diana Rebaza, Jesús Gamboa, Rolando Salazar, Juan Carlos Orosco, Ana Vargas, quienes me demostraron su gran apoyo en momentos difíciles, y con quienes comparto muy gratos momentos.

A la señora Rosa Sacsa, por el gran apoyo, cariño y aprecio que me demuestra constantemente.

A Jorge Ramos, por su amistad y gran consideración.

A todas las personas que de una u otra manera colaboraron con la realización de esta investigación.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

I. INTRODUCCIÓN

1.1 PROBLEMA DE INVESTIGACIÓN

La fuga de clientes, conocida en el idioma inglés como Churn, es un problema de gran preocupación para los diferentes sectores de la economía de un país. Este problema está presente en casi todas las empresas, sean estas de la industria, banca, telecomunicaciones, etc.

En el sector telecomunicaciones y específicamente en telefonía móvil, el Churn es uno de los más sensibles, debido a que el abandono o cambio de operador por parte de los usuarios se produce en medio de agresivas estrategias de marketing que buscan captar clientes por parte de las empresas que operan actualmente en el mercado y también por las empresas nuevas que están ingresando a ella.

Barrientos (2011) informa que uno de los principales problemas de cambiar una línea de telefonía está asociado a factores generados principalmente por el servicio entregado por la empresa tales como calidad de la señal, cobertura, servicio al cliente, precios, etc.

Según cifras del Osiptel, a abril del 2017, el mercado de telefonía móvil peruano registra 36 millones 99 mil líneas activas**,** y 4 operadoras nacionales para elegir, a la vez todos los meses alrededor de 170 mil clientes migran de una a otra compañía de telefonía móvil, generando un gran intercambio entre las operadoras por captación de clientes.

Ante la necesidad de prever e identificar a los posibles clientes que fugan, los modelos de predicción son una de las herramientas y soporte clave para identificarlos y entender el patrón y el motivo que les lleva a prescindir del servicio.

En un torneo para la medición y comprensión de la exactitud predictiva de los modelos de la pérdida de clientes, en la que compitieron 33 modelos de predicción de fuga, la regresión logística y los árboles de clasificación fueron los de mejor desempeño (Neslin *et al.* 2006).

En la actualidad se hace necesaria la creación de nuevos modelos que permitan predecir con el mayor índice de desempeño y la menor tasa de error, la fuga de los actuales y futuros clientes; es así que se busca entrenar modelos que pueden ser más óptimos que los modelos clásicos como es el caso de la Regresión Logística.

Recientemente, un número importante de investigaciones en minería de datos y específicamente en máquinas de aprendizaje, se están ocupando de desarrollar nuevos métodos de potenciación de clasificadores que permitan la mejora de las técnicas de predicción. Una alternativa para predecir es la metodología Boosting que es una familia de algoritmos de predicción basadas en aprendizajes, entre los algoritmos Boosting uno de los más representativos es el Adaboost, lo que según Pérez (2014) es conocida por su gran capacidad de mejorar el rendimiento de cualquier algoritmo de aprendizaje y por minimizar el error de predicción en forma eficiente, en este caso, las fugas o intercambios de clientes entre operadoras de telefonía celular. Escobar y Loas (2015) realizaron un estudio sobre la detección temprana sobre fuga o deserción de estudiantes con 260000 observaciones, utilizando una gran variedad de técnicas de predicción de minería de datos (9 técnicas en total), donde el Adaboost fue el más sobresaliente con la menor tasa de error.

Hadad *et al.* (s.f) manifiestan que el problema del desbalanceo es un tema de creciente interés en el aprendizaje automático, debido a sus grandes efectos sobre los resultados obtenidos y las aplicaciones en donde se puede encontrar esta situación. El conjunto de datos desbalanceados es el que presenta un desequilibrio notable en el número de categorías pertenecientes a cada clase, provocando un sesgo en el desempeño de los clasificadores hacia el reconocimiento de las clases más numerosas.

Ante esto surge la interrogante de cuál de las técnicas estadísticas, Adaboost o Regresión Logística para datos asimétricos predice mejor la fuga de clientes en la empresa de telefonía móvil de este estudio (por motivos de privacidad se mantiene en reserva el nombre de la empresa de telefonía, esto a razón de las propias políticas de la empresa en estudio), con los indicadores de mejor desempeño y métricas de precisión.

Por lo tanto en esta investigación se pretende comparar la capacidad predictiva, identificando las variables que afectan el comportamiento de la fuga de clientes en telefonía móvil, para utilizarlas como variables de entrada en los modelos predictivos propuestos.

Para comparar los dos modelos se utilizó el concepto de “mejor desempeño” el cual se mide a través de un conjunto de indicadores recomendados para datos asimétricos, tales como: la precisión, recall (sensibilidad), F-measure y el AUC (área debajo de la curva).

Para este estudio se utilizaron los registros de los últimos 6 meses (Mayo - Octubre del 2017) de clientes de una empresa internacional de telefonía móvil que tiene presencia importante en el Perú, tomando como principal área los registros nacionales del programa postpago.

1.2 JUSTIFICACIÓN DE LA INVESTIGACIÓN

Generalmente en toda empresa para establecer planes y estrategias de retención no sólo es necesario saber cuándo los clientes están a punto de irse, sino también por qué, ya que solo conociendo los motivos sabremos cómo retenerlos. Aquí es donde juegan los elementos “causas y variables” y donde los profesionales responsables de fidelización y predicción, deben responder adecuadamente.

Pérez (2014) detalla que por diferentes estudios en empresas al momento de identificar “variables importantes” se sabe que los clientes antiguos tienen menor propensión al “Churn” que los recién captados; que los de mayor edad suelen plantear menos problemas que los más jóvenes, etc. Por lo tanto, al identificar un modelo de predicción y sus variables respectivas, y combinando las distintas causas y perfiles claramente definidas, la empresa podrá determinar qué acciones de retención llevar a cabo.

El estudio de la fuga de clientes busca evitar que estos migren a las otras operadoras, es aquí donde un modelo de fuga de clientes se hace fundamental para conocer con precisión por qué los clientes deciden retirarse definitivamente o migrar hacia otros operadores. Si los responsables de la toma de decisiones la empresa logran identificar que variables influyen para que un cliente decida quedarse o simplemente fugue, seguramente cambiaría cuanto antes la estrategia de negocios para enfocar menos esfuerzos hacia la retención y focalizarse en atacar las verdaderas causas del retiro.

Este trabajo de tesis permitirá indagar en el pasado de los clientes y descubrir que comportamientos comunes existen antes del retiro, se podrá predecir los volúmenes de retiro, e incluso detectar cuanto antes aquellos clientes que pueden estar pensando en prescindir de

los productos y servicios de la empresa, esto será posible tomando muestras y estudiando los patrones de comportamiento en clientes que se han dado de baja en el pasado.

Por lo tanto, conociendo el problema que afronta la empresa, y la disponibilidad de esta al proporcionar un gran volumen de datos a través de una extensa data, se hace menester el conocimiento y la aplicación de la minería de datos o data mining, la que contiene en sí una metodología estándar a seguir para llegar a la predicción de los clientes a través de diferentes y múltiples algoritmos tales como el Adaboost (Máquinas de aprendizaje adaptativo) la cual se pondrá a prueba y medirá su desempeño en comparación con otro modelo eficiente de gran uso (regresión logística).

1.3 OBJETIVOS DE LA INVESTIGACIÓN

1.3.1 OBJETIVO GENERAL

Identificar entre el algoritmo Adaboost desbalanceado y la Regresión Logística asimétrica aquella que tiene mayor precisión en la predicción de fuga de clientes en el sector de telefonía móvil mediante indicadores de mejor desempeño y rendimiento.

1.3.2 OBJETIVOS ESPECÍFICOS

- Realizar la limpieza de datos, el análisis de valores perdidos, y la detección de posibles valores atípicos en la base de datos.
- Determinar las variables de mayor importancia e incidencia en la decisión de fuga.
- Comparar las metodologías de Adaboost desbalanceado y regresión logística Asimétrica mediante técnicas de muestreo y ajuste de algoritmo en función al mejor desempeño.

II. REVISIÓN DE LITERATURA

2.1 Definición de fuga de clientes (Churn)

Según Pérez (2014), en el marco de las telecomunicaciones, el Churn se define como "la acción de cancelar el servicio prestado por la compañía". En esta cancelación, el cliente decide renunciar a la empresa (voluntaria) o en todo caso la empresa puede expulsarlo por algunas irregularidades o por no cumplir con las obligaciones o pagos acordados (involuntaria). Por lo tanto se detalla el Churn en función a la decisión del cliente en abandonar la empresa.

Según Huang, citado por Barrientos y Ríos (2013), en el sector de las telecomunicaciones también se define al Churn como aquel término "usado para describir colectivamente el cese de servicios de la suscripción de un cliente, donde el cliente es alguien que se ha unido a la compañía por al menos un periodo de tiempo. Un Churner o fugado es un cliente que ha dejado la compañía".

Según Lejenue (2001) el Churn se define como: "La propensión de clientes a efectuar el cese de los negocios que tenga con una compañía en un periodo de tiempo determinado"

En esta investigación se utiliza el término de Churn relacionado con el concepto definido por Huang hacia el sector de telecomunicaciones.

2.1.2 Tipos de Churn

Según Muñoz (2016), básicamente se pueden encontrar dos tipos de Churn:

- **Pasivo:** Es la tasa de Churn de todos aquellos clientes que no se dan de baja de manera directa, a estos se les acaba su periodo de suscripción y no deciden renovar. Esto ocurre cuando el usuario no está interesado en el producto, olvidándose inclusive de que cuenta con el servicio o producto.

- **Activo:** Es la tasa de Churn donde los usuarios toman la decisión de terminar suscripción al servicio y buscan los mecanismos formales para darse de baja. (Generalmente suelen llamar por teléfono)

Para este estudio, el Churn de clientes son los que voluntariamente decidieron cambiarse de línea y simplemente no renovaron o renunciaron a su servicio, es decir clientes activos que renuncian al servicio.

2.2 La clasificación

Brownlee (2017) define a la clasificación como la tarea de predecir una etiqueta de clase discreta, donde a menudo un problema de dos clases se le denomina clasificación binaria, mientras que a problemas de más de dos clases se le denomina clasificación múltiple.

Según Obregón (2016) la clasificación en el marco de las ciencias computacionales se enfrenta a dos tipos de problemas bajo diferentes perspectivas: conocer anticipadamente el número de categorías o grupos en los que se desea hacer una clasificación de datos, de tal forma que hay que definir reglas para clasificar futuras observaciones; o bien establecer determinado número de clases dentro de un grupo de datos obtenidos.

En este caso se utilizó el enfoque de clasificar observaciones futuras en clases o categorías ya conocidas, es decir conociendo las clases como variable principal, clasificar nuevas observaciones a estas categorías.

2.2.1 El enfoque estadístico

Obregón (2016) también explica que la primera fase, conocida como clásica, deriva de los trabajos de Fisher en discriminación lineal. La segunda fase o fase moderna utiliza modelos más flexibles, muchos de los cuales tratan de proporcionar una estimación de la distribución conjunta de las características de cada clase, que a su vez puede dar lugar a una regla de clasificación.

Los métodos estadísticos se caracterizan por estar inmersos en una probabilidad explícita, la cual proporciona la probabilidad de pertenencia a cada clase en lugar de una clasificación simple. Por otro lado, se asume que las técnicas serán utilizadas por estadísticos lo cual

implica cierta intervención humana con la transformación y selección de variables, así como la metodología para la estructuración del problema general.

En cuanto al enfoque estadístico en esta investigación se vio reflejada principalmente en la regresión logística asimétrica

2.3 Metodologías de minería de datos

Según Rouse (2008) la minería de datos es el proceso de clasificar grandes conjuntos de datos para identificar patrones y establecer relaciones para resolver problemas a través del análisis de datos. Las herramientas de minería de datos permiten a las empresas predecir las tendencias futuras.

En la minería de datos, se formulan reglas de asociación para identificar patrones frecuentes, técnicas de clasificación, regresión, aplicación de diferentes algoritmos etc., para luego utilizarlas para la toma de decisiones y la comprensión de diferentes fenómenos.

Las técnicas de minería de datos se establecen a partir de diferentes metodologías, lo cuales aseguran resultados coherentes en función al resultado que se busca obtener. Entre las diferentes metodologías se mencionarán a las dos más usadas, tal es el caso de las metodologías SEMMA y CRISP-DM.

2.3.1 Metodología SEMMA

El acrónimo SEMMA proviene de Sample, Explore, Modify, Model, Assess, el cual fue desarrollado por SAS Institute. Esta metodología se refiere al proceso de selección, exploración y modelamiento de volúmenes grandes de datos, es decir la conducción de un proyecto de minería de datos.

Figura 1: Metodología SEMMA. Fuente: Adaptado de Santos y Azevedo (2005)

En la Figura 1 es posible observar la metodología SEMMA, la cual según Santos y Azevedo (2005) consta de cinco etapas:

1. Muestreo: etapa inicial donde se procede a preparar y recolectar los datos para su posterior exploración. El subconjunto de datos debe ser representativa, extrayendo una porción de un gran conjunto de datos lo suficientemente grande como para contener la información significativa, pero lo suficientemente pequeña como para manipularla rápidamente.

2. Exploración: es una de las etapas más trabajosas y consiste en la exploración de los datos mediante la búsqueda de tendencias y anomalías imprevistas con el fin de comprender y obtener ideas.

3. Modificación: esta etapa consiste en la modificación de los datos creando, seleccionando y transformando las variables para enfocar el proceso de selección del modelo. En esta etapa también se destaca la imputación de datos perdidos y la reducción de dimensiones.

4. Modelamiento: esta etapa consiste en modelar los datos al permitir que el software busque automáticamente una combinación de datos que prediga de manera confiable un resultado deseado para posteriormente proceder a la selección de modelos, pudiendo aplicar más de uno a la vez, para luego comparar los resultados obtenidos.

5. Evaluación: en esta etapa final se analizan los datos mediante la evaluación de la utilidad y fiabilidad de los hallazgos estimando qué tan bien funciona.

2.3.2 Metodología CRISP-DM

La metodología CRISP-DM (Cross Industry Standard Process for Data Mining) es un modelo de procesos para proyectos de minería de datos que incluye una guía que suele venir de las experiencias propias y también de los procedimientos estándar conocidos, las cuales pueden estar estructuradas en fases bidireccionales, puesto que al desarrollar una fase es posible revisar en forma total o parcial las anteriores.

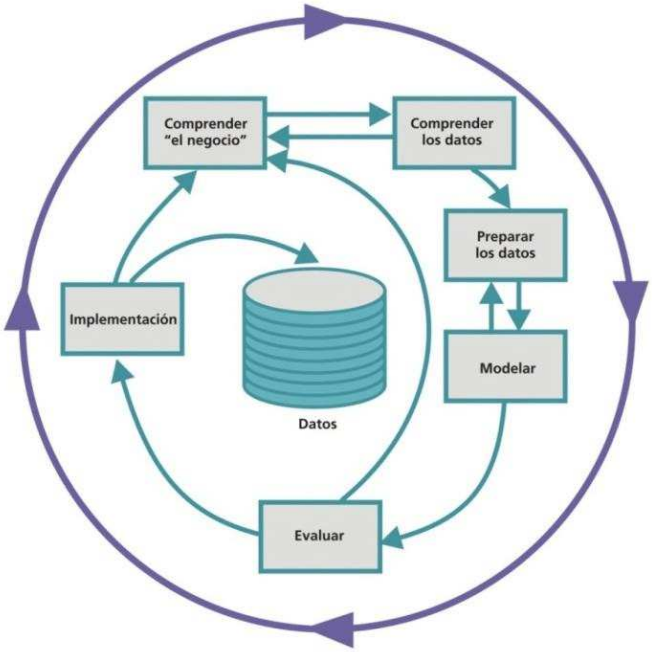


Figura 2: Etapas de la metodología CRISP-DM. Fuente: Chapman et al. (2000)

En la figura 2 se aprecia como la metodología se estructura en seis fases. Según Moine, citado por Jaramillo y Paz-Arias (2015) el desarrollo de esta serie de fases o etapas funcionan de manera cíclica e iterativa, cada una cuenta con tareas generales y específicas que permiten cumplir con los objetivos del proyecto.

Las fases son:

1. Comprensión del Negocio: fase inicial cuyo objetivo es la comprensión de los requisitos y objetivos del cliente, es quizás el paso más importante de la metodología, convierte el conocimiento en objetivos técnicos y después en un plan de proyecto.

2. Comprensión de los Datos: busca establecer un primer contacto con el problema, contiene algunas tareas como la recolección de datos, teniendo claro desde qué lugar fueron obtenidos, definiendo su calidad y estableciendo las relaciones más resaltantes para luego identificar las primeras hipótesis. Esta fase junto a las próximas dos, son quizás las que demandan mayor esfuerzo y tiempo en un proyecto de minería de datos.

3. Preparación de los Datos: después de la recolección de datos se procede a su preparación, para adaptarlos a las técnicas de minería de datos que se utilizarán posteriormente, esta etapa incluye la limpieza de los datos, preparándolos para la fase de modelación, incluyendo técnicas de normalización, discretización, tratamiento de valores perdidos, generación de

variables adicionales, integración de diferentes orígenes de datos, cambios de formato, entre otros (Rendón y Acosta 2006).

4. Modelamiento: se seleccionan las técnicas de modelado apropiadas para el proyecto. Esta selección se debe realizar en función de los siguientes criterios: que sea apropiada al problema, disponer de los datos adecuados, cumplir con los requisitos del problema, tiempo adecuado para obtener un modelo y conocimiento de la técnica.

5. Evaluación: se procede a la evaluación del modelo considerando el cumplimiento de los criterios de éxito del problema. Además, debe considerarse que la fiabilidad calculada para el modelo se aplique solo para los datos sobre los que se realizó el análisis. Es preciso revisar el proceso, teniendo en cuenta los resultados obtenidos, para poder repetir algún paso anterior, en el que se haya posiblemente cometido algún error. Es importante considerar que se pueden emplear múltiples herramientas para la interpretación de los resultados. Luego, si el modelo generado es válido en función de los criterios de éxito establecidos en la fase anterior, se procede a la explotación del modelo.

6. Implementación: luego de que el modelo ha sido construido y validado, se procede a transformar el conocimiento obtenido en acciones dentro del proceso de negocio.

2.3.3 El Aprendizaje automático

Según Moya (2016) el Aprendizaje Autónomo es una rama de la Inteligencia Artificial (IA) que tiene como objetivo crear sistemas capaces de aprender por ellos mismos a partir de un conjunto de datos (data set), sin ser programados de forma explícita evolucionado a partir del estudio de reconocimiento de patrones y la teoría del aprendizaje computacional en la inteligencia artificial. Las máquinas de aprendizaje exploran el estudio y construcción de algoritmos que se pueden aprender y a través de ellos hacer predicciones sobre los datos (Kearns 1988).

Su objetivo es general fuentes de clasificación lo suficientemente simples como para ser comprendidas fácilmente.

La limitación de este tipo de clasificación radica en su sistema de planteamiento: a mayor complejidad o dificultad del problema de clasificación mayor debe ser la cantidad de datos a proporcionar previamente para su entrenamiento. El aprendizaje automático se emplea en una amplia gama de tareas de computación, donde el diseño y la programación explícitos algoritmos es inviable; ejemplos de aplicaciones incluyen el filtrado de correo no deseado, reconocimiento óptico de caracteres.

Para la presente investigación se usó el aprendizaje automático a través del algoritmo Adaboost desbalanceado.

2.3.3.1 Características del Aprendizaje automático

Obregón (2016) manifiesta que las características, ventajas y desventajas del aprendizaje automático moderno son las siguientes:

- Buena exactitud en el marco de las predicciones.
- Métodos totalmente automáticos y de uso general.
- Métodos que se adapta a una gran cantidad de datos.
- Buena interacción entre teoría y práctica.

Desventajas importantes:

- Es necesario gran cantidad de datos para el proceso.
- Es imposible llegar a una precisión perfecta.
- Dificultad para expresar.

Principales ventajas:

- Permite el manejo de datos multidimensionales y multigrados en entornos dinámicos.
- Permite reducir el ciclo de tiempo y la utilización eficiente de recursos.
- No es necesario de un experto.
- Métodos baratos y flexibles.

Posibles errores de clasificación:

- Error de entrenamiento: es la proporción de casos de entrenamiento mal clasificados.
- Error de test: es la proporción de casos de prueba mal clasificados.

- Error generalizado: se refiere a la probabilidad de clasificar erróneamente una nueva muestra aleatoria.

2.3.3.2 Tipos de Aprendizaje automático

Según Sancho (2015) los algoritmos automáticos en función a su tipo de salida, y a como aborde los tratamientos, se puede agrupar de la siguiente manera:

- **Aprendizaje supervisado**

Se presenta con ejemplos de entradas y sus salidas deseadas, a cargo de un "maestro", y el objetivo es aprender una regla general que mapea las entradas a las salidas. Es tipo de clasificación es el que se enfocará en esta investigación.

- **Aprendizaje no supervisado**

En el proceso no hay etiquetas se dan al algoritmo de aprendizaje, dejándola a su propia estructura de encontrar en su entrada. Aprendizaje sin supervisión puede ser un objetivo en sí mismo (descubrir patrones ocultos en los datos) o un medio hacia un fin.

- **Aprendizaje semisupervisado**

Combina los dos algoritmos tratados anteriores, enfocándose en ejemplos clasificados y no clasificados.

- **Aprendizaje por refuerzo**

El algoritmo aprende inspeccionando el entorno que le rodea y en forma continua con el flujo de información bajo dos direcciones, realizando un proceso de ensayo-error, y fortaleciendo las acciones que adquieren una respuesta positiva en el entorno.

- **Transducción**

Similar al aprendizaje supervisado, pero su objetivo no es construir de forma explícita una función, sino únicamente tratar de predecir las categorías de los siguientes ejemplos basándose en los ejemplos de entrada, sus respectivas categorías y los ejemplos nuevos al sistema. Es decir, estaría más cerca del concepto de aprendizaje supervisado dinámico.

- **Aprendizaje multitarea**

Engloba todos aquellos métodos de aprendizaje que usan conocimiento previamente aprendido por el sistema de cara a enfrentarse a problemas parecidos a los ya vistos.

2.4 Algoritmos basados en el máquinas de aprendizaje

Hay múltiples tipos de algoritmos de clasificación basados en aprendizaje automático. A la hora de elegir uno u otro hay que tener en cuenta ciertas consideraciones y lineamientos.

2.4.1 Consideraciones al elegir un algoritmo

Una de las incertidumbres al momento de elegir un algoritmo es saber cuál de ellos elegir, algunos estudios indican que eso puede depender de la naturaleza de los datos, la calidad, el tamaño, o dependiendo del tiempo que se disponga, a continuación se presenta las siguientes consideraciones:

- **Precisión**

Según Rohrer (2016) a veces es difícil obtener la respuesta más precisa. Generalmente una aproximación ya es útil. Por ejemplo se puede reducir el tiempo de procesamiento considerablemente al hacer uso de métodos más aproximados. Una de las ventajas es que tiende evitar el sobreajuste.

- **Tiempo de entrenamiento**

Rohrer (2016) comenta que al momento de elegir un algoritmo es importante ver la variación de tiempo para entrenar el modelo. Generalmente este tiempo depende de la precisión. Si el tiempo es limitado la elección del algoritmo es determinante, más aún cuando el conjunto o la data son grande.

- **Linealidad**

Aunque no siempre sucede lo ideal es buscar el algoritmo más sencillo. Gran parte de los algoritmos de aprendizaje automático trabajan en función a una clasificación lineal, suponiendo que las clases tienen la opción de estar separados mediante una línea recta. Para este estudio como ejemplo se realizará la regresión logística.

- **Cantidad de parámetros**

Para Rohrer (2016) al momento de configurar un algoritmo los parámetros juegan un papel muy importante, estos pueden afectar el comportamiento del algoritmo, inmersos a la tolerancia de errores y cantidad de iteraciones. Generalmente cuando un algoritmo tiene parámetros con números grandes, estos requieren mayor número de pruebas, y errores para llegar a la mejor predicción y combinaciones.

En el caso de que un algoritmo tenga muchos parámetros una ventaja radica en que el algoritmo tiene mayor flexibilidad. Se dice que constantemente se puede lograr una precisión muy alta cuando se halle la combinación correcta de configuraciones de parámetros.

- **Cantidad de características**

Al momento de tratar cientos tipos de datos, la cantidad de características puede exceder a la cantidad de puntos de datos. Un ejemplo de ello es el caso de los datos textuales o genética. Un gran número de características puede dificultar el aprendizaje de algunos algoritmos provocando demora en el tiempo de entrenamiento.

Analizado todo esto, en esta presentación se buscará conseguir el equilibrio aunque habrá casos en los que predominarán unas u otras.

2.5 El modelo de regresión logística binario

La regresión logística formula un análisis utilizado para predecir una variable categórica en función a una o varias variables predictoras. El análisis de regresión logística está inmersa en el grupo que usa una función de enlace llamada logit (Hosmer y Lemeshow 2002).

En la regresión logística los datos son distribuidos binomialmente de la siguiente forma:

$$Y_i \sim B(p_i, n_i),\ para\ i = 1, ..., m$$

Siendo los números n_i ensayos de Bernoulli conocidos, y p_i probabilidades de éxito desconocidas. Estos logits de las probabilidades binomiales desconocidas forman el modelo general de regresión logística, modelada bajo una función lineal:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = x_i^T\beta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p = \eta$$

La regresión logística suele usarse para correlacionar la probabilidad de una variable cualitativa binaria (pudiendo categorizar los valores como "0" y "1") con un conjunto de variables escalares x_i. La probabilidad aproximada de pertenencia a cualquiera de las dos categorías en el suceso se aproxima a través de una función logística de la siguiente manera:

$$p_i = \text{logit}^{-1}(\eta) = \frac{e^{\eta}}{1+e^{\eta}} = \frac{1}{e^{-\eta}+1}$$

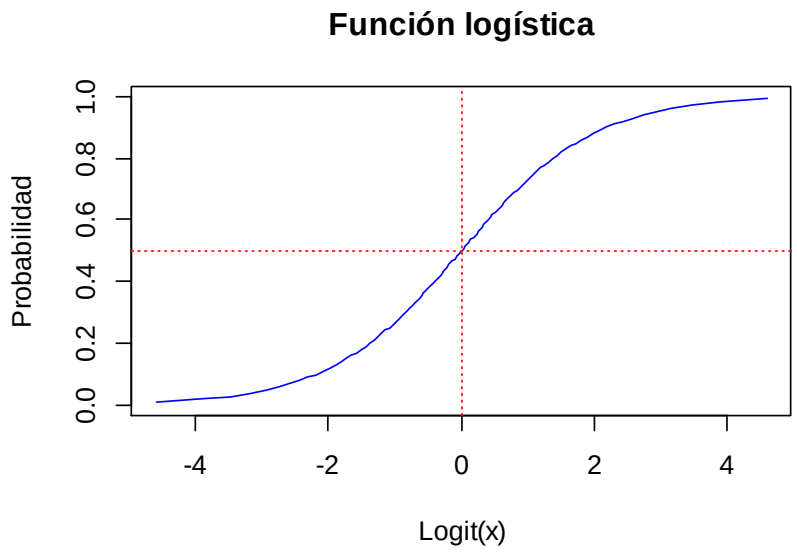


Figura 3: Función logística. Fuente: Green (2003)

El gráfico considera a p_i como una función de η_i llamada curva de respuesta o probabilidad de éxito, y tiene una forma simétrica centrada en 0.5

Esta función logística es usada para hallar la probabilidad de pertenencia de uno u otro grupo, sin embargo esta función es óptima cuando los datos son simétricos, es decir cuando la variable dependiente categórica tiene una cantidad equilibrada de “ceros” y “unos”, cosa que en muchas situaciones prácticas no se cumple. Tal es el caso de esta investigación sobre la fuga de clientes, donde hay una gran diferencia entre los que fugan y los que se mantienen con el servicio.

2.6 El modelo de regresión logística asimétrica

Cuando hay presencia de datos desbalanceados en la variable de respuesta los enlaces simétricos pueden ser inadecuados. Según Nagler, citado por Dávila et al. (2015), indica que el uso de la regresión logística sobre datos desbalanceados llevará a que un individuo cuya probabilidad de 0.5 de éxito sea más susceptible a variaciones en las variables regresoras, provocando una distribución sesgada. En esta investigación, las respuestas no son simétricas en torno a 0.5 por lo tanto se justifica el uso del enlace asimétrico.

Para subsanar el inconveniente de datos de respuesta asimétrica se han propuesto diferentes variantes, incluyendo la regresión logística Bayesiana relacionado al método de cadenas de Markov, modelos Power (Bazan et al. 2016) y simulación de Monte Carlo (Lunn et al. citado por Komori et al. 2015). Las otras posibles soluciones están relacionadas con tipos de muestreo para equilibrar la variable de respuesta.

Parte de esta investigación se desarrolló utilizando modelos que incorporan un parámetro para equilibrar el efecto de asimetría que existe en los datos.

Komori et al. (2015) proponen una metodología para los datos asimétricos en un modelo de regresión logística con covariables de efectos fijos y aleatorios. A continuación se detalla el procedimiento a trabajar con el modelo de efectos fijos:

Teniendo en cuenta el modelo de regresión logística clásica detallado a continuación

$$P_0(y=1|x,\eta)=\frac{\exp(\eta(x))}{1+\exp(\eta(x))} \tag{1}$$

donde $P_0(y=1|x,\eta)$ es una probabilidad condicional de y = 1 dado x .

De lo anterior los autores introducen un parámetro “k” adicional para proponer un modelo de regresión logística asimétrica (MRLA)

$$P_k(y=1|x,\eta)=\frac{\exp\{\eta(x)\}+\mathrm{k}}{1+\exp(\eta(x))+\mathrm{k}} \tag{2}$$

Teniendo en cuenta que: $k/(1+k)<p_k(t=1|x;\eta)<1$

El parámetro “k” corresponde a una extensión de una constante en un modelo logístico mixto de tres parámetros usado en psicometría descritos como:

$$P_{3PL}(y=1|\,x,z;\eta_2)=c+(1-c)\frac{\exp\{\eta_2(x,z)\}}{1+\exp\{\eta_2(x,z)\}} \tag{3}$$

Si se toma $\eta_2(x,z)=\eta(x,z)-\log(1+k)$ y $c=\kappa/\left(1+\kappa\right)$, esto coincide con el ALRM en la ecuación 2.

Se tiene en cuenta que si k = 0 (c = 0), se reduce a LRM en la ecuación 1

La función de máxima verosimilitud para el modelo está dado por:

$$L(\beta;k)=\prod_{i=1}^{n}P_k(y=1|\,\eta_i)^{y_i}\,P_k(y=0|\,n_i)^{1-y_i}$$

La función log-likelihood para el modelo está dado por:

$$lp(\beta,k)=\sum_{i=1}^{n}y_i\log\left[P_k(y=1|\,\eta_i)\right]+\sum_{i=1}^{n}(1-y_i)\log\left[1-P_k(y=1|\,\eta_i)\right]$$

Donde: $P_k(y=1|\,\eta)=\dfrac{\exp(\eta)+\mathrm{k}}{1+\exp(\eta)+\mathrm{k}}$

La obtención del valor óptimo de “k” se lleva a cabo durante la maximización de L (β; K). El parámetro “k” es esencial para la estimación de β en el MRLA. En la ecuación de estimación se también se deriva la probabilidad marginal de MRLA, dando una función de peso como w (eta) = exp (η (x) / {exp (η (x)) + k} esto hace que se ajuste el tamaño efectivo de la muestra y se equilibre los tamaños de muestra para las dos categorías de la variable de respuesta. Si η adquiere un gran valor negativo para las observaciones de y = 0, w (η) tiene un valor cercano a cero, de lo contrario se mantiene a uno.

2.7 El modelo Power Logit

Según Bazan et al (2017), cuando hay presencia de datos desbalanceados, los enlaces simétricos como logit o probit son inapropiados e inflexibles para ajustarse a la asimetría en la curva de respuesta y probablemente pueden llevar a una especificación errónea. Para

superar los problemas de desbalance propone un conjunto de enlaces de potencia sesgados entre los cuales se encuentra el Power logit el cual incluye un parámetro de potencia λ. Este modelo inicialmente fue introducido por Prentice en 1976 (Gaudar et al. 1993). El modelo viene dado por:

$$F_p(\eta) = P_\lambda(y = 1 \mid x, \eta) = \left[\frac{\exp(\eta)}{1 + \exp(\eta)} \right]^\lambda$$

Donde λ > 0, el modelo se reduce a una regresión logística clásica cuando λ=1

El valor de lambda se obtiene y es el que maximiza la función de verosimilitud dada por:

$$L(\beta; \lambda) = \prod_{i=1}^{n} P_\lambda(y = 1 \mid \eta_i)^{y_i} P_\lambda(y = 0 \mid n_i)^{1-y_i}$$

La función log-likelihood para el modelo está dado por:

$$lp(\beta, \lambda) = \sum_{i=1}^{n} y_i \log\left[G(\eta_i)^\lambda \right] + \sum_{i=1}^{n} (1 - y_i) \log\left[1 - G(\eta_i)^\lambda \right]$$

Donde: $p_i = F_p(\eta_i) = G(\eta_i)^\lambda, \quad i = 1, ..., n.$

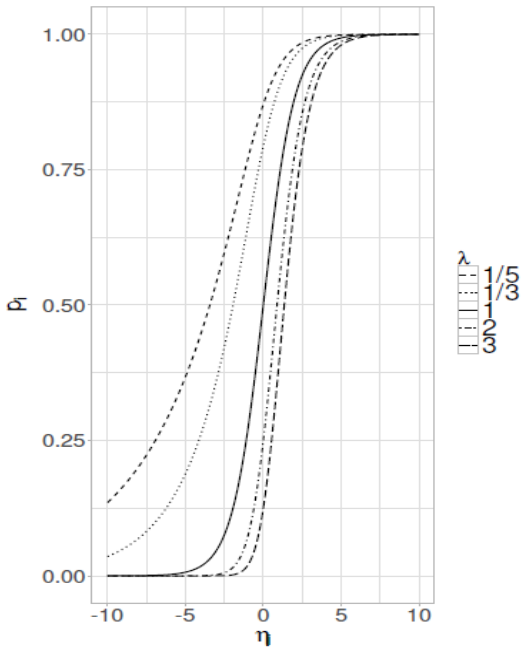


Figura 4: Función link power logit para diferentes valores de λ. Fuente: Bazán et al. (2017)

En la figura 4 se observa diferentes valores de lambda para modelos Power Logit, el valor de lambda óptimo permite ajustar el mejor modelo.

2.8 El algoritmo Boosting

Zhi-Hua (2012) define al Boosting como una máquina de aprendizaje a través de un meta-algoritmo para reducir principalmente el sesgo en el aprendizaje supervisado, consta de una familia de algoritmos de aprendizaje automático que convierten clasificadores débiles en clasificadores fuertes (clasificadores más eficientes). Un clasificador débil se correlaciona solo ligeramente con la verdadera clasificación). Por el contrario, un clasificador fuerte está arbitrariamente bien correlacionado con la verdadera clasificación.

La principal diferencia entre los algoritmos boosting es su método de puntuación en los datos de entrenamiento e hipótesis de ponderación. El Adaboost creado por Freund y Schapire (1996) es el más significativo históricamente, ya que era el primer algoritmo que podía adaptarse a los clasificadores débiles. Sin embargo, hay otros algoritmos más recientes como: LPBoost, TotalBoost, BrownBoost, xgboost, MAdaboost, LogitBoost, y otros.

Las características de este método son las siguientes:

- Rápido
- Facilidad para programar
- Flexibilidad para combinarse con otros algoritmos de aprendizaje
- No se sobre-ajusta fácilmente

2.9 Método de clasificación mediante el algoritmo Adaboost

Adaboost deriva de “AdaptativeBoosting”, es decir, Boosting adaptativo, es un algoritmo de aprendizaje automático presentado por Freund y Schapire en unos de sus principales artículos relacionados con este algoritmo.

Obregón (2016) en su tesis explica que existen variadas versiones tales como: Adaboost, Adaboost.M1, Adaboost.M2, etc., siendo estos tres populares porque fueron creadas originalmente por los mismos autores.

Al igual que todos los algoritmos boosting, consiste en entrenar en forma iterativa una serie de clasificadores débiles tal que cada nuevo clasificador de mayor atención a los datos mal clasificados en los clasificadores que se dieron anteriormente, para que de esa manera se

pueda combinar todos el conjunto de clasificadores débiles y obtener un clasificador cuyo rendimiento se similar a los fuertes.

Al inicio se da a todas las observaciones un mismo peso, y para lograr que cada nuevo clasificador de mayor importancia a los datos clasificados equivocadamente por los antecesores se utilizan funciones que ponderan la importancia en relación a cada dato en el proceso del entrenamiento del clasificador. De esta manera, los datos que se han clasificado correctamente pierden peso a favor de los que fueron clasificados erróneamente, intentando conseguir que los nuevos clasificadores se enfoquen en aquellos datos clasificados erróneamente.

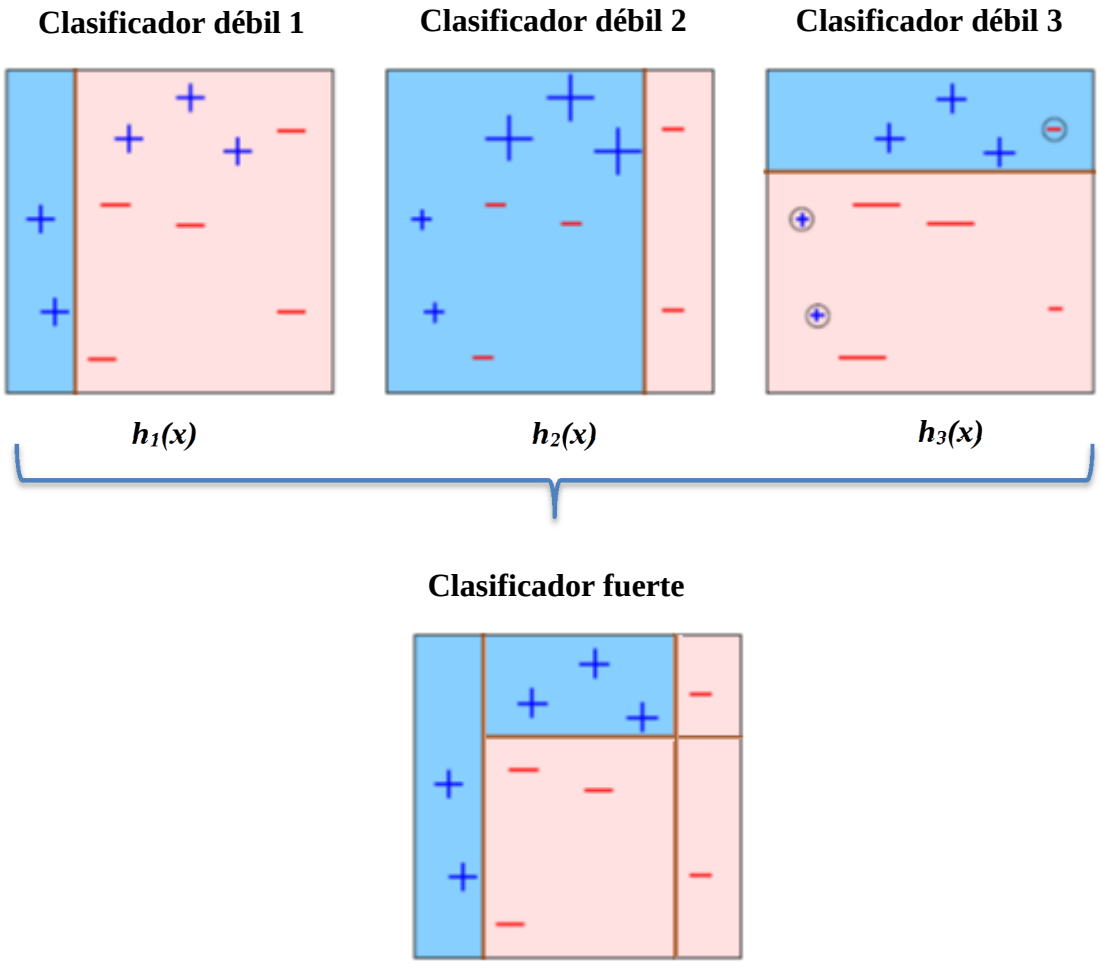


Figura 5: Formación de 3 clasificadores débiles con el método Adaboost. Fuente: Schapire (1996**)**

Finalmente el algoritmo combina todos los clasificadores débiles, obteniendo mayor peso en la votación final los que hayan evidenciado mejor rendimiento. De la combinación de todos

los clasificadores, la clasificación del grupo se proyectará hacia una clasificación correcta y lo más exacta posible de todos los datos.

2.9.1 Fundamentos del Algoritmo Adaboost

Este algoritmo es el inicial de la familia de algoritmos Adaboost. Conocido por excelencia generalmente para datos de dos categorías, el cual tuvo sus inicios en 1995.

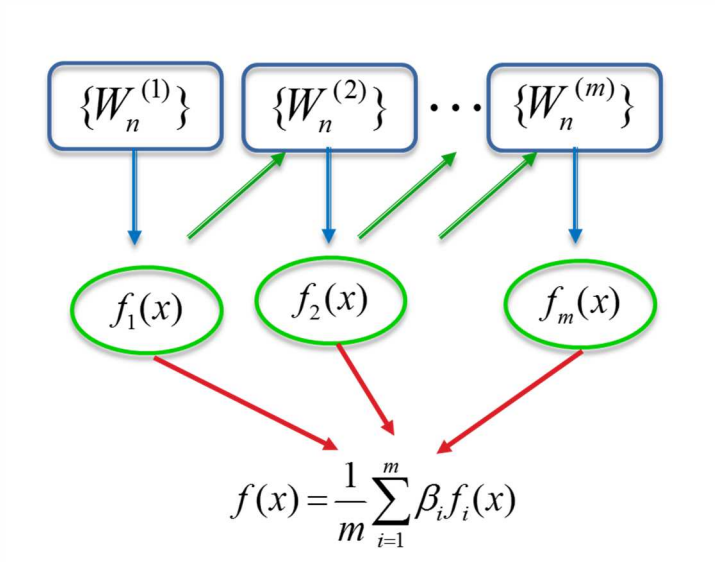


Figura 6: Mecanismo del Adaboost. Fuente: Chiu (2015)

En la figura 6 se aprecia el mecanismo del algoritmo Adaboost, cada clasificador débil asume un peso "W" mayor en diferentes iteraciones, al final en conjunto se forma el clasificador fuerte.

Sea : $(x_1, y_1), ..., (x_m, y_m)$ donde $x_i \in X$, $y_i \in Y = \{-1, +1\}$

Iniciar $D_1(i) = 1/m$.

Para $t = 1, ..., T$:

- Entrenar la distribución del clasificador débil D_t
- Obtener la hipótesis débil $h_t : X \to \{-1, +1\}$ con error

$$\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$$

- Escoger $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
- Actualizar:

$$D_{t+1}(i) = \frac{D_t(i)}{z_t} \times \begin{cases} e^{-\alpha_t} & \text{si } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{si } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) \exp(\alpha_t y_i h_t(x_i))}{z_t}$$

Donde z_t es un factor de normalización (Elegido para que D_{t+1}sea a distribución).

La hipótesis final será:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Figura 7: Forma del algoritmo Adaboost Original

La figura 7 muestra un enfoque general de la metodología del algoritmo Adaboost. A continuación, se detalla su funcionamiento en forma general:

1. Al inicio, se toma los datos que pertenecen a la muestra de entrenamiento, dando un peso inicial, cuando T=1 (T hace referencia al tiempo de interacción), todos los ejemplos tienen la misma probabilidad. En las iteraciones siguientes, es más probable seleccionar ejemplos más difíciles (aquellos que hacen fallar al clasificador).

2. Luego se realizan T iteraciones, donde para cada una se entrena un algoritmo de aprendizaje débil que conlleva una hipótesis con un error (ε). Finalmente los pesos son actualizados para cada ensayos según su clasificación y repitiendo esto en forma iterativa.

3. Al finalizar las T iteraciones se logra el clasificador final, cuyo error de entrenamiento se ha minimizado.

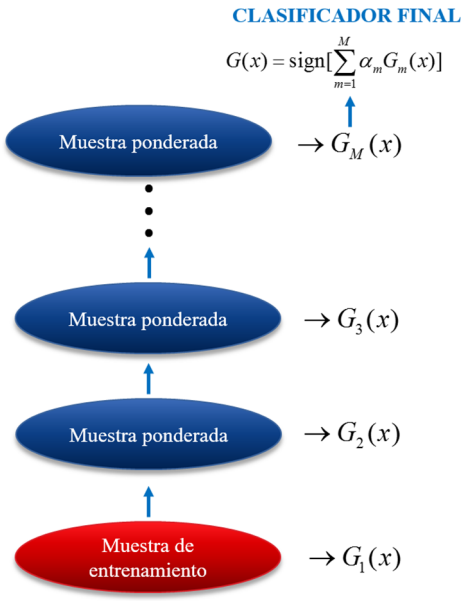


Figura 8: Proceso de clasificación del algoritmo Adaboost. Fuente: Llew Mason (2008).

La figura 8 muestra el esquema estructural de la forma de clasificación del algoritmo Adaboot.

2.10 Metodologías para el Adaboost con datos desbalanceados

Una de las dificultades que se presenta en la clasificación, es la frecuencia de los casos en los que una de las clases es mucho más frecuente que la otra, dando resultados y predicciones imprecisas. Tales situaciones se conocen como asimetría de los clasificadores (datos desbalanceados). Este problema se presenta con más frecuencia en problemas de clasificación binaria que en problemas de clasificación de múltiples niveles. Ante estas desventajas, diferentes investigadores han propuesto diferentes métodos alternativos de solución.

Según Landesa (2014) en lugar de buscar hipótesis que se ajusten lo mejor posible al total de los datos, se debe centrar la atención en clases que se consideran más valiosas o que sean menos frecuentes. Según esto, la relación entre Adaboost y los problemas frecuentes de asimetría tienen relevancia práctica especial, dado que Adaboost es el algoritmo de aprendizaje que utiliza el sistema propuesto por Viola y Jones (2004) para detección de objetos en imágenes lleva consigo en forma implícita problemas de carácter asimétrico.

En función a esto se hace necesario incorporar al algoritmo características o propiedades asimétricas, razón a esto conforme ha ido pasando el tiempo se ve en la literatura muchas variantes.

Para tal fin Landesa (2014) en su estudio detalla que han aparecido diferentes propuestas hacia un Adaboost asimétrico sin embargo estas son muy heterogéneas, presentan propiedades y diferencias muy difusas, y en muchos casos están basadas en modificaciones heurísticas del algoritmo inicial. Esto puede presentar una dificultad para el investigador, ya que se puede crear un conjunto de algoritmos carente de un marco general capaz de clasificar, analizar y discutir sus propiedades sobre una base común y objetiva.

2.10.1 Variantes asimétricas al Adaboost

El enfoque de las diferentes variantes para el Adaboost asimétrico en la literatura parte las diferentes alternativas en tres conjuntos, en función al tipo de mecanismo usado para contrarrestar la asimetría:

A posteriori: Es una modificación de un clasificador simétrico después de haber sido entrenado.

Heurístico: Utiliza manipulaciones directas en la actualización de pesos del algoritmo Adaboost (sienta esta una consecuencia de minimización inmerso en el Adaboost y no un punto de inicio).

Teórico: Enfocado en una derivación de la teoría.

El esquema quedaría resumido tal como se muestra en el cudro 1.

Cuadro 1: Mecanismo para contrarrestar la asimetría

Mecanismo	Algoritmos
A posteriori	Adaboost con Modificación del Umbral
Heurísticos	AdaCost, CSB0, CSB1 y CSB2 AdaC1, AdaC2 y AdaC3
Teóricos	Cost-Sensitive Adaboost

Fuente: Landesa (2014)

2.10.2 Variante asimétrica Asymboost

Chu y Yunqian (2012) enfocan su estudio para el Adaboost asimétrico, detallando el algoritmo Asymboost el cual fue un intento de Viola y Jones para resolver este problema, donde la idea es poner énfasis en los falsos negativos (clasificar verdaderos como positivos) más que falsos positivos (clasificar negativos como positivos). De esta manera tanto el falso positivo y el falso negativo obtienen la misma pérdida en el algoritmo inicial. Esta pérdida simétrica es reemplazada por una función de pérdida asimétrica (asumiendo que los falsos negativos son k veces más importantes que los falsos positivos):

$$A_{Loss(i)} = \begin{cases} \sqrt{K} \ if \ y_i = +1, \ y \ H(x_i) = -1 \\ \frac{1}{\sqrt{K}} \ f \ y_i = -1, \ y \ H(x_i) = +1 \\ 0 \qquad\qquad\qquad\qquad en \ otro \ caso \end{cases}$$

Esta nueva función de pérdida se incorpora en forma preponderada amortizando el coste asimétrico en cada iteración de Adaboost e incorporándose fácilmente al algoritmo.

2.11 Métodos para equilibrar datos desbalanceados (asimétricos)

Brownlee (2015) expone algunos métodos para hacer frente a los datos desequilibrados los cuales son conocidos como “Métodos de muestreo”, teniendo como objetivo equilibrar la distribución mediante algún mecanismo propio. Esta modificación se puede realizar mediante la alteración del tamaño de datos original de tal manera que se equilibre la proporción.

Kunal (2016) recomienda los siguientes métodos más utilizados para el tratamiento de conjuntos de datos asimétricos, los cuales se detallan a continuación:

- Sub muestreo
- Sobre muestreo
- Sobre muestreo en minorías sintéticas
- Costo de Aprendizaje Sensible

2.11.1 El submuetreo

Según Haibo y Yunqian (2013) el submuestreo aleatorio elimina los casos de clases mayoritaria de los datos de entrenamiento, lo cual reduce el número de observaciones de clase de la mayoría para equilibrar el conjunto de datos. Este método es uno de los mejores cuando la data en conjunto es grande ya que reduce el número de muestras de entrenamiento mejorando de tiempo de ejecución y almacenamiento.

El método de submuestreo se divide en tipo Aleatorio y tipo Informativo. Kunal (2016) detalla que en el submuestreo aleatorio elige observaciones de la clase que en su mayoría han sido eliminados hasta que la data en conjunto se equilibra. En cambio el submuestreo informativo realiza la selección especificado previamente algún criterio para eliminar las observaciones donde la clase mayoritaria.

Según Liu, Wu y Zhou (2006) dentro del submuestreo informativo, los algoritmos EasyEnsamble y BalanceCascade producen buenos resultados, siendo estos algoritmos sencillos y fáciles de entender combinando resultados y entrenando secuencialmente a los clasificadores.

2.11.2 El sobremuestreo

Este método se realiza con la clase minoritaria. Se comienza replicando las observaciones minoritarias para equilibrar la distribución de las clases al aumentar aleatoriamente las observaciones de la clase minoritarias. Esto se hace hasta que las observaciones de la clase mayoritaria y minoritaria se compensen.

Al igual que el submuestreo, este método también se puede dividir en dos tipos: sobremuestreo aleatorio e informativo.

Kunal (2016) describe que una posible ventaja de este método es que no dirige a ninguna pérdida de información. El inconveniente es que, el método no hace sino aumentar el sobremuestreo de casos replicadas en el conjunto de la data original, lo cual termina incluyendo múltiples observaciones de muchos tipos, produciéndose un posible sobreajuste.

2.11.3 Sobre muestreo de minorías sintéticas (SMOTE)

Según Haibo y Yunqian (2013) esta técnica de sobremuestreo de las minorías sintéticas (SMOTE) incrementa los datos mediante la inclusión de nuevos casos de clase minoritaria no replicados de los grupos de línea que unen a los cinco vecinos más cercanos de la clase menor.

Kunal (2016) menciona que en lugar de replicar y añadir observaciones de la clase minoritaria, erradica los desequilibrios mediante la inclusión de datos artificiales, convirtiéndose en un método muy potente.

El algoritmo SMOTE crea datos artificiales basados en las similitudes de las muestras minoritarias espacio con características similares. Dicho de otra manera, genera un conjunto de observaciones aleatorias cuya clase minoritaria puede cambiar el clasificador con sesgo hacia el aprendizaje de clase minoritaria.

La forma en la que funciona es la siguiente:

1. Inicia tomando la diferencia entre el vector que contiene la muestra en consideración y su vecino más cercano.
2. Multiplica esta diferencia por un número aleatorio entre 0 y 1
3. Añade la función del vector en estudio
4. Selecciona un punto aleatorio en la línea del segmento entre las dos características.

2.11.4 Aprendizaje Sensible al Costo (CSL)

Haibo y Yunqian (2013) informan que este método de aprendizaje sensible al costo se centra en el problema del aprendizaje desequilibrado mediante el uso de diferentes matrices de costos que describen los costos de clasificar erróneamente cualquier ejemplo de un dato en particular.

El método inspecciona el coste asociado a la clasificación con errores en las observaciones. A diferencia de los otros métodos no distribuye los datos en forma equilibrada. El método ataca el problema de aprendizaje desequilibrado mediante la utilización de matrices de

costos describiendo el costo de mala clasificación en un ámbito particular. Algunas investigaciones han demostrado que este método algunas veces supera a los métodos de muestreo. Dicho de otra manera, este método proporciona la alternativa más probable para la toma de muestras.

Kunal (2016) explica que la matriz de coste es parecida a la matriz de confusión. Donde el enfoque central se da a los falsos positivos y falsos negativos. No hay penalización por los costos asociados con los verdaderos positivos y los verdaderos negativos esto a medida que se identifican correctamente.

El objetivo de este método es elegir un clasificador con el coste total más bajo.

Cuadro 2: Matriz de costos

		Predicho	
		Positivo	Negativo
Actual	Positivo	0	C(FN)
	Negativo	C(FP)	0

Fuente: Kunal (2016)

Total Cost = C(FN)xFN + C(FP)xFP

Dónde:

- FN es el número de observaciones positivas predichas erróneamente.
- FP es el número de observaciones negativos predichos erróneamente.
- C(FN) y C(FP) representa los costos relacionados a los falsos negativos y falsos positivos, respectivamente. Tomando en consideración que, C(FN)> C(FP).

2.12 Métodos de evaluación en la clasificación

Es una de las partes más fundamentales y más difíciles en el ámbito de la clasificación, Obregón (2016) comenta que se pueden dar muchos casos posibles en los que un clasificador no tiene una validez adecuada: se puede tener un clasificador que se haya ajustado a la perfección a la muestra de entrenamiento (bajo error del clasificador en el entrenamiento) pero luego muestre un gran error a la hora de clasificar la muestra de test (gran error de clasificación).

Ante estas posibilidades se proporciona las herramientas necesarias para analizar la validez y rendimiento de los clasificadores.

Las herramientas elegidas para esta investigación son las siguientes:

2.12.1 Técnica de validación cruzada de k iteraciones

Según Muñoz (2015) consiste en dividir los datos en varios subconjuntos, posteriormente se selecciona uno de ellos como prueba y con los demás lo utiliza para entrenar el modelo de clasificación. El proceso se repite hasta evaluar cada subconjunto de datos.

En la validación cruzada de K iteraciones o K-fold cross-validation los datos de muestra se dividen en varios subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que se evalúa a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional. En la práctica, la elección del número de iteraciones depende de la medida del conjunto de datos. Lo más común es utilizar la validación cruzada de 10 iteraciones (10-fold cross-validation)

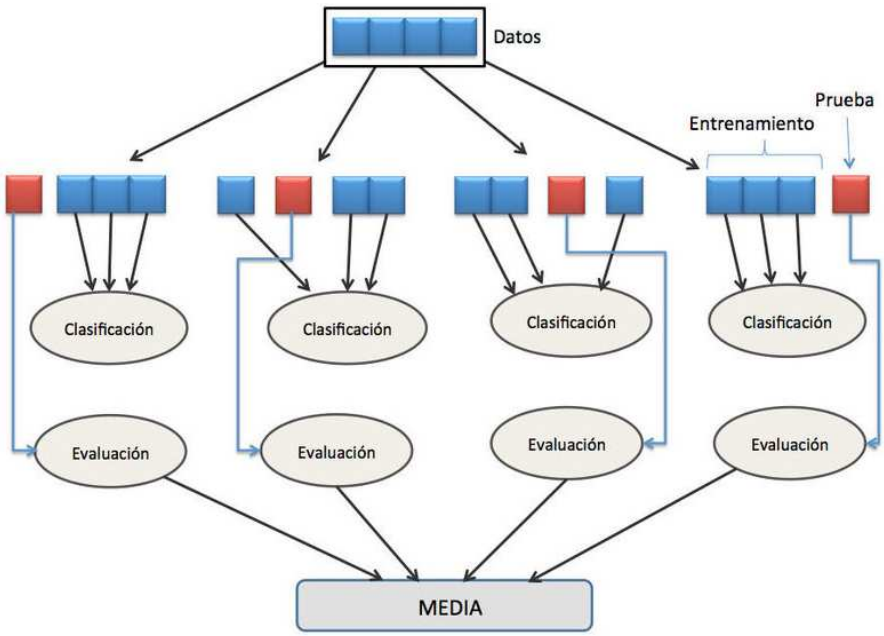


Figura 9: Validación cruzada de K iteraciones con K=4. Fuente: Lang

2.12.2 Medidas para evaluar la eficiencia de los modelos asimétricos

La tasa de error no es la métrica que se debe usar cuando se trabaja con un conjunto de datos desbalanceados. Diferentes estudios demostraron que esta medida es engañosa para datos desbalanceados. Hay indicadores que han sido diseñados para las clases desbalanceadas

Error de clasificación por re-sustitución

Según Obregón (2016) es el error del clasificador obtenido de la muestra de entrenamiento. Por sí sólo este error no es determinante, ya que para muestras posteriores presentará un error de clasificación más elevado, de magnitud desconocida. A priori es deseable un error lo más bajo posible. Sin embargo como se dijo al inicio esta métrica es engañosa.

Precisión: Es una medida de la exactitud de un clasificador , es decir es una medida de la corrección lograda en la predicción positiva, es decir, de las observaciones predecidas como positivas, cuando son realmente positivas.

Recall (sensibilidad): Es una medida de la integridad de un clasificador, mide las observaciones reales que se etiquetan (predice) correctamente, es decir, cuántas observaciones de clase positiva se etiquetan correctamente.

F-measure (medida F): Es un promedio ponderado entre la precisión y el Recall.

Curva ROC: Es posible comparar el desempeño de un clasificador con otro mediante el área bajo la curva ROC. Es la métrica de evaluación más utilizada. La curva ROC se forma trazando la tasa de verdaderos positivos (sensibilidad) y la tasa de falsos positivos (especificidad). El área bajo la curva (AUC) será la medida de mayor desempeño para evaluar los clasificadores. Esta medida es la más recomendada para datos desbalanceados, y será la principal medida para la toma de decisiones al momento de elegir los modelos. Sin embargo se recalca que a pesar que las curvas ROC han sido (y siguen siendo) ampliamente utilizados en la literatura científica, estas deberían complementarse con otras curvas tales como la curva PRC (curva de precisión y sensibilidad).

2.12.3 Matriz de confusión

Conocida como matriz de error, es una matriz que valora la capacidad predictiva de un modelo de clasificación.

Hair et al. (1999) describen que se construye tabulando de forma cruzada el miembro del grupo correcto con el miembro del grupo predicho, donde los números de la diagonal de la matriz representan clasificaciones correctas y los números fuera de la diagonal son clasificaciones incorrectas.

Esta matriz es una tabla de contingencia de dos variables la cual muestra los verdaderos positivos, los falsos positivos, verdaderos negativos y falsos negativos (FN) que se han obtenido en el proceso de clasificación para cada clase.

2.13 Análisis de valores perdidos

En el análisis de datos, es muy frecuente encontrar valores perdidos en las observaciones, estos datos faltantes pueden involucrar desde algunas de las variables de algunos de los sujetos seleccionados hasta la totalidad de los datos de algunos de los individuos seleccionados, siendo inevitable en diferentes estudios de investigación, independientemente de su diseño metodológico (Duran 2005).

Para subsanar el problema de valores perdidos, se han propuesto diferentes metodologías, entre las cuales se especifica la imputación de datos.

Viada et al. (2016) recopilaron distintos estudios de autores, formulando diferentes tipos de técnicas de imputación, mostradas a continuación:

2.13.1 Técnicas fundamentadas en información externa

Fundamentadas en variables relacionadas con una encuesta perteneciente a otras bases de datos o reglas previas.

a) **Métodos deductivos:** cuando los datos faltantes se deducen con cierto grado de certidumbre de otros registros completos del mismo caso, siguiendo algunas reglas específicas.
b) **Tablas Look-up:** se hace uso de una tabla con información relacionada, como base de data externa para imputar los datos faltantes.

2.13.2 Técnicas determinísticas

Al repetir la imputación en varias unidades bajo las mismas condiciones, producirá las mismas respuestas.

a) **Imputación de la media o moda:** si la variable es cuantitativa se reemplaza el o los datos con el promedio, mientras que para variables cualitativas se reemplaza con la moda.

b) **Imputación de media de clases:** las respuestas de cada variable son agrupadas en clases disjuntas con diferentes medias, y a cada registro faltante se le imputará con la media respectiva de su grupo.

c) **Imputación por regresión:** se ajusta un modelo lineal clásico que especifique a "y" como variable a imputar, para un conjunto X de variables auxiliares que se deben disponer.

d) **Imputación mediante el vecino más cercano:** se basa en la suposición de que los individuos cercanos en un mismo espacio tienen características similares. La aplicación requiere de una medida de distancia.

e) **Algoritmo EM (Expectation Maximization):** basada en la función de máxima verosimilitud, permite obtener estimaciones máximo verosímiles (MV) de los parámetros cuando hay datos incompletos con unas estructuras determinadas.

f) **Redes Neuronales:** son sistemas de información procesados, que reconocen patrones de los datos sin algún valor perdido para aplicarlo a la data a imputar. Estas redes son más usadas para variables cualitativas que cuantitativas, siendo más adecuadas cuando la distribución es no lineal.

2.13.3 Técnicas aleatorias o estocásticas

Son aquellas que cuando se repite el método de imputación bajo las mismas condiciones para una unidad, producen resultados diferentes.

a) **Imputación aleatoria de un caso seleccionado:** en cada caso en una celda faltante, se selecciona un donante aleatoriamente para ser reemplazado al dato faltante.

b) **Imputación secuencial Hot-Deck:** cada caso es procesado secuencialmente. Si el primer registro tiene un dato faltante, este es reemplazado por un valor inicial para imputar, pudiendo ser obtenido de información externa. Si el valor no está perdido, éste será al valor inicial y es usado para imputar el subsiguiente dato faltante.

c) **Imputación jerárquica Hot-Deck:** similar al método secuencial anterior. En esta se organizan dentro de clases haciendo uso de variables auxiliares en forma de una estructura jerárquica. Si el donante no es encontrado en un nivel de clasificación, las clases pueden ser colapsadas en grupos más anchos hasta que el donante sea encontrado.

d) **Imputación por regresión aleatoria:** primero se realiza un procedimiento de regresión, posteriormente un término residual es incluido para imputar los diferentes valores de "y".

III. MATERIALES Y MÉTODOS

3.1 Materiales

- Una computadora Toshiba Intel Corei5 de 64 bits.
- Una Impresora hp Laser Jet P1102w.
- Programa R versión 3.3.2, siendo de uso principal los paquetes Fastboot versión 4.1, lme4 versión 1.1-12, caret 6.0-78, ROSE versión 0.0-3, entre otros.

3.2 Descripción del caso

La presente investigación pretende comparar dos modelos de predicción para la fuga de clientes de una empresa del sector de telefonía móvil. Puesto que el número de clientes que se mantienen en el servicio tiene una gran ventaja numérica en relación a los que fugan, se utilizó los modelos de regresión logística asimétrica y el algoritmo Adaboost para datos desbalanceados. Como se trabajó con una gran cantidad de datos, se espera que el algoritmo Adaboost sea el que tenga mayor rendimiento y desempeño, de esta manera se podrá adaptar el modelo a los datos y proponer una posible implementación futura a la empresa.

3.3 Población

La presente investigación se trabajó con información de una empresa de telefonía móvil del área de postpago con un conjunto de datos para el entrenamiento de los modelos de los registros de los últimos 6 meses del año 2017.

3.4 Identificación de las variables

En base a la experiencia de empresas y referencias de estudios similares se tomó en consideración las siguientes variables en la aplicación de ambas técnicas:

Variable independiente

- **Y=Churn:** Abandono del cliente, el cual puede ser por voluntad propia, o porque la empresa decidió cancelarle el contrato, se tomará como medida dicotómica (1: Fugado, 2: no fugado)

Variables independientes

Mediante la reducción de variables se utilizó las siguientes variables independientes:

- **Minutos de uso (MOU):** (Minutes Of Use), es el ratio de tiempo hablado mensual y o datos.
- **Días de deuda:** Número de días promedio por mes que adeuda el cliente en los últimos 6 meses.
- **Reclamos:** Número de reclamos mensual en los últimos 6 meses.
- **Tipo de Reclamos:** Tipo de reclamo más frecuente en los últimos 6 meses.
- **Sexo:** Género del cliente
- **Edad:** Edad en años.
- **Procedencia:** Lugar de procedencia (1: Lima, 2: Provincia)
- **Rol:** Es el rol del cliente, el cual puede ser líder, seguidor o marginal (se calcula en base a mensajes, llamadas, transferencias entre los usuarios de la misma empresa)
- **Comunidad:** Variable representada por niveles según el porcentaje de relación del cliente con todos sus contactos, calculado en función a las redes, seguidores, llamadas, etc., con clientes de la empresa (internos), ejemplo: grado 1 (0-25% interna), grado 2 (26-50% interna), grado 3 (50-75% interna), grado 4 (76-100% interna).
- **Plan renuncia:** Es la cantidad de planes renunciados del usuario durante su periodo de vida.
- **Antigüedad:** Es el tiempo en días desde que el cliente se afilió a la empresa
- **Canal:** Es el medio mediante el cual se vendió el plan al cliente (1: centro de la empresa, 2: vendedor individual, 3: Ejecutivo de la compañía, 4: no identificado)
- **Tipo cliente:** Es la valoración que la empresa asigna al cliente, (1: bajo, 2: medio bajo, 3: medio, 4: medio alto , 5: alto)

- **Número mensajes:** El número de mensajes mensual
- **Llamadas:** Es el número de llamadas mensual
- **Kilovatios:** Es la cantidad de Kilovatios de uso mensual (uso de datos, correo, internet, etc)
- **Ingresos (ARPU):** Equivale al gasto promedio que un usuario tiene con el servicio de telefonía móvil.
- **Nota de pago:** Es la nota del cliente mensual en función a su comportamiento de pago y el tiempo que demora un cliente en pagar sus cuentas.

3.4.1 Muestra

La muestra total constó de 80300 registros los cuales se dividieron en muestra de entrenamiento y muestra de validación para los modelos propuestos (regresión logística asimétrica y Adaboost desbalanceado), distribuyéndose de la siguiente manera:

- **Muestra de entrenamiento:** se utilizó el 70 por ciento equivalente a 56210 registros.
- **Muestra de validación:** se utilizó el 30 por ciento equivalente a 24090 registros.

La información de estas 80300 observaciones equivalen netamente a usuarios naturales, descartando a las instituciones o empresas afiliadas al servicio.

3.3 Metodología de investigación

3.3.1 Tipo de investigación

El tipo de investigación es de carácter explicativo predictivo. En los dos modelos propuestos de comparación se pretende explicar la importancia de las variables y predecir con la mínima tasa de error si un cliente fugará o no.

3.3.2 Diseño de la investigación

El diseño de la investigación fue no experimental de tipo longitudinal, debido a que se trabajó con los resultados obtenidos en un periodo de 6 meses. De estos se recopilaron 18 variables independientes de naturaleza cuantitativa discreta y continua, y cualitativa nominal y ordinal.

Las variables se utilizaron en el modelo de regresión logística asimétrica y el algoritmo Adaboost desbalanceado.

3.3.3 Formulación de la hipótesis

La hipótesis principal del presente trabajo de investigación es la siguiente:

El algoritmo Adaboost desbalanceado tiene mayor precisión que la regresión logística asimétrica en la predicción de fuga de clientes en la empresa de telefonía móvil mediante los indicadores de mejor desempeño y rendimiento.

3.4 Metodología aplicada

Los pasos que se realizaron para llevar a cabo este trabajo se detallan a continuación:

1. Análisis exploratorio de las variables a considerar.
2. Limpieza y análisis de datos perdidos y atípicos.
3. Metodologías de muestreo para equilibrar la respuesta desbalanceada.
4. Estimación de indicadores de predicción con el método de regresión logística asimétrica.
5. Estimación de indicadores de predicción con el algoritmo Adaboost desbalanceado.
6. Comparación de resultados obtenidos con el método de regresión logística asimétrica y el algoritmo Ababoost desbalanceado.
7. Selección del mejor modelo.

IV. RESULTADOS Y DISCUSIÓN

4.1 ANÁLISIS EXPLORATORIO DE DATOS

Para el procedimiento se utilizó la base de datos del centro de cómputo de la empresa de telefonía móvil. El primer proceso consistió en estructurar las bases de datos en una sola matriz, resumiendo y promediando los reportes de los últimos 6 meses, luego se pasó a la codificación de variables categóricas (tipo de reclamo, procedencia, etc), posteriormente se realizó la limpieza de datos, la cual consistió en identificar los valores perdidos, eliminando las variables o registros donde había exceso de valores perdidos y también en pocos casos se imputó algunos valores. Para evitar la influencia de valores atípicos se normalizó algunas variables. Finalmente, la base de datos quedó con 80300 registros con una variable dependientes categórica dicotómica (Fuga) y 18 variables independientes. Cabe recalcar que en la base de datos solo se consideraron a clientes personales del servicio de post pago y no a instituciones afiliadas a la empresa.

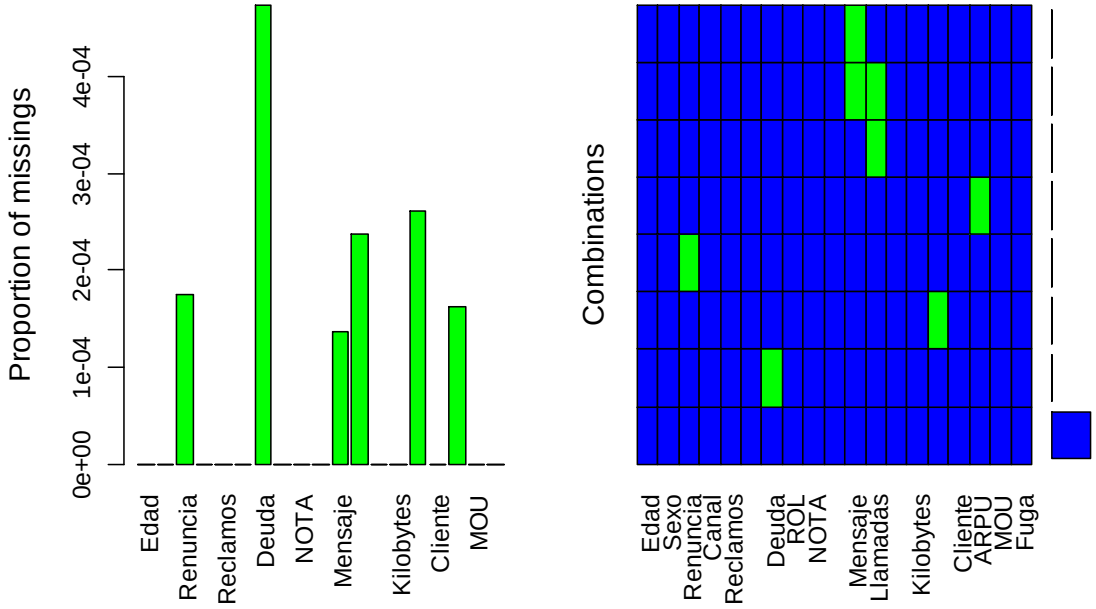


Figura 10: Identificación de valores perdidos

En la figura 10 se observa la presencia de valores perdidos en algunas variables, sin embargo el porcentaje no es grande por lo que no fue necesario pasar a eliminarlas. Las demás variables no presentan valores perdidos. Existe un porcentaje muy pequeño donde la variable Mensaje y Llamadas están perdidas en forma conjunta. Cabe recalcar que la empresa recopila casi toda la información de sus clientes por lo que los valores perdidos no son tan frecuentes.

Cuadro 3: Distribución de valores perdidos según variables

Variable	Conteo
Renuncia	14
Deuda	38
Mensaje	11
Llamadas	19
Antigüedad	21
ARPU	13

Fuente: Elaboración propia

En la Cuadro 3 se muestra el total de variables que contienen valores perdidos, estos en conjunto representa el 0.144% del total. Como siguiente procedimiento se pasó a identificar

si hay valores perdidos en forma multivariantes para posteriormente pasar a imputar los datos.

Cuadro 4: Análisis de combinación de valores perdidos multivariantes

Missings in combinations of variables:			
	Combinations	Count	Percent
0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0		80191	99.86425903
0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1:0:0		13	0.01618929
0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1:0:0:0:0		21	0.02615193
0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1:0:0:0:0:0:0		12	0.01494396
0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1:0:0:0:0:0:0:0		4	0.00498132
0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:**1:1**:0:0:0:0:0:0:0		7	0.00871731
0:0:0:0:0:0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0		38	0.04732254
0:0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0		14	0.01743462

Fuente: Elaboración propia

En la Cuadro 4 se observa la combinación de valores perdidos para las diferentes variables, hay una combinación donde hay dos variables que contienen valores perdidos en forma conjunta (Mensaje y llamadas). Por lo que se pasó a analizar en forma gráfica para ver si se debe imputar en forma bivariada.

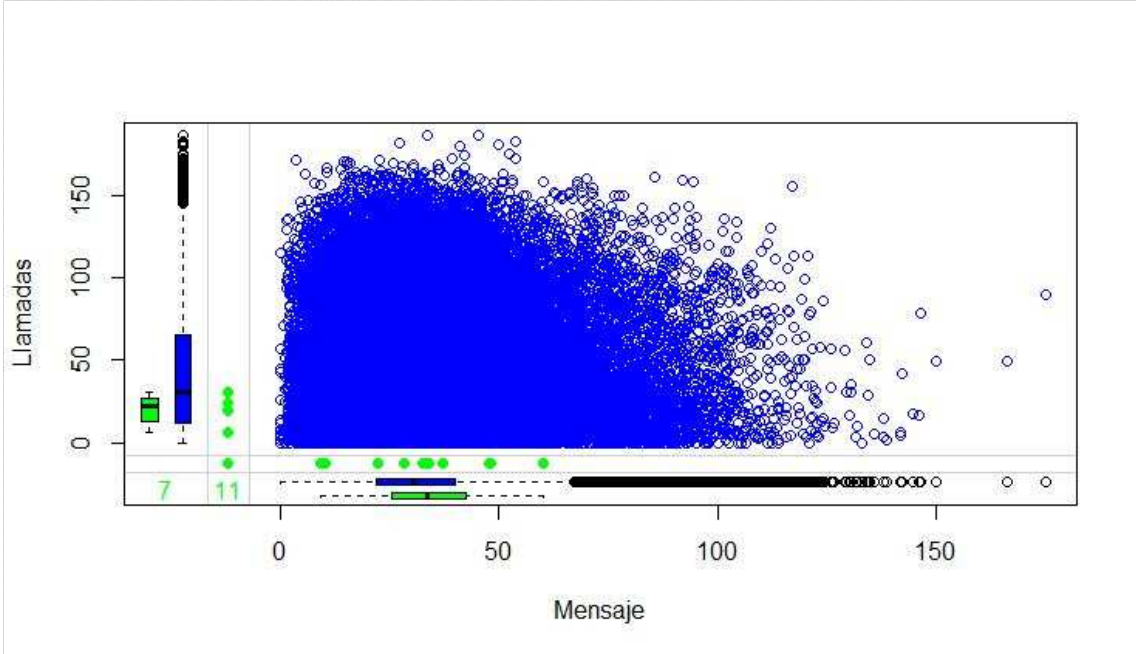


Figura 11: Valores perdidos en forma conjunta para las variables Mensaje y llamadas

La figura 11 muestra puntos que indican si debe o no imputar los datos en forma conjunta.

El diagrama de caja verde a la izquierda muestra la distribución de la variable “Llamadas” con ausencia de la variable “Mensajes”, mientras que el gráfico de caja azul muestra la distribución de los puntos de datos restantes, de la misma manera para el cuadro de la variable “Mensaje” se traza en la parte inferior del gráfico.

Si la suposición de los datos de que son completamente al azar es correcta, entonces se espera que los diagramas de caja verde y azul sean muy similares. En este caso los gráficos están casi superpuestos por lo que se asume que los 7 datos perdidos en forma conjunta se deben al azar.

Después del análisis de valores perdidos se pasó a imputar los pocos datos perdidos, de tal manera que si la variable es de naturaleza cuantitativa se imputó con el promedio y si era cualitativa con la moda.

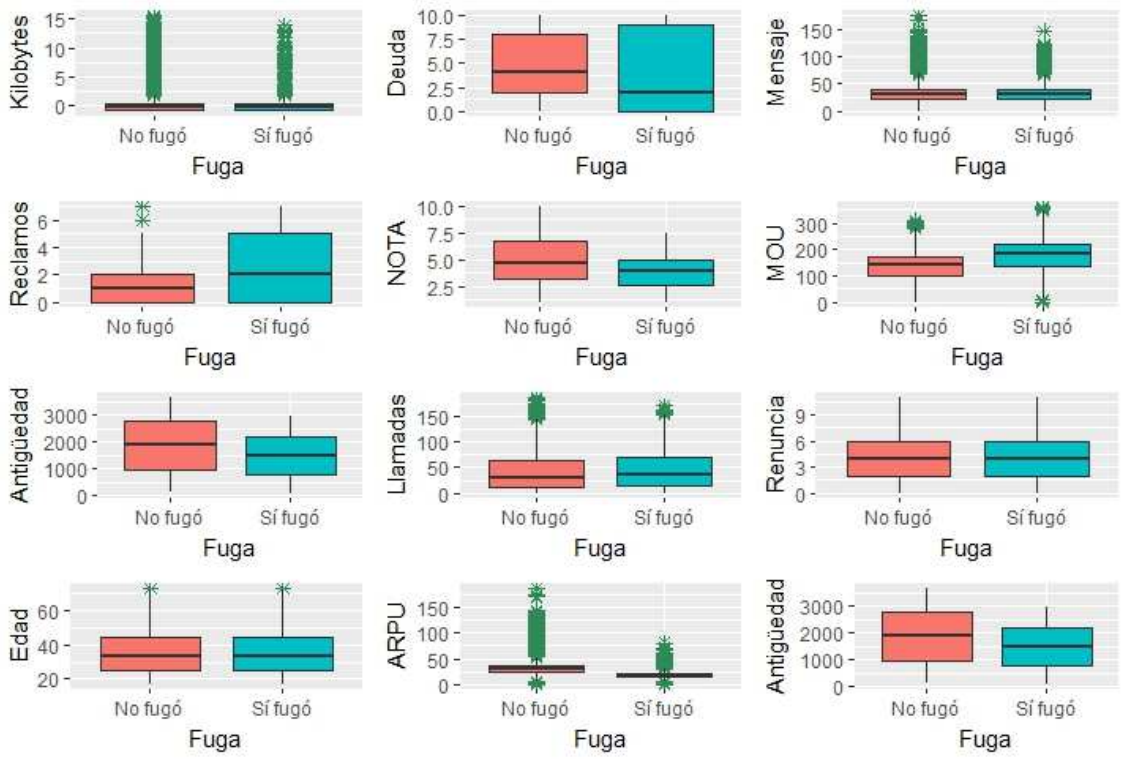


Figura 12: Diagrama de cajas para las variables cuantitativas

En la figura 12 se observa a través de diagramas de cajas que hay algunas variables que presentan gran cantidad de valores atípicos, principalmente en las variable Kilobytes, Mensaje y ARPU, con lo cual, para evitar su influencia se realizó una trasformación por medio de la normalización Z-score.

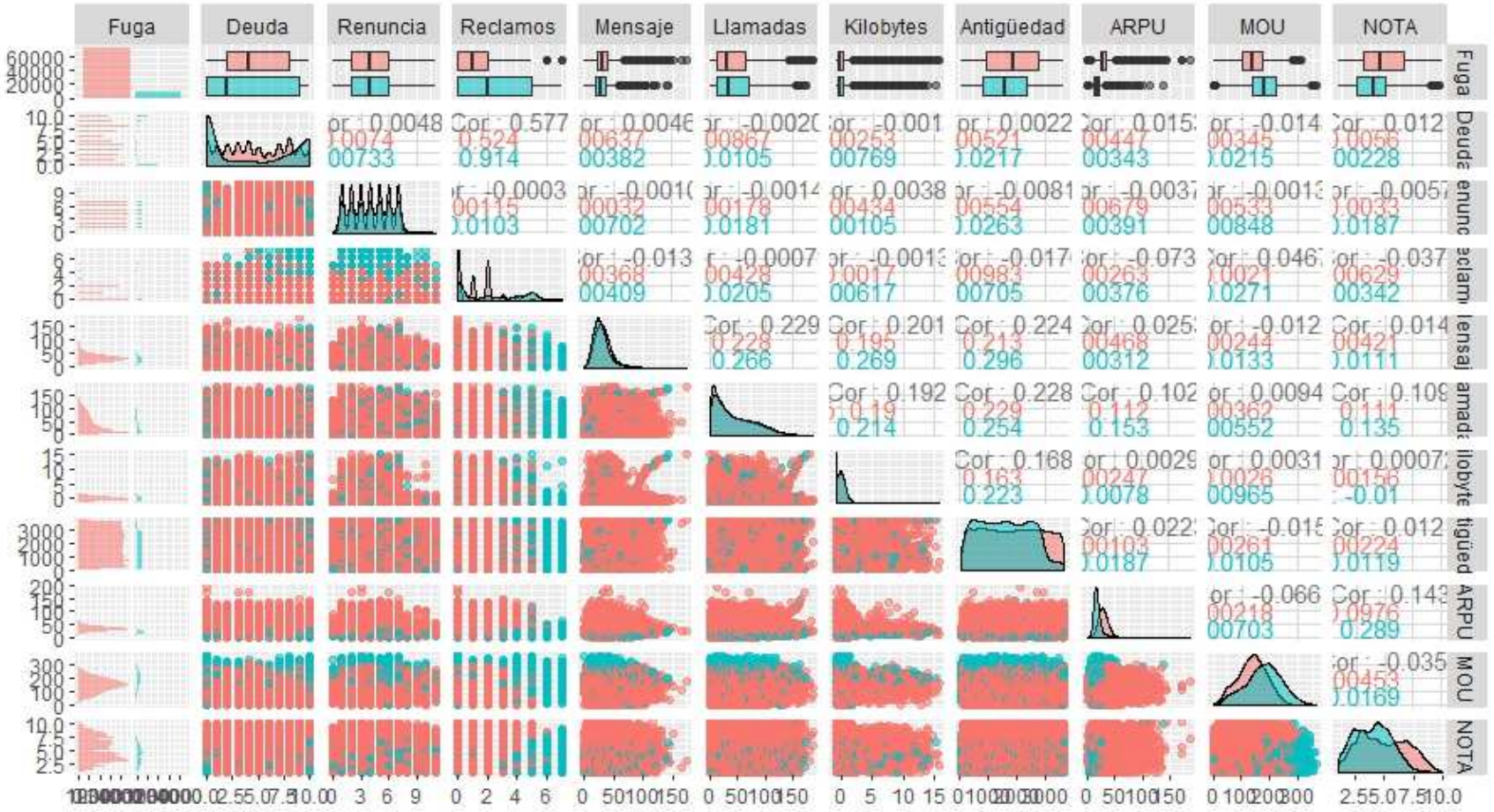


Figura 13: Matriz de correlaciones para las variables cuantitativas

En la figura 13 se observa las correlaciones entre las variables, los puntos rojos son la categoría de “No fuga” y los celestes la categoría “Sí fuga”, la variables “Reclamos” tiene una correlación media con la variable “Deuda”, en las demás variables las correlaciones son muy bajas, esto es bueno puesto que un requisita de la regresión logística es que no haya correlación entre las variables independientes.

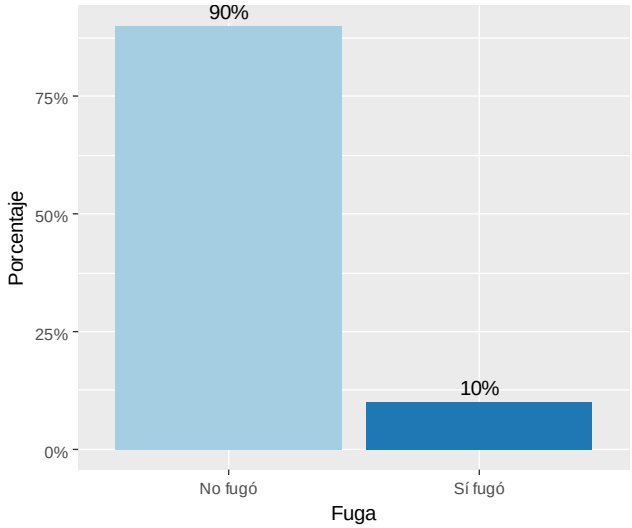


Figura 14: Distribución de la variable de respuesta “Fuga”

Cuadro 5: Distribución de la variable “Fuga” según el tipo de usuario

		Frecuencia	Porcentaje
Usuario	No fugó	72249	89.97
	Sí fugó	8051	10.03
	Total	80300	100

Fuente: Elaboración propia

En el cuadro 5 y la figura 14 se puede apreciar la distribución según el tipo de usuario. Se puede observar una marcada diferencia, presencia de usuarios que fugaron es muy poca (10.03%), realizar un modelo con estos datos desbalanceados podría conllevar un alto grado de sobreajuste, afectando a los indicadores de precisión, bajo este enfoque que el análisis, interpretaciones, medidas de desempeño y formación de modelos deben ir enfocados en técnicas para subsanar el problema de desbalance.

Antes de realizar el proceso de selección y formación de los modelos es importante realizar un análisis descriptivo para ver el comportamiento de los clientes y algunos patrones que puedan dar evidencias sobre la decisión de un cliente de fugar o no.

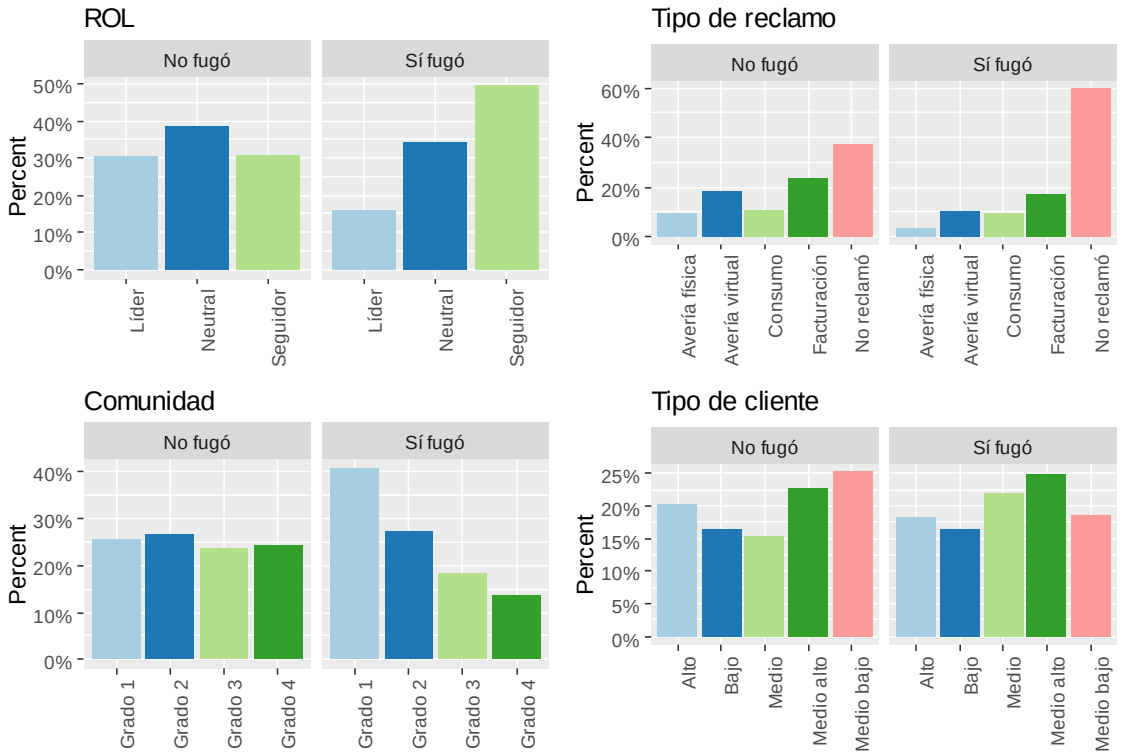


Figura 15: Gráficas descriptivas en variables cualitativas

En la figura 15, se aprecia la distribución de los usuarios que fugaron y los que no. En relación al tipo de reclamo, ambos grupos de usuarios generalmente no reclaman, sin embargo, para el caso de los reclamos más frecuentes, los que fugaron en su mayoría reclamaron por motivos de facturación, seguida de averías virtuales.

En el caso de la variable rol del usuario, los usuarios que fugaron en su mayoría son seguidores, eso quiere decir que tienen mayor comunicación, siguen a otros usuarios, los cuales pueden ser de la propia empresa o de otro servidor.

En cuanto a la variable comunidad, se refleja que los clientes que fugaron en su mayoría realizan contacto (redes, seguidores, llamadas, etc) de grado 1 (0-25% interna) y grado 2 (26-50% interna), es decir realizan llamadas no a usuarios internos (de la propia empresa) sino a usuarios de otras empresas o compañías, lo que se conoce como offnet.

En el caso de tipo de cliente, la empresa define a cada uno según su potencial de consumo. Para el caso de los usuarios que fugaron, no se encontró una diferencia marcada entre los tipos de clientes, en cambio para los que no fugan y mantienen la fidelidad en la empresa estos se caracterizan por ser en su mayoría clientes medio alto y medio bajo.

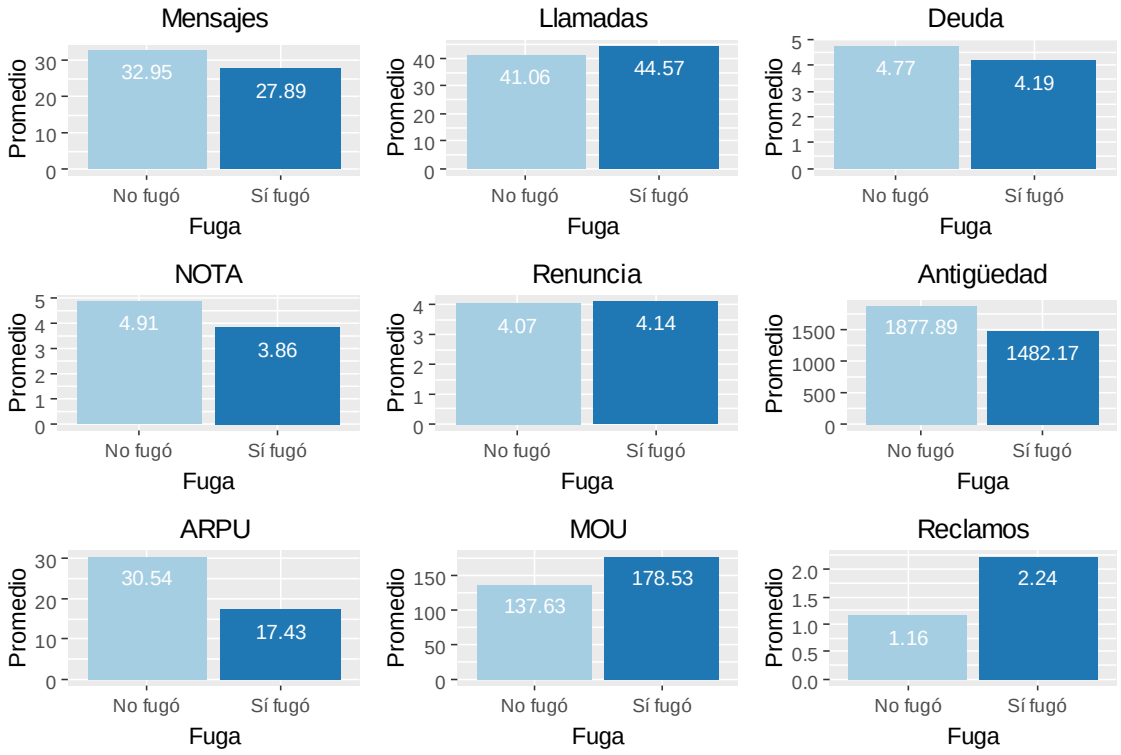


Figura 16: Gráficas descriptivas en variables cuantitativas

La figura 16 distribuye a cada variable cuantitativa según la condición del cliente (fuga o no). Hay algunas variables que no muestran un patrón marcado para diferenciar a los que fugan, sin embargo en otra como la variable “Reclamos” se ve claramente que los que fugaron tuvieron un mayor número promedio de reclamos, es aquí donde la empresa puede enfocarse para mantener a sus clientes.

Otra de las variables relevantes para entender la fuga es el ARPU, es decir gasto promedio que un usuario tiene con el servicio de telefonía móvil es bajo para los clientes que fugan, lo que manifiesta que un cliente que gasta poco en el servicio es probable que fugue. Otra variable relevante es el MOU, manifestando que los clientes que fugaron tienen un ratio de tiempo hablado mensual menor en relación a los que no fugan.

Los modelos creados más adelante dan un enfoque general sobre la importancia de cada variable y su grado de significancia para entender al cliente que fuga y al que se mantiene fiel en la empresa.

4.2 SELECCIÓN DEL TAMAÑO DE MUESTRA PARA LOS DATOS DE ENTRENAMIENTO Y DE PRUEBA (HOLDOUT METHOD)

Cuadro 6: Distribución del tamaño de entrenamiento y de prueba

Datos	No Fugó	Sí Fugó	Total	Porcentaje
Datos prueba	50554	5656	56210	70%
Datos test	21695	2395	24090	30%

Fuente: Elaboración propia

Para evitar el sobre ajuste en los modelos generados (que el modelo se ajuste muy bien a los datos pero no es útil en el ajuste de otros) como se cuenta con suficiente data se dividió la muestra en dos submuestras. El cuadro 6 muestra la distribución de la data de prueba y de entrenamiento, es decir con el 70% de los datos se construirán los modelos y con el 30% restante se evaluaran.

Se tuvo en cuenta que como el método de retención no es preciso al 100% debido a la variación de los resultados se complementó con la validación cruzada la cual se realizó para cada modelo.

4.3 MÉTODOS DE MUESTREO PARA EQUILIBRAR LOS DATOS DESBALANCEADOS

Para hacer frente a los datos desbalanceados se determinó algunas técnicas para remediar el problema. Esto se realizó mediante diferentes métodos de muestreo implementados en el paquete Caret del software R. Los diferentes métodos de muestreo buscan equilibrar las categorías de la variable de respuesta categórica “Fuga”, aclarando que esto se realizó no en los datos totales sino solo en la base de entrenamiento.

Cuadro 7: Métodos de muestreo y procedimientos para equilibrar los datos de prueba para el modelo de regresión logística y el algoritmo Adaboost

Métodos de muestreo	No Fugó	Sí Fugó	Total	Procedimiento
Sin muestreo	50554	5656	56210	Datos originales
Sobremuestreo (down)	50554	50554	101108	Iguala la categoría alta
Submuestreo (up)	5656	5656	11312	Iguala la categoría baja
Smote	22624	16968	39592	Muestra de minoría sintética

Fuente: Elaboración propia

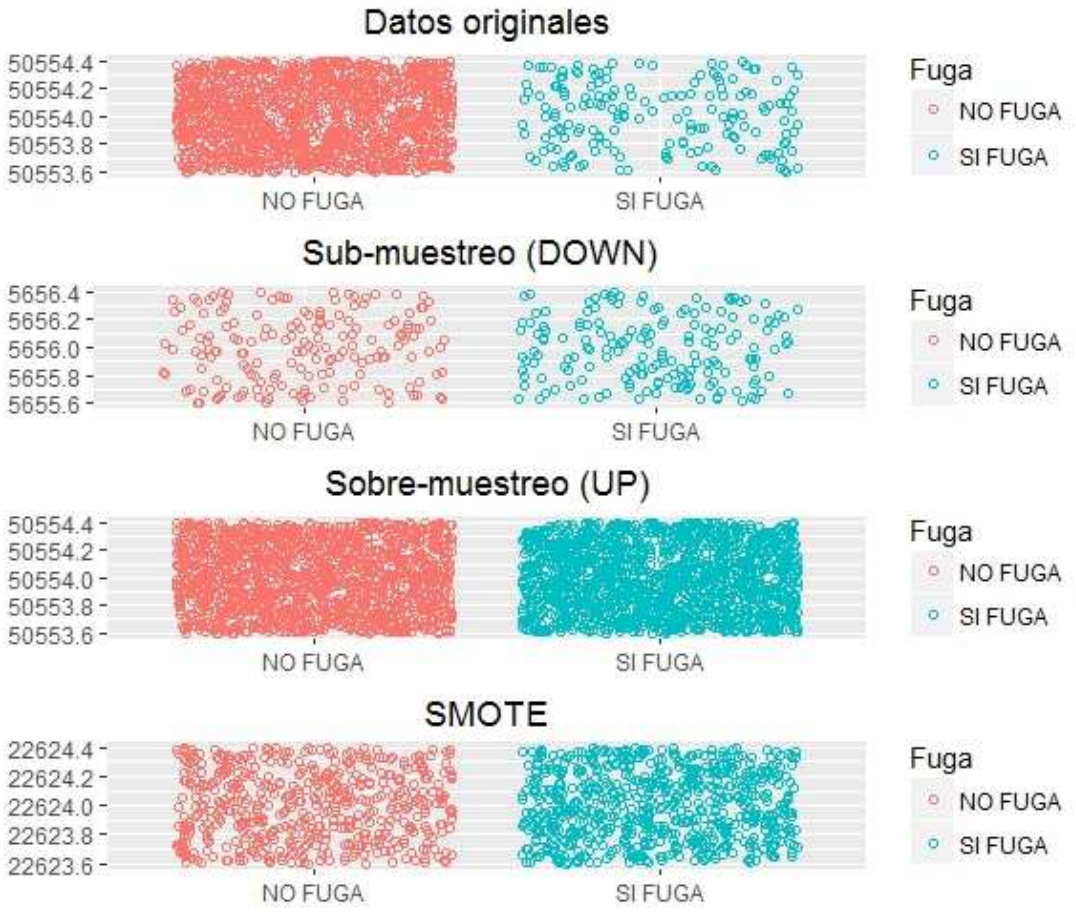


Figura 17: Distribución de la variable de respuesta “Fuga” según tipo de muestreo

El cuadro 7 muestra las diferentes opciones para trabajar los modelos especificados, si se trabaja con los datos originales de la data de entrenamiento (sin muestreo), habría un gran desequilibrio ya que los que fugaron solo representan el 10% aproximadamente del total.

Para tratar tal desbalance se comenzó equilibrando los datos mediante la técnica del sobre-muestreo, el cual aumentó la base de datos de entrenamiento hasta igualar o equilibrar la categoría más baja con la categoría más alta, es decir seleccionó registros de los que fugaron hasta igualar la cantidad de la categoría que no fugaron, con ello la base de datos se incrementó a 101108 registros.

El siguiente método que se realizó fue el sub-muestreo, el cual se realiza sin reemplazo, reduciendo los datos de la categoría más alta hasta igualar el tamaño de la categoría más baja (los que fugaron). En este caso la categoría menor cuenta con 5656 registros y al igualar ambas categorías, la base de entrenamiento se ha reducido a 11312 registros.

En los casos anteriores, a pesar de que el conjunto de datos está equilibrado, puede haber inconvenientes al haber perdido información significativa de la muestra, el sobre-muestreo ocasiona una cantidad de observaciones repetidas mientras que el sub-muestreo se priva de información importante sobre los datos originales.

Por último se trabajó con el método SMOTE el cual es un método hibrido de sobre-muestreo la cual utiliza muestras sintéticos de la clase minoritaria, equilibrando las clases a 22624 y 16968 respectivamente.

La figura 17, detalla visualmente la distribución de los datos según el tipo de muestreo, mostrando desequilibrio en los datos originales, equilibrio con el sub-muestreo pero con reducción de datos a la categoría minoritaria, también equilibrio con el sobre-muestreo aumentando los datos a la categoría mayoritaria, por último muestra el método SMOTE, en la cual los tamaños para la variable categórica "Fuga" es casi la misma.

4.4 FORMULACIÓN DE MODELOS MEDIANTE MÉTODOS DE MUESTREO PARA LA REGRESIÓN LOGÍSTICA Y EL ALGORITMO ADABOOST

Cuadro 8: Matrices de confusión comparativa para los modelos de regresión logística mediante métodos de muestreo

Datos originales

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	497	322
	No fugo	1898	21373

Sub-muestreo (DOWN)

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	1966	5018
	No fugo	429	16677

Sobre-muestreo (UP)

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	1969	5039
	No fugo	426	16656

Smote

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	1776	3876
	No fugo	619	17819

Fuente: Elaboración propia

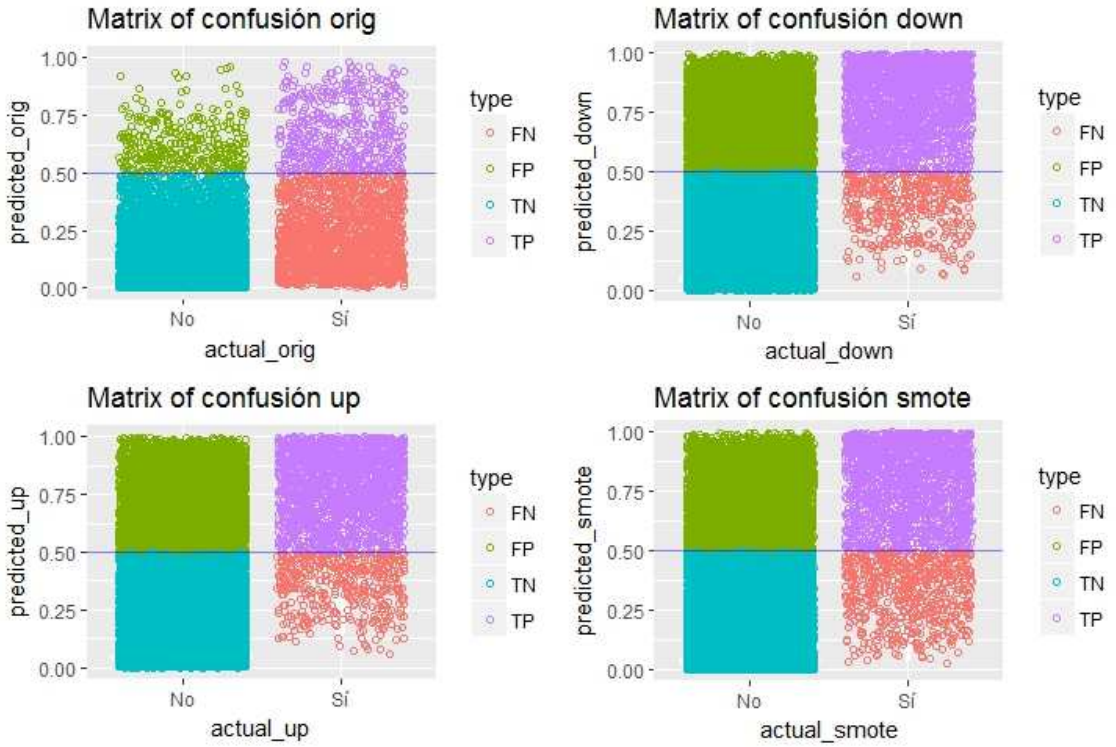


Figura 18: Gráficas de matrices de confusión para los modelos de regresión logística mediante los métodos de muestreo

En la cuadro 8 se muestra las diferentes matrices de confusión para los modelos de regresión logística mediante los métodos de muestreo, en cada método de muestreo la cantidad de verdaderos positivos aumentaron en comparación con los datos originales, también disminuyeron los falsos negativos, sin embargo los falsos positivos aumentaron, la figura 18 muestra la dispersión de cada valor clasificado, el color morado muestra los verdaderos positivos, corroborando que con los métodos de muestreo los Verdaderos positivos aumentan, para tener una idea más clara, se pasó a analizar las métricas recomendadas para datos desbalanceados.

Cuadro 9: Medidas de desempeño para los diferentes métodos de muestreo

Medidas	**Sin ajuste**	**Sub-muestreo**	**Sobre-muestreo**	**SMOTE**
Exactitud (Accuracy)	0.90785	0.77389	0.77314	0.81341
Tasa de Error	0.09215	0.22611	0.22686	0.18659
Precisión	0.60684	0.28150	0.28096	0.31423
Recall	0.20752	0.82088	0.82213	0.74154
F measure	0.30927	0.41923	0.41880	0.44141

Fuente: Elaboración propia

En el cuadro 9 se aprecia las diferentes medidas para los tipos de muestreo especificados, para poder elegir el modelo adecuado se tiene que analizar cada una de las métricas, en cuanto a la exactitud (Accurcy), para todos los métodos de muestreo esta métrica bajó, y las tasas de error aumentaron en comparación con los datos originales, sin embargo al trabajar con datos desbalanceados estas métricas no son las más adecuadas.

En cuanto a la precisión, se manifiesta la presencia de falsos positivos, sin embargo el Recall (sensibilidad) aumentó considerablemente en cada método, siendo las de mejor las que se obtiene con el método de sub-muestreo (0.8208) y sobre muestreo (0.82213), esto es importante porque manifiestan menor presencia de falsos negativos. En cuanto a las medidas F-measure, todos los métodos aumentaron moderadamente, lo cual es bueno puesto que un valor alto de F-Measure indica que el modelo funciona mejor en la clase positiva (los que fugaron).

Si bien es cierto, las últimas 3 métricas últimas son las más recomendables para datos desbalanceados, estas pueden seguir siendo ineficaces al responder las preguntas importantes de clasificación. La precisión no dice nada sobre la predicción negativa, y el recall de los resultados es más interesante para conocer aspectos positivos reales.

Ante todo esto nace la necesidad de trabajar con una mejor métrica para satisfacer aspectos de mejor desempeño, tal métrica es la curva ROC puesto que es la métrica de evaluación más utilizada para este tipo de datos.

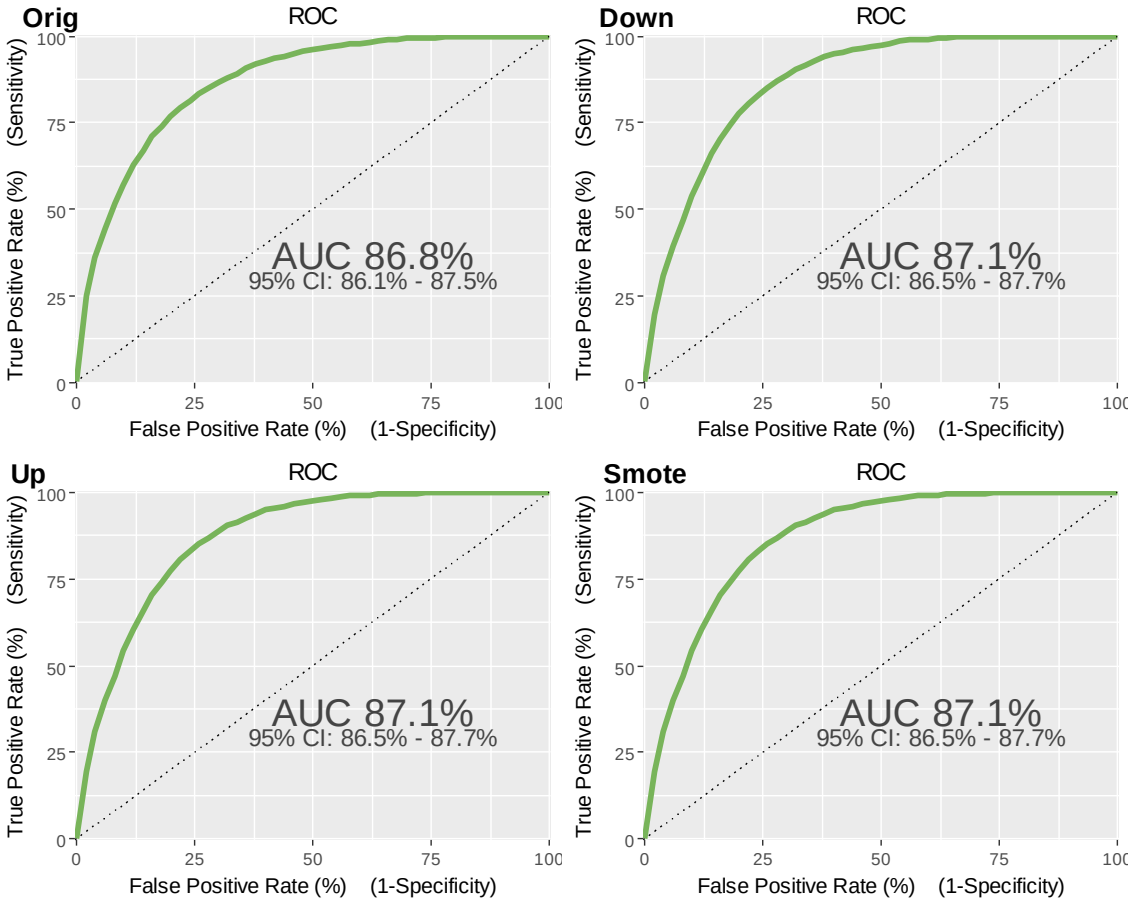


Figura 19: Comparación de curvas ROC para los modelos de regresión logística mediante los métodos de muestreo

La figura 19, muestra el AUC (área bajo la curva) para la regresíón logística en cada método de muestreo. Al evaluar esta métrica se aprecia que casi no hay diferencia entre los métodos, esto también se observa en los intervalos al 95% de confianza. Cada punto en el gráfico ROC, corresponde al rendimiento de un único clasificador en la distribución dada. Cuanto mayor es el área bajo la curva ROC, mayor es la precisión. Los resultados arrojaron valores realtivamente altos de AUC, y a pesar de que estas curvas pueden tener pocas deficiencias, se sabe que en más del 90% estas curvas funcionan bastante bien.

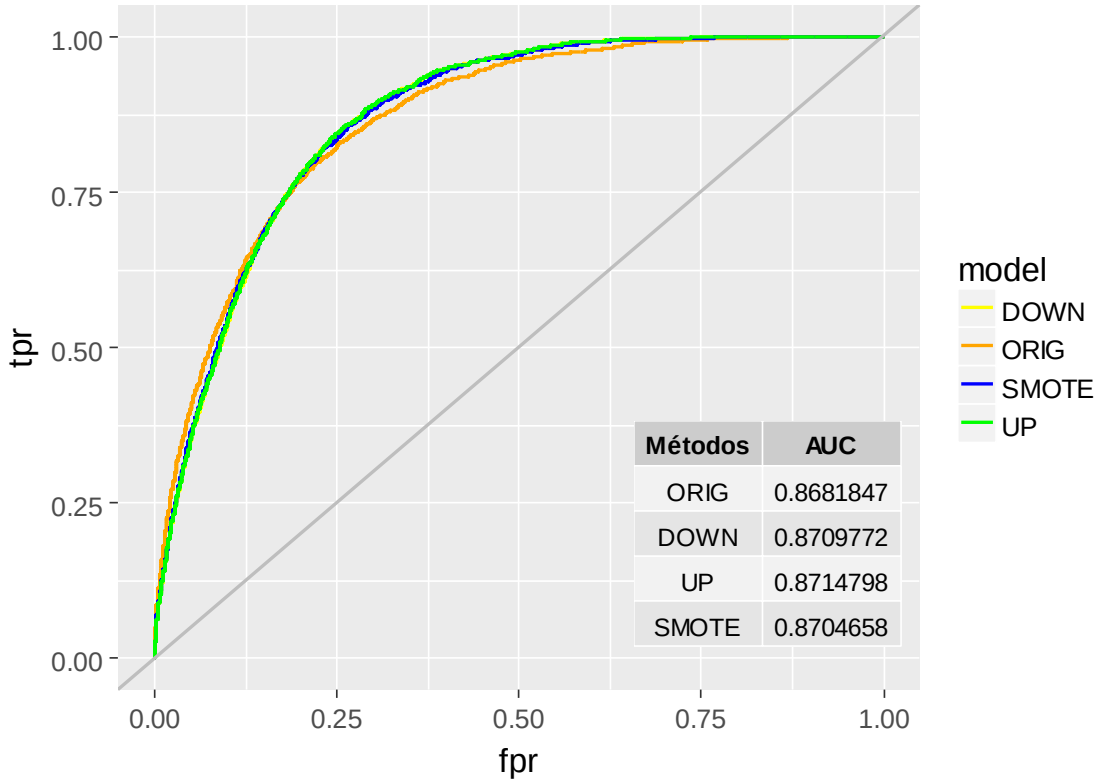


Figura 20: Curvas ROC mediante los métodos de muestreo en los modelos de regresión logística

La figura 20 muestra las diferentes curvas ROC en forma conjunta para los modelos de regresión logística mediante muestreo. Se observa que con los métodos de muestreo el AUC aumentó ligeramente, no habiendo mucha diferencia entre ellos, la métrica AUC para cada modelo es alta, por lo que se puede optar cualquier método de muestreo para construir el modelo.

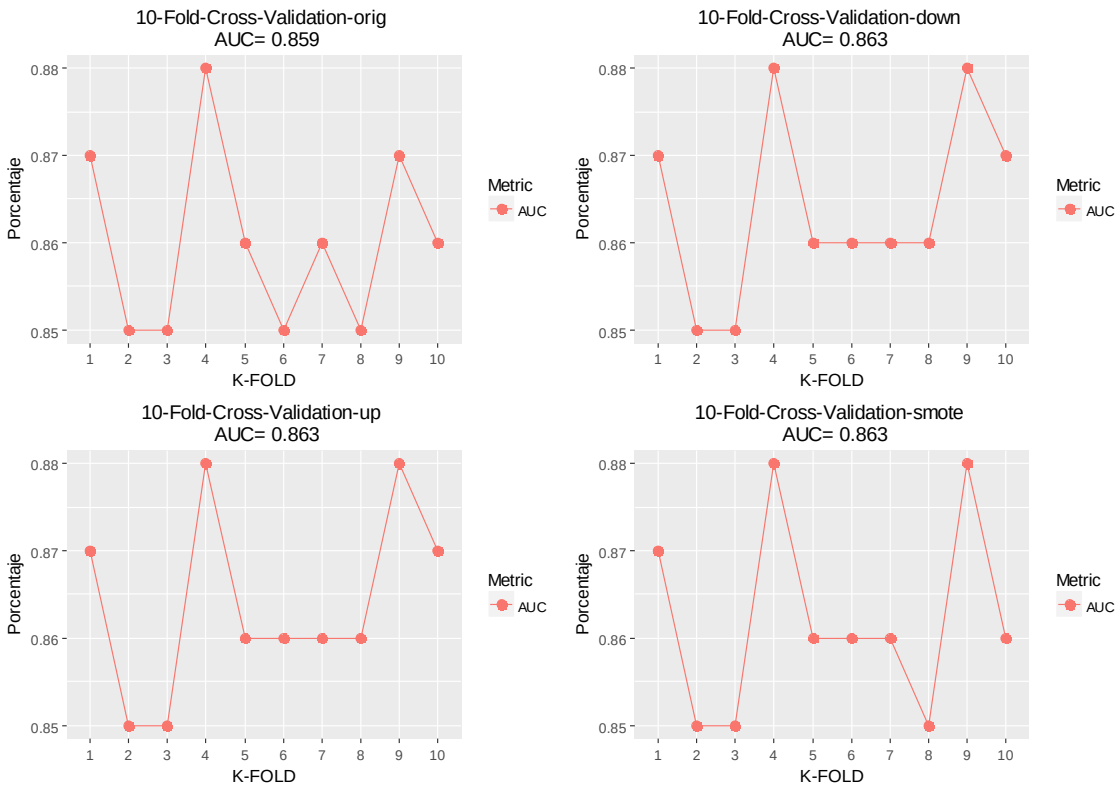


Figura 21: Validación cruzada en el AUC para los modelos de regresión logística mediante los métodos de muestreo

La figura 21 muestra la validación cruzada para el AUC, en los modelos de regresión logística mediante los métodos de muestreo, para evaluar cada modelo se formó Kfold, que representan a K grupos que son asignados de manera aleatoria para posteriormente construir modelos y evaluarlos repetitivamente.

Para cada método se realizó 10 particiones, cada punto graficado en cada figura corresponde a cada uno de los porcentaje del área bajo la curva (AUC) al dejar una de las 10 particiones fuera del conjunto de entrenamiento y utilizarlo como conjunto de datos de prueba. Finalmente cada uno de los 10 puntos fue promediado obteniendo el AUC general para cada modelo indicado.

Se observa que no hubo mucha variación entre cada método de muestreo al utilizar la regresión logística, la validación cruzada para cada modelo fue semejante en comparación con el método de retención realizado en la figura 17.

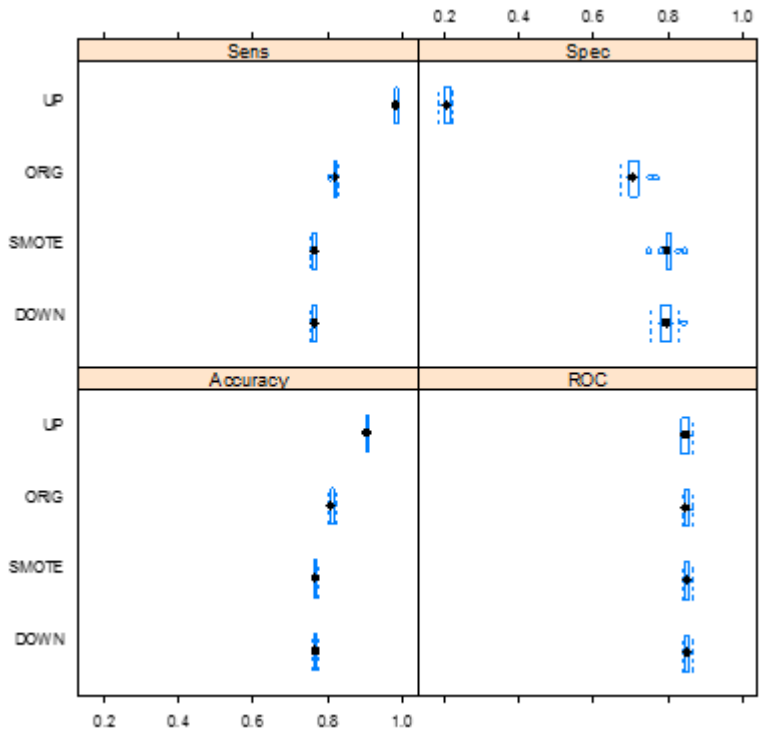


Figura 22: Validación de diferentes métricas para los modelos de regresión logística mediante los métodos de muestreo.

La figura 22 presenta las diferentes métricas de validación cruzada segmentado por cada método de muestreo, en ella se puede apreciar que utilizando los datos originales, la sensibilidad (Recall) es la más alta, sin embargo la especificidad es baja, es decir presenta menor cantidad de falsos negativos, pero mayor cantidad de falsos positivos. Al realizar los métodos de muestreo la sensibilidad baja relativamente (incremento leve de falsos negativos), pero la especificidad aumenta considerablemente (menor cantidad de falsos positivos). La tasa de error de predicción (Accuracy) es mejor en los datos originales, sin embargo al tener en cuenta que se trabajó con datos desbalanceados esta métrica no es confiable. Al final se aprecia las curvas ROC vistas anteriormente las cuales tiene la métrica AUC casi semejante.

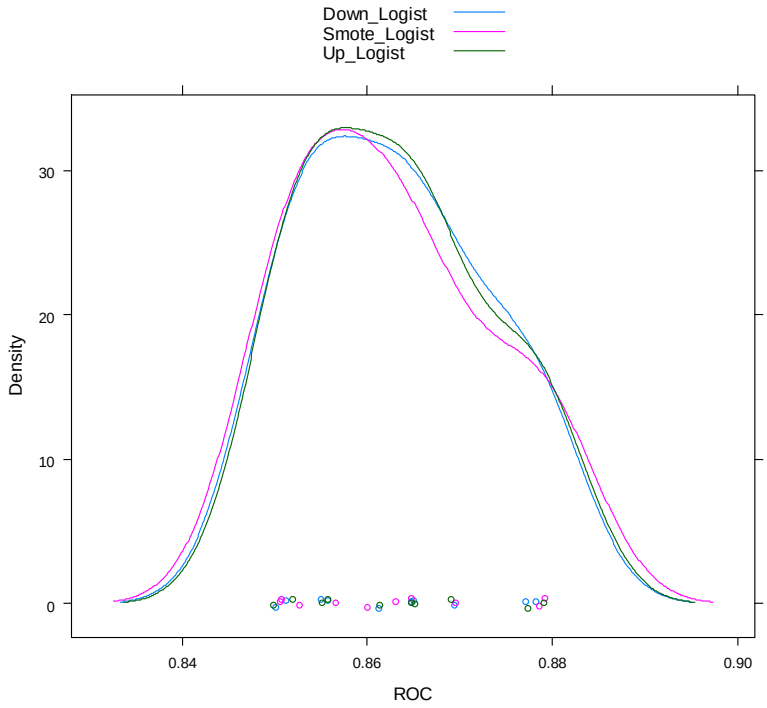


Figura 23: Densidades para los modelos de regresión logística mediante los métodos de muestreo

La figura 23 muestra las densidades para cada método de muestreo mediante la regresión logística, en estas líneas se evaluó el comportamiento de cada densidad, donde el modelo óptimo debería ser el de mayor altura y a la vez el de menor variabilidad en la curva ROC (cuando no existe mucha amplitud en el ancho de la curva). Se observa también que los tres modelos logísticos presentan un comportamiento similar en variabilidad, concluyendo que los modelos son estables.

Cuadro 10: Matrices de confusión comparativa para los modelos Adaboost mediante métodos de muestreo

Datos originales

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	975	318
	No fugo	1420	21377

Sub-muestreo (DOWN)

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	2165	3763
	No fugo	230	17932

Sobre-muestreo (UP)

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	979	318
	No fugo	1416	21377

Smote

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	1799	1841
	No fugo	596	19854

Fuente: Elaboración propia

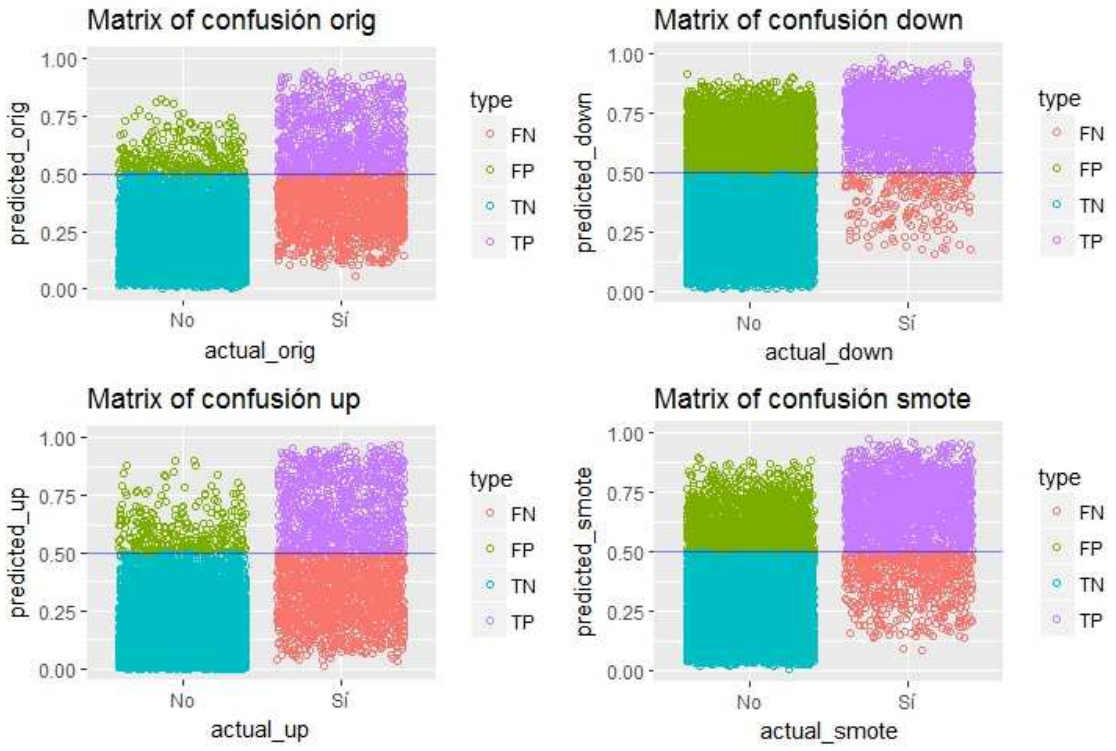


Figura 24: Gráficas de matrices de confusión para los modelos Adaboost mediante los métodos de muestreo

En el cuadro 10 se muestra las diferentes matrices de confusión para los modelos con el algoritmo Adaboost equilibrados mediante los métodos de muestreo, en cada método de muestreo la cantidad de verdaderos positivos aumentaron en comparación con los datos originales, y con los métodos de sub-muestreo y SMOTE, los falsos negativos disminuyeron considerablemente, la figura 24 muestra la dispersión de cada valor clasificado, el color morado muestra los verdaderos positivos, corroborando que con los métodos de muestreo los verdaderos positivos aumentan, para los demás colores hay variaciones en cada método de muestreo por lo que para tener una idea más clara, se pasó a analizar las métricas recomendadas para datos desbalanceados.

Cuadro 11: Medidas de desempeño para los diferentes métodos de muestreo en el algoritmo Adaboost

Medidas	**Sin ajuste**	**Sub-muestreo**	**Sobre-muestreo**	**SMOTE**
Exactitud (Accuracy)	0.92785	0.83425	0.92802	0.89884
Tasa de Error	0.07215	0.16575	0.07198	0.10116
Precisión	0.75406	0.36522	0.75482	0.49423
Recall	0.40710	0.90397	0.40877	0.75115
F score	0.52874	0.52025	0.53034	0.59619

Fuente: Elaboración propia

En el cuadro 11 se aprecia las diferentes medidas para los tipos de muestreo especificados, para poder elegir el modelo adecuado se tiene que analizar cada una de las métricas, en cuanto a la exactitud (Accurcy), para todos los métodos esta fue alta, y las tasas de error son menores son relativamente bajas, siendo el sobre-muestreo la de mejor desempeño en estas dos métricas, sin embargo como se dijo antes al trabajar con datos desbalanceados estas métricas son engañosas.

En cuanto a la precisión, existen problemas de falsos positivos, especialmente con los métodos de sub-muestreo (0.3652) y SMOTE (0.494), en cuanto al Recall (sensibilidad) el mejor desempeño se obtiene con el método de sub-muestreo (0.904) y SMOTE (0.7511), es decir menor presencia de falsos negativos. En cuanto a las medidas F-measure, todos los métodos tienen una precisión moderada pareja.

En cuanto a las medidas F-measure, en todos los métodos se mantiene igual, teniendo un ligero aumento con el método SMOTE, esta medida ponderada es media indicando que los modelos funcionaron mejor en la clase positiva (los que fugaron).

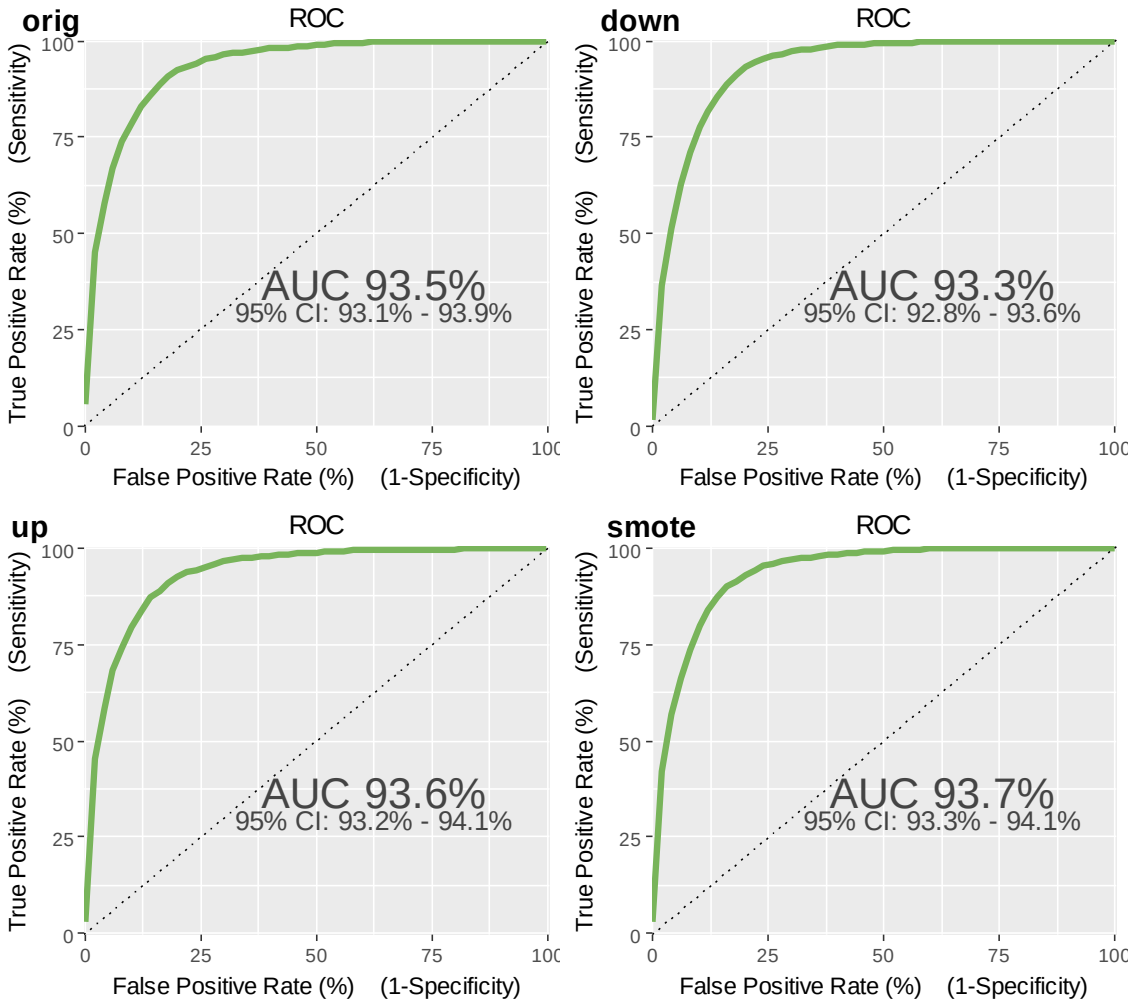


Figura 25: Comparación de curvas ROC para los modelos Adaboost mediante los métodos de muestreo

La figura 25, muestra el AUC (área bajo la curva) para el algoritmo Adaboost en cada método de muestreo. Cada método da un AUC aproximado del 93%, esto también se observa en los intervalos de confianza al 95% de confianza. Los resultados arrojaron valores significativamente altos, pudiendo elegir cualquier método de muestreo para compararlo con la regresión logística.

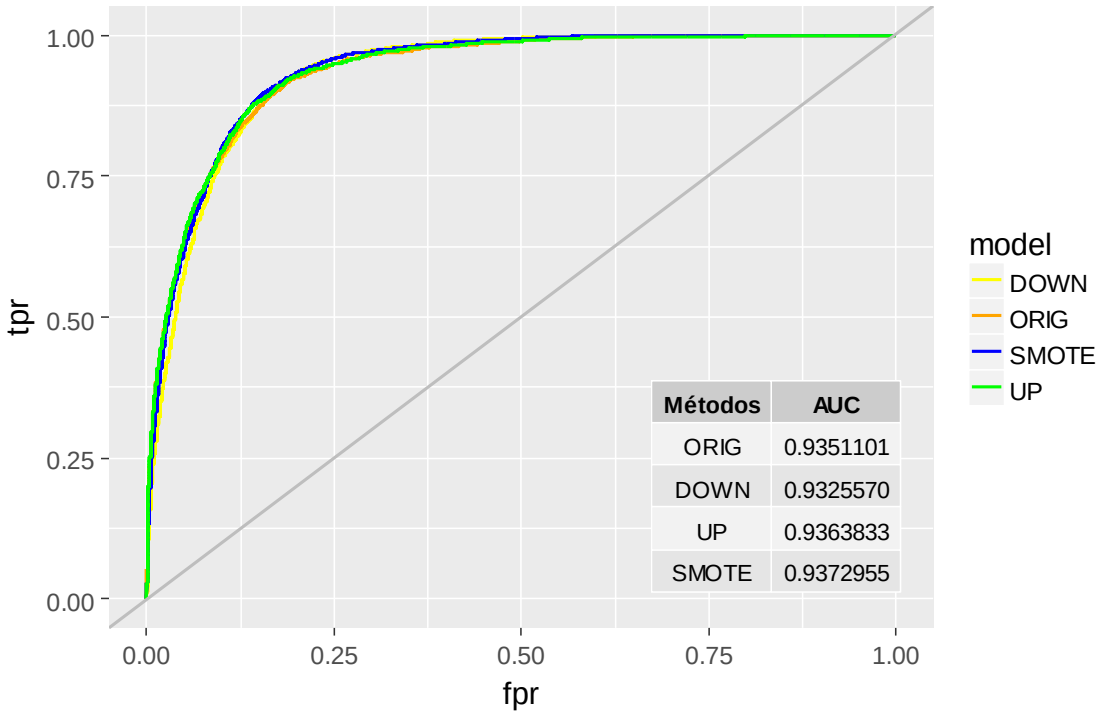


Figura 26: Curvas ROC mediante los métodos de muestreo en los modelos mediante el algoritmo Adaboost

La figura 26 muestra las diferentes curvas ROC en forma conjunta para el algoritmo Adaboost mediante los métodos de muestreo. El área es similar en casi todos los casos, y los valores altos, lo que significa que para este caso cualquier método de muestreo trabaja eficientemente.

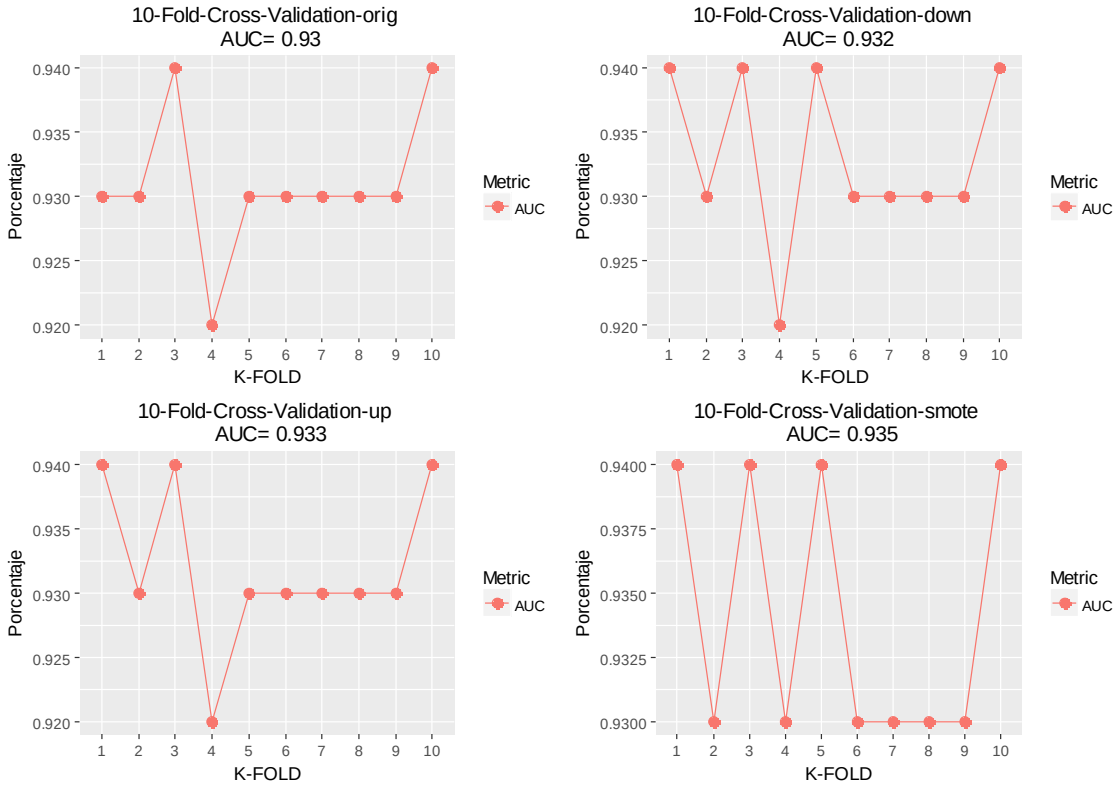


Figura 27: Validación cruzada en el AUC para los modelos con el algoritmo Adaboost mediante los métodos de muestreo

La figura 27 muestra la validación cruzada para el AUC, en los modelos con el algoritmo Adaboost mediante los métodos de muestreo, para cada método se realizó 10 particiones, cada punto graficado en cada figura corresponde a cada uno de los porcentaje del área bajo la curva (AUC) al dejar una de las 10 particiones fuera del conjunto de entrenamiento y utilizarlo como conjunto de datos de prueba. El promedio en cada uno de los 10 puntos para cada método da resultados semejantes al que se vio con el método de retención.

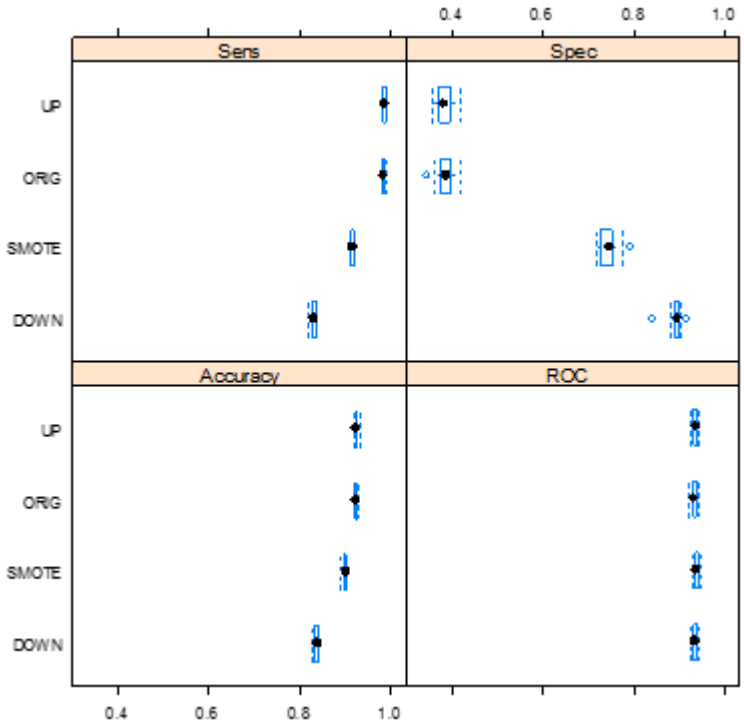


Figura 28: Validación de diferentes métricas para el algoritmo Adaboost mediante los métodos de muestreo.

En la figura 28 se aprecia que utilizando los datos originales y el método de sobre muestreo (UP) poseen la sensibilidad (Recall) más alta, sin embargo la especificidad es baja, es decir presenta menor cantidad de falsos negativos, pero mayor cantidad de falsos positivos. La tasa de error de predicción (Accuracy) es parecida en todos lo métodos, sin embargo se recalca que al tener en cuenta que se trabajó con datos desbalanceados esta métrica no es confiable.

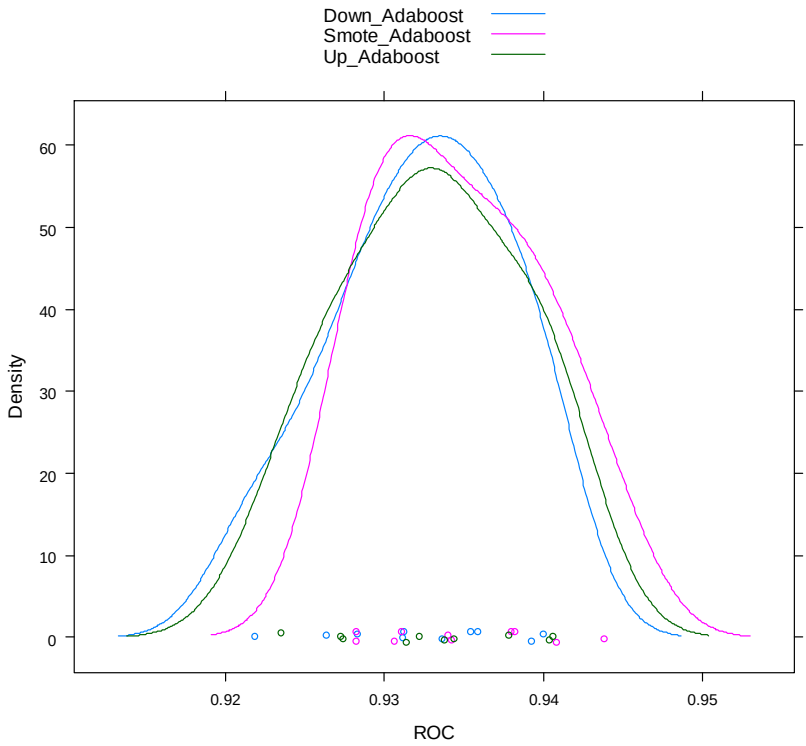


Figura 29: Densidades para los modelos Adaboost mediante los métodos de muestreo

La figura 29 muestra las densidades para cada método de muestreo mediante el algoritmo Adaboost, se observa también que los tres modelos presentan un comportamiento similar en variabilidad, habiendo una pequeña ventaja para el método SMOTE, sin embargo esta diferencia no es relevante, concluyendo que los modelos con el algoritmo Adaboost son estables.

4.5 COMPARACIÓN DE LOS MODELOS DE REGRESIÓN LOGÍSTICA Y EL ALGORITMO ADABOOST MEDIANTE LOS MÉTODOS DE MUESTREO

Cuadro 12: Comparación de las medidas de desempeño con los métodos de muestreo para los modelos de regresión logística y el algoritmo Adaboost

Métricas	Sin muestreo			Sub-muestreo			Sobre-muestreo			SMOTE		
	Logís-tica		Ada-boost	Logís-tica		Ada-boost	Logís-tica		Ada-boost	Logís-tica		Ada-boost
Exactitud	0.908	<	0.928	0.774	<	0.834	0.773	<	0.928	0.813	<	0.899
Error	0.092	>	0.072	0.226	>	0.166	0.227	>	0.072	0.187	>	0.101
Precisión	0.607	<	0.754	0.282	<	0.365	0.281	<	0.755	0.314	<	0.494
Recall	0.208	<	0.407	0.821	<	0.904	0.822	>	0.409	0.742	<	0.751
F score	0.309	<	0.529	0.419	<	0.52	0.419	<	0.53	0.441	<	0.596

Fuente: Elaboración propia

En la Cuadro 12 se muestra una comparación entre los métodos de muestreo para los modelos de regresión logística y el algoritmo Adaboost, se observa que en cuanto a la precisión el algoritmo Adaboost fue superior a la regresión logística en todos los métodos de muestreo, eso quiere decir que el algoritmo Adaboost tiene mayor capacidad para detectar a la proporción dentro de los datos clasificados como fugados que en verdad lo son.

También se observa que solamente el Recall (sensibilidad) para el método de sobre muestreo en la regresión logística fue superior, en las demás medidas (Precisión, y F measure) el algoritmo Adaboost tuvo mejor desempeño, por lo cual hasta aquí se pudo concluir que el Algoritmo Adaboost tiene la capacidad de mejor clasificación en la fuga de clientes para la empresa de telefonía evaluada. La conclusión final para los métodos de muestreo se ve a continuación con el análisis comparativo de las curvas ROC para ambos modelos.

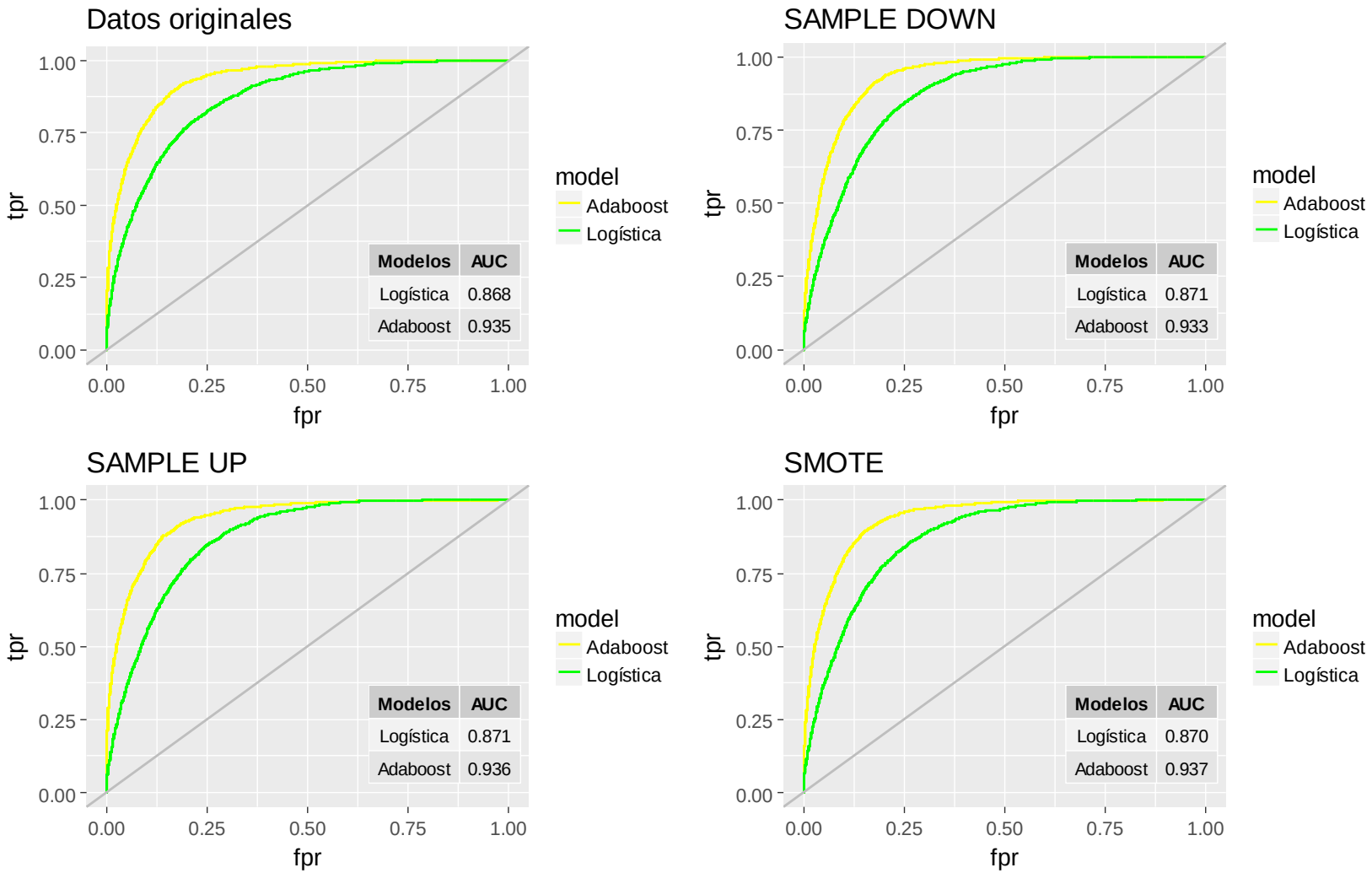


Figura 30: Comparación del AUC en los modelos de regresión logística y el algoritmo Adaboost mediante los métodos de muestreo

En la figura 30 se muestra la comparación final entre el algoritmo Adaboost y la regresión logística utilizando métodos de muestreo. En todos los modelos utilizando muestreo al igual que con los datos originales, el algoritmo Adaboost fue superior, el AUC sobrepasa el 93% entre las técnicas de muestreo. Con estos resultados y con las medidas vistas anteriormente se concluye que para los datos de la empresa de telefonía la mejor técnica de clasificación es el algoritmo Adaboost.

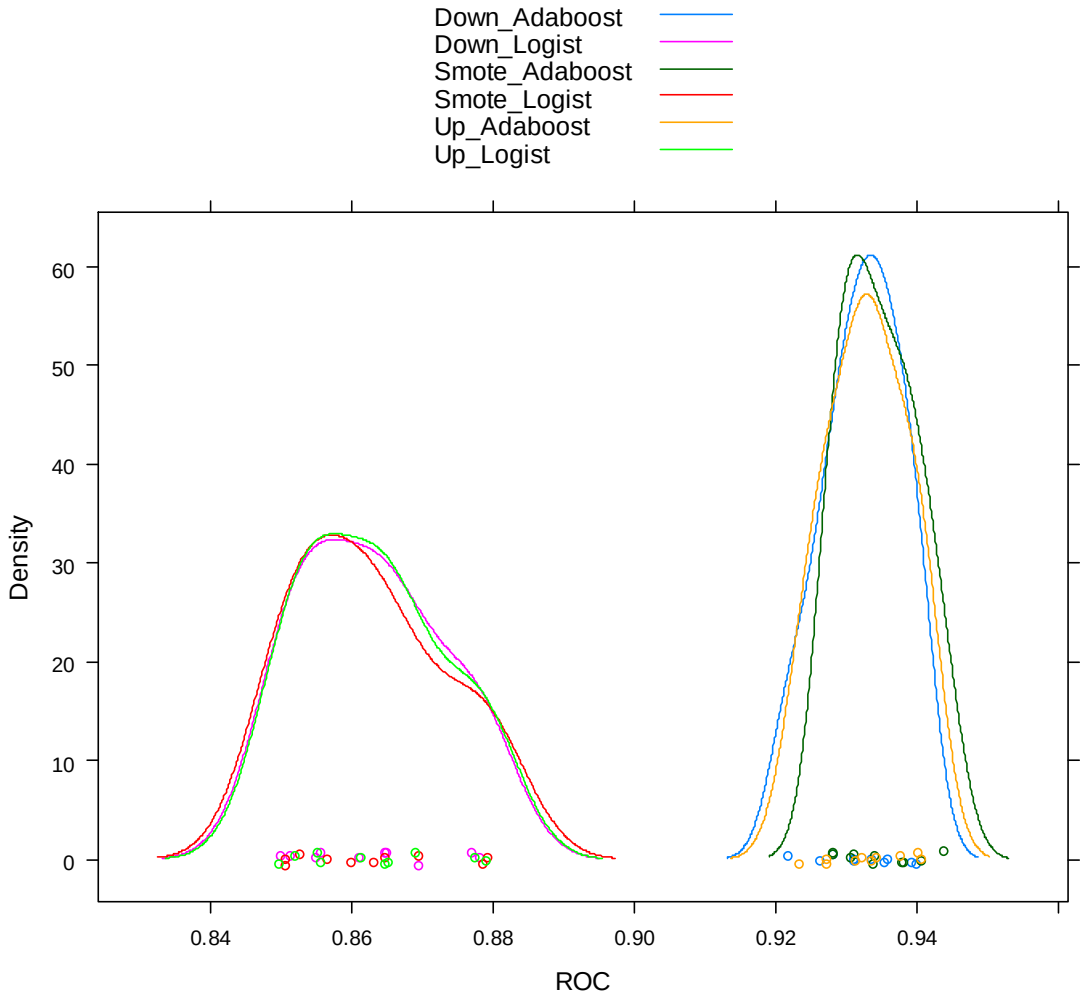


Figura 31: Comparación de densidades en los modelos de regresión logística y el algoritmo Adaboost mediante los métodos de muestreo

En la figura 31 se compara las densidades de todos los modelos mediante métodos de muestreo, los modelos utilizando el algoritmo Adaboost tienen mayor precisión en la curva Roc y menor variabilidad, concluyendo que estos son más óptimos para la predicción de futa en la empresa.

4.6 MODELOS MEDIANTE AJUSTE A NIVEL DE FUNCIÓN O ALGORITMO

Anteriormente se analizó cada modelo mediante técnicas de muestreo, esto con el fin de remediar el problema del desbalance. En esta sección no se utilizó métodos de muestreo, sino que se construyó y analizó cada modelo a nivel de función o algoritmo, lo que significa que se hizo algún ajuste o modificación en la función de enlace (para la regresión logística) y en el algoritmo Adaboost, posteriormente se pasó a comparar cada modelo con sus métricas de eficiencia.

El modelo logístico asimétrico

Este es el primer modelo logístico ajustado que se propone, el cual se llamó Logit Asym para efectos de resumen.

Cuadro 13: Ajuste de modelos asumiendo diferentes valores de K para la regresión logística

$\eta(x) = \beta_0 + \beta_1 x_1 + ... + \beta_i x_i$		
Escenario 1	$P(y=1\|x) = \frac{\exp\{\eta(x)\}}{1+\exp\{\eta(x)\}}$	$k = 0.0$
Escenario 2	$P(y=1\|x) = \frac{\exp\{\eta(x)\}+0.01}{1.01+\exp\{\eta(x)\}}$	$k = 0.01$
Escenario 3	$P(y=1\|x) = \frac{\exp\{\eta(x)\}+0.02}{1.02+\exp\{\eta(x)\}}$	$k = 0.02$
Escenario 4	$P(y=1\|x) = \frac{\exp\{\eta(x)\}+0.05}{1.05+\exp\{\eta(x)\}}$	$k = 0.05$
Escenario 5	$P(y=1\|x) = \frac{\exp\{\eta(x)\}+0.008}{1.008+\exp\{\eta(x)\}}$	$k = 0.008$

Fuente: Elaboración propia

En la Cuadro 13 se ajusta la regresión logística con algunos valores de K, en cada escenario se modificó la función, como siguiente paso se corrieron los modelos y se pasó analizar cada escenario.

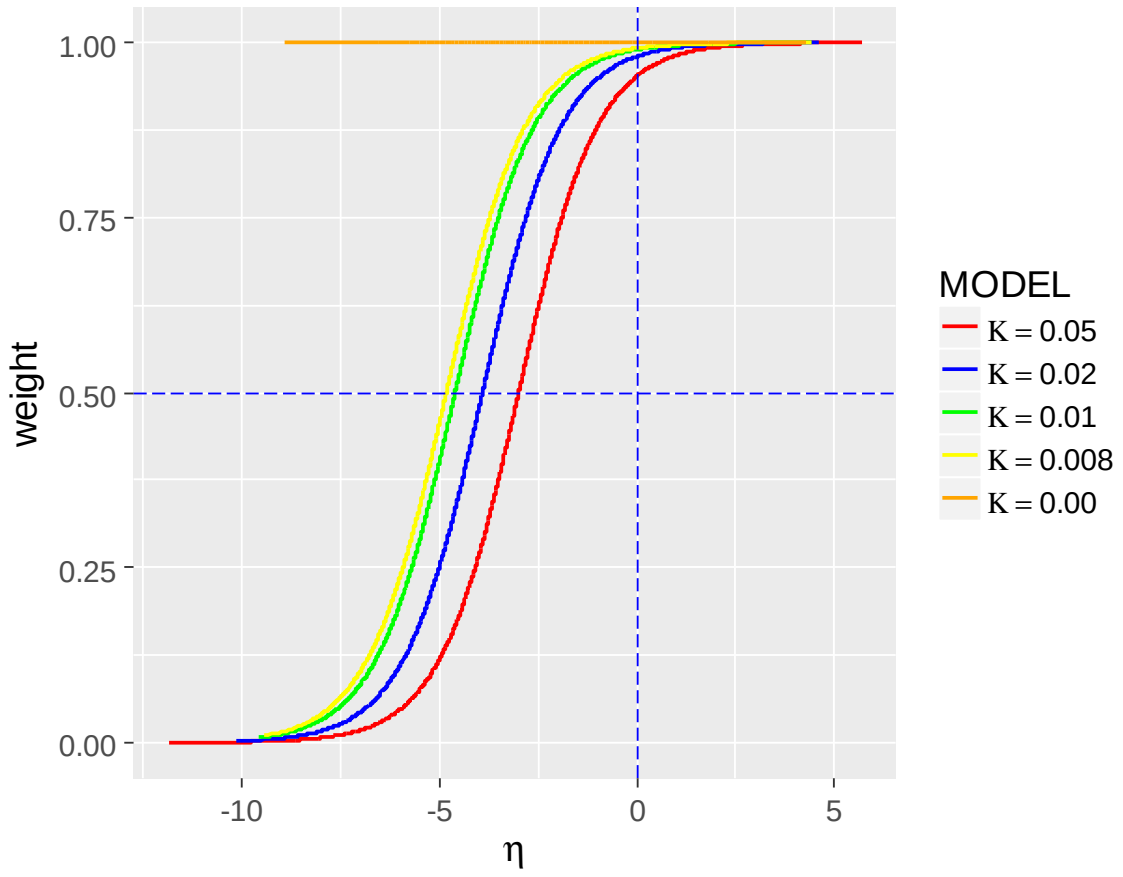


Figura 32: Funciones logísticas con valores de Kappa ajustados.

Cuadro 14: Pesos en los niveles de la variable de respuesta Fuga para diferentes valores Kappa

K	AIC	$\sum_{yi=0} w(\eta_i)$	$\sum_{yi=1} w(\eta_i)$	Razón
k= 0.05	28314.08	14588.98	3947.21	3.70
k = 0.02	27438.96	26829.26	4917.27	5.46
k = 0.01	27182.36	34298.37	5274.46	6.50
k = 0.08	27133.73	36315.08	5348.64	6.79
k = 0.00	26949.5	50554.00	5656.00	8.938

Fuente: Elaboración propia

En la figura 32 al igual en el cuadro 14 se indica que el número de clientes que no fugaron y los que sí, es de 50554 y 5656, respectivamente (se tiene en cuenta que se trabajó con los datos de entrenamiento), lo que indica claramente un conjunto de datos desequilibrados. La función de ponderación w (η_i) para cada η_i (i = 1,…, 56210) cuando K = 0, equivale al modelo de regresión logística simple; por lo tanto, el valor de w (η_i) es igual a 1 para todas

las observaciones (color naranja de la gráfica). Sin embargo, al aumentar el valor de K, se observa que el peso para los que no fugaron disminuye drásticamente con el aumento de η_i. El uso de K permite compensar las grandes diferencias entre los tamaños de muestra de las observaciones que no fugaron y los que sí lo hicieron.

Los tamaños de muestra efectivos calculados como la suma de w (η_i) son 14588.98 y 3947.21 en el caso de K = 0.05, lo que da como resultado una razón de tamaño de muestra de 3.7. Este valor es mucho menor en relación al tamaño de muestra original (k=0) de 8.9. Probando diferentes valores de K cuando la probabilidad marginal alcanza un valor máximo, y tomando la en consideración el AIC y la razón de tamaño entre los que fugaron y los que no, el valor óptimo de K para este caso es 0.02. Otro aspecto de K es que tiende a aumentar la probabilidad de fuga, que a menudo es subestimada por un modelo de regresión simple cuando el tamaño de la muestra es altamente desequilibrado. Con esto se comprueba que $P_K (y = 1 \mid \eta_i) \geq P_0 (y = 1 \mid \eta_i)$ puesto que se asume que $K \geq 0$.

Se recalca que el término K puede considerarse como una variable, que no es observable en realidad, pero tiene una impacto en la probabilidad de los que si fugan.

Detallado lo anterior, se elige el modelo de regresión logística con parámetro K=0.02, este modelo será comparado con los otros para finalmente elegir el más óptimo.

Cuadro 15: Coeficiente del modelo de regresión logística con parámetro K=0.02

Coefficients:	Estimate	Std.Error	z value	valuePr(>|z|)	sig.
(Intercept	-4.2840	0.1460	-29.334	0.0000	***
Sexo2	0.0603	0.0383	1.575	0.1152	
Edad	0.0009	0.0015	0.57	0.5688	
Deuda	-0.3870	0.0091	-42.745	0.0000	***
Renuncia	0.0262	0.0091	2.867	0.0041	**
Canal2	-0.0509	0.0477	-1.066	0.2864	
Canal3	-0.0597	0.0478	-1.25	0.2114	
Canal4	-0.0045	0.1027	-0.043	0.9654	
Reclamos	1.1770	0.0197	59.861	0.0000	***
Tipo.Reclamo1	-0.7556	0.0648	-11.664	0.0000	***
Tipo.Reclamo2	-1.1660	0.0534	-21.833	0.0000	***
Tipo.Reclamo3	-1.7380	0.0701	-24.776	0.0000	***
Tipo.Reclamo4	-2.3750	0.1259	-18.866	0.0000	***
Mensaje	-0.0015	0.0012	-1.256	0.2092	
Llamadas	0.0045	0.0005	8.458	0.0000	***
Kilobytes	-0.0216	0.0189	-1.143	0.2531	
Antigüedad	0.0000	0.0000	-2.371	0.0178	*
ROL2	0.8530	0.0606	14.086	0.0000	***
ROL3	1.7410	0.0594	29.295	0.0000	***
Comunidad.L	-1.1480	0.0445	-25.803	0.0000	***
Comunidad.Q	0.0661	0.0421	1.57	0.1165	
Comunidad.C	-0.0690	0.0409	-1.688	0.0914	.
Cliente.L	0.7620	0.0433	17.615	0.0000	***
Cliente.Q	0.0529	0.0439	1.205	0.2284	
Cliente.C	0.1350	0.0419	3.223	0.0013	**
Cliente^4	0.0057	0.0445	0.129	0.8975	
ARPU	0.0054	0.0016	3.403	0.0007	***
MOU	0.0020	0.0003	6.656	0.0000	***
Procedencia2	0.0132	0.0382	0.344	0.7307	
NOTA	0.0833	0.0091	9.192	0.0000	***
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

Fuente: Elaboración propia

El cuadro 15 muestra los coeficientes del modelo de regresión con parámetro K=0.02, y la significancia de cada variable. Se observa variables altamente significativas y otras que no, sin embargo se mantuvo todas variables para efecto de compararlas en los otros modelos. Como hay diferentes variables significativas fue necesario la realización de una gráfica para analizar la importancia de cada.

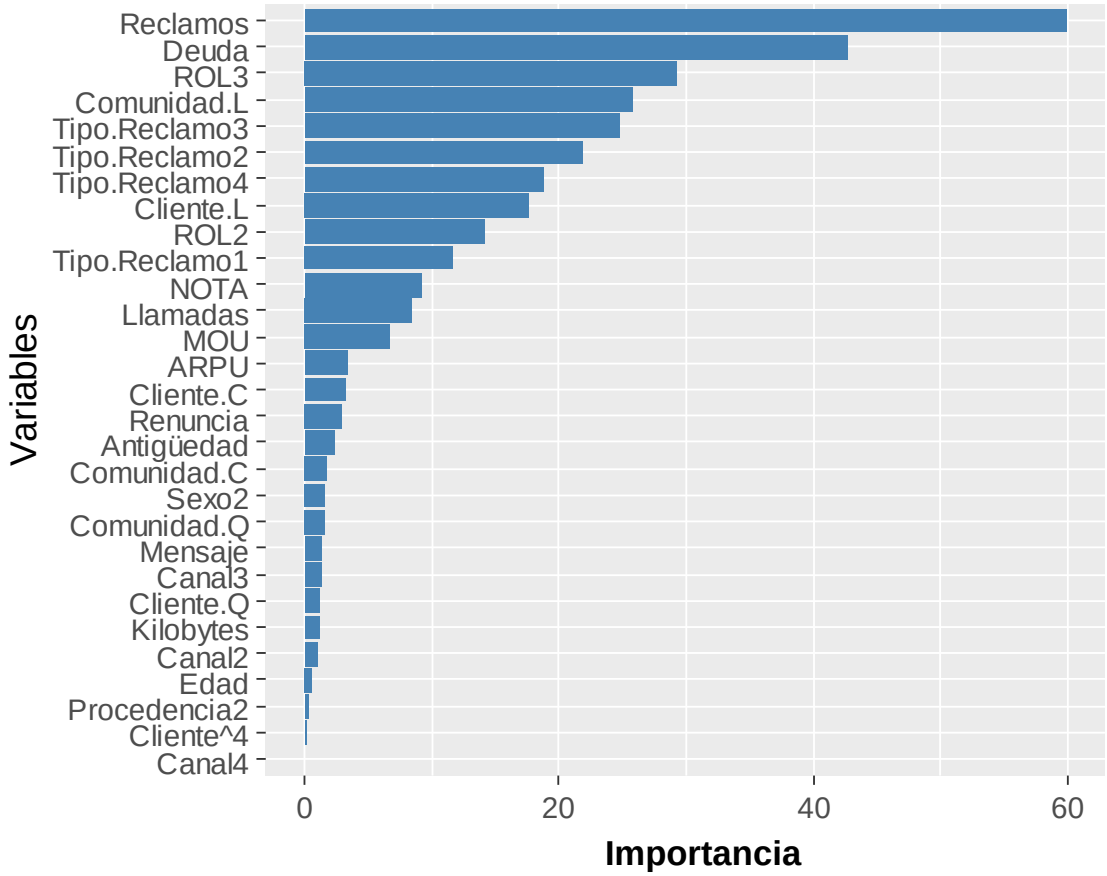


Figura 33: Distribución de variables para el modelo de regresión logística con parámetro K=0.02 según grado de importancia

La figura 33 detalla la distribución de cada variable, siendo el número de reclamos la más importante, en la gráfica descriptiva inicial se observó que los que fugaron fueron los que en promedio más reclamos realizaron, sigue la variable deuda (días de deuda promedio), la gráfica descriptiva anterior no mostró mucha diferencia entre los que fugan o no para esta variable, sin embargo se observa que es determinante para entender el comportamiento de los clientes. Otras variables a tomar en cuenta son Rol, Comunidad, Tipo de reclamo, etc. Cada variable contribuye a formar el modelo por lo que se prefirió mantenerlas para efectos de comparación con los otros modelos.

El modelo Power Logit

Al igual que el modelo anterior a este modelo se ajusta en la forma de su distribución, acá el parámetro establecido es Lambda.

Cuadro 16: Ajuste de modelos asumiendo diferentes valores del parámetro λ para el modelo logístico Power Logit

$\eta(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_i x_i$		
Escenario 1	$F_p(\eta) = \left[\frac{\exp\{\eta(x)\}}{1+\exp\{\eta(x)\}}\right]^{1/4}$	$\lambda = 1/4$
Escenario 2	$F_p(\eta) = \left[\frac{\exp\{\eta(x)\}}{1+\exp\{\eta(x)\}}\right]^{1/2}$	$\lambda = 1/2$
Escenario 3	$F_p(\eta) = \left[\frac{\exp\{\eta(x)\}}{1+\exp\{\eta(x)\}}\right]^{3/2}$	$\lambda = 3/2$
Escenario 4	$F_p(\eta) = \left[\frac{\exp\{\eta(x)\}}{1+\exp\{\eta(x)\}}\right]^{5/2}$	$\lambda = 5/2$
Escenario 5	$F_p(\eta) = \left[\frac{\exp\{\eta(x)\}}{1+\exp\{\eta(x)\}}\right]^{3.5}$	$\lambda = 3.5$

Fuente: Elaboración propia

En la Cuadro 16 se ajusta los modelos Power Logit para algunos valores de λ, en cada escenario se modificó la función, como siguiente paso se corrieron los modelos y se pasó analizar cada escenario.

Cuadro 17: AIC y AUC para modelos Power Logit en diferentes valores de λ

Lambda	**AIC**	**AUC**
λ = 1/4	27364.74	0.860
λ = 1/2	27149.63	0.865
λ = 3/2	26860.51	0.869
λ = 5/2	26785.23	0.871
λ = 3.5	26785.23	0.870

Fuente: Elaboración propia

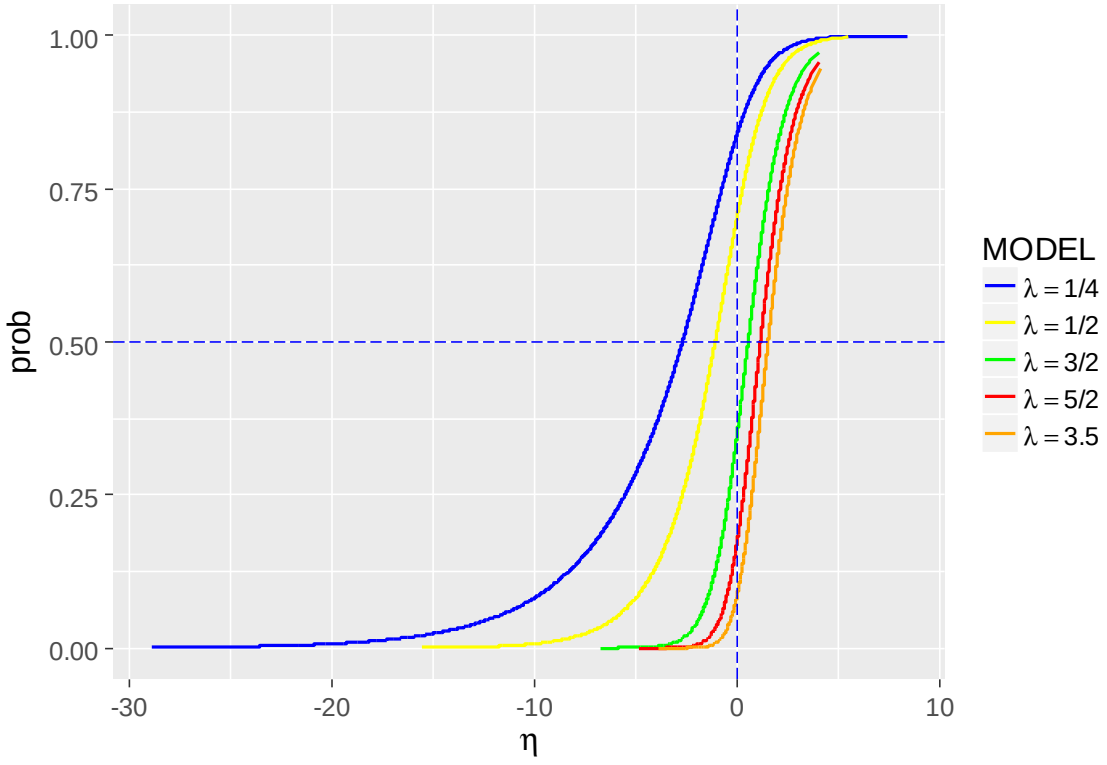


Figura 34: Distribución de funciones Power Logit para diferentes valores de λ

La figura 34 muestra cada función Power logit en diferentes valores de λ, se observa que cada función es asimétrica, es decir no pasan por el punto central en el predictor 0 con probabilidad 0.5.

Según Bazán et.al (2017) la forma de interpretación y elección de cada valor de λ depende de cómo estén balanceados los éxitos en la variable de respuesta. Se debe usar $0 < \lambda < 1$ cuando la proporción de éxitos (Fuga = sí) es mayor que 0.5, $\lambda = 1$, cuando la proporción de éxitos y fracasos esta equilibrada, y $\lambda > 1$, cuando la proporción de éxitos (Fuga = sí) es menor que 0.5.

Como en este caso se cuenta con pocos valores de éxito (Fuga = sí), entonces se optó por tomar un valor de $\lambda > 1$, eligiendo el valor $\lambda = 5/2$, esto en relación a los resultados del cuadro 17 el cual muestra el AIC menor y la curva ROC mayor en comparación de los otros valores.

Por lo tanto el modelo Power Logit elegido es el que usa el valor de $\lambda = 5/2$, este modelo es el que se comparó con lo Logit Asym y el Adaboost Asym.

Cuadro 18: Coeficiente del modelo de Power Logit con λ=5/2

Coefficients:	Estimate	Std.Error	z value	valuePr(>|z|)	sig.
(Intercept	-1.1820	0.0758	-15.5890	0.0000	***
Sexo2	0.0263	0.0207	1.2680	0.2049	
Edad	0.0002	0.0008	0.1910	0.8489	
Deuda	-0.2787	0.0049	-57.1960	0.0000	***
Renuncia	0.0136	0.0050	2.7500	0.0060	**
Canal2	-0.0152	0.0259	-0.5880	0.5564	
Canal3	-0.0300	0.0259	-1.1610	0.2458	
Canal4	-0.0392	0.0558	-0.7030	0.4823	
Reclamos	0.7288	0.0106	68.6360	0.0000	***
Tipo.Reclamo1	-0.4126	0.0349	-11.8240	0.0000	***
Tipo.Reclamo2	-0.6187	0.0275	-22.5080	0.0000	***
Tipo.Reclamo3	-0.8889	0.0330	-26.9570	0.0000	***
Tipo.Reclamo4	-1.1470	0.0502	-22.8500	0.0000	***
Mensaje	-0.0004	0.0007	-0.6580	0.5108	
Llamadas	0.0024	0.0003	8.1470	0.0000	***
Kilobytes	-0.0057	0.0104	-0.5510	0.5815	
Antigüedad	0.0000	0.0000	-1.9910	0.0465	*
ROL2	0.4107	0.0289	14.1980	0.0000	***
ROL3	0.9001	0.0286	31.4920	0.0000	***
Comunidad.L	-0.5821	0.0220	-26.4860	0.0000	***
Comunidad.Q	0.0554	0.0215	2.5750	0.0100	*
Comunidad.C	-0.0460	0.0213	-2.1560	0.0311	*
Cliente.L	0.4009	0.0232	17.3050	0.0000	***
Cliente.Q	0.0246	0.0237	1.0370	0.2998	
Cliente.C	0.0822	0.0226	3.6370	0.0003	***
Cliente^4	0.0156	0.0240	0.6520	0.5143	
ARPU	0.0027	0.0009	3.0960	0.0020	**
MOU	0.0012	0.0002	7.4830	0.0000	***
Procedencia2	-0.0003	0.0207	-0.0150	0.9884	
NOTA	0.0448	0.0049	9.1500	0.0000	***
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

Fuente: Elaboración propia

El cuadro 18 muestra los coeficientes del modelo de regresión Power Logit con λ=5/2, y la significancia de cada variable. Al igual que el modelo anterior se observa variables altamente significativas y otras que no, sin embargo se mantuvo todas variables para efecto de compararlas en los otros modelos. Las significancia de las variables fueron muy similares al modelo Logit Asym.

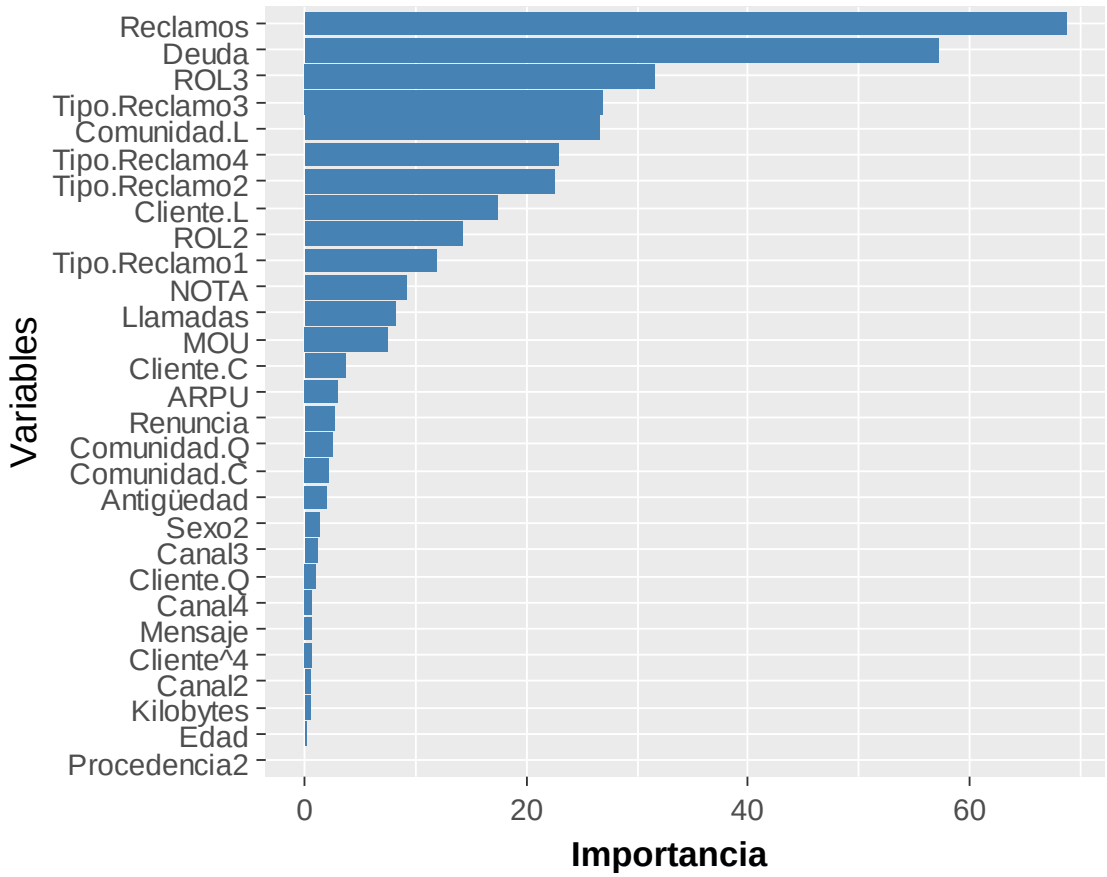


Figura 35: Distribución de variables para el modelo de Power Logit con parámetro λ=5/2 según grado de importancia.

La figura 35 detalla la distribución de cada variable, siendo el número de reclamos la más importante, seguida de la variable deuda (días de deuda promedio), otras variables a tomar en cuenta son Rol, Comunidad, Tipo de reclamo, etc. La importancia de estas variables son muy semejantes a las vistas en el modelo Logit Asym, cada variable contribuye a formar el modelo por lo que se prefirió mantenerlas para efectos de comparación con los otros modelos.

El algoritmo Adaboost desbalanceado (Adaboost Asym)

En el algoritmo Adaboost se utilizó uno de los métodos para contrarrestar el efecto de datos desbalanceados, el cual consiste en ajustar el peso de la clase (costos de clasificación errónea) de tal manera que sea más sensible a la clase que si fugó (la minoritaria). Teóricamente al realizar el procedimiento la clase minoritaria ganó importancia (se impuso un mayor costo cuando se cometieron errores en la clase de los que fugan).

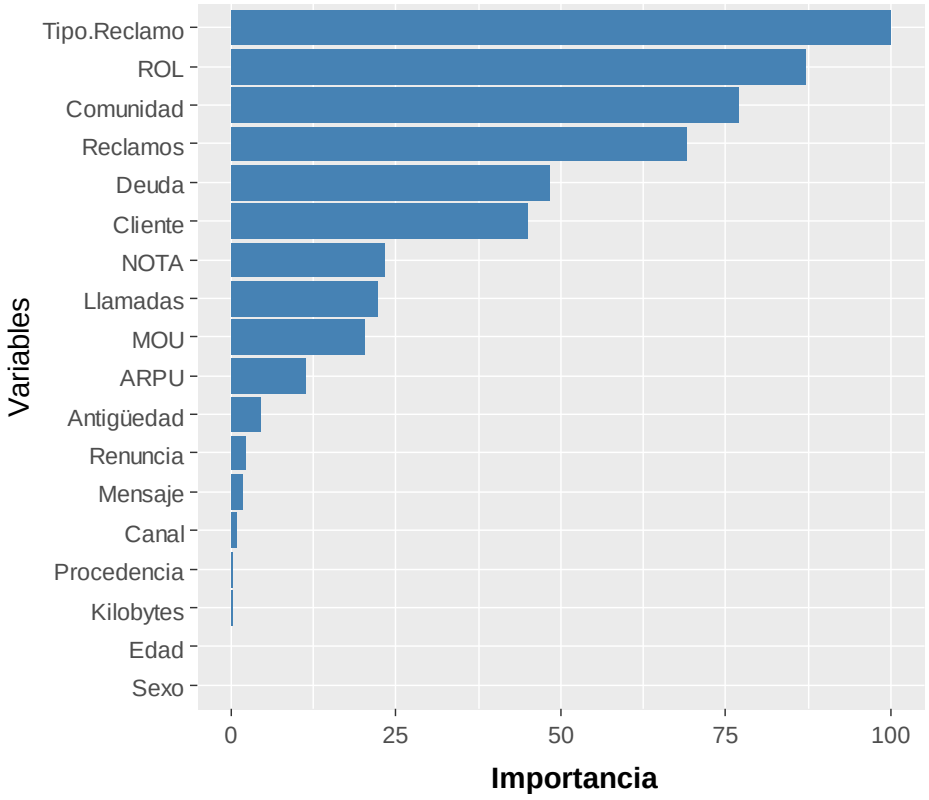


Figura 36: Importancia de variables en el algoritmo Adaboost Asym

Después que se implementó el algoritmo Adaboost, se identificó cada variable según su grado de importancia. En la Figura 36 se observa que el tipo de reclamo es la más importante, anteriormente en la parte descriptiva se observó que el reclamo más frecuente fue por problemas de Facturación, y que generalmente los que fugan prefieren ya no reclamar y abandonar el servicio. Otras variables importantes para implementación del algoritmo son el ROL, Comunidad y número de reclamos, observando que hubo cierta diferencia con los modelos de regresión logística.

COMPARACIÓN DE LOS MODELOS DE REGRESIÓN LOGÍSTICA ASYMÉTRICA Y EL ALGORITMO ADABOOST

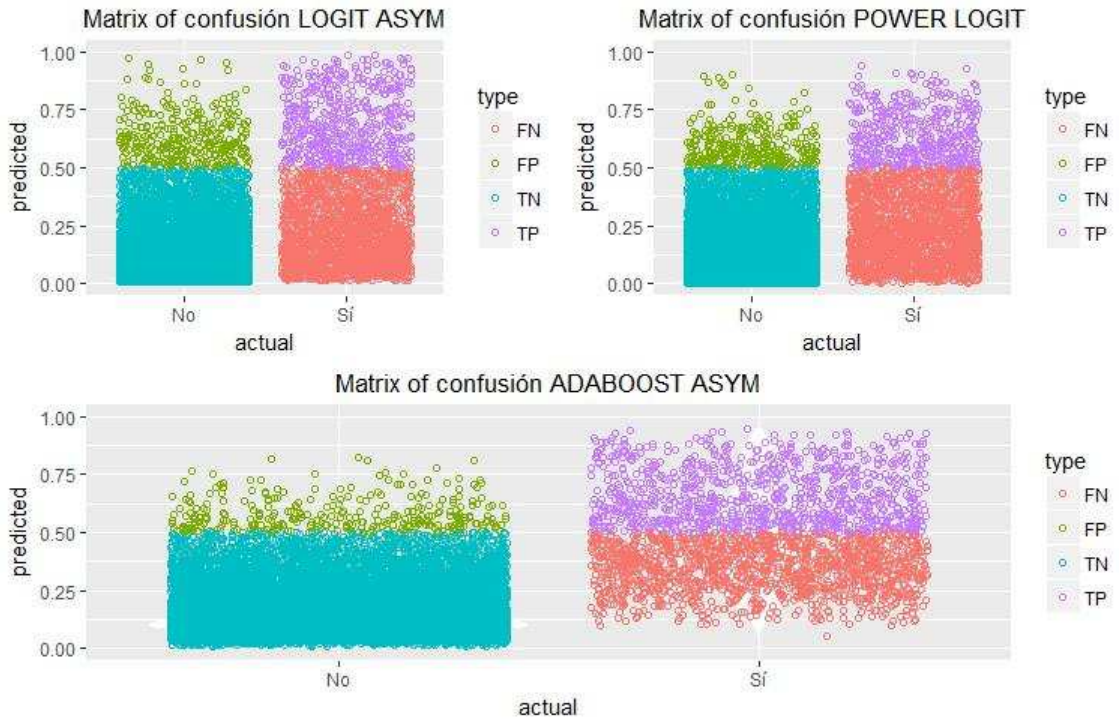


Figura 37: Gráficas de matrices de confusión para los modelos de regresión logística asimétrica y el algoritmo Adaboost asimétrico

Cuadro 19: Matrices de confusión comparativa para los modelos de regresión logística asimétrica y el algoritmo Adaboost desbalanceado.

Logística Asym, K=0.02

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	540	312
	No fugo	1855	21383

Manuel

Power Logit, λ=5/2

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	447	306
	No fugo	1948	21389

Adaboost Asym

		Clase real	
		Sí fugó	**No fugó**
Predicción	**Sí fugó**	975	318
	No fugo	1420	21377

Fuente: Elaboración propia

La figura 37 muestra que en los tres modelos los verdaderos positivos identificados con color morado aumentaron (especialmente en el Logit Asym y Adabost Asym), los falsos positivos identificados con color verde disminuyeron considerablemente (principalmente en el Adaboost Asym), en los modelos Logit Asym y Power logit hay más concentración de falsos negativos (color rojo). Los 3 modelos parecen identificar correctamente a los verdaderos positivos. Hasta acá se sospecha que el Adaboost Asym es el de mayor rendimiento.

En el cuadro 19 se muestra las matrices de confusión para cada modelo, identifica con más precisión a los que realmente fugaron, es decir cuenta con la mayor cantidad de verdaderos positivos. En cuanto a los verdaderos negativos (los que realmente no fugan), los tres modelos identificaron en forma similar. Para identificar el mejor modelo se utilizó estas tablas y más adelante se presentó las métricas de desempeño.

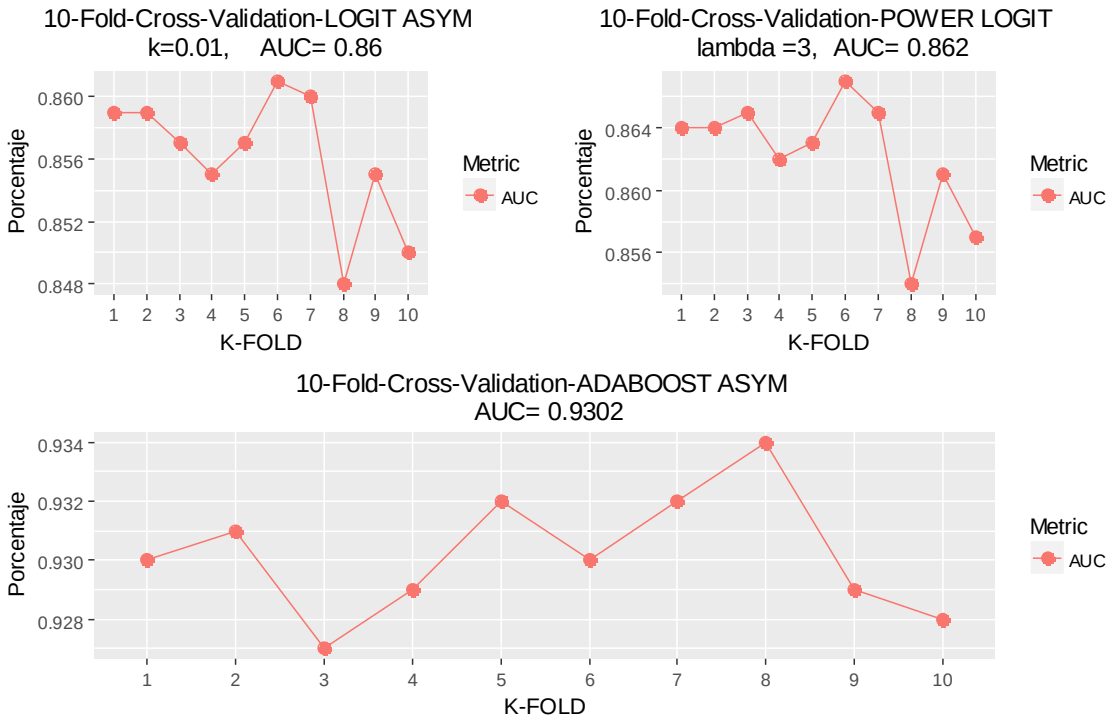


Figura 38: Validación cruzada en el AUC para los modelos de regresión logística asimétrica y el algoritmo Adaboost asimétrico

La figura 38 muestra la validación cruzada para el AUC, para los modelos propuestos, para cada método se realizó 10 particiones, cada punto graficado en cada figura corresponde a cada uno de los porcentaje del área bajo la curva (AUC) al dejar una de las 10 particiones

fuera del conjunto de entrenamiento y utilizarlo como conjunto de datos de prueba. Al promediar los 10 puntos y obtener un AUC general se observa que algorimto Adaboost fue superior y que no hubo mucha diferencia entre el Logit Asym y el Power Logit, las conclusiones mediante estos resultados son semejantes a los obtenidos con el método de retención.

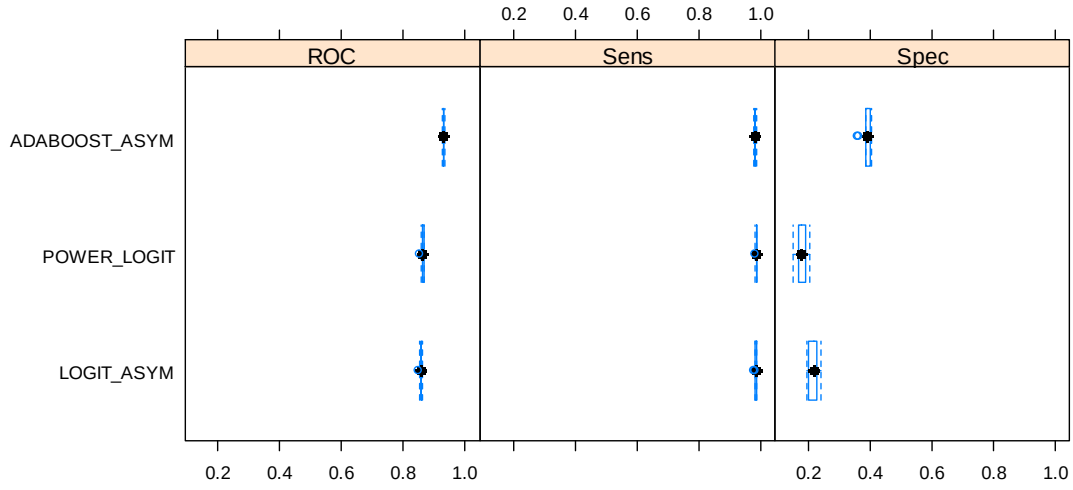


Figura 39: Validación cruzada de diferentes métricas para los modelos logísticos asimétricos y el algoritmo Adaboost asimétrico.

En la figura 39 se observa que mediante la validación cruzada, la sensibilidad (Recall) fue similar en los 3 modelos, es decir presentan cantidad similar de falsos negativos, en cuanto a la especificidad el algoritmo Adabost Asym fue superior, es decir presenta menor cantidad de falsos positivos. El AUC mediante las curvas ROC son los mismos que la figura 3.

Cuadro 20: Medidas comparativas de desempeño de los 3 modelos ajustados

Medidas	**Logit Asym**	**Power Logit**	**Adaboost Asym**
Exactitud (Accuracy)	0.91005	0.90643	0.92785
Tasa de Error	0.08995	0.09357	0.07215
Precisión	0.63380	0.59363	0.75406
Recall	0.22547	0.18664	0.40710
F measure	0.33261	0.28399	0.52874

Fuente: Elaboración propia

El cuadro 20 muestra que el modelo Adaboost Asym es superior a los modelos de regresión logística asimétrica en todas las métricas de desempeño, con lo cual hasta ahora se llegó a la

conclusión de que este modelo es el mejor para el caso de la empresa de telefonía móvil estudiado. Como comparación final se analizó las curvas ROC para dar la conclusión final sobre el mejor modelo.

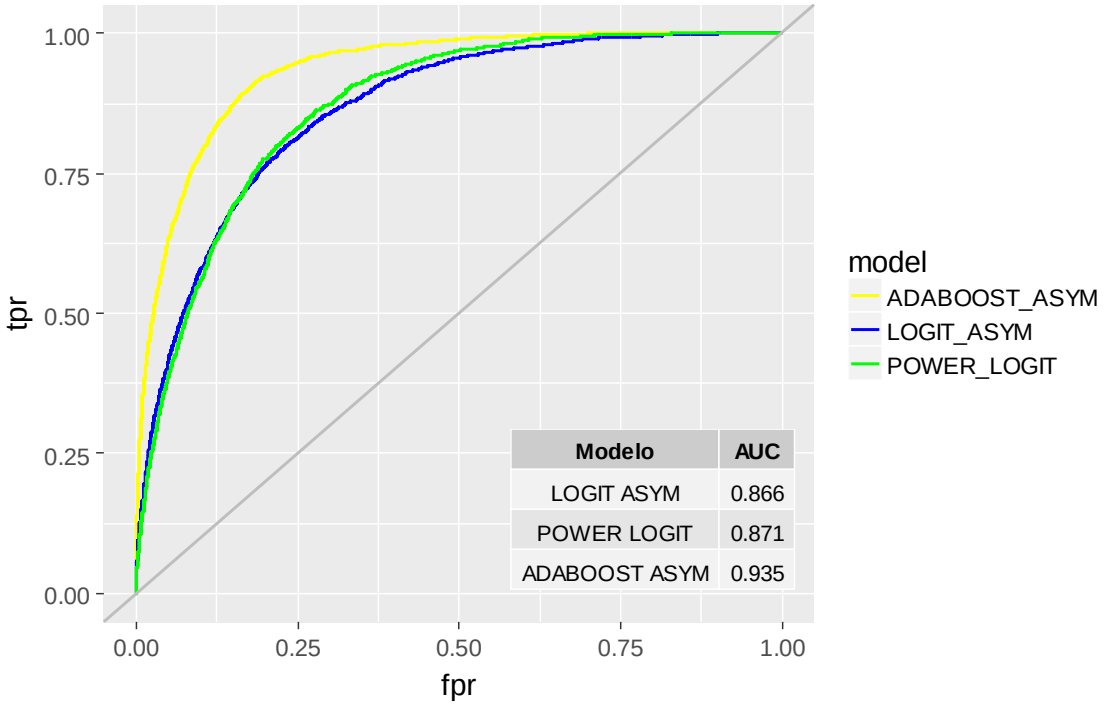


Figura 40: Comparación del AUC en los modelos de regresión logística asimétrica y el algoritmo Adaboost asimétrico

En la figura 40 se muestra la comparación final entre el algoritmo Adaboost y la regresión logística. Se observa que el algoritmo Adaboost fue superior, el AUC sobrepasa el 93%, no habiendo mucha diferencia en los modelos logísticos ajustados. Con estos resultados y con las medidas vistas anteriormente se concluye que para los datos de la empresa de telefonía la mejor técnica de clasificación es el algoritmo Adaboost, a esta conclusión también se llegó al analizar los modelos mediante los métodos de muestreo.

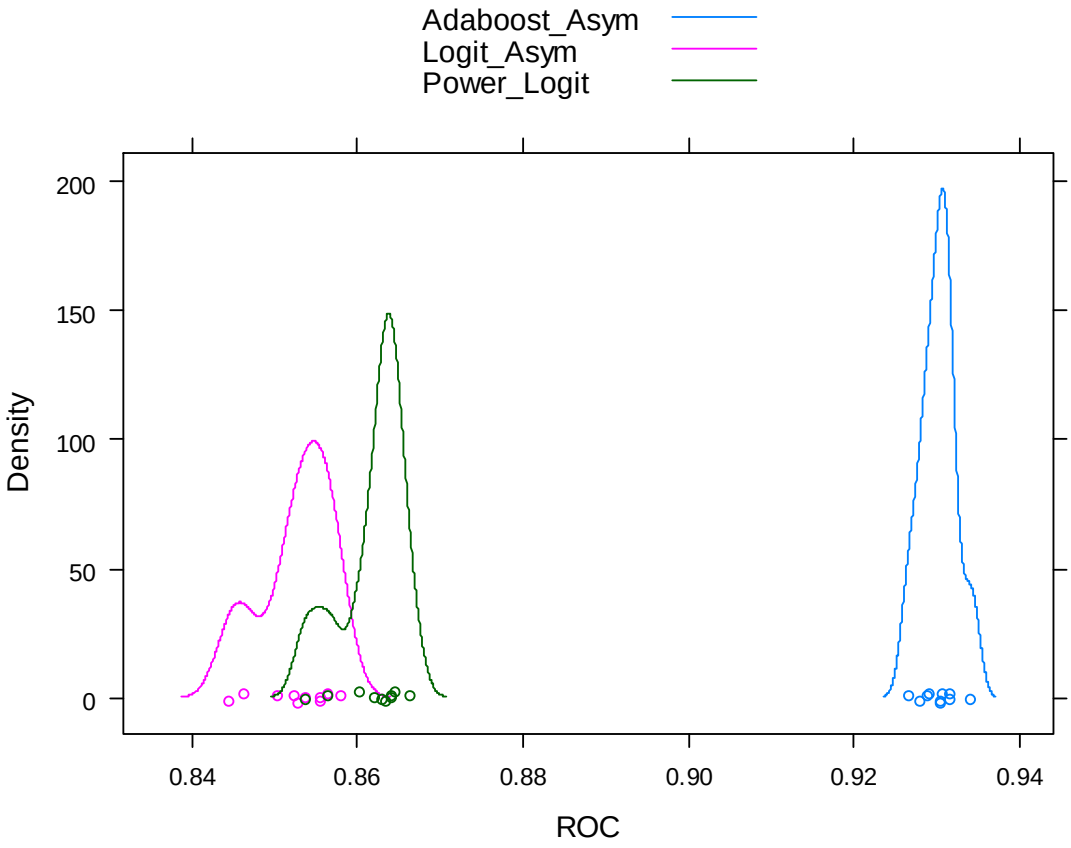


Figura 41: Comparación de densidades en los modelos de regresión logística y el algoritmo Adaboost mediante ajuste de algoritmo y función

En la figura 41 se comparan las densidades de todos los modelos mediante ajuste de algoritmo y función, el modelo utilizando el algoritmo Adaboost tienen mayor precisión en la curva ROC y menor variabilidad. En la regresión logística, el Power Logit tiene mayor precisión, sin embargo la variabilidad es similar al modelo logit asimétrico, ante estas evidencias se concluye que el modelo óptimo para la predicción de fuga de clientes en la empresa es el algoritmo Adaboost.

V. CONCLUSIONES

Con el fin de comparar la eficiencia de cada modelo y poder elegir entre cuál de las dos propuestas (logística o Adaboost) sería el más adecuado para predecir la fuga de clientes de la empresa de telefonía, se realizó dos procedimientos de clasificación, el primer procedimiento fue mediante métodos de muestreo (sub-muestreo, sobre-muestreo y SMOTE) para equilibrar la categoría de éxito desbalanceada. En la regresión logística se obtuvo resultados semejantes en los métodos para cada una de las métricas de desempeño adecuadas en datos desbalanceados, en cuanto a la precisión no se obtuvieron resultados muy altos, sin embargo para el Recall (sensibilidad) los resultados fueron muy buenos, especialmente en el sub-muestreo (0.8208) y sobre-muestreo (0.822). En cuanto a la medida principal de desempeño, se obtuvieron resultados de AUC prácticamente semejantes en los 3 métodos de muestreo (0.871 aprox.) concluyendo que para la regresión logística cualquiera de los métodos de muestreo es adecuado.

En cuanto al algoritmo Adaboost, también se subsanó el desbalance mediante métodos de muestreo, se obtuvo una buena precisión con el método sobre-muestreo (0.7548), con los otros dos métodos la precisión fue baja. En cuanto al Recall, el método más óptimo fue sub-muestreo (0.9039) y un F-measure (media armónica entre la precesión y el Recall) similar para cada método de muestreo, sobresaliendo ligeramente el SMOTE (0.596). El desempeño final se midió con el AUC, obteniendo resultados muy similares con los 3 métodos de muestreo (0.97 aprox.), concluyendo que para los datos de la empresa, la modelación con el algoritmo Adaboost da resultados similares con cualquier método de muestreo.

Tanto para la regresión logística como para el algoritmo Adaboost, los modelos mediante métodos de muestreo fueron probados mediante la validación cruzada, eligiendo 10 iteraciones para cada método, y promediando en una sola medida de desempeño las 10 iteraciones, los resultados arrojaron medidas de desempeño muy similares al método de

retención, donde la medida principal (AUC) fue de 0.86 aprox. en la regresión logística y 0.93 aprox. en el algoritmo Adaboost.

En el primer procedimiento al comparar la regresión logística y el algoritmo Adaboost mediante métodos de muestreo, se demostró que el algoritmo Adaboost tuvo mejor rendimiento en casi todas las medidas de desempeño, por lo tanto se concluye que si desea realizar modelos de clasificación para predecir la fuga de clientes en la empresa de telefonía móvil propuesta, la mejor técnica mediante métodos de muestreo es el algoritmo Adaboost.

Como segundo procedimiento para comparar la eficiencia de cada modelo y elegir entre el más adecuado para clasificar y predecir la fuga de clientes de la empresa de telefonía se utilizó técnicas de ajuste a nivel de función o de algoritmo, es decir no se utilizó métodos de muestreo sino que se modificó la función logística y el algoritmo Adaboost a efecto de alterar los pesos de la parte desbalanceada y equilibrar internamente las categorías.

En la regresión logística se ajustó la función de distribución acumulada con un término K (Kappa) llamando al modelo Logit asimétrico (Logit Asym), el valor de K es un valor el cual puede considerarse como una variable, teniendo esta un impacto importante en la probabilidad de los que fugan. Se probó diferentes valores de K, y para elegir el valor óptimo se tuvo en cuenta la suma de los pesos de los que fugan y los que no, el AIC y principalmente el AUC. El valor óptimo elegido fue K=0.02, con esto la razón de la suma de pesos bajó de ser de 8.9 (los que no fugan son 8.9 veces más que los que fugan) a 5.4. El AIC fue 27438.96 y el AUC equivalente a 0.866.

Otro modelo ajustado en regresión logística fue el modelo llamado Power Logit el cual utiliza un término λ (lambda) en la función de distribución acumulada, también se probó diferentes valores de λ y se eligió el valor óptimo teniendo en cuenta que para valores desbalanceados donde la categoría de éxito es minoritaria es recomendable un valor de λ mayor a 1, aparte de ello se evaluó el AIC y el AUC. El valor óptimo elegido fue λ=2.5, con este valor el AIC fue 26785.23 el fue menor a los demás valores propuestos de lambda y el

AUC equivalente a 0.871, teniendo también un mejor desempeño en comparación con los demás valores de lambda propuestos.

En cuanto al Algoritmo Adaboost desbalanceado (Adaboost Asym), se ajustó el peso de la clase minoritaria cuyo costo de clasificación fue errónea, es decir se impuso mayor costo a los falsos positivos. Este algoritmo de clasificación fue comparado con las técnicas de regresión logística ajustada.

En los dos modelos de regresión logística ajustada (Logit Asym y Power Logit) así como en el Adaboost Asym se realizó la validación cruzada, donde los modelos logísticos arrojaron valores similares para el AUC (0.86 y 0.862), siendo superados por el algoritmo Adaboost Asym (0.9302), estos resultados fueron muy semejantes a los obtenidos con el método de retención, concluyendo que los modelos se ajustan bien a los datos.

Parte del procedimiento también consistió en identificar las variables significativas y el grado de importancia de cada una de ellas. En los modelos logísticos, las variables significativas y de mayor importancia fueron el número de reclamos, días de deuda, tipo de reclamo, y el Rol del cliente. Estas variables son las determinantes para entender el comportamiento y la decisión del cliente de fugar o no. En cuanto a algoritmo Adaboost, se identificó que las variables relevantes para la clasificación de clientes fueron el tipo de reclamo, ROL, comunidad, número de reclamos, días de deuda y tipo de cliente. Cabe recalcar que las otras variables también fueron importantes para formular los modelos de clasificación, sin embargo su aporte o grado fue menor.

Al comparar las medidas de desempeño de los dos modelos logísticos ajustados con el algoritmo adaboost asimétrico se identificó que tanto en precisión, Recall y F-score, el algoritmo Adaboost fue superior (0.754, 0.407 y 0.528 respectivamente), de la misma manera con la medida AUC (0.93) el algoritmo Adaboost llevó la ventaja, concluyendo que si se desea crear un modelo para la empresa de telefonía propuesta sin utilizar muestreo el algoritmo Adaboost asimétrico es el mejor en comparación con la regresión logística.

Como decisión final se concluye que tanto a nivel de algoritmo como a nivel de muestreo, el algoritmo Adaboost es superior y más eficiente que la regresión logística para la clasificación de clientes que fugan en la empresa de telefonía propuesta. Este modelo es fácil de implementar a través del software libre R u otro software comercial.

VI. RECOMENDACIONES

1. A pesar de que el algoritmo Adaboost es muy eficiente, computacionalmente puede ser muy costoso en relación al tiempo de procesamiento, por lo que se recomienda realizar un método de paralelismo o particionamiento de datos, esto a razón de que el tiempo de procesamiento hasta crear el modelo y la realización de la validación cruzada es muy extenso.

2. El umbral de clasificación para esta investigación fue 0.5, esto a efectos de que todos los modelos tengan el mismo punto de corte de clasificación, entonces se recomienda probar con otros puntos de corte para ajustar al máximo los modelos.

3. Las curvas ROC no son al 100% seguras puesto que pueden ser engañosas si no se tiene cuidado tal como proporcionar resultados de rendimiento excesivamente optimistas en datos altamente asimétricos, por lo tanto como complemento se recomienda utilizar el PPROC (área bajo la curva entre la Precisión y el Recall), puesto que puede ser una medida más implícitamente informativa.

4. Se recomienda probar el modelo elegido con otros modelos, a través de otros algoritmos de aprendizaje, esto a razón de identificar a cuál de los modelos trabajados se aproxima mejor el desempeño de los indicadores.

5. Aparte del R, utilizar otro software complementario como Python, como ejemplo se encuentra implementado el adaMEC (Adaboost calibrado) el cual puede dar un rendimiento diferente.

6. Para comparar los modelos también se recomienda utilizar métodos de estimación de los errores mediante boostrap.

VII. REFERENCIAS BIBLIOGRÁFICAS

Barrientos, F. 2011. Diseño e implementación de una metodología de predicción de fuga de clientes en una compañía de telecomunicaciones. Memoria para optar al título de ingeniero civil industrial. Departamento de Ingeniería Industrial. Universidad de Chile.

Barrientos, F; Ríos, S. 2013. Aplicación de Minería de Datos para Predecir Fuga de Clientes en la Industria de las Telecomunicaciones. Revista Ingeniería de Sistemas. Volumen XXVII.

Bazán J. 2017. New distributions and new links function to binary regression with unbalanced data. Seminário - Universidad Nacional Agraria La Molina.

Breiman, L. 1998. Arcing Classifier, Annals of Statistics, vol. 26, no. 3, pp. 801-849

Brownlee, J. 2015. 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset. Machine Learning Process. Consultado el 13 de nov. 2016. Disponible en: http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Brownlee J. 2017. Difference between classification and Regression in Machine Learning. Consultado el 24 de febrero del 2018. Disponible en: https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/

Cha, Z.; Yunqian, M. 2012. Ensemble Machine Learning. Methods and Applications. Dordrecht New York.

Chapman P.; Clinton J.; Kerber R.; Khabaza T.; Reinartz T.; Shearer C.; Wirth R. 2000. CRISP-DM 1.0: Step-by-step data mining guide. USA: SPSS Inc., CRISP-DM Consortium.

Chiu, Y. 2015. Machine Learning with R Cookbook. Birmingham B3 2PB, UK. ISBN 978-1-78398-204-2

Dávila, N.; García, D; Pérez, J; Gómez, E. 2015. An Asymmetric Logit Model to explain the likelihood of success inacademic results. Revista de Investigación Educativa, 33(1), 27-45.

Davis J. 2007. Magic Numbers for Sales Management. Wiley. ISBN 8126513543. Pp. 107-109

Duran P. 2005. Los datos perdidos en estudios de investigación ¿son realmente datos perdidos? Archivos argentinos de pediatría. versión On-line ISSN 1668-3501. Consultado el 01 de mar. 2018. Disponible en: http://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S0325-00752005000600015

Freund, Y.; Schapire, R. 1996. Experiments with a New Boosting Algorithm. Machine Learning: Proceedings of the Thirteenth International Conference. Murray Hill, NJ 07974-0636.

Hadad, A.; Evin, D.; Drozdowicz, B. s.f. Modelo para el Tratamiento de Datos Desbalanceados basado en Redes Neuronales Autoorganizadas. Universidad Nacional de Entre Ríos.

Haibo, H.; Yunqian, M. 2013. Imbalanced Learning: Foundations, Algorithms, and Applications. Hoboken, New Jersey, John Wiley & Sons.

Hair, J.; Anderson R.; Tatham R.; Black W. 1999. Análisis Multivariante (5ta edición). Prentice Hall Iberia , Madrid , ISBN : 84-8322-035-0.

Hosmer, D.; Lemeshow, S. 2000. Applied Logistic Regression (2nd edición). Wiley. ISBN 0-471-35632-8.

Kearns, M. 1988. Thoughts on Hypothesis Boosting, Manuscript unpublished (Machine Learning class project, December 1988).

Komori, O.; Eguchi, S.; Ikeda, S.; Okamura, H.; Ichinokawa, M.; Nakayama, S. 2015. An asymmetric logistic regression model for ecological data. Methods in Ecology and Evolution 2016, 7, 249–260.

Kriegler, B.; Berk, R. 2007. Estimación de las personas sin hogar en Los Ángeles (estimación espacial en pequeñas áreas).

Kunal J. 2016. Practical Guide to deal with Imbalanced Classification Problems in R. Analytics Vidhya. Learn Everything About Analytics. Consultado el 13 de agt. 2016. Disponible en: https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

Lang, J. Predictors tutorial, Bioinformatic Department Projects. Consultado el 8 de enero 2017. Disponible en: http://docs.bioinfo.cipf.es/projects/1/wiki/Predictors_methods

Lejenue, M. 2001. Measuring the impact of data mining on Churn Management. Research Internet, 11, pp. 374-384.

Liu, X. Y.; Wu, J.; Zhou, Z. H. 2006. Exploratory undersampling for class-imbalance learning. Washington, USA, IEEE Computer Society.

Mason, L.; Baxter, J.; Bartlett, P.; Frean, M. 2000. Boosting Algorithms as Gradient Descent. pp. 512-518.

Moya, M. 2016. ¿Qué es el Machine Learning? Consultado el 25 de Nov. 2016. Disponible en: http://jarroba.com/que-es-el-machine-learning/

Muñoz, J. 2016. Cómo reducir el Churn rate de tu SaaS. Consultado el 13 de sept. 2016. Disponible en: https://tribescale.com/es/blog/como-reducir-el-Churn-rate-de-tu-saas/

Neslin, S.; Gupta, S.; Kamakura, W.; Lu, J.; Mason, C. 2006. Defection Detection: Measuring and Understanding the Predictive Accuracy of Customer Churn Models, Journal of Marketing Research, vol XLIII, pp. 204-211.

Obregón, S. J (2016). Desarrollo de una Herramienta de Diagnóstico de Fallos en Motores de Inducción Mediante la técnica Adaboost. Trabajo fin de Máster para obtener el título de Ingeniero Industrial. Universidad de Valladolid.

Osiptel 2017. Reporte estadístico. Se intensifica desconcentración del mercado de telefonía móvil. Consultado el 25 de Febr. 2018. Disponible en: https://www.osiptel.gob.pe/Archivos/Publicaciones/reporteestadistico_abril2017/files/assets/basic-html/index.html#1

Pérez, V. P. 2014. Modelo de predicción de Fuga de clientes de telefonía Móvil Post pago. Memoria para Optar al Título de Ingeniero Civil Industrial. Departamento de Ingeniería Industrial. Universidad de Chile.

Polikar, R. 2006. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine, 6(3):21–45.

Rendón M.; Acosta J. 2006. Estudio sobre el estado de las soluciones ICT y de los casos prácticos de aplicación de la minería de datos a nivel mundial en al menos 5 casos representativos. Proyecto de Grado para optar el título de Ingeniero de Sistemas. Universidad EAFIT. Medellín.

Rohrer, B. 2016. Cómo elegir algoritmos para Aprendizaje automático de Microsoft Azure. Consultado el 12 de sept. 2017. Disponible en: https://docs.microsoft.com/es-es/azure/machine-learning/machine-learning-algorithm-choice

Rouse M. 2008. What type of data mining has your organization embraced? AWS analytics tools help make sense of big data. Consultado el 01 de mar. 2018. Disponible en: http://searchsqlserver.techtarget.com/definition/data-mining

Sancho, C. 2015. "Introducción al Aprendizaje Automático". Universidad de Sevilla. Consultado el 18 de sept. 2016. Disponible en: http://www.cs.us.es/~fsancho/?e=75.

Santos, M.; Azevedo, C. 2005. Data Mining – Descoberta de Conhecimento em Bases de Dados. FCA Publisher.

Viada C.; Bouza C.; Ballesteros J.; Fors M.; Robaina M.; Uranga R. 2016. Revisión sistemática de los métodos de imputación de datos faltantes. In book: Toma de decisiones en la salud humana, medio ambiente y desarrollo humano, Edition: Tomo 2, Chapter: Capítulo 12. Consultado el 01 de mar. 2018. Disponible en: https://www.researchgate.net/publication/289248594_REVISION_SISTEMATICA_DE_LOS_METODOS_DE_IMPUTACION_DE_DATOS_FALTANTES

Viola, P.; Jones, M. 2004. Robust Real-Time Face Detection. International Journal of Computer Vision.

Zhi-Hua, Z. 2012. Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC. p. 23. ISBN 978-1439830031.

ANEXOS

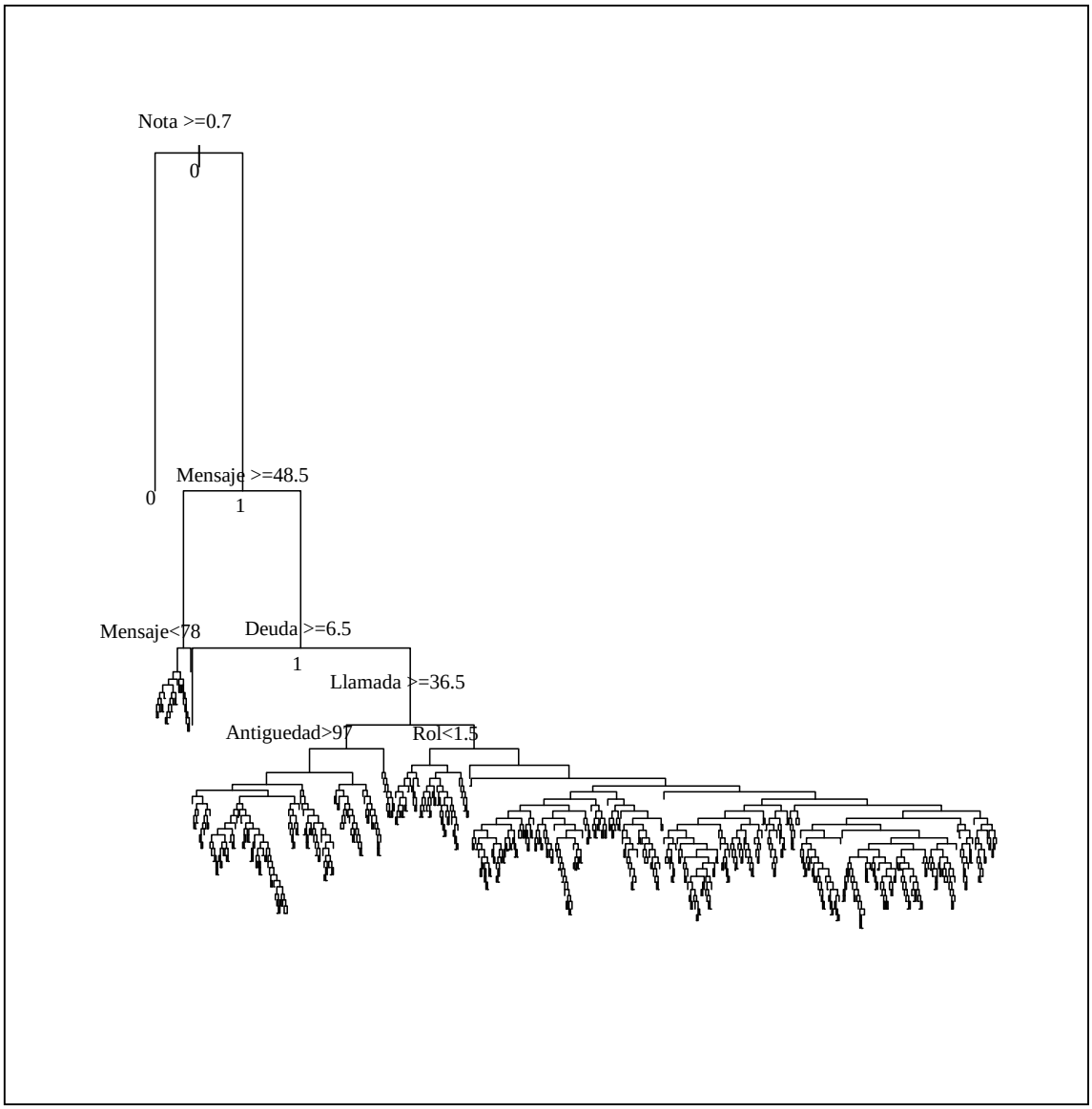


Figura 38: Árbol inicial que utiliza el modelo Adboost en 100 interacciones

Códigos en R

```
#=================================================#
#=====SELECCIÓN DE MUESTRAS DE ENTRENAMIENTO Y DE PRUEBA=====#
#=================================================#

# Conversión de factores
#------------------------
MOVIL<-read.delim("clipboard")
MOVIL$Kilobytes<-rescaler(MOVIL$Kilobytes,type="sd")
MOVIL$Fuga<-as.factor(MOVIL$Fuga)
MOVIL$Tipo.Reclamo<-as.factor(MOVIL$Tipo.Reclamo)
MOVIL$Sexo<-as.factor(MOVIL$Sexo)
MOVIL$Procedencia<-as.factor(MOVIL$Procedencia)
MOVIL$ROL<-as.factor(MOVIL$ROL)
MOVIL$Comunidad<-ordered(as.factor(MOVIL$Comunidad))
MOVIL$Canal<-as.factor(MOVIL$Canal)
MOVIL$Cliente<-ordered(as.factor(MOVIL$Cliente))

# Muestras de entrenamiento y de prueba
#---------------------------------------
set.seed(1983)
#Selección datos de entrenamiento
tamano.total <- nrow(MOVIL)
tamano.entreno <- round(tamano.total*0.70)
datos.indices <- sample(1:tamano.total , size=tamano.entreno)
datos.entreno <- MOVIL[datos.indices,]

#Selección datos de prueba
datos.test<-MOVIL[-datos.indices,]
table(datos.entreno[,1])
```

```
#-------------------------------------------------------------------------------------------------#
# ----------------------------------Balance de datos en muestreo-------------------------------#
#-------------------------------------------------------------------------------------------------#

#  Sub-muestreo (Down)
#--------------------

library(caret)

set.seed(1983)

down_train <- downSample(x = datos.entreno[, -1],y = datos.entreno$Fuga)

table(down_train$Class)

#  Subre-muestreo (Up)
#--------------------

set.seed(1983)

up_train <- upSample(x = datos.entreno[, -1],y = datos.entreno$Fuga)

table(up_train$Class)

#  Sobre-muestreo de minorias sintéticas (SMOTE)
#-----------------------------------------------

library(DMwR)

set.seed(1983)

smote_train <- SMOTE(Fuga ~.,data  = datos.entreno)

head(smote_train)

table(smote_train$Fuga)


#=================================================#
#=================CREACIÓN DE MODELOS=============#
#=================================================#

# Cargando Paquetes
#------------------

library(caret) # for model-building

library(DMwR) # for smote implementation

library(purrr) # for functional programming (map)

library(pROC) # for AUC calculations

library(fastAdaboost)
```

```
library(devtools)
library(parallel)
library(doParallel)

# INICIO DE PROCESO PARALELO
#---------------------------
cluster <- makeCluster(4) # convention to leave 1 core for OS
registerDoParallel(cluster)

#-------------------------------------------------------------------------------------------------------#
# ------------------------ Modelos Adaboost para cada tipo de muestreo ----------------------#
#-------------------------------------------------------------------------------------------------------#
fiveStats = function(...) c (twoClassSummary(...), defaultSummary(...))

# Modelo original
#--------------------
orig_cv.ctrl = trainControl( method = "cv", number = 10 ,
                             classProbs = TRUE,
                             summaryFunction = fiveStats,
                             allowParallel = TRUE)
set.seed(1983)
orig_adaboost <- train(Fuga ~ .,
                       data = datos.entreno,
                       method = "adaboost",
                       trControl = orig_cv.ctrl,
                       metric = "ROC")
# Modelo down
#-----------------
down_cv.ctrl = trainControl( method = "cv", number = 10 ,
                             classProbs = TRUE,
                             summaryFunction = fiveStats,
                             sampling = "down",
```

```
                        allowParallel = TRUE)
set.seed(1983)
down_adaboost <- train(Fuga ~ .,
                       data = datos.entreno,
                       method = "adaboost",
                       trControl = down_cv.ctrl,
                       metric = "ROC")
# Modelo up
#-------------------
up_cv.ctrl = trainControl( method = "cv", number = 10 ,
                           classProbs = TRUE,
                           summaryFunction = fiveStats,
                           sampling = "up",
                           allowParallel = TRUE)
set.seed(1983)
up_adaboost <- train(Fuga ~ .,
                     data = datos.entreno,
                     method = "adaboost",
                     trControl = up_cv.ctrl,
                     metric = "ROC")


# Modelo smote
#-------------
smote_cv.ctrl = trainControl( method = "cv", number = 10 ,
                              classProbs = TRUE,
                              summaryFunction = fiveStats,
                              sampling = "smote",
                              allowParallel = TRUE)
set.seed(1983)
smote_adaboost <- train(Fuga ~ .,
                        data = datos.entreno,
```

```
                    method = "adaboost",
                    trControl = smote_cv.ctrl,
                    metric = "ROC")

#-------------------------------------------------------------------------------------------------#
# ------------------------ Modelos Logísticos para cada tipo de muestreo ----------------------#
#-------------------------------------------------------------------------------------------------#

# Modelo original
#------------------------------
origL_cv.ctrl = trainControl( method = "cv", number = 10 ,
                        classProbs = TRUE,
                        summaryFunction = fiveStats )
set.seed(1983)
orig_logis<- train(Fuga ~ .,
                data = datos.entreno,
                method = "glm",
                family=binomial(link=logit),
                trControl = origL_cv.ctrl,
                metric = "ROC")

# Modelo down
#-----------------------------
downL_cv.ctrl = trainControl( method = "cv", number = 10 ,
                        classProbs = TRUE,
                        summaryFunction = fiveStats,
                        sampling = "down")
set.seed(1983)
down_logis <- train(Fuga ~ .,
                 data = datos.entreno,
                 method = "glm",
                 family=binomial(link=logit),
                 trControl = downL_cv.ctrl,
                 metric = "ROC")
```

```
# Modelo up
#---------------------------
upL_cv.ctrl = trainControl( method = "cv", number = 10 ,
                            classProbs = TRUE,
                            summaryFunction = fiveStats,
                            sampling = "up")
set.seed(1983)
up_logis <- train(Fuga ~ .,
                  data = datos.entreno,
                  method = "glm",
                  family=binomial(link=logit),
                  trControl = upL_cv.ctrl,
                  metric = "ROC")

# Modelo smote
#---------------------------
smoteL_cv.ctrl = trainControl( method = "cv", number = 10 ,
                               classProbs = TRUE,
                               summaryFunction = fiveStats,
                               sampling = "smote")
set.seed(1983)
smote_logis <- train(Fuga ~ .,
                     data = datos.entreno,
                     method = "glm",
                     family=binomial(link=logit),
                     trControl = smoteL_cv.ctrl,
                     metric = "ROC")

#FIN PROCESO PARALELO
#--------------------
stopCluster(cluster)
registerDoSEQ()
```

```
#-------------------------------------------------------------------------------------------------------#
# -------------------------Modelos de Regresión Logística Asimétrica-------------------------#
#-------------------------------------------------------------------------------------------------------#

#Link para la regresión logística Asimétrica (Logit Asym)
#-----------------------------------------------------------------
link.eta<-function(k){
  linkfun <- function(mu){
    A=(1+k)*mu-k
    log(A/(1-mu))
  }
  linkinv <- function(eta)(exp(eta)+k)/(1+exp(eta)+k)
  mu.eta <- function(eta)exp(eta)/(1+exp(eta)+k)^2
  valideta <- function(eta) TRUE
  link <- paste0("link.eta(", k, ")")
  structure(list(linkfun = linkfun, linkinv = linkinv,
                 mu.eta = mu.eta, valideta = valideta, name = link),class = "link-glm")
}

#Link para la regresión Power  Logit
#-----------------------------------------
library(BinaryEPPM)
powerlogit(power = 1)
link.eta1<-function(lambda){
  linkfun<-function (mu)
  {
    wkv <- exp(log(mu)/lambda)
    log(wkv) - log(1 - wkv)
  }
  linkinv<-function (eta)
  {
    1/((1 + exp(-eta)))^lambda
  }
  mu.eta<-function (eta)
  {
    lambda * exp(-eta)/((1 + exp(-eta)))^(lambda + 1)
```

```
  }
  valideta<-function (eta) TRUE
  structure(list(linkfun = linkfun, linkinv = linkinv,
                 mu.eta = mu.eta, valideta = valideta, name = powerlogit(1)),class = "link-glm")
}
#Construcción de Modelo Logit Asym
#-----------------------------------------
K=0.02
fiveStats = function(...) c (twoClassSummary(...), defaultSummary(...))
cv.ctrl = trainControl( method = "cv", number = 10 ,
                        classProbs = TRUE,
                        summaryFunction = fiveStats )
set.seed(1983)
Logis.Asy = train ( Fuga ~ .,
                    data = datos.entreno ,
                    method = "glm",
                    family=binomial(link=link.eta(K)),
                    trControl = cv.ctrl,
                    metric = "ROC")

#Construcción de Modelo Power Logit
#-----------------------------------------
lambda=2
set.seed(1983)
Power.logit = train (Fuga ~ .,
                     data = datos.entreno ,
                     method = "glm",
                     family=binomial(link=link.eta1(lambda)),
                     trControl = cv.ctrl,
                     metric = "ROC")

#-------------------------------------------------------------------------------------------------#
# ------------------------------ Modelos Adaboost asimétrico (weight) ------------------------#
#-------------------------------------------------------------------------------------------------#
#Inicio forma paralela
#--------------------
```

```
cluster <- makeCluster(4) # convention to leave 1 core for OS
registerDoParallel(cluster)
set.seed(1983)
ctrl <- trainControl(method = "cv",
                     number = 10,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     allowParallel = TRUE)

model_weights <- ifelse(datos.entreno$Fuga == "Sí",
                        (1/table(datos.entreno$Fuga)[1]) * 0.5,
                        (1/table(datos.entreno$Fuga)[2]) * 0.5)
library(fastAdaboost)
# Build weighted model
weighted_fit <- train(Fuga ~ .,
                      data = datos.entreno,
                      method = "adaboost",
                      verbose = FALSE,
                      weights = model_weights,
                      metric = "ROC",
                      trControl = ctrl)

#Fin forma Paralela
#------------------------
stopCluster(cluster)
registerDoSEQ()
```