



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

**Modelo de predicción de deserción de clientes de
servicio postpago de telefonía móvil en empresas de
telecomunicaciones**

TESIS

Para optar el Grado Académico de Doctor en Ingeniería de
Sistemas e Informática

AUTOR

David Andres FREIRE VERA

ASESOR

Dr. Ciro RODRIGUEZ RODRIGUEZ

Lima, Perú

2025



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Freire, D. (2025). *Modelo de predicción de deserción de clientes de servicio postpago de telefonía móvil en empresas de telecomunicaciones*. [Tesis de doctorado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	David Andres Freire Vera
Tipo de documento de identidad	Cédula de identidad
Número de documento de identidad	EC / 0923799001
URL de ORCID	https://orcid.org/0009-0001-4930-8888
Datos de asesor	
Nombres y apellidos	Ciro Rodriguez Rodriguez
Tipo de documento de identidad	DNI
Número de documento de identidad	06445707
URL de ORCID	https://orcid.org/0000-0003-2112-1349
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Carlos Edmundo Navarro Depaz
Tipo de documento	DNI
Número de documento de identidad	08482690
Miembro del jurado 1	
Nombres y apellidos	Luzmila Elisa Pró Concepción
Tipo de documento	DNI
Número de documento de identidad	08862360
Miembro del jurado 2	
Nombres y apellidos	Luis Soto Soto
Tipo de documento	DNI
Número de documento de identidad	22409532
Miembro del jurado 3	
Nombres y apellidos	Ciro Rodriguez Rodriguez
Tipo de documento	DNI

Número de documento de identidad	06445707
Datos de investigación	
Línea de investigación	C.0.5.8. Inteligencia Artificial
Grupo de investigación	Tecnologías de la Información aplicados a la ciencia de los datos - ITDATA
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	País: Ecuador Provincia: Guayas Ciudad: Guayaquil Latitud: -2.223333 Longitud: -79.8875
Año o rango de años en que se realizó la investigación	2022 - 2024
URL de disciplinas OCDE	Ingeniería de Sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE DOCTOR EN INGENIERÍA DE SISTEMAS E INFORMÁTICA

A los diez (10) días del mes de abril de 2025, siendo las 11:00 am., se reunieron en el Auditorio, Profesor: Alfredo Celso Alva Bravo, el Jurado de Tesis conformado por los siguientes docentes:

Dr. Carlos Edmundo Navarro Depaz (Presidente)
Dra. Luzmila Elisa Pró Concepción (Miembro)
Dr. Luis Soto Soto (Miembro)
Dr. Ciro Rodríguez Rodríguez (Miembro asesor)

Se inició la Sustentación invitando al candidato a Doctor **DAVID ANDRES FREIRE VERA**, para que realice la exposición oral de la tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas e Informática, siendo la Tesis intitulada:

“MODELO DE PREDICCIÓN DE DESERCIÓN DE CLIENTES DE SERVICIO POSTPAGO DE TELEFONÍA MÓVIL EN EMPRESAS DE TELECOMUNICACIONES”

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

..... DIECISIETE (17) MUY BUENO

Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga al Magister **DAVID ANDRES FREIRE VERA** el Grado Académico de Doctor en Ingeniería de Sistemas e Informática.

Siendo las 12:10 horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.

Dr. Carlos Edmundo Navarro Depaz
(Presidente)

Dra. Luzmila Elisa Pró Concepción
(Miembro)

Dr. Luis Soto Soto
(Miembro)

Dr. Ciro Rodríguez Rodríguez
(Miembro Asesor)



UNIVERSIDAD NACIONAL MAYOR DE
SAN MARCOS
Universidad del Perú, Decana de América

CERTIFICADO DE SIMILITUD

Yo, **Ciro Rodriguez Rodriguez** en mi condición de asesor acreditado con **Dictamen N° 000401-2023-UPG-VDIP-FISI/UNMSM** de la tesis, cuyo título es “**Modelo de predicción de deserción de clientes de servicio postpago de telefonía móvil en empresas de telecomunicaciones**”, presentado por el magíster **David Andres Freire Vera** para optar el grado de **Doctor en Ingeniería de Sistemas e Informática**, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el **10 %** de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado correspondiente.

Lima, 26 de febrero de 2025

Firma del Asesor

Nombres y Apellidos del asesor: **Ciro Rodríguez Rodríguez**

DNI: **06020241**

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a todas las personas que, con su apoyo y dedicación, contribuyeron significativamente no solo en la elaboración de esta tesis, sino también a lo largo del extenso y desafiante camino hacia la obtención de mi título de doctorado. Su respaldo fue fundamental para superar las vicisitudes que surgieron en este proceso. Asimismo, agradezco profundamente a Dios por haberlos puesto en mi vida, ya que fueron luz y guía que me impulsaron a seguir adelante hasta alcanzar esta importante meta.

También quiero agradecer a mis tutores, el Doctor Ciro y Doctor Mauricio, por su invaluable guía y dedicación durante todo este proceso. Sus enseñanzas me proporcionaron las bases y pautas para llevar a cabo tanto mi tesis como mi artículo científico asociado.

DEDICATORIA

Esta tesis está dedicada a mis padres, a mi esposa Solange Gálvez e hijos Josue, Josias e Iris.

A mi amada esposa, quien vio en mí un potencial que ni siquiera yo mismo imaginaba. Para ser honesto, nunca pensé en llegar hasta este punto, pero admiro profundamente tu fe en mí y tu capacidad para impulsarme a seguir adelante en este arduo y desafiante camino. Gracias por estar siempre a mi lado.

A mis amados hijos, recuerden siempre que el cielo es el límite. Atrévanse a soñar en grande, apunten alto y no teman a las caídas ni a lo desafiante que pueda parecer el camino. Confío en que podrán superar cualquier reto que la vida les presente. Los amo profundamente.

Amados Padres, David y Mery, he culminado este importante camino y confieso que, a lo largo de él, he tenido tanto aciertos como errores, y que nada ha sido fácil. Sin embargo, agradezco profundamente a Dios por tenerlos presentes en mi vida. Aunque no nos veamos con frecuencia, quiero que sepan que este logro también les pertenece, porque su amor y enseñanzas siempre han sido mi guía.

ÍNDICE GENERAL

AGRADECIMIENTO	III
DEDICATORIA	IV
ÍNDICE GENERAL	V
LISTA DE CUADROS	IX
LISTA DE FIGURAS	XI
RESUMEN	XII
ABSTRACT	XIII
CAPÍTULO I: INTRODUCCIÓN.....	1
1.1 Situación problemática	1
1.2 Formulación del problema	4
1.2.1 <i>Problema general</i>	4
1.2.2 <i>Problemas específicos</i>	4
1.3 Justificación teórica.....	4
1.4 Justificación práctica.....	5
1.5 Objetivos.....	7
1.5.1 <i>Objetivo General</i>	7
1.5.2 <i>Objetivos Específicos</i>	7
CAPÍTULO 2: MARCO TEÓRICO.....	8
2.1 Bases teóricas o enfoques teóricos	8
2.1.1 <i>Deserción</i>	8
2.1.2 <i>Factores y/o variables de deserción</i>	10

2.1.3	<i>Segmentación de mercado</i>	16
2.1.4	<i>Predicción de deserción</i>	18
2.1.5	<i>Net Promoter Score (NPS)</i>	30
2.2	Marco conceptual	31
CAPÍTULO 3: METODOLOGÍA DE LA INVESTIGACIÓN		36
3.1	Hipótesis general.....	36
3.2	Hipótesis específicas	36
3.3	Identificación de variables	37
3.3.1	Variable Independiente	37
3.3.2	Variable dependiente	37
3.4	Desglose de hipótesis	38
3.5	Tipo y diseño de investigación	40
3.6	Unidad de análisis.....	40
3.7	Población de estudio	40
3.8	Tamaño de la muestra	42
3.9	Selección de la muestra	43
3.10	Técnicas de recolección de datos	44
3.11	Técnicas de análisis de datos	46
CAPÍTULO 4: DESARROLLO DE LA PROPUESTA.....		50
4.1	Construcción del conjunto de datos (Telco y no Telco).....	50
4.1.1	Identificación de variables del dataset Telco	50

4.1.2	Asociación de variables con factores de deserción	50
4.1.3	Unificación de dataset Telco con no Telco.....	51
4.2	Diseño de modelo de aprendizaje automático	53
4.2.1	Análisis exploratorio de variables	53
4.2.2	Pre – Procesamiento de datos	57
4.2.3	Selección de variables.....	58
4.2.4	Ejecución de algoritmos de predicción	60
4.2.5	Selección de algoritmos de predicción	62
4.2.6	Ensamble de datos.....	63
4.3	Validación de modelo de aprendizaje automático.....	63
4.3.1	Validación cruzada	63
4.3.2	Validación con prueba t de student.....	64
4.3.3	Validación con test de McNemar.....	65
CAPÍTULO V: RESULTADOS Y DISCUSIÓN		66
5.1	Resultados.....	66
5.1.1	Construcción del dataset (Telco y no Telco).....	66
5.1.2	Diseño del modelo de aprendizaje automático.....	69
5.1.3	Validación del modelo de aprendizaje automático	70
5.2	Discusión	71
5.2.1	Sobre construcción del dataset (Telco y no Telco).....	72
5.2.2	Sobre el diseño del modelo de aprendizaje automático	73

5.2.3	Sobre la Validación del modelo de aprendizaje automático	74
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES		75
6.1	Conclusiones.....	75
6.2	Recomendaciones	77
REFERENCIAS BIBLIOGRÁFICAS		79
Anexo: Código Python		97

LISTA DE CUADROS

Tabla 1 Impacto de deserción de clientes en empresas de América Móvil	6
Tabla 2 Factores relacionados con el producto	12
Tabla 3 <i>Factores relacionados con el servicio</i>	13
Tabla 4 Factores relacionados con el cliente	14
Tabla 5 <i>Factores externos</i>	16
Tabla 6 <i>Algoritmos basados en aprendizaje automático en procesos de predicción de churn</i>	21
Tabla 7 <i>Algoritmos basados en aprendizaje profundo en procesos de predicción de churn</i>	25
Tabla 8 <i>Algoritmos híbridos en procesos de predicción de churn</i>	30
Tabla 9 Matriz de operacionalización de variables	37
Tabla 10 Matriz de resultados de modelos (F1 Score)	54
Tabla 11 Variables con baja correlación usando técnicas estadísticas	59
Tabla 12 Variables con baja correlación usando algoritmos de machine learning	59
Tabla 13 Variables con alta correlación usando técnica greedy stepwise	60
Tabla 14 Algoritmos de predicción sin datos balanceados	61
Tabla 15 Algoritmos de predicción sin datos balanceados	61
Tabla 16 Matriz de resultados de modelos (F1 Score)	62
Tabla 17 Búsqueda de hiperparámetros	62
Tabla 18 Validación cruzada	64
Tabla 19 Selección de variables (dataset original - Parte 1)	66
Tabla 20 Selección de variables (dataset original - Parte 2)	67

Tabla 21 Matriz de variables seleccionadas	68
Tabla 22 Validación cruzada	70

LISTA DE FIGURAS

Figura 1. Suscripciones a telefonía celular móvil (por cada 100 personas)	2
Figura 2. Tipos de deserción	9
Figura 3. Factores de deserción de clientes en empresas de Telco	11
Figura 4. Tipos de segmentación de mercado	17
Figura 5. Tipos de segmentación de mercado	18
Figura 6. Calificación Net Promoter Score	31
Figura 7. Número de abonados postpago en empresas Telco en Ecuador...	41
Figura 8. Desglose de muestra seleccionada.....	41
Figura 9. Asociación de variables del dataset con factores de deserción	51
Figura 10. Desglose de muestra seleccionada	52
Figura 11. Desglose de variables del dataset.....	53
Figura 12. Desglose de variables – Parte 1	55
Figura 13. Desglose de variables – Parte 2	56
Figura 14. Desglose de variables – Parte 3.....	57
Figura 15. Proceso de imputación de datos	58
Figura 16. Nuevas variables para reducir la variabilidad	58
Figura 17. Ensamble de Algoritmos (Sin ajuste de Hiperparámetros).....	63
Figura 18. Resultado de prueba t de student	64
Figura 19. Resultado de test de McNemar	65
Figura 20. Diseño de Modelo de predicción de deserción de clientes postpago en empresas Telco.....	69
Figura 21. Ensamble de Algoritmos (Con ajuste de Hiperparámetros)	70
Figura 22. Secuencia de pasos del Modelo de predicción	71

RESUMEN

El sector de telecomunicaciones ha experimentado un crecimiento exponencial desde el año 2000, alcanzando 5.31 billones de usuarios hasta el 2022 y generando ingresos de 1.07 trillones de dólares, sin embargo, este crecimiento conduce a una intensa competencia entre las empresas para la adquisición de clientes, especialmente en servicios móviles, lo que a menudo resulta en una deserción de clientes "churn".

Por ello, se vienen desarrollando múltiples investigaciones que presentan algoritmos de predicción, que ayuden a tomar acciones con antelación para evitar la fuga de clientes, sin embargo, no existe homogeneidad entre las variables utilizadas en el conjunto de datos, el diseño de pasos para obtención del modelo y su comprobación, por lo cual el objetivo de esta investigación consiste en determinar cómo contribuye el uso del modelo de aprendizaje automático a la predicción de la deserción de clientes postpago en empresas de telecomunicaciones, considerando las variables del conjunto de datos (pertenecientes o no al sector de telecomunicaciones), el diseño del modelo y su validación a través de técnicas como la validación cruzada, prueba t de Student y el test de McNemar, demostrando con un p-value cercano a 0 que el modelo supera significativamente a los modelos base, y alcanzando un F1 score de 84% y un recall de 74%.

El estudio concluye que el uso de aprendizaje automático contribuye significativamente a predecir la deserción de clientes, proponiendo mejoras en la integración de datos externos y exploración de nuevas técnicas de ensamblaje. La implementación del modelo puede extenderse a otros sectores, sugiriendo incluir variables macro y microeconómicas para una visión más completa del cliente.

Palabras clave: deserción de clientes, telecomunicaciones, predicción de deserción, aprendizaje automático, modelos de predicción, conjunto de datos

ABSTRACT

The telecommunications sector has experienced exponential growth since 2000, reaching 5.31 billion users by 2022 and generating revenues of \$1.07 trillion, however, this growth leads to intense competition between companies for customer acquisition, especially in mobile services, often resulting in customer churn.

For this reason, multiple investigations have been developed that present prediction algorithms that help to take actions in advance to avoid customer churn. However, there is no homogeneity between the variables used in the data set, the design of steps to obtain the model and its verification, so the objective of this research is to determine how the use of the machine learning model contributes to the prediction of postpaid customer churn in telecommunications companies, considering the variables of the data set (belonging or not to the telecommunications sector), the design of the model and its validation through techniques such as cross-validation, Student's t-test and the McNemar test, demonstrating with a p-value close to 0 that the model significantly outperforms the base models, and reaching an F1 score of 84% and a recall of 74%.

The study concludes that the use of machine learning contributes significantly to predicting customer churn, proposing improvements in the integration of external data and exploration of new assembly techniques. The implementation of the model can be extended to other sectors, suggesting the inclusion of macro and microeconomic variables for a more complete view of the client.

Keywords: customer churn, telecom, churn prediction, machine learning, prediction models, dataset.

CAPÍTULO I: INTRODUCCIÓN

1.1 Situación problemática

El sector de telecomunicaciones (Telco) posee 5.31 billones de usuarios hasta el 2022, generando 1.07 trillones de dólares de ingreso anuales (Global Overview Report, 2022), y sus principales prestaciones de servicios son los servicios móviles (de voz y de datos) y fijos (Lera, 2000), siendo el servicio móvil el que ha evolucionado de forma exponencial.

Esta evolución ha consolidado a la industria del servicio móvil como una de las principales en los países desarrollados, impulsando un creciente número de operadores telefónicos (Ahmad et al., 2019) y de clientes (Banco Mundial, 2022).

Desde el año 2000, el número de suscripciones a servicios de telefonía móvil ha ido incrementado significativamente, a tal punto que, para el año 2022, por cada 100 clientes se tienen 108 suscripciones, como se muestra en la Figura 1 (Banco Mundial, 2022). En ese contexto, este incremento ha intensificado la competencia por la captación de clientes, lo que, según estudios como el de Esteves & Mendes-Moreira (2016), es una de las causas principales que provoca la deserción de clientes de telefonía móvil.

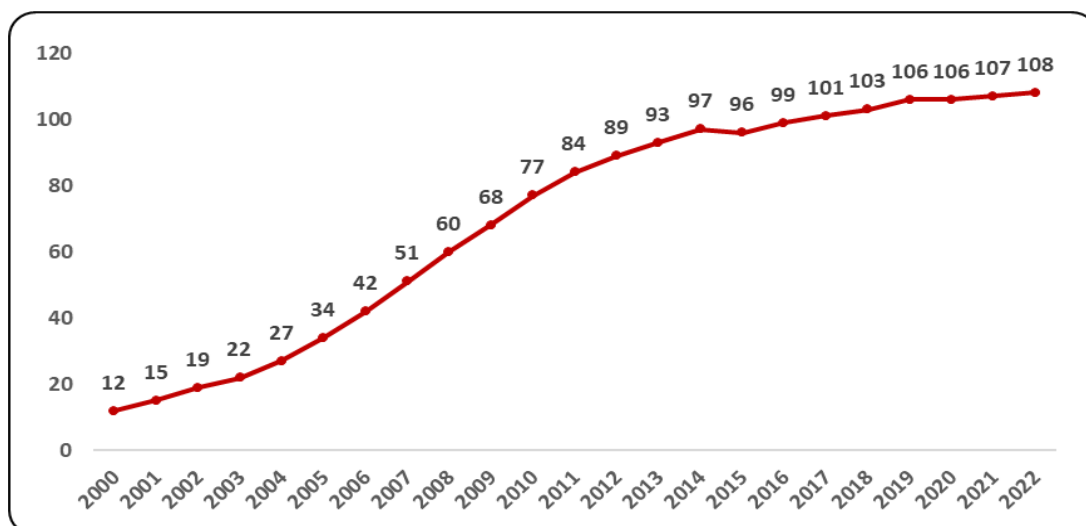


Figura 1. Suscripciones a telefonía celular móvil (por cada 100 personas)

Fuente. Banco Mundial (2022)

La deserción, conocida como “churn”, es un fenómeno ampliamente reconocido en la industria de telecomunicaciones y autores como Chouiekh & El Haj (2020) la definen como la situación en la que un cliente deja de comprar productos, de utilizar el servicio o se cambia de proveedor, mientras que Ahmad et al. (2019) enfatizan la necesidad de minimizar esta fuga mediante estrategias de retención y fidelización, las cuales deben revisar características y analizar comportamientos para predecir la posible renuncia al servicio, lo cual es reforzado por Kim et al. (2020), que demuestran que es 5 veces más costoso traer un nuevo cliente que retenerlo, subrayando así la importancia de ese indicador en la industria.

Debido a esta situación, las operadoras de telecomunicaciones enfrentan el desafío de reducir la tasa de deserción, la cual implica pérdidas significativas. Por ejemplo, América Móvil (AM), en el año 2022, presentó tasas de deserción en Brasil, Ecuador, México, Colombia, Chile y Perú del 2.9%, 3.1%, 3.3%, 4%, 4.1% y 4.6%, respectivamente (América Móvil, 2022). Una situación similar ocurre en otras operadoras de telefonía móvil como Telefónica Movistar que en el año 2020 presentó tasas de deserción en Alemania, Ucrania, Brasil e Hispanoamérica (Argentina, Chile, Uruguay, México, Venezuela, Perú, Colombia y Ecuador) de 1.8%, 1.5%, 3.1%, 3.0%, respectivamente (Telefónica Movistar, 2020). Reflejando de esta manera el

impacto económico y competitivo que la deserción puede tener en el sector de telecomunicaciones.

Para abordar este problema, autores como Esteves & Mendes-Moreira (2016), indican que las empresas se deben orientar a anticipar que cliente es más propenso a renunciar o cambiar de operadora telefónica, para lo cual se mencionan 2 métodos: el proactivo y el reactivo, donde el primero identifica con antelación a los clientes que podrían desertar, mientras que el segundo actúa cuando el cliente ya ha renunciado (Bhale & Bedi, 2021). Siendo el método proactivo considerado como la mejor estrategia, ya que ofrece un periodo de tiempo para implementar estrategias de retención (Retana et al., 2016).

Identificar con antelación que cliente finalizará su relación con la empresa se conoce como predicción (Zhang et al., 2022). Esta predicción se basa en el uso de algoritmos avanzados capaces de reconocer patrones y extraer conocimiento de grandes volúmenes de datos (Han et al., 2012). Estos algoritmos de predicción se utilizan para la construcción de modelos de predicción, los cuales pueden ser de aprendizaje automático o profundo. El aprendizaje automático aprende patrones a partir de datos, lo cual le permite hacer predicciones precisas, mientras que el aprendizaje profundo se basa en redes neuronales artificiales para extraer características complejas, lo cual los hace ideales para tareas como procesamiento de imágenes y lenguaje natural.

La predicción o pronóstico puede variar acorde al tipo de cliente, lo cual es diferenciado por las operadoras de telecomunicaciones en dos grandes grupos, los clientes prepago y los postpago. Los clientes postpago, son aquellos que tienen un contrato formal con la operadora (Bhale & Bedi, 2021) y suelen proporcionar más datos útiles para los modelos, permitiendo una predicción más precisa. En contraste, los clientes prepago pueden requerir diferentes enfoques para la recolección y análisis de datos debido a la naturaleza de su relación más flexible con la operadora (Azeem & Usman, 2018).

1.2 Formulación del problema

1.2.1 Problema general

¿Cómo el modelo de aprendizaje automático contribuye a la predicción de la deserción de clientes postpago en empresas de telecomunicaciones?

1.2.2 Problemas específicos

- ¿Cómo la construcción de un conjunto de datos, que incorpora variables Telco y no Telco influye en el modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones?
- ¿Cómo los componentes en el diseño del modelo de aprendizaje automático tienen un impacto en la predicción de deserción de clientes postpago en empresas de telecomunicaciones?
- ¿Cómo la validación del modelo de aprendizaje automático usando métricas de clasificación para clases desequilibradas, hacen confiable la predicción de la deserción de clientes postpago en empresas de telecomunicaciones?

1.3 Justificación teórica

La literatura actual ha desarrollado numerosos trabajos de predicción de deserción de clientes de empresas de telecomunicaciones, centrándose en conjuntos de datos (datasets) que reflejan aspectos relacionados con el servicio provisto por la operadora. Entre las variables más estudiadas se encuentran las llamadas entrantes y salientes (Jain et al., 2022), los megas consumidos (Mandák & Hančlová, 2019), el monto facturado mensual (Khan et al., 2019), los cargos diurnos y nocturnos (Dalli, 2022), el soporte técnico (Fujo et al., 2022), entre otros.

Estos factores están limitados a la información que el cliente proporciona a la operadora durante un tiempo determinado; sin embargo, estas variables no identifican el comportamiento subyacente del individuo. Por esta razón, se propone complementar el dataset con información proveniente de otras instituciones como agencias de tránsito, de impuestos, servicios de seguridad social o a través de encuestas que ayuden a identificar nuevas características subyacentes del individuo, con el objetivo de mejorar la precisión en la predicción de deserción de clientes en empresas de telecomunicaciones.

Incorporar datos de múltiples fuentes permitirá una comprensión más holística del comportamiento del cliente, lo que podría revelar patrones no evidentes a partir de los datos tradicionales proporcionados por la operadora. Esto no solo optimizaría la capacidad predictiva de los modelos, sino que también permitiría a las empresas de telecomunicaciones diseñar estrategias más efectivas de retención, ajustadas a las necesidades y comportamientos reales de sus clientes.

1.4 Justificación práctica

Esta investigación contribuye al desarrollo de un modelo de aprendizaje automático diseñado para anticipar qué clientes rescindirán su contrato, lo que facilita la implementación de estrategias de retención y fidelización (Abarca Sánchez et al., 2022). Esto tiene como objetivo prevenir la renuncia, mitigar el impacto económico asociado, promover la extensión del tiempo de permanencia del cliente y, en consecuencia, mejorar directamente la rentabilidad de la empresa (Bhale & Bedi, 2020).

Por esta razón, varios investigadores han desarrollado modelos de predicción de deserción (Al-Mashraie et al., 2020), (Liu, Fan, et al., 2022), (Yu et al., 2018), dada la significativa pérdida económica que esta representa. Un ejemplo de esto se visualiza en la Tabla 1, donde se incluyen indicadores como la cantidad de abonados activos y desertores, junto con su respectivo

cálculo porcentual, el ingreso promedio por usuario (ARPU) y la estimación del impacto económico calculado en función de la tasa de churn y la cantidad de abonados que se desvinculan (América Móvil, 2022).

Esta información es fundamental para identificar los factores que influyen en la renuncia de clientes y, de este modo, comprender de manera holística todos los elementos que afectan el proceso de deserción en empresas de telecomunicaciones. La capacidad de predecir y abordar de manera proactiva la deserción permite a las empresas no solo retener a sus clientes actuales, sino también optimizar sus estrategias de negocio y mejorar su posición competitiva en el mercado.

Tabla 1

Impacto de deserción de clientes en empresas de América Móvil

País	% clientes desertores		# Abonados Activos		# Abonados desertores		ARPU		Impacto estimado (ingreso) *		Moneda
	1T 2022	1T 2021	1T 2022 (Millón)	1T 2021 (Millón)	1T 2022 (Mil)	1T 2021 (Mil)	1T 2022	1T 2021	1T 2022 (Millón)	1T 2021 (Millón)	
México	3.30%	3.50%	80.9	78.3	2672	2740	162	152	432.9	416.6	Peso mexicano
Brasil	2.90%	3%	71.8	66.3	2082	1990	22	22	45.8	43.78	Real
Colombia	4%	4.10%	35.7	33.4	1428	1371	16017	17035	22878	23370	Peso colombiano
Perú	4.60%	4.10%	12.1	11.1	555	458	23	23	12.7	10.54	Sol peruano
Ecuador	3.10%	3.30%	8.7	8.1	271	267	7	8	1.8	2.1	Dólar
Chile	4.10%	6.20%	7.1	6.5	292	408	4038	4845	1181	1976	Peso chileno
Argentina	1.60%	1.00%	23.5	21.8	376	218	473	344	177	75.08	Peso argentino

Fuente. Elaboración propia tomada de (América Móvil, 2022).

* Impacto calculado con el churn y abonados que desertan.

1.5 Objetivos

1.5.1 Objetivo General

Determinar cómo contribuye el uso del modelo de aprendizaje automático a la predicción de la deserción de clientes postpago en empresas de telecomunicaciones.

1.5.2 Objetivos Específicos

1. Construir un conjunto de datos, incorporando variables Telco y no Telco que influyan en el modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones.
2. Diseñar el modelo de aprendizaje automático para predecir la deserción de clientes postpago en empresas de telecomunicaciones.
3. Validar el modelo de aprendizaje automático usando métricas de clasificación para clases desequilibradas, haciendo confiable la predicción de la deserción de clientes postpago en empresas de telecomunicaciones.

CAPÍTULO 2: MARCO TEÓRICO

La importancia de predecir la deserción de clientes de empresas de telecomunicaciones ha sido ampliamente estudiada y en este capítulo se definen los conceptos complementarios para poder entender de una manera holística la investigación propuesta.

2.1 Bases teóricas o enfoques teóricos

2.1.1 Deserción

La deserción representa una situación en la que el cliente deja de comprar productos o utilizar los servicios ofertados por una compañía, y específicamente en el contexto de las telecomunicaciones, se refiere a cuando el cliente abandona el servicio provisto o se cambia de operadora (Chouiekh & El Haj, 2020).

Ahn et al., (2020), describen la deserción como la propensión de los clientes a dejar de hacer negocios con una empresa en un tiempo o contrato definido. Esto implica que, aunque el cliente pueda estar satisfecho con el servicio, factores externos o internos pueden influir en su decisión de abandonar la compañía.

Adhikary & Gupta, (2021), indican que, un cliente se convierte en “desertor” cuando interrumpe su suscripción y se pasa a un proveedor de servicios de la competencia independientemente de la empresa proveedora de servicios. Este cambio puede deberse a una variedad de razones, como

ofertas más atractivas por parte de la competencia o experiencias negativas con la compañía original.

Por su parte, Lazarov & Capota, (2007) clasifican la deserción en dos tipos: voluntaria e involuntaria. La deserción voluntaria se da, como su nombre lo indica, cuando el cliente decide voluntariamente cancelar el servicio, mientras que la deserción involuntaria sucede cuando la empresa rescinde o finaliza el contrato por mora o fraude. Esta clasificación (ver Figura 2) permite a las empresas entender mejor las causas de la deserción y ajustar sus estrategias en consecuencia.

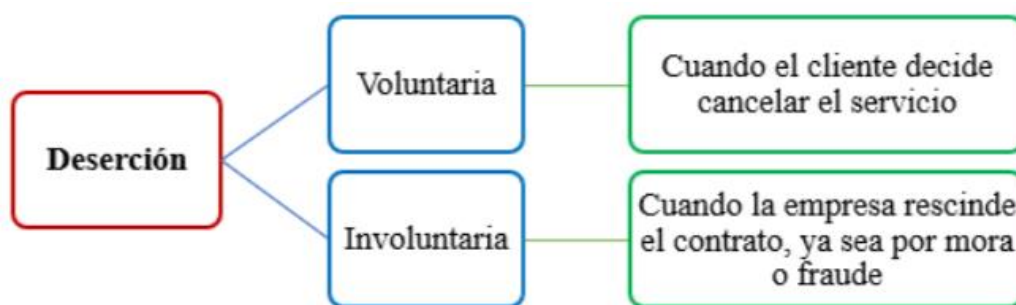


Figura 2. Tipos de deserción

Fuente. Elaborado por el autor.

Adicionalmente, Mandák & Hančlová (2019) identifican 3 dimensiones clave a considerar en el estudio de la deserción:

- **Quién:** Esta dimensión determina qué clientes son posibles desertores, analizando características demográficas, comportamientos previos y patrones de consumo. Con la identificación de clientes en riesgo, las empresas pueden enfocar sus esfuerzos de retención de manera más efectiva.
- **Cuando:** Esta dimensión determina el momento en que se producirá la deserción, lo cual permite a la organización anticiparse y tomar acciones preventivas, siendo una herramienta crucial para la

planificación estratégica, ya que ofrece una ventana de oportunidad para intervenir antes de que el cliente decida abandonar el servicio.

- **Por qué:** Esta dimensión explica las razones por las cuales se produce el abandono o deserción, basándose en el análisis de factores como la insatisfacción, problemas con el servicio, competencia, cambios en las necesidades del cliente, entre otros; permitiendo comprender las causas subyacentes del churn, para poder desarrollar estrategias de mejora del servicio y de satisfacción del cliente.

El objetivo de considerar estas dimensiones es proporcionar una vista holística e integral del cliente desde diferentes perspectivas relacionadas con la deserción, lo cual contribuye significativamente al proceso de retención, dado que la pérdida de un cliente impacta directamente en la rentabilidad de la organización (Bhale & Bedi, 2020). Además, el abordar el problema de la deserción desde estas tres perspectivas, permite a las empresas de telecomunicaciones implementar estrategias más precisas y efectivas para retener a sus clientes y mantener una ventaja competitiva en el mercado.

Estas dimensiones no solo ofrecen un marco para analizar la deserción de clientes, sino que también facilitan la identificación de áreas específicas donde las empresas pueden intervenir para mejorar la experiencia del cliente y reducir las tasas de deserción, permitiéndole a la organización, la optimización de sus recursos y el aseguramiento de una relación más sólida y duradera con sus clientes.

2.1.2 Factores y/o variables de deserción

Los factores relacionados con la deserción consisten en identificar las razones detrás de la pérdida de clientes (Soeini & Rodpysh, 2012), y acorde al estado del arte presentado por Freire et al. (2024), el proceso de deserción de clientes en telecomunicaciones puede analizarse a través de cuatro perspectivas o

aristas principales: los factores relacionados con el producto, el servicio, el cliente y los factores externos, las cuales se ilustran en la Figura3.



Figura 3. Factores de deserción de clientes en empresas de Telco

Fuente. Elaborado por el autor.

Estas perspectivas proporcionan un marco integral para desglosar y comprender las múltiples facetas que contribuyen a la deserción de clientes. Su análisis podría ayudar a las empresas de telecomunicaciones identificar áreas problemáticas y a diseñar estrategias específicas para abordar de manera efectiva los factores que influyen en este proceso y que llevan a los clientes a abandonar el servicio.

Factores de deserción relacionados al producto

Según Sandhya et al. (2019), los elementos relacionados con el producto adquirido por el cliente, tales como los planes de red, los planes internacionales, las tarifas mensuales, la duración del plan, los cobros de llamadas diarios, las tarifas de llamadas internacionales y los cargos de llamadas nocturnas, son factores que pueden provocar la deserción, ya que la insatisfacción con alguno de estos elementos puede llevar a que los clientes busquen alternativas que mejor se adapten a sus necesidades y expectativas, es decir, pueden conllevar a una renuncia.

Estos factores están detallados en la Tabla 2, proporcionando un análisis más específico de las variables asociadas al producto.

Tabla 2
Factores relacionados con el producto

Factor	Variables Telco	Concepto
Plan	Plan internacional	Aplica cuando el cliente realiza negocios internacionales.
	Número servicio	Número de servicio asignado al cliente.
	Nombre del plan	Nombre del plan contratado por el cliente.
	Tipo de producto	Tipo de producto contratado por el cliente.
	Duración	Duración del contrato del cliente.
Equipos	Protección	Servicio de protección de equipo ante siniestros.
	-	Nombre del dispositivo del cliente.
	Modelo	Modelo del dispositivo del cliente.
	Tipo	Tipo de dispositivo del cliente (Ejemplo: Smartphone).
Facturación	-	Monto total a cancelar por el cliente.
	Cargos nocturnos	Recargos por consumos realizados durante la noche.
	Precio plan	Costo monetario del plan contratado.
	Internacional	Recargos por consumos realizados internacionalmente.
	Adicional	Recargos por consumos que exceden los términos del plan contratado.
Ofertas	-	Ofertas vigentes disponibles para el cliente.
Recarga	-	Recargas adicionales fuera del plan contratado.

Fuente. Elaboración propia tomada de (Freire et al., 2024).

Factores de deserción relacionados con el servicio

Bhattacharyya & Dash (2020) argumentan que la decisión de un cliente de abandonar una operadora está influenciada por factores internos relacionados con el servicio, tales como las quejas, el valor y la calidad percibida, además de la experiencia del cliente. Asimismo, destacan que la satisfacción del cliente desempeña un papel crucial, ya que impacta directamente en lo que denominan "boca a boca" (WOM), refiriéndose a las opiniones y comentarios que los clientes comparten sobre la empresa. Estas opiniones pueden ser tanto favorables (WOM positivo) como desfavorables (WOM negativo). En el caso de que sean desfavorables, pueden perjudicar significativamente la reputación de la empresa e impactar en el proceso de deserción.

Riyadh et al. (2021) refuerzan esta perspectiva al señalar que, para comprender los motivos y causas de la deserción, es esencial profundizar en las interacciones que el cliente mantiene con la empresa.

Estos factores están detallados en la Tabla 3, proporcionando un análisis más específico de las variables relacionadas con el servicio.

Tabla 3
Factores relacionados con el servicio

Factor	Variable Telco	Concepto
Servicio al Cliente	-	Servicio al cliente provisto por la operadora.
	Ineficiente construcción de relación	Problemas en la relación entre el cliente y la empresa.
Calidad	De la llamada	Calidad de llamadas realizadas por el cliente.
	Del servicio	Calidad del servicio recibido por el cliente.
	Del internet	Calidad de la conexión a internet que recibe el cliente.
Cobertura del Servicio	-	Cobertura del servicio proporcionado por la operadora.
Reclamos	-	Reclamos realizados por el cliente.
	Equipos	Reclamos relacionados con el dispositivo del cliente.
Canal de ventas	-	Canales de venta ofrecidos por la operadora

Fuente. Elaboración propia tomada de (Freire et al., 2024).

Factores de deserción relacionados con el cliente:

Tong et al. (2017), sugieren que es fundamental comenzar con un análisis de la situación inicial del cliente para identificar los aspectos que impulsan su lealtad. Además, Kassem et al. (2020), añaden que factores como los saldos no utilizados (consumos de minutos, internet o mensajes), la edad del cliente y los minutos gratuitos entregados también influyen en el proceso de deserción.

Estos factores están detallados en la Tabla 4, proporcionando un análisis más específico de las variables relacionadas con el cliente.

Tabla 4
Factores relacionados con el cliente

Factor	Variable Telco	Concepto
Relación con la empresa	Tiempo de suscripción	Tiempo total de suscripción del cliente con la empresa
	Múltiples líneas	Número de líneas contratadas por el cliente
	Customer id	Código de identificación del cliente
	Estado	Estado del cliente (Ej. activo, inactivo, etc.)
	Conexiones	Número de personas allegadas al cliente que utilizan el mismo servicio
	Forma pago	Método utilizado por el cliente para pagar el servicio
	Tipo uso del servicio	Tipo de uso que el cliente le da al servicio contratado
	Mora	Tiempo que el cliente ha permanecido en impago
	Lealtad	Nivel de lealtad del cliente hacia la empresa
	Transacciones	Cantidad de transacciones realizadas por el cliente
Demografía	Penalizaciones	Penalizaciones recibidas por el cliente
	Área	Zona de residencia del cliente
	Ciudad residencia	Ciudad donde reside el cliente
	Género	Género del cliente (Ej. Masculino, femenino, etc.)
	Dependencias	Número de personas dependientes del cliente
	Edad	Edad del cliente
	Otros servicios	Otros servicios contratados por el cliente. (Ej. Internet Fijo)
	Estado civil	Estado civil del cliente
	Profesión	Profesión del cliente
	Crédito	Historial crediticio del cliente
	Estatus social/ viajes	Nivel socioeconómico y número de viajes del cliente
	Hijos	Número de hijos del cliente
	Vehículo propio	Indica si el cliente posee vehículo propio
	Educación	Nivel de educación alcanzado por el cliente
	Dirección	Lugar de residencia del cliente
	Curiosidad	Afinidad del cliente por probar otros servicios
	Casa propia	Indica si el cliente es propietario de una vivienda
	Uso wireless	Indica si el cliente utiliza conexiones wifi
	Provincia	Provincia de residencia del cliente
Consumo de Voz	-	Consumo de servicios de voz del cliente
	Buzón	Llamadas de buzón de voz
	Internacional	Llamadas internacionales realizadas
	Día / Noche	Llamadas realizadas durante el día y/o la noche

Consumo de Navegación	Onnet / Offnet	Llamadas realizadas a números de la misma operadora y/o de otras operadoras
	Entrantes / Salientes	Llamadas recibidas y/o realizadas
	Excedentes	Exceso de minutos del plan contratado
	Fijos / Gratis	Llamadas a números fijos y/o gratuitos
	Roaming entrante / saliente	Llamadas recibidas y/o realizadas fuera del país
	Cargos	Llamadas con costo
	Identificador de llamadas	Uso del servicio de identificación de llamadas
	Conferencias	Llamadas realizadas en modo conferencia
	Seguridad	Nivel de seguridad en la navegación de los clientes
	Respaldos	Respaldos de la información del cliente
Consumo de SMS	-	Consumo de internet por parte del cliente
	Upload / Download	Subidas y descargas de datos realizadas por el cliente
	2G / 3G / 4G	Navegación en las redes 2G, 3G y/o 4G
	-	Consumo de mensajes de texto del cliente
	Entrantes / Salientes	Mensajes de texto entrantes y/o salientes del cliente
Consumo de MMS	Nacional / Internacional	Mensajes nacionales y/o internacionales realizados
	Onnet / Offnet	Mensajes enviados a números de la misma operadora y/o de otras operadoras
	Entrantes / Salientes	Mensajes multimedia recibidos y/o enviados
	-	Consumo general de mensajes multimedia

Fuente. Elaboración propia tomada de (Freire et al., 2024).

Factores externos

Uner et al. (2020) atribuye la deserción a factores relacionados a la imagen de la compañía, la variedad de servicios propuestos, los clubes de beneficios, y las nuevas campañas realizadas con el fin de promover la marca. Además, Bhale & Singh Bedi (2021) incorporan factores externos, como los beneficios obtenidos en sectores diferentes al de telecomunicaciones, tales como el acceso a televisión, películas y series por internet, donde la ausencia de estos también pueden influenciar en la deserción de un cliente. Estos factores externos se pueden observar en la Tabla 5.

Estos factores están detallados en la Tabla 5, proporcionando un análisis más específico de las variables relacionadas con los factores externos que pueden influenciar en el proceso de deserción.

Tabla 5
Factores externos

Factor	Variable Telco	Concepto
Beneficios externos	Streaming movies	Beneficios de películas por internet
	Streaming tv	Beneficios de televisión por internet
	Beneficios	Beneficios provistos por alianzas
Imagen de la compañía	Imagen de la compañía	Imagen de la marca (Ej. Branding, BTL, ATL)
	Publicidad	Publicidad que realiza la compañía
	Marca	Marca de la compañía
Otros	No identificados	Factores no identificados
	Mes	Fecha de evaluación de la información
	Tipo	Indica si el cliente posee servicio prepago o postpago
Compañeros, amigos, familia desertores	-	Indica si el cliente posee compañeros, amigos o familiares que han renunciado

Fuente. Elaboración propia tomada de (Freire et al., 2024).

2.1.3 Segmentación de mercado

La segmentación de mercado es un proceso que implica dividir los datos en segmentos homogéneos con el fin de crear subgrupos que compartan características similares (Dolnicar et al., 2018). Esta técnica ayuda considerablemente a la identificación de comportamientos específicos basados en diversas variables, como los registros de detalles de llamadas (Golubev et al., 2017) o los perfiles de clientes.

Uno de los métodos utilizados para detectar estos patrones de uso es el Fuzzy c-means clustering o clustering difuso (Bose & Chen, 2015), el cual permite una clasificación flexible de los datos en diferentes categorías, adaptándose a la variabilidad inherente en los comportamientos de los clientes.

Además de Fuzzy c-means, existen otras técnicas de segmentación, como la optimización de k-means mediante el método de enjambre de partículas, la cual facilita el análisis de conglomerados y ayuda a mejorar la precisión de la segmentación, además de que se puede utilizar en combinación con el método de clasificación C5.0 para formular reglas de clasificación más robustas (S.-T. Wang, 2018).

Asimismo, la integración de k-means con árboles de decisión también ofrece un enfoque diferente para el análisis de agrupamientos, lo cual le permite interpretar de mejor manera los resultados obtenidos (Khalili-Damghani et al., 2018).

En la Figura 4, se detallan los tipos de agrupación utilizados en la segmentación de mercado mencionada.



Figura 4. Tipos de segmentación de mercado

Fuente. Elaborado por el autor.

La agrupación no jerárquica organiza observaciones en grupos o clusters con el objetivo de minimizar ciertos criterios de evaluación, y a diferencia de los métodos jerárquicos, este enfoque no crea una estructura de árbol, sino que particiona directamente el conjunto de datos en un número predefinido de clusters. Sin embargo, uno de los desafíos de este método es que el usuario debe especificar el número de clusters a generar, lo cual puede influir significativamente en los resultados y la interpretación de los datos. (Jin & Han, 2017)

Por otro lado, el agrupamiento particional hace referencia a un método de agrupación basado en particiones, es decir, el conjunto de datos se divide en grupos separados, donde cada uno contiene al menos una observación, y cada observación pertenece exclusivamente a un único grupo. Luego e esto, se implementan algoritmos específicos que optimizan un criterio definido, como la minimización de la variabilidad interna dentro de los clusters o la maximización de las distancias entre ellos, permitiendo un análisis de datos más preciso y estructurado. (Renuka Devi et al., 2019).

Ambas técnicas, la agrupación no jerárquica y el agrupamiento particional son métodos ampliamente utilizados en análisis de datos debido a su flexibilidad y eficacia para organizar información compleja, y aunque sus enfoques son diferentes, ambas técnicas permiten descubrir patrones y estructuras subyacentes en los datos, siendo herramientas valiosas en disciplinas como la minería de datos, el aprendizaje automático y la estadística, sin embargo, la elección de estás depende de las características del conjunto de datos y de los objetivos específicos del análisis.

2.1.4 Predicción de deserción

El identificar con antelación que cliente renunciará o finalizará su relación con la empresa, se conoce como predicción o pronóstico de deserción (Zhang et al., 2022), y para lograr esto, se basa en la ejecución de algoritmos (ver Figura 5), los cuales pueden ser individuales o híbridos, y su selección depende de las características del problema y de los datos disponibles.



Figura 5. Tipos de segmentación de mercado
Fuente. Elaborado por el autor

2.1.4.1 Algoritmos individuales.

Los algoritmos individuales son aquellos que, en la predicción, utilizan exclusivamente un único enfoque algorítmico para analizar y procesar datos. Estos algoritmos se dividen en dos grandes categorías según la metodología empleada: algoritmos basados en aprendizaje automático (machine learning, ML) y aquellos basados en aprendizaje profundo (deep learning, DL).

Aprendizaje automático

En el 2018, diversos estudios se enfocaron en su aplicación, como Random Forest, el cual obtuvo resultados significativos en dos estudios, por ejemplo, Malleswari et al., (2018) donde alcanzaron una precisión del 87%, mientras que Bell & Mgbemena, (2018) obtuvieron un 73.53%.

Adicionalmente, en el mismo año, otros algoritmos individuales fueron explorados, por ejemplo, Manjupriya & Poornima, (2018) y Jaisakthi et al., (2018) que utilizaron Árboles de decisiones y Boosting, destacando su efectividad. Por otro lado, Óskarsdóttir et al., (2018), adoptaron un enfoque diferente al emplear una técnica basada en series temporales, logrando una precisión del 88.8% de predicción.

En el 2019, se mantiene el uso de los algoritmos previamente mencionados, como Random Forest (Nagaraj et al., 2019), Árboles de decisiones (Deng et al., 2019) y Boosting (Ahmed & Maheswari, 2019); sin embargo, se incorporan algoritmos como XGboost cuyo trabajo alcanza un 79.8% de precisión (Pamina et al., 2019), Naive Bayes cuya investigación obtuvo un 84% (Sharm et al., 2019), SVM con un desempeño del 95.5% de precisión (Suguna et al., 2019), entre otros.

En 2020, se continuaron utilizando los algoritmos previamente mencionados; sin embargo, las técnicas de representación basada en grafos y Particle Swarm Optimization (PSO) presentaron un enfoque

distinto. La representación basada en grafos permitió capturar relaciones complejas entre entidades, presentando una perspectiva heterogénea y definiendo diferentes meta-caminos probabilísticos en gráficos de llamadas aumentados para predecir la deserción, obteniendo un 75% de precisión en el área bajo la curva (AUC). Por su parte, PSO, una técnica inspirada en el comportamiento colectivo de enjambres alcanzó una precisión de 89.44%, demostrando su capacidad de optimización en modelos predictivos.

En 2021, la técnica innovadora en predicción de churn que destacó fue el Bat Algorithm, el cual fue inspirado en el comportamiento de los murciélagos, mostrando un gran desempeño y subrayando la diversificación en metodologías de predicción de churn. Su enfoque distinto obtuvo una precisión de 99.66% sobre un dataset de 699 registros.

En 2022, se mantuvo el uso de los algoritmos previamente establecidos; sin embargo, la tendencia se orientó a una mayor integración de técnicas avanzadas de optimización y modelado, como Catboost que alcanzo un 81.8% de precisión sobre un dataset de 7000 registros (Lalwani et al., 2022) y AdaGrad (técnica de optimización basada en gradientes adaptativos) que mejoró la convergencia en problemas de alta dimensionalidad, logrando una precisión de 92.6% sobre un dataset de 7000 registros (Akbar & Chowanda, 2022).

En 2023, se continuaron utilizando los algoritmos mencionados anteriormente, como XGBoost con una precisión de 81.2% (Swetha & Dayananda, 2023), Naive Bayes con 98% (Amin et al., 2023) y SVM con 95% sobre un dataset de 51.047 registros (Babatunde et al., 2023). Estos resultados reflejan una consolidación en el uso de modelos ya establecidos, con mejoras en su desempeño.

La Tabla 6 presenta un resumen consolidado de los algoritmos individuales, basados en aprendizaje automático, que han sido utilizados para la predicción de churn en empresas de telecomunicaciones, abarcando investigaciones realizadas entre el 2018 y 2023, acorde al estudio provisto por Freire et al., (2024).

Tabla 6

Algoritmos basados en aprendizaje automático en procesos de predicción de churn

Algoritmo	Referencia	Algoritmo	Referencia
Adaboost	(Sabbeh, 2018)	Gradiente Boosted	(Bugajev et al., 2022)
AdaGrad	(Akbar & Chowanda, 2022)		(Mandić & Kraljević, 2022)
ALB (Adaptive logitboost)	(Hemalatha & Mahalakshmi, 2020)		(Jaisakthi et al., 2018)
Decision Tree	(Acero-Charaña et al., 2021)		(Ahmed & Maheswari, 2019)
	(Höppner et al., 2020)		(Nguyen et al., 2021)
	(Manjupriya & Poornima, 2018)		(Xu et al., 2021)
	(Deng et al., 2019)	XGBoost	(Arshad et al., 2022)
Bat algorithm	(Li et al., 2021)		(Ahmad et al., 2019)
Catboost	(Lalwani et al., 2022)		(Pamina et al., 2019)
Graphs	(Mitrović & De Weerd, 2020)		(Alzubaidi & Al-Shamery, 2019)
KNN	(Sjarif et al., 2019)		(AL-Shatnwai & Faris, 2020)
PSO	(Faritha Banu et al., 2022)		(Beschi & Chenthur, 2020)
	(Amali & Arunkumar, 2020)		(Swetha & Dayananda, 2020)
	(Poornappriya & Durairaj, 2019)		(Swetha & Dayananda, 2023)
	(Vijaya & Sivasankar, 2019)		(Sari et al., 2023)
	(Li & Marikannan, 2019)	Naive Bayes	(Toor et al., 2020)
Logistic Regression	(Ramana et al., 2019)		(Sharm et al., 2019)
	(Makinde et al., 2021)		(Amin et al., 2019)
Time Series	(Zhang et al., 2022)		(Amali et al., 2019)
	(Óskarsdóttir et al., 2018)		(Amin et al., 2023)
	(Rani & Kant, 2020)	SVM	(Amin et al., 2020)
	(Edwine et al., 2022)		(Loukili et al., 2022)
	(Wu et al., 2021)		(Mohan et al., 2021)
	(Ullah et al., 2019)		(Praseeda & Shivakumar, 2021)
	(Nagaraj et al., 2019)		(Jyothi et al., 2020)
Random Forest	(Eria & Marikannan, 2019)		(Suguna et al., 2019)
	(Malleswari et al., 2018)		(Lee & Chen, 2023)
	(Bell & Mgbemena, 2018)		(Babatunde et al., 2023)
	(Colot et al., 2021b)		

Fuente. Elaboración propia tomada de (Freire et al., 2024).

Aprendizaje profundo

En 2018, los trabajos en redes neuronales trabajaron con datasets de volúmenes significativamente mayores que los de los algoritmos de aprendizaje automático utilizados en esa época. Un ejemplo destacado es el trabajo de (Azeem & Usman, 2018), que alcanzó una precisión del 71% en el área bajo la curva (AUC) sobre un conjunto de datos de 200,000 registros. Otro caso relevante es el de (Yu et al., 2018), que obtuvo una precisión del 73.29% sobre un conjunto de datos de 100,000 registros.

En 2019, se observaron avances significativos con la incorporación de nuevas arquitecturas como ANN, DNN, CNN y LSTM. Sin embargo, la tendencia de trabajar con grandes volúmenes de información no se mantuvo, ya que se utilizaron conjuntos de datos de tamaños más variados. Por ejemplo, los estudios de Sivasankar & Vijaya, (2019) y Almufadi et al., (2019) trabajaron con datasets de 100000 registros, alcanzando una precisión del 94.62% y 96%, respectivamente. Mientras que otros estudios, como los de Khan et al., (2019), Arora et al., (2019) y Nigam et al., (2019), emplearon conjuntos de datos más pequeños de 20468, 7043 y 4914 registros, con precisiones de 90.9%, 86.4% y 85.94%, respectivamente. Estos resultados demostraron que la precisión mejoró considerablemente en comparación con el año anterior.

En 2020, se lograron avances significativos en la predicción de pérdida de clientes (churn) en la industria de las telecomunicaciones, impulsando nuevas perspectivas que mejoraron la precisión y relevancia de las predicciones, ejemplo de esto fue el trabajo de Alboukaey et al., (2020) el cual subrayó la importancia de realizar predicciones diarias, cuando el patrón común eran las predicciones mensuales. Este enfoque de analizar el comportamiento dinámico de los clientes a nivel de días, permitió identificar cambios sutiles en los patrones de conducta antes de que los clientes decidieran abandonar

el servicio, logrando una mejora considerable en la oportunidad y la precisión de las predicciones. Por otro lado, Jafari-Marandi et al., (2020) introdujeron un criterio innovador denominado costo de clasificación errónea a través del modelo ChP-SOEDNN, que combina aprendizaje auto organizado y dirigido por error, el cual evalúa el impacto financiero de las decisiones predictivas.

Además, Chouiekh & El Haj, (2020) se enfocaron en los clientes de servicios prepagos, utilizando como fuente principal de datos los registros de detalle de llamadas (CDR). Su metodología basada en redes neuronales convolucionales profundas (DCNN), destacó el potencial del aprendizaje profundo para reducir los costos asociados a la pérdida de clientes, además de que su enfoque resaltó la importancia de segmentar los análisis según las características específicas de los clientes.

En 2021, los avances en la predicción de churn se centraron en abordar limitaciones técnicas y optimizar la precisión mediante enfoques innovadores. Por ejemplo, Garimella et al., (2021) desarrollaron un modelo basado en redes neuronales convolucionales profundas (DCNN), optimizadas mediante el algoritmo adaptive firefly-spider optimization (adaptive FSO), combinando técnicas de Spider Monkey Optimization (SMO), Firefly Algorithm (FA) y conceptos adaptativos, logrando una precisión del 99.76% en su investigación. Por otro lado, Ewieda et al., (2021) analizaron el impacto de la calidad del servicio en la satisfacción del cliente y su relación con el churn, utilizando técnicas de minería de datos (DM), alcanzando un 96.9% de precisión. Asimismo, Pustokhina et al., (2021) también propusieron un enfoque que combinó la técnica de selección de características CPIO-FS (Chaotic Pigeon Inspired Optimization) con un modelo de clasificación basado en LSTM y autoencoders apilados (SAE) , alcanzando una precisión del 95.56%, lo cual demostró su eficacia en la reducción de la complejidad computacional y la mejora de la clasificación de datos.

Además, Usman et al., (2021) llevaron a cabo un análisis comparativo entre aprendizaje automático (ML) y aprendizaje profundo (DL), destacando cómo el DL supera las limitaciones de ML, evidenciando que las arquitecturas de redes neuronales convolucionales ofrecen mejores resultados.

En 2022, también se hicieron hallazgos importantes en el campo de la predicción de churn en empresas Telco, por ejemplo, Abdulsalam et al., (2022) utilizaron un algoritmo de selección de características mejorado denominado Relief-F, el cual les permitió seleccionar características relevantes de un conjunto de datos, y complementarlo con redes neuronales artificiales (ANN), logrando una precisión del 93.88%. Además, Mahmood & Abdullah, (2022) realizaron un gran aporte trabajando con el dataset más grande estudiado en el año, el cual tenía 2500 millones de registros, alcanzando una precisión del 91.63%. Por otro lado, Zhai et al., (2022) propusieron un modelo basado en redes LSTM, el cual permitió aprender de las preferencias de los comportamientos de los usuarios, utilizando un mecanismo de atención múltiple para capturar información colaborativa de los comportamientos de los usuarios.

En 2023, los avances en modelos de predicción de churn en empresas Telco continuaron, por ejemplo, Zhao et al., (2023) propusieron un modelo que combina la clasificación supervisada y no supervisada mediante K-means, análisis de entropía y un clasificador basado en atención multi-cabezal con LSTM anidados. Este enfoque mejoró la segmentación de clientes, permitiendo una mayor precisión en la predicción del comportamiento. Por otro lado, Haridasan et al., (2023) desarrollaron un modelo basado en memoria bidireccional apilada (SBLSTM), el cual fue optimizado con el algoritmo de optimización aritmética (AOA); esta combinación logró un rendimiento sobresaliente en predicción con ajustes óptimos de hiper parámetros.

Además, Abualkishik et al., (2023) introdujeron un enfoque innovador utilizando Stacked GRU (SGRU) combinado con el algoritmo Wind-Driven Optimization y Golden Sine, el cual fue utilizado en la selección de características. Este modelo destacó en aplicaciones empresariales, especialmente en telecomunicaciones y comercio electrónico, porque pudo reducir la complejidad del procesamiento de datos masivos. Por último, Saha et al., (2023) compararon técnicas de ensamblado como AdaBoost, Random Forest y redes neuronales, demostrando que las arquitecturas de aprendizaje profundo (CNN y ANN) ofrecieron resultados más precisos, ya que alcanzaron un 99% de precisión.

La Tabla 7 presenta un resumen consolidado de los algoritmos individuales, basados en aprendizaje profundo, que han sido utilizados para la predicción de churn en empresas de telecomunicaciones, el cual abarca investigaciones realizadas entre el 2018 y 2023, acorde al estudio realizado por Freire et al., (2024).

Tabla 7

Algoritmos basados en aprendizaje profundo en procesos de predicción de churn

Algoritmo	Referencia	Algoritmo	Referencia
ANN	(Fujo et al., 2022)	CNN	(Usman et al., 2021)
	(Jafari-Marandi et al., 2020)		(Rabbah et al., 2022)
	(Ramesh et al., 2022)		(Almufadi et al., 2019)
	(Abdulsalam et al., 2022)		(Saha et al., 2023)
	(Sivasankar & Vijaya, 2019)		(Dalli, 2022)
	(Khan et al., 2019)		(Sudharsan & Ganesh, 2022)
DCNN	(Garimella et al., 2023)	NN	(Colot et al., 2021a)
	(Garimella et al., 2021)		(Ewieda et al., 2021)
	(Almufadi & Qamar, 2022)		(Azeem & Usman, 2018)
	(Chouiekh & El Haj, 2020)		(Abdi & Abolmakarem, 2019)
DL (golden sine algorithm)	(Abualkishik et al., 2023)		(Awang et al., 2018)
DNN	(Mirza et al., 2022)		(Yu et al., 2018)
	(Mahmood & Abdullah, 2022)		(Allam & Srinivas, 2019)
	(Nigam et al., 2019)		(Xu et al., 2023)
LSTM	(Alboukaey et al., 2020)		(Zhai et al., 2022)
	(Pustokhina et al., 2021)		(Zhao et al., 2023)
	(Arora et al., 2019)		(Haridasan et al., 2023)

Fuente. Elaboración propia tomada de (Freire et al., 2024).

2.1.4.2 Algoritmos híbridos

Son la combinación de algoritmos individuales, es decir, la unión de dos o más algoritmos. Esta combinación busca aprovechar las fortalezas de cada algoritmo para mejorar el rendimiento y la eficiencia en la resolución de problemas complejos. Al combinar diferentes enfoques (unión de varios algoritmos), se pueden obtener soluciones más robustas y precisas, superando las limitaciones inherentes de los algoritmos individuales.

Por ejemplo, en 2018, Shokouhyar et al., (2018) combinaron cuatro algoritmos: árbol de decisiones (DT), red neuronal artificial (ANN), K-Nearest Neighbors (KNN) y regresión logística, aplicándolos sobre un conjunto de datos de 53418 registros, obteniendo una precisión del 96.9%, superando considerablemente los resultados obtenidos con cada algoritmos por separado.

Otro ejemplo de ese mismo año fue el trabajo realizado por Vijaya & Sivasankar, (2018), quienes combinaron cinco algoritmos: DT, KNN, redes bayesianas (NB), máquina de soporte vectorial (SVM) y ANN, sobre un conjunto de datos de 100 mil registros, obteniendo una precisión del 95.1%.

En 2019, se mantuvieron las combinaciones de algoritmos, destacando el trabajo de Idris et al., (2019), que marcó la diferencia al integrar las capacidades de búsqueda de la programación genética (GP) y las capacidades de clasificación de AdaBoost. Este sistema, denominado ChP-GPAB, aborda el problema del desequilibrio en los conjuntos de datos mediante un método de submuestreo basado en optimización por enjambre de partículas (PSO), logrando una distribución imparcial del conjunto de entrenamiento. El sistema ChP-GPAB no solo mejora la identificación de clientes propensos a abandonar, sino que también interpreta el comportamiento de abandono, logrando un área bajo la curva (AUC) de 0.91.

Otro ejemplo de ese mismo año fue el trabajo realizado por Brmez & Znidarsic, (2019), quienes combinaron SVM con Random Forest, añadiendo al SVM un núcleo (kernel) de función de base radial (SVM rbf), logrando alcanzar una precisión del 73% sobre un conjunto de datos de 16516 registros.

En 2020, Ajitha et al., (2020) propusieron un modelo híbrido para predecir el churn. Este modelo combina regresión logística y perceptrón multicapa (MLP), ejecutándose en paralelo y comparando los resultados a nivel de confrontación. El sistema automatizado no solo predice con alta precisión qué clientes pueden abandonar, sino que también reporta las razones del abandono y almacena los informes en una base de datos HBASE (base de datos distribuida y orientada a columnas, diseñada para manejar grandes volúmenes de datos no estructurados o semiestructurados en tiempo real, perteneciente al ecosistema de hadoop).

Otro trabajo destacado de 2020 fue realizado por Alzubaidi & Al-Shamery, (2020), quienes desarrollaron un modelo de predicción de abandono denominado Project Pursuit Random Forest (PPForest). Este modelo se distingue por la utilización de análisis discriminante para mejorar la construcción de árboles de decisión oblicuos (PPtree). La metodología propuesta combina SVM y análisis discriminante lineal (LDA) para lograr divisiones lineales de variables, produciendo clasificadores más robustos que los Random Forest clásicos. El modelo PPForest basado en LDA mostró un rendimiento superior en medidas como precisión, logrando un 99.1% sobre un conjunto de datos de 71 mil registros.

En 2021, Karuppaiah & Gopalan, (2021) presentaron una alternativa distinta con un modelo denominado SCHIE (stacked CLV-based heuristic incorporated ensemble) para la predicción de abandono de clientes. Este modelo se compone de dos niveles de predicción

seguidos de una recomendación para reducir el churn. El primer nivel implica identificar modelos efectivos para predecir posibles abandonos. Luego, se realiza la segregación de resultados, predicción basada en el valor del ciclo de vida del cliente (CLV) y la selección de usuarios para ofertas. Los resultados experimentales indican una alta eficiencia en la predicción de clientes propensos y no propensos a abandonar, alcanzando un 88% de precisión sobre un conjunto de datos de 3333 registros. Se encontró que el modelo propuesto reduce la pérdida general hasta en un 50% en comparación con los modelos más avanzados.

En 2022, Al-Shourbaji et al., (2022) propusieron un nuevo enfoque de selección de características (Feature Selection) denominado ACO-RSA, que combina dos algoritmos meta heurísticos: optimización por colonia de hormigas (ACO) y algoritmo de búsqueda de reptiles (RSA). Este enfoque fue evaluado en siete conjuntos de datos de predicción de abandono de clientes de código abierto y mostró ser superior en comparación con otros algoritmos competidores como PSO, MVO y GWO, logrando un 90.4% de precisión sobre un dataset de 3333 registros.

Otro trabajo destacado de 2022 fue realizado por Liu, Ali, et al., (2022), quienes propusieron un modelo de predicción de abandono de clientes basado en un sistema de ensamblaje que incorpora técnicas de aprendizaje de clasificación y agrupamiento. El modelo híbrido basado en apilamiento integro métodos de agrupamiento como k-medoids con técnicas de clasificación como Gradient Boosted Trees (GBT), árboles de decisión (DT) y aprendizaje profundo (DL), alcanzando el 93.6% de precisión sobre un dataset de 50 mil registros.

En 2023, Pustokhina et al., (2023) presentaron un modelo mejorado denominado ISMOTE-OWELM, diseñado para la predicción de abandono de clientes. Este modelo combina la técnica de sobre

muestreo sintético mejorado (ISMOTE) con el aprendizaje extremo ponderado óptimo (OWELM), empleando el algoritmo de optimización de lluvia multi objetivo (MOROA). Los resultados del modelo demostraron una alta precisión, ya que alcanzaron valores de 0.94, 0.92 y 0.909 en tres conjuntos de datos distintos, evidenciando su capacidad para abordar problemas de desbalanceo y mejorar la eficacia en escenarios reales.

Otro trabajo destacado de 2023 fue realizado por Jiang et al., (2023), quienes presentaron un modelo híbrido de predicción de abandono, que integra tanto las consideraciones de retorno como las de costo, marcando un enfoque diferente en la optimización de estrategias de detección de churn. Este enfoque distintivo combina técnicas de sobre muestreo y selección de características, utilizando la búsqueda orbital atómica multi objetivo modificada para ajustar de manera óptima los pesos y umbrales iniciales del aprendizaje extremo, maximizando así el beneficio de las predicciones.

Finalmente, Khoh et al., (2023) desarrollaron un sistema de predicción de abandono basado en un modelo de aprendizaje en ensamblaje ponderado optimizado. Este modelo utilizó los algoritmos KNN, Catboost y Random Forest, y les aplicó el algoritmo de optimización de Powell para ajustar los pesos de influencia de los aprendices base, con lo cual logró una precisión del 84% y una puntuación F1 score del 83.42% en un conjunto de datos de 71047 registros.

La Tabla 8 presenta un resumen consolidado de los algoritmos híbridos, que han sido utilizados para la predicción de churn en empresas de telecomunicaciones, abarcando investigaciones realizadas entre el 2018 y 2023, acorde al estudio provisto por Freire et al., (2024).

Tabla 8
Algoritmos híbridos en procesos de predicción de churn

Algoritmo	Referencia	# Registros	Exactitud	Churn (%)
ACO+RSA	(Al-Shourbaji et al., 2022)	3.333	90,4	14,49
Adaboost + PSO	(Idris et al., 2019)	50.000	0,91*	7,3
C5.0 + Lineal Tree Model	(Jamalian & Foukerdi, 2018)	51306	0,67*	1,8
DT+ANN+KNN+RL	(Shokouhyar et al., 2018)	53.418	96,9	-
DT+ANN+SVM	(Beeharry & Fokone, 2022)	1.470	82,3	-
DT+LR+ANN+KNN+NB	(Ahmed et al., 2020)	5.000	97,6	14,3
DT + LR + FT + RF	(Usman-Hamza et al., 2022)	5.000	96,6	10,14
DT + LR + RF + SVM	(Al-Mashraie et al., 2020)	100.304	94,7	50
DT + MLP	(Awang et al., 2021)	272	95,4	-
DT + NB	(Mishachandar & Kumar, 2018)	20.469	70	-
DT+KNN+NB+SVM	(Vijaya et al., 2019)	3.333	89,4	14,49
DT+KNN+NB+SVM +ANN	(Vijaya & Sivasankar, 2018)	100.000	95,1	49,56
Ensemble learning	(Liu, Fan, et al., 2022)	51.047	-	28,82
Extreme learning machine	(Pustokhina et al., 2023)	3.333	94,6	14,49
Extreme learning machine	(Jiang et al., 2023)	-	-	-
GB + DT	(Tang et al., 2020)	51.047	60,7	28,82
Gboost + DT	(Q.-F. Wang et al., 2019)	66.059	0,84*	36,62
K med+GBT+DT+DL	(Bilal et al., 2022)	5.000	94,7	14,14
k-medoids+GBT +DT+DL	(Liu, Ali, et al., 2022)	50.000	93,6	7,93
KNN+Catboost+RF	(Khoh et al., 2023)	71.047	86	28,82
LDA + SVM	(Alzubaidi & Al-Shamery, 2020)	71.000	99,1	29
PSO + RWN	(Faris, 2018)	5.000	96,3	7,6
RF + GB	(Jain et al., 2022)	3.333	97	14,49
RL + MLP	(Ajitha et al., 2020)	-	-	-
SBM + RF	(Brmez & Znidarsic, 2019)	16.516	73	39,45
Stacked CLV (SCHIE)	(Karuppaiah & Gopalan, 2021)	3333	88	14,49
SVM + J48	(Salunkhe & Mali, 2018)	3.333	89	14,49

Fuente. Elaboración propia tomada de (Freire et al., 2024).

2.1.5 Net Promoter Score (NPS)

Un aspecto a considerar es el indicador NPS (Net Promoter Score) que se basa en una sola pregunta "¿Qué probabilidad hay de que recomiende la empresa X a un amigo o colega?" (Reichheld, 2003). Este indicador sirve para medir el nivel de satisfacción y lealtad de los clientes, permitiendo identificar

el porcentaje de clientes que estarían dispuestos a recomendar nuestra empresa.

Se trata de una métrica que compara dos grandes grupos, a los promotores, que son aquellos clientes satisfechos que probablemente recomendarán la empresa, con los detractores, quienes no están satisfechos con el servicio brindado.

En la Figura 6 se presenta la escala de clasificación de los resultados del NPS, mostrando la distribución de los clientes en estas categorías y proporcionando una visión clara de las áreas de mejora necesarias para incrementar la satisfacción y lealtad del cliente.

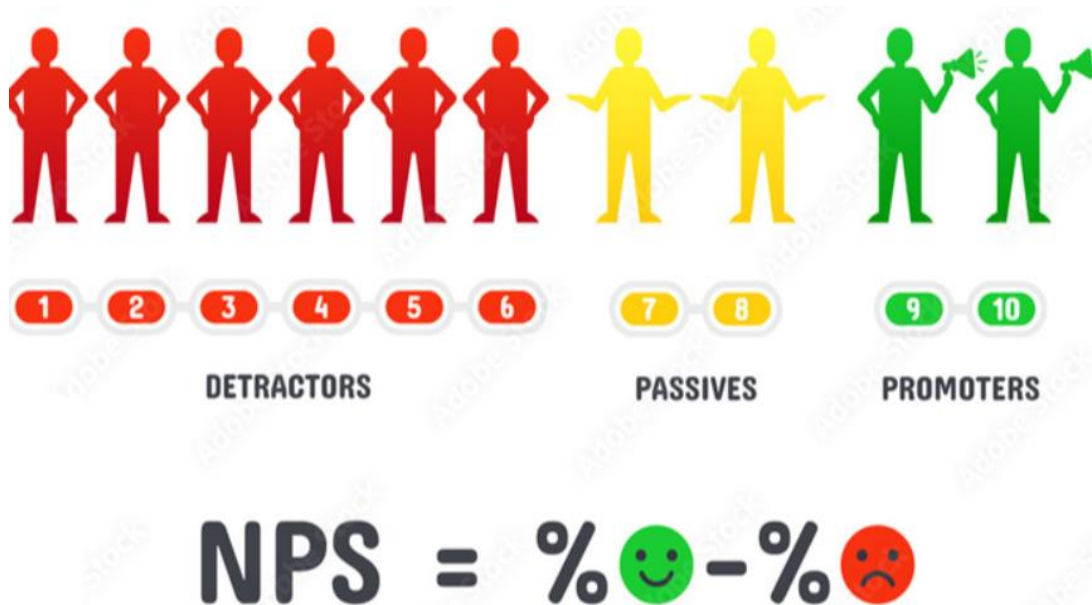


Figura 6. Calificación Net Promoter Score

Fuente: Tomado de (Reichheld, 2003)

2.2 Marco conceptual

Churn (Deserción de Clientes): Es la tasa de pérdida de clientes que dejan de utilizar los servicios de una operadora de telecomunicaciones en un período determinado. Es uno de los indicadores cruciales de las empresas

Telco, ya que un alto porcentaje de churn puede indicar problemas en la calidad del servicio o en la satisfacción de los clientes.

Tasa de deserción: También conocida como tasa de churn, y hace referencia al porcentaje que representa a los clientes que dejan de utilizar los servicios o que se cambian de una empresa Telco a otra, durante un período específico de tiempo.

Servicios móviles: Servicios de telecomunicaciones proporcionados a través de redes celulares, como la telefonía móvil, acceso a internet móvil, mensajería de texto, y otros servicios de datos.

Servicios fijos: Servicios de telecomunicaciones que requieren una conexión física, como la telefonía fija, internet de banda ancha a través de DSL o fibra óptica, y televisión por cable.

Abonado: Persona física o jurídica que ha contratado un servicio con la empresa de telecomunicaciones. Si el cliente aún está haciendo uso del servicio se considera abonado activo, en su defecto se considera abonado inactivo.

Cliente postpago: Cliente que tiene un contrato establecido con la operadora de telecomunicaciones, permitiéndole utilizar los servicios ofrecidos y pagar una factura mensual después de consumir dichos servicios.

Cliente prepago: Cliente que no tiene compromisos contractuales con la operadora y paga por adelantado antes de consumir los servicios.

Plan: Denominación de la oferta comercial que el cliente selecciona al contratar servicios de telecomunicaciones. Los planes pueden variar en términos de servicios incluidos, precios y beneficios adicionales.

Tarifa del plan: Costo en dólares asociado con la oferta comercial que el cliente ha elegido. La tarifa puede reflejar la cantidad de servicios incluidos y el nivel de servicio ofrecido.

Duración del plan: Período durante el cual el contrato del cliente está vigente. Puede variar desde contratos mensuales hasta acuerdos de largo plazo, e influye en las tarifas y beneficios ofrecidos.

Tipos de clientes: Clasificación de clientes en dos categorías principales: personas individuales, que contratan servicios para uso personal, y empresas, que requieren servicios para operaciones comerciales con planes más complejos y personalizados.

Tipos de servicios: Servicios de telecomunicaciones que incluyen opciones como servicios de voz, acceso a internet, o una combinación de ambos, dependiendo de las necesidades del cliente.

Tipo de planes: Variedad de planes ofrecidos por las operadoras que pueden incluir telefonía celular (voz y/o internet), módems para acceso a internet, netbooks con conectividad integrada, servicios de televisión, combinaciones de telefonía e internet por cable, y soluciones de comunicación máquina a máquina para aplicaciones industriales.

ARPU (Average Revenue Per User): Se refiere al Ingreso Promedio por Usuario y mide el ingreso promedio que una empresa genera por cada usuario o cliente durante un período específico (normalmente se mide por mes) y se obtiene dividiendo el ingreso total en el periodo de tiempo analizado contra el número total de abonados en ese mismo periodo.

Retención de clientes: Son las estrategias y acciones que se implementan para mantener a los clientes existentes y reducir la tasa de deserción. Estas acciones pueden incluir mejoras en el servicio al cliente, programas de lealtad, ofertas personalizadas, descuentos, entre otros.

Fidelización de clientes: Son las estrategias y acciones que se implementan para aumentar la lealtad de los clientes, incluyendo programas de recompensas, descuentos y/o beneficios personalizados, y mejoras en el servicio al cliente. Cabe recalcar que, la fidelización es crucial para la reducción de la deserción en empresas Telco y la maximización del valor de por vida del cliente.

Variables Telco: Son las variables específicas del sector de telecomunicaciones utilizadas en el análisis de datos, como el tiempo de uso del servicio, el tipo de plan, la frecuencia de llamadas realizadas y/o recibidas, el consumo de navegación de internet, los patrones de facturación, entre otros. Estas variables son fundamentales para la construcción de modelos predictivos y segmentación de clientes.

Variables No Telco: Se refieren a las variables externas al sector de telecomunicaciones que pueden influir en el comportamiento del cliente, como el número de establecimientos comerciales que posee el cliente, el tipo de vehículo, la ocupación laboral, el sueldo, entre otras. Cabe recalcar que, la integración de estas variables en modelos predictivos puede mejorar la precisión de estas.

Validación de modelo: Se refiere al proceso de evaluación de la precisión y efectividad de un modelo predictivo, para lo cual se utiliza un conjunto de datos de prueba. Cabe recalcar que, la validación de un modelo es fundamental para asegurar su funcionamiento en datos no vistos y de esta manera evitar el sobre ajustado a los datos de entrenamiento.

Clases desequilibradas: En algoritmos de clasificación se utilizan clases de dos tipos de valor, y se denomina desequilibrio, cuando ambas están proporcionalmente distantes entre sí. En el contexto de la predicción de deserción, puede haber muchos más clientes que no desertan comparados con los que sí lo hacen, lo que puede sesgar el rendimiento del modelo.

Confiabilidad de Predicción: Es una medida de cuán preciso y consistente es un modelo predictivo en sus estimaciones. En telecomunicaciones, el aseguramiento de la confiabilidad de las predicciones es esencial, ya que la toma de decisiones se basa en los datos, y el tener resultados sesgados puede afectar la retención de clientes y la optimización de servicios.

Aprendizaje Automático (Machine Learning): Se refiere al uso de algoritmos y modelos estadísticos que permiten a las computadoras mejorar en la realización de tareas a través de la experiencia. En telecomunicaciones, el aprendizaje automático se utiliza para predecir la deserción de clientes, optimizar redes, personalizar ofertas y mejorar la satisfacción del cliente.

Sobreajuste (overfitting): Es un problema que ocurre en modelos de aprendizaje profundo, cuando el modelo se ajusta demasiado a los datos de entrenamiento, es decir, solo funciona para el dataset en el que es probado y no sobre otros conjuntos de datos.

CAPÍTULO 3: METODOLOGÍA DE LA INVESTIGACIÓN

El presente trabajo corresponde a un tipo de investigación aplicada, de nivel explicativa, con enfoque cuantitativo, que busca predecir la deserción de clientes postpago de empresas de telecomunicaciones.

3.1 Hipótesis general

Un modelo basado en aprendizaje automático contribuye a la predicción de la deserción de clientes postpago en empresas de telecomunicaciones.

3.2 Hipótesis específicas

- Hipótesis 1 (H1): La construcción de un conjunto de datos que incorpora variables Telco y no Telco influyen significativamente en el modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones.
- Hipótesis 2 (H2): Los componentes en el diseño del modelo de aprendizaje automático, tienen un impacto significativo en la predicción de deserción de clientes postpago en empresas de telecomunicaciones.
- Hipótesis 3 (H3): La validación de un modelo de aprendizaje automático de predicción de deserción de clientes postpago en empresas de telecomunicaciones, incrementa significativamente la confiabilidad del modelo.

3.3 Identificación de variables

3.3.1 Variable Independiente

Modelo de predicción

3.3.2 Variable dependiente

Deserción de clientes

La Tabla 9 presenta la matriz de operacionalización de variables, en la cual se detallan las variables dependientes e independientes que serán utilizadas en el presente estudio. En esta matriz se describe cada variable, su definición conceptual y operacional, los indicadores que la representan y las escalas de medición empleadas, permitiendo una comprensión clara de cómo se abordará cada variable en el presente proceso de investigación, asegurando la consistencia y precisión en la recolección y análisis de los datos.

Tabla 9
Matriz de operacionalización de variables

VARIABLES	TIPO DE VARIABLES	DEFINICIÓN DE VARIABLE	DIMENSIONES O FACTORES	MEDICIÓN DE INDICADOR O CUMPLIMIENTO	DEFINICIÓN DE INDICADOR
Deserción de clientes	Dependiente	Clientes que renuncian al servicio o se cambian de una operadora a otra	Factores de deserción (Tiempo, recaudación, facturación, entre otros)	Porcentaje de clientes que desertan	Se divide el número de clientes que se van versus los que permanecen, obteniendo su relación porcentual
			Variables no Telco (salario, cantidad de establecimiento, gama de vehículo, entre otros)	Porcentaje de aporte al modelo	Indica el porcentaje de aporte que tienen las variables en el modelo de predicción
Modelo de predicción de deserción de clientes	Independiente	Técnicas y/o algoritmos para poder identificar con antelación que cliente renunciará	Algoritmos de predicción (Random Forest, XGBoost, entre otros)	F1 Score Recall	Indica el nivel de precisión que tiene el modelo al predecir.
			Ensamble de algoritmos de predicción (mezcla de algoritmos)	F1 Score Recall	Indica el nivel de precisión que tiene el modelo al predecir.

Fuente. Elaboración propia.

3.4 Desglose de hipótesis

Hipótesis general

- H0= Un modelo basado en aprendizaje automático no contribuye significativamente a la predicción de la deserción de clientes en empresas de telecomunicaciones.
- H1= Un modelo basado en aprendizaje automático contribuye significativamente a la predicción de la deserción de clientes en empresas de telecomunicaciones.

Para la comprobación de esta hipótesis se emplearán el test de McNemar y la prueba t de Student, las cuales son herramientas estadísticas que permiten evaluar cambios significativos en datos categóricos dependientes y diferencias en las medias de dos muestras, respectivamente, proporcionando un análisis robusto para la validación de la hipótesis, al identificar variaciones significativas y tendencias relevantes en los datos.

Hipótesis específica 1

- H0= Las variables Telco y no Telco no influyen significativamente en el modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones.
- H1= Las variables Telco y no Telco influyen significativamente en el modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones.

Para la comprobación de esta hipótesis, se analizarán las variables del dataset mediante los algoritmos de aprendizaje automático Random Forest y XGBoost. Estos algoritmos son empleados específicamente para identificar las variables más relevantes y optimizar el proceso de selección, lo que permite reducir la complejidad del modelo y mejorar

su rendimiento en tareas de clasificación, asegurando que el modelo esté alineado con los objetivos de la hipótesis planteada.

Hipótesis específica 2

- H0= Los componentes (features) en el diseño del modelo de aprendizaje automático no tienen un impacto significativo en la predicción de deserción de clientes postpago en empresas de telecomunicaciones.
- H1= Los componentes en el diseño del modelo de aprendizaje automático tienen un impacto significativo en la predicción de deserción de clientes postpago en empresas de telecomunicaciones.

Para la validación de esta hipótesis, se evaluará el rendimiento del modelo utilizando las métricas F1 Score y Recall, con el fin de determinar su efectividad en la clasificación y su capacidad para identificar correctamente los casos positivos.

Hipótesis específica 3

- H0= La validación de un modelo de aprendizaje automático de predicción de deserción de clientes postpago en empresas de telecomunicaciones, no hacen confiable el modelo.
- H1= La validación de un modelo de aprendizaje automático de predicción de deserción de clientes postpago en empresas de telecomunicaciones, hacen confiable el modelo.

Para la validación de esta hipótesis, se empleará la técnica denominada validación cruzada, la cual permite evaluar la capacidad de generalización del modelo, asegurando que su rendimiento sea consistente y confiable al ser aplicado a nuevos datos.

3.5 Tipo y diseño de investigación

La presente investigación es de tipo aplicada, con un diseño de nivel explicativo y un enfoque cuantitativo. Esta investigación busca medir la precisión del modelo predictivo de deserción de clientes mediante la recolección y análisis de datos, encontrando relaciones causales entre las variables que influyen en la deserción de clientes postpago en empresas Telco.

3.6 Unidad de análisis

Para probar el modelo de predicción de deserción de clientes, se definió como unidad de análisis a los clientes postpago, los cuales son personas naturales que tienen servicios de telefonía celular, excluyendo a los clientes que cuentan con servicios que no corresponden a telefonía celular, tales como: televisión, telefonía por cable, internet por cable, televisión satelital, comunicaciones máquina a máquina, internet de las cosas, tabletas, módems y computadoras portátiles.

Este enfoque permite centrar el estudio en un segmento específico del mercado que es de gran relevancia para las operadoras de telecomunicaciones.

3.7 Población de estudio

La población de estudio está conformada por el conjunto de clientes postpago de telefonía celular pertenecientes a las empresas de telecomunicaciones del Ecuador. Estas empresas están reguladas por la Agencia de Regulación y Control de las Telecomunicaciones del Ecuador (Arcotel), que es la entidad encargada de regular, controlar y manejar las cifras oficiales del sector (Arcotel, 2024). Según el último informe de Arcotel del año 2024, se presenta la distribución de clientes entre las tres principales empresas de telecomunicaciones del país (ver Figura 7).

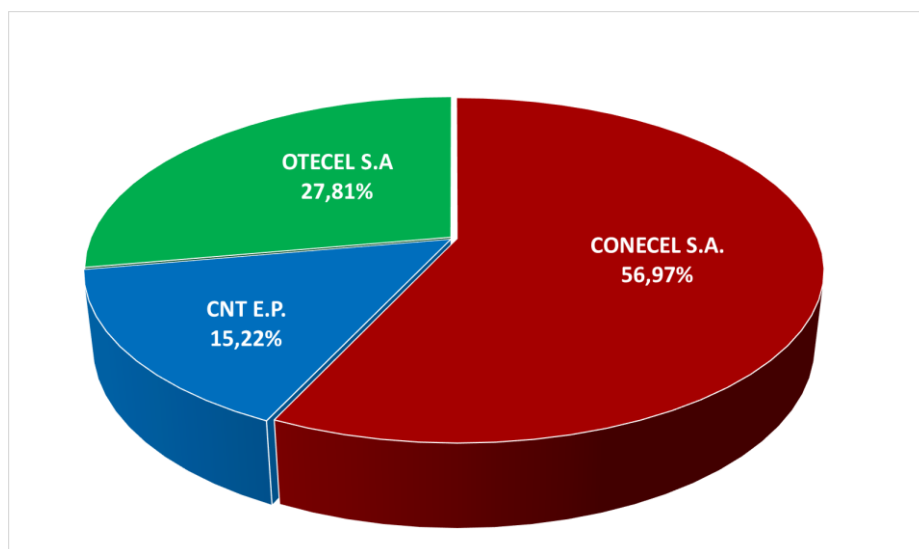


Figura 7. Número de abonados postpago en empresas Telco en Ecuador

Fuente: Elaboración propia tomada de (Arcotel, 2024)

Además, se han definido criterios de inclusión y exclusión detallados en la sección 3.9, delimitando la muestra a clientes postpago que sean personas menores a 65 años y que únicamente posean el servicio de telefonía móvil (ver Figura 8), asegurando que el análisis sea representativo de los usuarios más propensos a cambiar de proveedor y que el modelo predictivo sea aplicable a situaciones reales del mercado ecuatoriano).

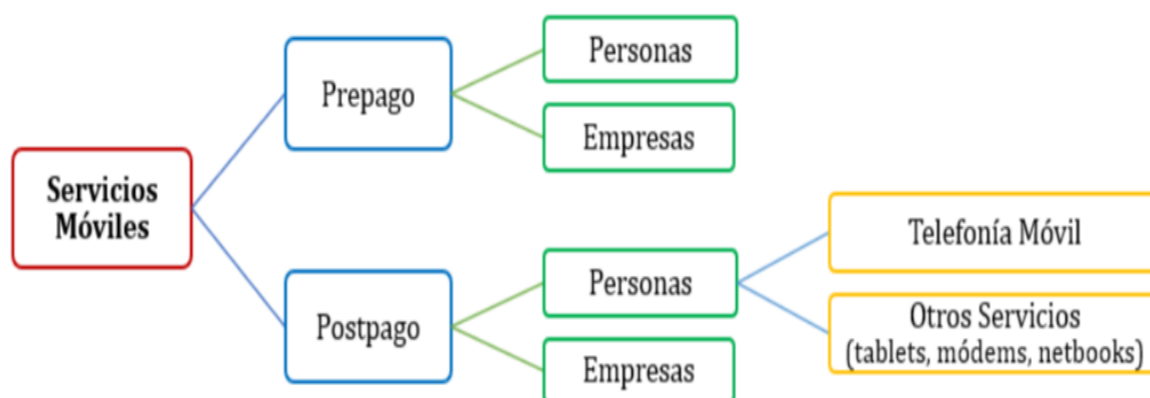


Figura 8. Desglose de muestra seleccionada

3.8 Tamaño de la muestra

Para la validación del modelo propuesto, se ha determinado que el tamaño de la muestra será de 77,520 registros. Este tamaño es una representación adecuada de la población de estudio y proporciona suficiente variabilidad en los datos.

El conjunto de datos será particionado en dos partes:

- **Conjunto de Entrenamiento:** Aproximadamente el 70-80% de los registros se asignarán al entrenamiento del modelo. Esta parte del conjunto de datos se utilizará para ajustar los parámetros del modelo y aprender patrones relevantes para la predicción de deserción de clientes en empresas Telco.
- **Conjunto de Validación:** El 20-30% restante de los registros se destinarán a la validación del modelo. Este subconjunto permite evaluar la capacidad del modelo para generalizar a nuevos datos, de tal manera que se asegura que las predicciones sean fiables y precisas, considerando que la validación es una etapa crítica, ya que ayuda a identificar problemas de sobreajuste y a realizar ajustes necesarios antes de la implementación final del modelo.

Este enfoque de particionamiento no solo garantiza que el modelo sea capaz de aprender efectivamente de los datos, sino que también proporciona una medida objetiva de su rendimiento, ya que se simula su aplicación en el mundo real. Cabe recalcar que, la elección del tamaño de la muestra y su particionamiento está alineada con las mejores prácticas en el desarrollo de modelos predictivos en el ámbito de las telecomunicaciones.

3.9 Selección de la muestra

La selección de la muestra será no probabilística y aleatoria, aplicando los siguientes criterios de exclusión, los cuales permitirán garantizar que los datos sean relevantes y significativos para el modelo de predicción de deserción de clientes postpago de empresas de telecomunicaciones:

- Clientes mayores a 65 años: En Ecuador, estos clientes son categorizados como tercera edad (adulto mayor) y reciben un beneficio del 50% de descuento en su facturación de servicio de telefonía móvil. Este descuento podría influir en su comportamiento de deserción de manera no representativa para el resto de la población.
- Cliente prepago: Se excluyen ya que no poseen un contrato con la operadora, es decir, no tienen un compromiso jurídico que los obligue a pagar una mensualidad por el servicio recibido, lo que puede resultar en patrones de uso y deserción diferentes.
- Clientes con un estado de inactividad o mora mayor a 45 días: Se excluyen aquellos clientes que tienen mora y cuyo servicio está limitado o restringido debido a impagos, ya que su comportamiento no es típico del cliente postpago activo.
- Clientes empresas: Se excluyen las personas jurídicas debido a que su comportamiento puede diferir significativamente del de las personas naturales, y sus necesidades de servicios de telecomunicaciones son generalmente más complejas.
- Servicios (planes) que no son de telefonía celular: Se excluyen planes de prueba, de máquina a máquina, de internet de las cosas, de geolocalización, de computadoras y de módems para centrar el análisis exclusivamente en la telefonía celular.

- Planes especiales: Se excluyen los planes de empleados y de clientes relacionados con la operadora telefónica (parte del mismo holding), ya que pueden tener condiciones preferenciales que afectan su comportamiento de deserción.
- Clientes con menos de 3 meses de facturación: Se excluyen clientes recientes para asegurar que el modelo de predicción tenga suficientes datos históricos para aprender y generar predicciones precisas.

Adicionalmente, se incluyeron otros conjuntos de datos, los cuales servirán para el enriquecimiento de la información y mejorar la precisión del modelo, los cuales son:

- Agencia de tránsito: Proporciona información relacionada con el o los vehículos que posee el cliente.
- Agencia de rentas internas: Proporciona información relacionada con el o los locales de negocio que posee el cliente.
- Agencia de seguridad social: Proporciona información acerca del trabajo, en relación de dependencia, que posee el cliente.
- Encuesta NPS: Mide el nivel de satisfacción del cliente con respecto al asesor y al centro de atención en el que fue atendido.

3.10 Técnicas de recolección de datos

Para la presente investigación, se utilizará una combinación de fuentes de datos privadas y públicas, con las cuales se construirá un conjunto de datos robusto y enriquecido que permita el desarrollo y validación del modelo de predicción de deserción de clientes postpago de telefonía móvil.

Las fuentes de datos son las siguientes:

- **Base de Datos Privada de una Operadora de Telecomunicaciones:**
Esta base de datos proporcionará información detallada sobre los clientes postpago de telefonía celular, incluyendo variables como: facturación, uso de servicios, tipo de plan, tipo de equipo, entre otros. Con esta información se puede identificar patrones de comportamiento, además de factores que influyen en la deserción de clientes postpago de servicio de telefonía móvil de empresas de telecomunicaciones.
- **Bases de Datos de Entidades Públicas:**
 - **Servicio de Rentas Internas (SRI):** Esta base de datos proporcionará información los locales de negocio que posee el cliente y su actividad comercial. Con estos datos se puede inferir información sobre la estabilidad económica y el perfil de riesgo de los clientes (*Datasets - Servicio de Rentas Internas, 2022*).
 - **Agencia Nacional de Tránsito (ANT):** Esta base de datos proporcionará información sobre la posesión de vehículos del cliente y su avalúo comercial. Con estos datos se puede inferir información acerca de la capacidad económica y nivel de endeudamiento del cliente (*Agencia Nacional de tránsito, 2022*).
 - **Instituto de Seguridad Social:** Esta base de datos proporcionará detalles sobre la afiliación laboral de los clientes, su cargo y remuneración mensual aproximada. Con estos datos se puede inferir la capacidad de pago y estabilidad laboral de los clientes (*Dataset - Instituto Ecuatoriano de Seguridad Social, 2022*).

- **Encuesta NPS (Net Promoter Score):** Esta encuesta realizada a clientes de la operadora de telecomunicaciones medirá el nivel de satisfacción del cliente respecto al servicio recibido, incluyendo la atención en los centros de servicio.

La integración de estas diversas fuentes de datos permitirá un análisis más completo y matizado, mejorando la capacidad del modelo para predecir con precisión la deserción de clientes. Al combinar datos internos con información externa relevante, se espera obtener un entendimiento más profundo de los factores que afectan la retención y satisfacción del cliente.

3.11 Técnicas de análisis de datos

En el marco de la investigación propuesta, se llevará a cabo un enfoque metodológico, el cual incluye las siguientes herramientas:

- **Análisis exploratorio de datos:** Esta herramienta tiene como objetivo explorar y comprender el comportamiento y las características intrínsecas del conjunto de datos. Permite identificar patrones, anomalías y relaciones entre variables, lo cual facilita una comprensión profunda del dataset (Wu et al., 2021).
- **Validación de variables:** Para garantizar la pertinencia de las variables seleccionadas, se emplearán las técnicas estadísticas Pearson (Sjarif et al., 2019), Kendall (Garimella et al., 2021) y Spearman (Xia et al., 2022). Estas técnicas se utilizarán para determinar la correlación entre la variable objetivo y las demás variables del dataset, asegurando que se mantengan aquellas que realmente aportan valor al modelo predictivo.
- **Evaluación de importancia de variables:** Para cuantificar la relevancia de cada variable en relación con la variable objetivo, se utilizarán

algoritmos de aprendizaje automático, como Random Forest (Jain et al., 2022) y XGBoost (AL-Shatnwai & Faris, 2020), los cuales pueden identificar las variables más influyentes que contribuirán a la optimización del modelo predictivo.

- Técnica greedy stepwise: Para contribuir con la selección de variables, se utilizará la técnica denominada greedy stepwise basada en Random Forest (Alzubaidi & Al-Shamery, 2020). Esta técnica evalúa iterativamente las características, añadiendo y eliminando variables en función del rendimiento del modelo, reduciendo así la dimensionalidad del dataset y manteniendo solo las variables significativas.
- Balanceo de datos: Para abordar el desequilibrio en los datos, se aplicarán métodos de balanceo como Smote, Adasyn y borderline Smote (AL-Shatnwai & Faris, 2020), los cuales generan nuevas instancias sintéticas para la clase minoritaria, mejorando de esta manera, la capacidad predictiva del modelo al reducir el sesgo inherente hacia la clase mayoritaria.
- División de dataset (Split): Se dividirá el conjunto de datos en dos partes, una para entrenamiento y otra para pruebas. Esto tiene como objetivo evitar la reutilización del conjunto de entrenamiento para pruebas y prevenir el sobreajuste mediante la aplicación de clasificadores (Loukili et al., 2022).
- Limpieza de datos: Autores como Lalwani et al., (2022), indican que se debe limpiar los datos, eliminando las inconsistencias y gestionando los valores nulos o faltantes en el dataset, lo cual es secundado por Usman et al., (2021).
- Codificación de Etiquetas (Label encoding): Esta herramienta tiene como objetivo, transformar las variables categóricas en representaciones numéricas para el procesamiento de datos en los

algoritmos de aprendizaje automático. Una de las técnicas utilizadas es Label Encoding, que convierte categorías en números enteros continuos (Fujo et al., 2022), por ejemplo, la variable categórica “género” se codifica en etiquetas 0 y 1 (Allam & Srinivas, 2019). Otra técnica utilizada es One-Hot Encoding, que transforma cada valor categórico en una variable binaria individual, permitiendo al modelo capturar mejor la información inherente a estas variables, por ejemplo, para la variable “establecimiento”, se crean características individuales para cada valor categórico, con valores Si (1) o No (0), generando nuevas columnas o variables al dataset (Ullah et al., 2022).

- **Procesamiento:** Para la predicción de deserción, se seleccionarán algoritmos de aprendizaje automático validados en la literatura, tales como XGBoost (AL-Shatnwai & Faris, 2020), Random Forest (Ullah et al., 2019), LGBM (Sari et al., 2023). Estos modelos serán entrenados para calcular la probabilidad de deserción, proporcionando un valor cuantitativo que indique la propensión de un cliente a desertar.
- **Identificación de verdaderos positivos (TP):** Indica que parte de los datos se marca adecuadamente como positiva y Fujo et al., (2022) lo conceptualizan como un cliente que es churn y que fue clasificado como churn, se denomina “True Positive”.
- **Identificación de ratio de verdaderos positivos (TPR):** Se calcula dividiendo los verdaderos positivos sobre los positivos actuales (AP). (Praseeda & Shivakumar, 2021).

$$TPR = TP / AP \quad (1)$$

- **Identificación de falsos positivos (FP):** Indica que parte de los datos se marcan falsamente como positivos y Fujo et al., (2022) lo conceptualizan como un cliente que no es churn y que fue clasificado como churn, se denomina “False Positive”.

- Identificación de ratio de falsos positivos (FPR): Se calcula dividiendo los falsos positivos sobre los negativos actuales (AN). (Praseeda & Shivakumar, 2021).

$$\text{FPR} = \text{FP} / \text{AN} \quad (2)$$

- Identificación de falsos Negativos (FN): Indica que parte de los datos se marcan falsamente como negativos y Fujo et al., (2022) lo conceptualizan como un cliente que es churn y que fue clasificado como no churn, se denomina “False Negative”.
- Precisión: También se denomina valor predictivo positivo, y muestra que parte de los datos de predicción es positiva.

$$\text{Precisión} = \text{TP} / (\text{TP} + \text{FP}) \quad (3)$$

- Recall: Indica la puntuación real del algoritmo, reflejando la capacidad del proceso para identificar qué proporción de instancias relevantes han sido correctamente identificadas por el modelo de predicción.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (4)$$

- Exactitud (accuracy): Indica cuan cerca esta una medida del valor verdadero o aceptado.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (5)$$

- Ensamble de algoritmos: Para mejorar la precisión predictiva, se aplicarán técnicas de ensamblado de algoritmos, también conocidas como métodos híbridos. Estos combinan múltiples modelos de aprendizaje automático, potenciando las fortalezas de cada uno y compensando sus debilidades, con el objetivo de optimizar el rendimiento general en la predicción de la deserción (Beeharry & Fokone, 2022).

CAPÍTULO 4: DESARROLLO DE LA PROPUESTA

4.1 Construcción del conjunto de datos (Telco y no Telco)

En esta sección se abordan los pasos seguidos para validar las hipótesis y alcanzar los objetivos propuestos en esta investigación.

4.1.1 Identificación de variables del dataset Telco

Acorde a lo indicado en el capítulo 3, referente a la población de estudio y la selección de muestra, se obtuvo un dataset que cumple con los criterios de exclusión establecidos y que consta de 77.520 registros, de los cuales 1255 corresponden a la clase objetivo, lo que representa un churn del 1,62%. Este dataset Telco, consta de 67 variables, de las cuales 51 son numéricas y 16 son categóricas.

Las variables numéricas incluyen métricas como renta, megas navegados, cantidad de llamadas, entre otros, mientras que las variables categóricas comprenden factores como el comportamiento de pago, marca y modelo del dispositivo, entre otros.

4.1.2 Asociación de variables con factores de deserción

Una vez identificadas las variables del dataset, se procede a asociarlas con los factores de deserción previamente identificados en el estudio de Freire-Vera et al., (2024), donde se asocian las 67 variables del dataset Telco, excluyendo la variable churn, lo cual se puede visualizar en la Figura 9.

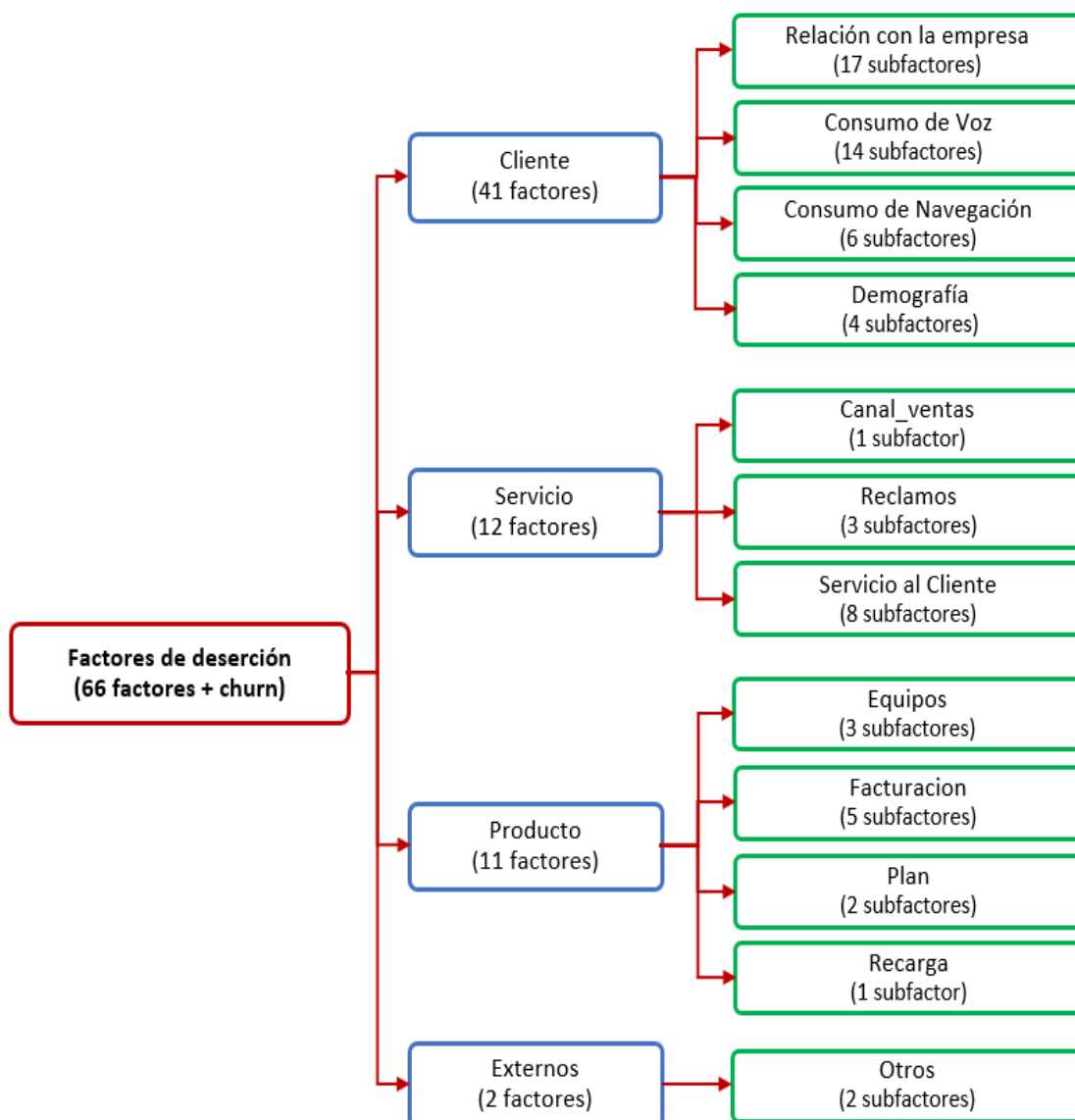


Figura 9. Asociación de variables del dataset con factores de deserción

4.1.3 Unificación de dataset Telco con no Telco

Adicional al dataset compuesto por variables Telco, se obtuvieron 4 fuentes de datos que aportan con variables al dataset principal, las cuales son: Agencia de rentas internas, agencia de seguridad social, agencia de tránsito y una encuesta realizada para obtener el indicador NPS. Esta fusión dio como resultado un dataset final conformado por variables Telco y variables que no pertenecen al sector de telecomunicaciones (variables no Telco), lo cual se puede visualizar en la Figura 10.

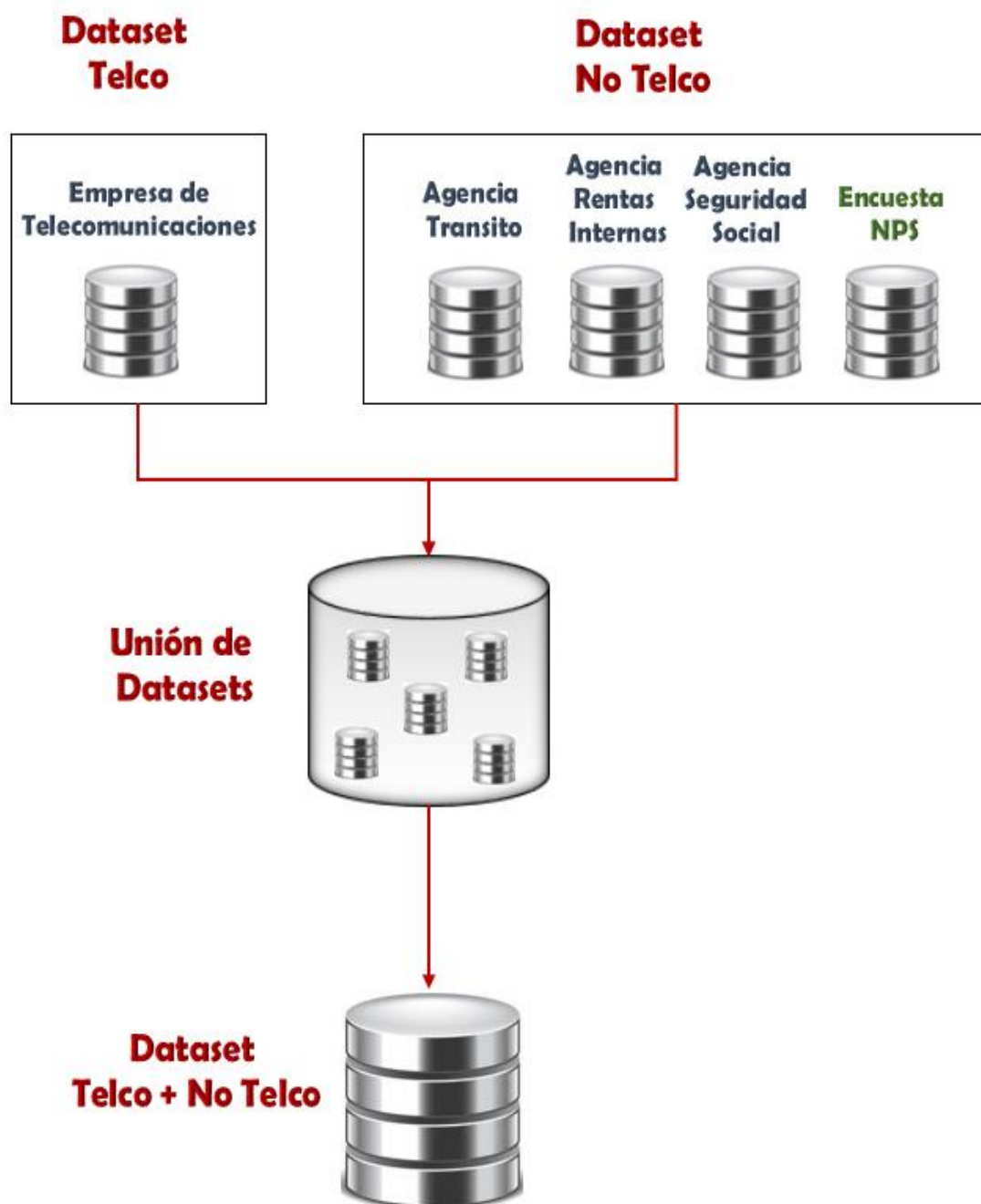


Figura 10. Desglose de muestra seleccionada

El dataset final cuenta con 87 variables, de las cuales 56 son numéricas, 30 son categóricas y la variable objetivo (churn). En la Figura 11, se puede visualizar la distribución de variables por cada uno de los datasets, siendo el de telecomunicaciones el que aporta con la mayor cantidad de variables (77%), mientras que los otros 3 aportan con el 23%.

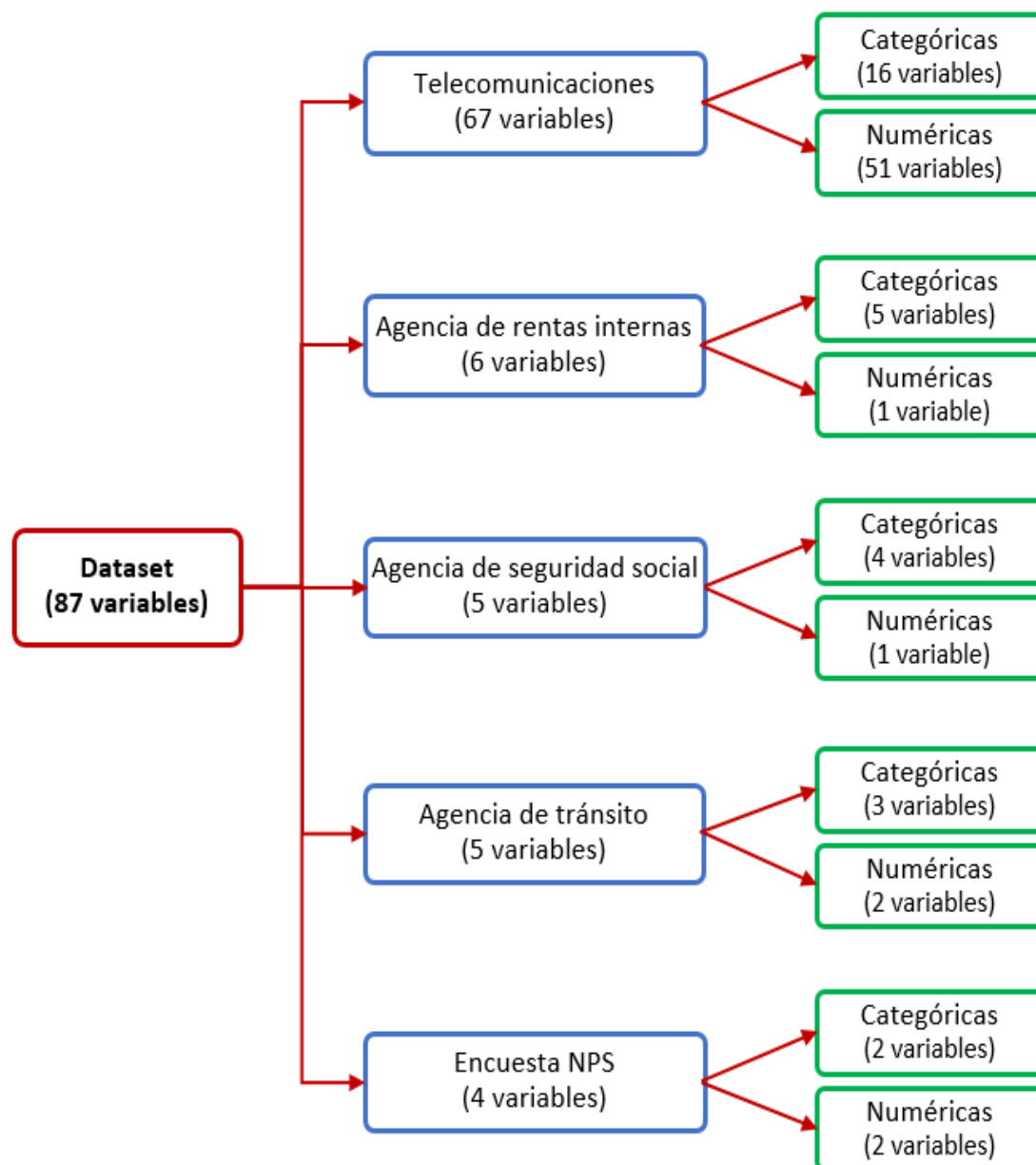


Figura 11. Desglose de variables del dataset

4.2 Diseño de modelo de aprendizaje automático

4.2.1 Análisis exploratorio de variables

Una vez obtenido el conjunto de datos, se lleva a cabo un análisis exploratorio con el objetivo de comprender la distribución de valores de cada variable, identificar datos anómalos, detectar variables con un alto porcentaje de valores nulos y examinar la dispersión de los datos.

En una fase inicial, se identifican las variables que presentan un exceso de valores nulos, las cuales se resumen y presentan en la Tabla 10. Este paso ayuda a decidir acciones adecuadas para su tratamiento, tales como la imputación, transformación o eliminación, dependiendo de su impacto en el análisis y en el modelo a desarrollar.

Tabla 10

Matriz de resultados de modelos (F1 Score)

Variable	Valores Nulos	Total de Registros	% de Participación
NPS_ASESOR_CATEG	76.992	77.520	99%
NPS_CAC_CATEG	76.983	77.520	99%
MORA_FIJO	71.001	77.520	92%
ANTIG_ULTIMO_FICIAMIENTO	67.585	77.520	87%

Luego de esto, se realiza un análisis detallado de las variables del conjunto de datos. En la Figura 12, se destaca la presencia de variables relacionadas con la cantidad de minutos realizados (por ejemplo, “cantidad_min_internos_ppa”), la facturación, la edad, la provincia y la antigüedad, además se visualizan las distribuciones de datos que pueden ser dispersas o concentradas en un único punto, lo que puede complicar la identificación de patrones diferenciales significativos para el modelo de predicción de deserción de clientes.

Debido a esto, se recurre a representaciones gráficas de los datos, lo cual permite visualizar de manera clara la dispersión o concentración de las variables. Esta visualización facilita el proceso de descartar variables que muestran patrones similares y concentrarse en aquellas que presentan patrones diferenciales, los cuales pueden aportar de manera significativa al modelo de predicción de deserción. Este enfoque ayuda a mejorar la capacidad del modelo para captar las complejidades inherentes en los datos, además de su precisión predictiva.



Figura 12. Desglose de variables – Parte 1

En la Figura 13, se puede visualizar un análisis de dispersión y concentración de datos en variables relacionadas con características como el tipo de vehículo, número de establecimiento, tarifa, cantidad de líneas prepago y exceso en la fecha máxima de pago. Este análisis permite identificar patrones significativos, por ejemplo, se observa que las variables “total_mb_2g_m”, “total_mb_3g_m” y “total_mb_4g_m” pueden ser combinadas en una sola variable que represente el total de megas navegados por el cliente. Además, se sugiere que las variables categóricas como “cat_comportamiento_pago”, “nps_categ”, “producto”, “device_type”, “nivel_socioeconomico” y “descripción_forma_pago”, sean transformadas mediante técnicas como Label Encoding. Esta transformación convierte las categorías en valores numéricos, lo cual es crucial para su inclusión en modelos de predicción, especialmente aquellos basados en algoritmos que requieren entradas numéricas. La metodología para aplicar Label Encoding se detalla en el

capítulo 3, asegurando que se mantenga la consistencia metodológica a lo largo del estudio.



Figura 13. Desglose de variables – Parte 2

En la Figura 14, se identifican datos anómalos en las variables, “ingreso_excedente_granel”, “ingresos_adicional_paq” y “megas_navegados”. La presencia de estos valores atípicos es un aspecto crítico que considerar, ya que pueden distorsionar los resultados y afectar la precisión de los modelos predictivos. De manera similar, las variables “modelo”, “tipo_empresa_laboral” y “ocupación” presentan problemas de dispersión y concentración de datos. Estas características pueden complicar tanto el análisis como la predicción dentro de los modelos utilizados.

Es importante identificar con antelación los patrones de los datos que se poseen, ya que son la base para los siguientes pasos de construcción del modelo de predicción de deserción de clientes. Con esta información, se sabrá que técnica implementar a cada una de las variables, por ejemplo, limpieza y preprocesamiento de datos, normalización o transformación de variables, y el uso de métodos robustos que sean menos sensibles a los valores atípicos,

incluso técnicas de reducción de dimensionalidad para manejar la dispersión excesiva y mejorar la interpretabilidad de las variables.

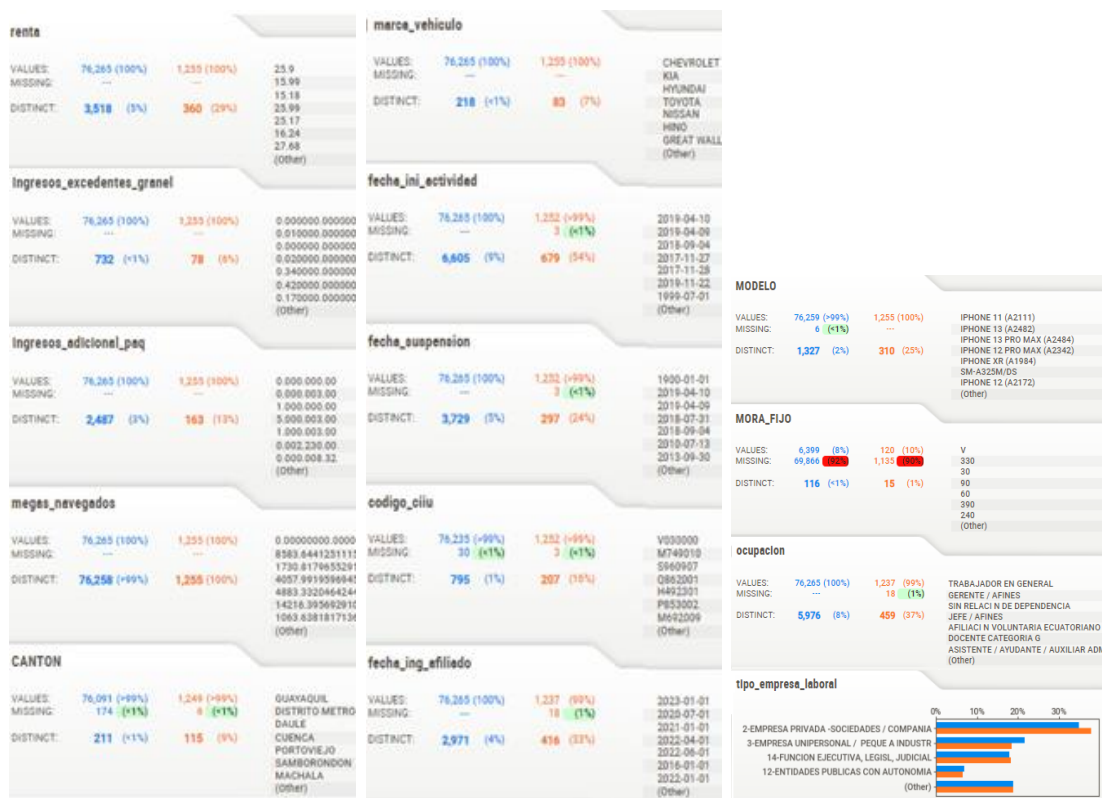


Figura 14. Desglose de variables – Parte 3

4.2.2 Pre – Procesamiento de datos

Una vez realizado el análisis exploratorio, se visualizan los tipos de datos de las variables, buscando la coherencia. Por ejemplo, las fechas se estandarizan en formato yyyy-mm-dd, la edad se convierte en valor numérico, y se verifica que las variables numéricas no contengan caracteres alfabéticos.

Posteriormente, se procede a dividir lógicamente el dataset en dos partes: una con variables numéricas y otra con variables categóricas. Esta separación facilita las imputaciones, aplicando la media y la moda respectivamente (ver Figura 15), con el fin de eliminar los datos nulos cuando la proporción en comparación con el total de la variable es mínima.



Figura 15. Proceso de imputación de datos

Adicional, se toman las variables con distribución de datos dispersa y/o centradas en un solo valor, y se procede a *crear nuevas variables* (ver Figura 16), agrupando una o más variables, buscando la distribución normal de datos.



Figura 16. Nuevas variables para reducir la variabilidad

4.2.3 Selección de variables

Culminado el pre - procesamiento de datos, se procede a identificar las variables que no aporten a los algoritmos de predicción de deserción de clientes postpago en empresas de telecomunicaciones, para lo cual se realizó

la eliminación de 3 variables con datos anómalos y de 4 variables con más del 80% con ausencia de valores o valores nulos, acorde a lo identificado en el análisis exploratorio.

Luego se ejecutaron las técnicas de correlación estadísticas Spearman, Pearson, Kendall (ver Tabla 11), eliminando 3 variables con correlaciones cercanas o iguales a 0 con respecto a la variable objetivo (rango mayor a -0.005 y menor a 0.005).

Tabla 11

Variables con baja correlación usando técnicas estadísticas

Variable	Correlación spearman	Correlación pearson	Correlación kendall
CANTIDAD_PROGR_COMPETENCIA	-0.000714	-0.001024	-0.000714
DIAS_SUSPENSION	-0.001596	-0.001078	-0.001596
PORC_MIN_COMPETENCIA	-0.000695	0.001070	-0.000693

Además, se ejecutaron algoritmos de clasificación XGBoost y Random Forest (ver Tabla 12), eliminando 6 variables con correlaciones cercanas o iguales a 0 con respecto a la variable objetivo (rango mayor a -0.001 y menor a 0.001).

Tabla 12

Variables con baja correlación usando algoritmos de machine learning

Variable	Valor_x	Valor_y
CANT_VISITAS_CENTROS_ATENCION	0.044590	0.000000
SOLICITUD_PORTABILIDAD_POR_DNI	0.009278	0.000000
MODULO_NEGOCIACION	0.001008	0.000000
NPS_CAC	0.000987	0.008348
SDA_SERVICIO	0.000582	0.000000
SDA_IDENTIFICACION	0.000463	0.000000

Adicional, se ejecutó la técnica greedy stepwise (ver Tabla 13), la cual determino las mejores correlaciones.

Tabla 13*Variables con alta correlación usando técnica greedy stepwise*

Variable	Score	Variable	Score
ANIO_VEHICULO	0.9896	PORC_MIN_INTERNOS	0.9890
FACTURACION	0.9895	CUPO_PRECALIFICADO	0.9890
LLAMADAS_CALL_CENTER_DNI	0.9895	PORC_MIN_INTERNOS_PPA	0.9890
LLAMADAS_CALL_CENTER_SERVICIO	0.9894	SDA_IDENTIFICACION	0.9890
TOTAL_MB	0.9894	MINS_ENTRANTES	0.9889
NPS_CAC	0.9893	CANT_LLAMADAS_ENTRANTES	0.9889
INBOUND_SERVICIO	0.9893	INBOUND_DNI	0.9889
NPS_ASESOR	0.9892	PORC_MIN_INTERNOS_POST	0.9889
PORC_RECAUDACION	0.9892	SUSP_COBRANZA	0.9886
FACTURACION_FIJO	0.9892	DIAS_USO_VOZ	0.9886
SOLICITUD_PORTABILIDAD_POR_SERVICIO	0.9891	VISITAS_INTENCION_RENUNCIA	0.9884
CANT_LLAMADAS_SALIENTES	0.9891	PORC_COMPETENCIA	0.9868
SDA_SERVICIO	0.9891	CANT_VISITAS_CENTROS_ATENCION	0.9866
MODULO_NEGOCIACION	0.9891	MINUTOS_SALIENTES	0.9836
CANT_LLAM_CALLCENTER	0.9890	VALOR_PAGADO	0.9821
SOLICITUD_PORTABILIDAD_POR_DNI	0.9890	ACTIVACIONES_PAQUETE	0.9819
VISITAS_POR_RENUNCIA	0.9890		

Culminado el proceso de selección de variables, se toman todas las variables categóricas, y se aplican dos técnicas de encoding, una donde se coloca de forma vertical los valores de las variables ESTABLECIMIENTO, ANTIGUEDAD_CLTE, GAMA_ASIGNADA_DISPOSITIVO, CAT_COMPORTAMIENTO_PAGO, CLASIFICACION_EMPRESA, CLASIF_VEHICULO, GAMA_VEHICULO, y otra donde se transforman los atributos nominales en números a CANT_PERSONAS_MISMA_OPE, CANT_PERSONAS_OTRA_OPE, LINEAS_PPA, SALARIO, USO_MB, TARIFA, CANT_PERSONAS_PPA, CALIFICACION_CLTE.

4.2.4 Ejecución de algoritmos de predicción

Una vez obtenido el dataset, se procede a la ejecución de algoritmos de predicción, para lo cual se utilizó Random Forest, Decisión Tree, XGBoost, CatBosst, LGBM, Regresión logística y Adaboost sobre el dataset sin aplicar la técnica de balanceo de datos.

En la Tabla 14 se puede observar los resultados obtenidos a nivel de F1 score y recall, donde los algoritmos XGBoost, CatBoost y LGBM obtuvieron la puntuación más alta.

Tabla 14

Algoritmos de predicción sin datos balanceados

MODELO	ROC AUC	F1 SCORE	RECALL	TRUE POSITIVE	FALSE POSITIVE	TRUE NEGATIVE	FALSE NEGATIVE
Decision Tree	0.83899	0.64419	0.68525	172	111	15142	79
Random Forest 1	0.79864	0.73891	0.59761	150	5	15248	101
Logistic Regression 1	0.72676	0.60800	0.45418	114	10	15243	137
XGBoost	0.85840	0.82568	0.71713	180	5	15248	71
Catboost 1	0.84631	0.80184	0.69322	174	9	15244	77
LGBM	0.85631	0.81735	0.71314	179	8	15245	72
Logistic Regression 2	0.86348	0.24634	0.80478	202	1187	14066	49
Random Forest 2	0.78071	0.71032	0.56175	141	5	15248	110
BRRF	0.90569	0.35497	0.86055	216	750	14503	35
AdaBoost	0.71265	0.57372	0.42629	107	15	15238	144
CatBoost 2	0.86232	0.82727	0.72510	182	7	15246	69

Adicional a esto, se realiza una nueva ejecución aplicando varias técnicas de balanceo de datos (Smoteen, Adasyn, BorderlineSmote y Smote), sobre los 3 algoritmos que tuvieron la puntuación más alta de F1 score (Catboost, XGBoost y LGBM).

En la Tabla 15 se puede observar los resultados obtenidos a nivel de F1 score de las 4 técnicas de balanceo aplicadas.

Tabla 15

Algoritmos de predicción sin datos balanceados

TIPO BALANCEADOR	ADASYN	Borderline SMOTE	SMOTEENN	SMOTE
CatBoost	0.807425	0.806452	0.818584	0.820046
LGBM	0.807256	0.816327	0.815618	0.807339
XGB	0.800926	0.809195	0.792952	0.792627

4.2.5 Selección de algoritmos de predicción

Después de la ejecución de algoritmos con y sin balanceo de datos, se realizó un comparativo para indicar cuál es la mejor técnica por utilizar. En la Tabla 16, se visualiza que la ejecución de los algoritmos sin aplicar técnicas de balanceo obtuvo el F1 score más alto, por ende, esta sería la mejor opción.

Tabla 16

Matriz de resultados de modelos (F1 Score)

MODELO	F1_SCORE	F1_SCORE - Técnicas de Balanceo			
	Sin Balanceo	ADASYN	BorderlineSMOTE	SMOTEENN	SMOTE
XGBoost	0.825688	0.800926	0.809195	0.792952	0.792627
LGBM	0.817352	0.807256	0.816327	0.815618	0.807339
CatBoost	0.827273	0.807425	0.806452	0.818584	0.820046

Después de seleccionar los tres algoritmos, se realizó una búsqueda de hiperparámetros con el objetivo de optimizar la precisión. En la Tabla 17 se presentan los rangos de valores explorados, así como los hiperparámetros óptimos seleccionados para los algoritmos de predicción XGBoost, LGBM y CatBoost.

Tabla 17

Búsqueda de hiperparámetros

Modelos	Parametros	Valores buscados	Valores seleccionados	Recall	F1 Score
XGB	colsample_bytree	[0.8, 0.9, 1.0]	1.0	0.756972	0.842572
	gamma	[0.1, 0.25, 0.5]	0.5		
	learning_rate	[0.1, 0.01, 0.001]	0.1		
	max_depth	[3, 5, 7, 9]	9		
	scale_pos_weight	[1, 5, 7.85, 10]	7.85		
	subsample	[0.8, 0.9, 1.0]	0.9		
LGBM	colsample_bytree	[0.8, 0.9, 1.0]	0.9	0.729084	0.833713
	min_child_samples	[10, 20, 30]	10		
	num_leaves	[31, 63, 127]	127		
	subsample	[0.8, 0.9, 1.0]	0.8		
CatBoost	border_count	[15, 25, 35]	35	0.709163	0.809091
	depth	[5, 7, 9]	5		
	iterations	[350, 400, 415]	415		
	l2_leaf_reg	[1, 3]	3		
	learning_rate	[0.15, 0.9]	0.15		

4.2.6 Ensamble de datos

La etapa final, consiste en ensamblar los 3 algoritmos seleccionados (XGBoost, CatBoost y LGBM). Esta acción es conocida como algoritmos híbridos (unión de 2 o más algoritmos de predicción).

En la Figura 17 se realiza una primera ejecución, ensamblando los 3 algoritmos mencionados sobre el dataset, sin realizar ningún ajuste de hiperparámetros.

```
Confusion Matrix:
[[15245    8]
 [   71  180]]

Classification Report:
              precision    recall  f1-score   support

     0       1.00      1.00      1.00     15253
     1       0.96      0.72      0.82        251

 accuracy          0.99          0.99          0.99     15504
 macro avg          0.98          0.86          0.91     15504
 weighted avg          0.99          0.99          0.99     15504
```

Figura 17. Ensamble de Algoritmos (Sin ajuste de Hiperparámetros)

4.3 Validación de modelo de aprendizaje automático

4.3.1 Validación cruzada

Culminada la etapa de diseño y ejecución del modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones, se procede con la realización de la validación cruzada la cual se puede observar en la Tabla 18:

Tabla 18
Validación cruzada

Modelos	Parámetros	Valores buscados	Valores seleccionados	Mejor F1_Score
CatBoost	border_count	randint (32, 255)	204	0.839729
CatBoost	depth	randint (1, 10)	5	
CatBoost	iterations	randint (100, 800)	425	
CatBoost	l2_leaf_reg	randint (1, 9)	2	
CatBoost	learning_rate	uniform (0.01, 1)	0.33367923640424	
LGBM	colsample_bytree	uniform (0.8, 0.2)	0.88508728772328	
LGBM	min_child_samples	randint (10, 30)	10	
LGBM	num_leaves	randint (31, 127)	50	
LGBM	subsample	uniform (0.8, 0.2)	0.84848194648301	
XGB	colsample_bytree	uniform (0.8, 0.2)	0.82296736494784	
XGB	gamma	uniform (0.1, 0.9)	0.64955803819746	
XGB	learning_rate	uniform (0.001, 0.9)	0.26076749791623	
XGB	max_depth	randint (3, 10)	7	
XGB	n_estimators	randint (50, 500)	431	
XGB	scale_pos_weight	randint (1, 10)	4	
XGB	subsample	uniform (0.8, 0.2)	0.86412843843190	

4.3.2 Validación con prueba t de student

Finalizada la validación del modelo mediante validación cruzada, se procedió a comparar el rendimiento del modelo propuesto con un modelo de referencia básico. Para este propósito, se seleccionó el algoritmo AdaBoost como modelo base, con el objetivo de determinar si la contribución propuesta presenta significancia estadística.

Los resultados de esta comparación se presentan en la Figura 18, donde se observa un p-value de 0.

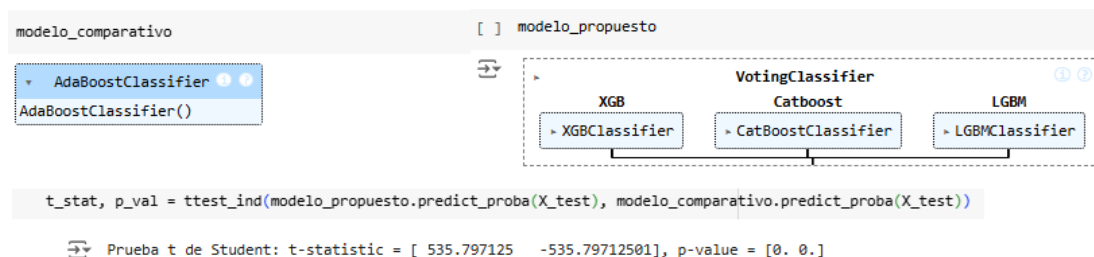


Figura 18. Resultado de prueba t de student

4.3.3 Validación con test de McNemar

Adicional a la validación con la prueba t de student, se realizó el test de McNemar, basándonos en la misma validación de tomar un algoritmo (AdaBoost) como modelo base, para determinar si la contribución propuesta presenta significancia estadística. Los resultados de esta comparación se presentan en la Figura 19, donde se observa un p-value muy cercano a 0.



Figura 19. Resultado de test de McNemar

CAPÍTULO V: RESULTADOS Y DISCUSIÓN

5.1 Resultados

5.1.1 Construcción del dataset (Telco y no Telco)

El dataset utilizado para la predicción contó con 87 variables, las cuales pasaron por un proceso de selección representado en la Tabla 19 y Tabla 20, donde se visualiza una matriz con el motivo de cada eliminación de variable.

Tabla 19

Selección de variables (dataset original - Parte 1)

Variable	Data anómala	Agrupación	Exceso Nulos	Estadística	XGBoost y/o RF	Greedy Stepwise	
CAT_COMPORTAMIENTO_PAGO						x	
NIVEL_SOCIOECONOMICO						x	
RENTA						x	
DIAS_EXCESO_MAXPAGO			x			x	
VALOR_PAGADO							
CANT_LINEAS_POSTPAGO				x		x	
CANT_LINEAS_PREPAGO		x				x	
ACTIVACIONES_PAQUETE							
TARIFA_CUPO		x				x	
CALIFICACION_CLIENTE		x				x	
DESCRIPCION_FORMA_PAGO						x	
TOTAL_MB_2G_M		x				x	
TOTAL_MB_3G_M		x				x	
TOTAL_MB_4G_M		x				x	
INGRESOS_EXCEDENTES_GRANEL	x					x	
INGRESOS_ADICIONAL_PAQ	x					x	
MINUTOS_SALIENTES							
MEGAS_NAVEGADOS	x					x	
CANT_VISITAS_CENTROS_ATENCION					x		
CANTON		x				x	
PROVINCIA		x				x	
ANTIG_ULTIMO_FINANCIAMIENTO			x			x	
PORC_COMPETENCIA							
CANTIDAD_PROGR_COMPETENCIA				x		x	
VISITAS_INTENCION_RENUNCIA							
DIAS_SUSPENSION				x		x	
SUSP_COBRANZA							
MODELO		x				x	
MARCA		x				x	
DEVICE_TYPE						x	

Cabe recalcar que en el análisis no participa la variable objetivo “churn”, y las variables que indican la identificación del cliente, que son “MES” y “id”.

Tabla 20

Selección de variables (dataset original - Parte 2)

Variable	Data anómala	Agrupación	Exceso Nulos	Estadística	XGBoost y/o RF	Greedy Stepwise	
DIAS_USO_VOZ						X	
DIAS_USO_MB							
DIAS_USO_VOZ_Y_MB		X					
ESTADO						X	
PRODUCTO						X	
EDAD_CLIENTE		X				X	
ANTIGUEDAD_CLIENTE		X				X	
CANTIDAD_SITIO		X				X	
CUPO_PRECALIFICADO							
CANT_LLAMADAS_SALIENTES							
CANT_LLAMADAS_ENTRANTES							
MINS_ENTRANTES							
PORC_MIN_COMPETENCIA				X		X	
CANTIDAD_MIN_INTERNOS_POS		X				X	
CANTIDAD_MIN_INTERNOS_PPA		X				X	
CANTIDAD_MIN_INTERNOS_OTRAS_OPE		X				X	
PORC_MIN_INTERNOS							
PORC_MIN_INTERNOS_POST							
PORC_MIN_INTERNOS_PPA							
VISITAS_POR_RENUNCIA							
MODULO_NEGOCIACION					X		
SOLICITUD_PORTABILIDAD_POR_SERVICIO							
SOLICITUD_PORTABILIDAD_POR_DNI					X		
LLAMADAS_CALL_CENTER_SERVICIO							
LLAMADAS_CALL_CENTER_DNI							
INBOUND_SERVICIO							
INBOUND_DNI							
SDA_SERVICIO					X		
SDA_IDENTIFICACION					X		
PORC_RECAUDACION							
CANT_LLAM_CALLCENTER							
MORA_CTA						X	
MORA_FIJO			X			X	
FACTURACION_FIJO							
FACTURACION							
NPS_CAC_CATEG			X			X	
NPS_CAC					X		
NPS_ASESOR_CATEG			X			X	
NPS_ASESOR							
CLASE_VEHICULO		X				X	
TIPO_VEHICULO		X				X	
MARCA_VEHICULO						X	
ANIO_VEHICULO							
AVALUO_VEHICULO		X				X	
ESTADO_CONTRIBUYENTE						X	
CLASE_CONTRIBUYENTE						X	
FECHA_INI_ACTIVIDAD						X	
FECHA_SUSPENSION						X	
NUMERO_ESTABLECIMIENTO		X				X	
CODIGO_CIIU						X	
TIPO_EMPRESA_LABORAL		X				X	
FECHA_ING_AFILIADO						X	
OCUPACION						X	
SUELDO		X				X	

Las variables que no fueron seleccionadas por ninguna de las técnicas de descarte conforman el dataset final que se utilizará en la ejecución de los algoritmos de predicción. Este dataset final incluye 26 variables originales y 14 nuevas (variables creadas), las cuales se puede observar en la Tabla 21.

Tabla 21

Matriz de variables seleccionadas

#	Descripción	Nombre de Variable
1	Variable objetivo	Churn
2	Variable dataset original	Visitas_Intencion_Renuncia
3	Variable dataset original	Cant_Llamadas_Entrantes
4	Variable dataset original	Porc_Min_Internos_Ppa
5	Variable dataset original	Llamadas_Call_Center_Dni
6	Variable dataset original	Facturacion_Fijo
7	Variable dataset original	Valor_Pagado
8	Variable dataset original	Susp_Cobranza
9	Variable dataset original	Mins_Entrantes
10	Variable dataset original	Visitas_Por_Renuncia
11	Variable dataset original	Inbound_Servicio
12	Variable dataset original	Facturacion
13	Variable dataset original	Activaciones_Paquete
14	Variable dataset original	Dias_Uso_Voz
15	Variable dataset original	Porc_Min_Internos
16	Variable dataset original	Solicitud_Portabilidad_Por_Servicio
17	Variable dataset original	Inbound_Dni
18	Variable dataset original	Nps_Asesor
19	Variable dataset original	Minutos_Salientes
20	Variable dataset original	Cupo_Precalificado
21	Variable dataset original	Porc_Min_Internos_Post
22	Variable dataset original	Llamadas_Call_Center_Servicio
23	Variable dataset original	Porc_Recaudacion
24	Variable dataset original	Anio_Vehiculo
25	Variable dataset original	Porc_Competencia
26	Variable dataset original	Cant_Llamadas_Salientes
27	Variable dataset original	Cant_Llam_Callcenter
28	Nuevas Variables	Gama_Asignada_Dispositivo
29	Nuevas Variables	Clasif_Vehiculo
30	Nuevas Variables	Cant_Personas_Ppa
31	Nuevas Variables	Clasificacion_Empresa
32	Nuevas Variables	Antiguedad_Clte
33	Nuevas Variables	Salario
34	Nuevas Variables	Calificacion_Clte
35	Nuevas Variables	Gama_Vehiculo
36	Nuevas Variables	Cant_Personas_Misma_Ope
37	Nuevas Variables	Uso_Mb
38	Nuevas Variables	Cant_Personas_Otra_Ope
39	Nuevas Variables	Lineas_Ppa
40	Nuevas Variables	Tarifa
41	Nuevas Variables	Establecimiento

5.1.2 Diseño del modelo de aprendizaje automático

Culminada la etapa de selección de variables, se obtuvo el dataset final, el cual pasa por procesos de ejecución, selección, ensamble y validación de algoritmos de predicción, con el fin de dar como resultado un Modelo de predicción de deserción de clientes postpago de empresas de Telecomunicaciones.

En la Figura 20, se diagrama los pasos seguidos en el proceso de la creación del modelo de predicción de deserción.

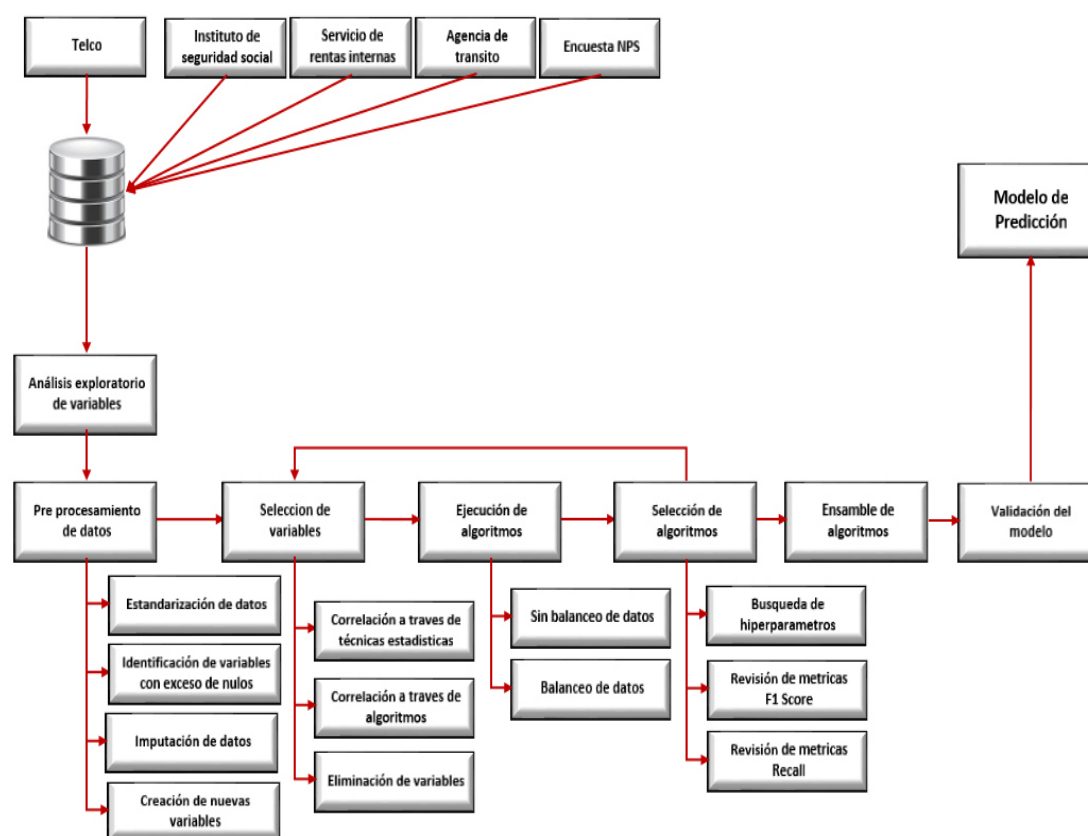


Figura 20. Diseño de Modelo de predicción de deserción de clientes postpago en empresas Telco

En la Figura 21, se puede observar el resultado de la ejecución del diseño mostrado en la Figura 20, donde se logró un F1 score de 84% y un recall de 74% sobre la clase objetiva, dando un promedio macro de 92%.

```

Confusion Matrix:
[[15246   7]
 [   65 186]]

Classification Report:
              precision    recall  f1-score   support

     0       1.00      1.00      1.00     15253
     1       0.96      0.74      0.84       251

 accuracy          0.98      0.87      0.92     15504
 macro avg          0.98      0.87      0.92     15504
 weighted avg       1.00      1.00      1.00     15504

```

Figura 21. Ensemble de Algoritmos (Con ajuste de Hiperparámetros)

5.1.3 Validación del modelo de aprendizaje automático

Después de realizado el ensemble de algoritmos con el respectivo ajuste de hiperparámetros, se procede con la ejecución de la validación cruzada, lo cual se puede observar en la Tabla 22, donde se obtuvo un F1 Score de 0.839.

Tabla 22

Validación cruzada

Técnica	Modelo	Parametro	Valor	F1 Score
Voting Classifier	CatBoost	border_count	204	0.839729
	CatBoost	depth	5	
	CatBoost	iterations	425	
	CatBoost	l2_leaf_reg	2	
	CatBoost	learning_rate	0.3336792364042437	
	LGBM	colsample_bytree	0.8850872877232834	
	LGBM	min_child_samples	10	
	LGBM	num_leaves	50	
	LGBM	subsample	0.8484819464830161	
	XGB	colsample_bytree	0.8229673649478407	
	XGB	gamma	0.6495580381974694	
	XGB	learning_rate	0.2607674979162302	
	XGB	max_depth	7	
	XGB	n_estimators	431	
	XGB	scale_pos_weight	4	
	XGB	subsample	0.864128438431902	

5.2 Discusión

El resultado de esta investigación permite confirmar que el modelo de aprendizaje automático propuesto contribuye a la predicción de la deserción de clientes postpago en empresas de telecomunicaciones, proporcionando una vista holística que contribuye a identificar con antelación que cliente desertará, habiendo pasado por un proceso de validación de factores y/o variables que conforman el dataset, mismo que pasa por las diferentes etapas del diseño de modelo propuesto, y que finalmente es validado a través de una técnica denominada “validación cruzada” (ver Figura 22).

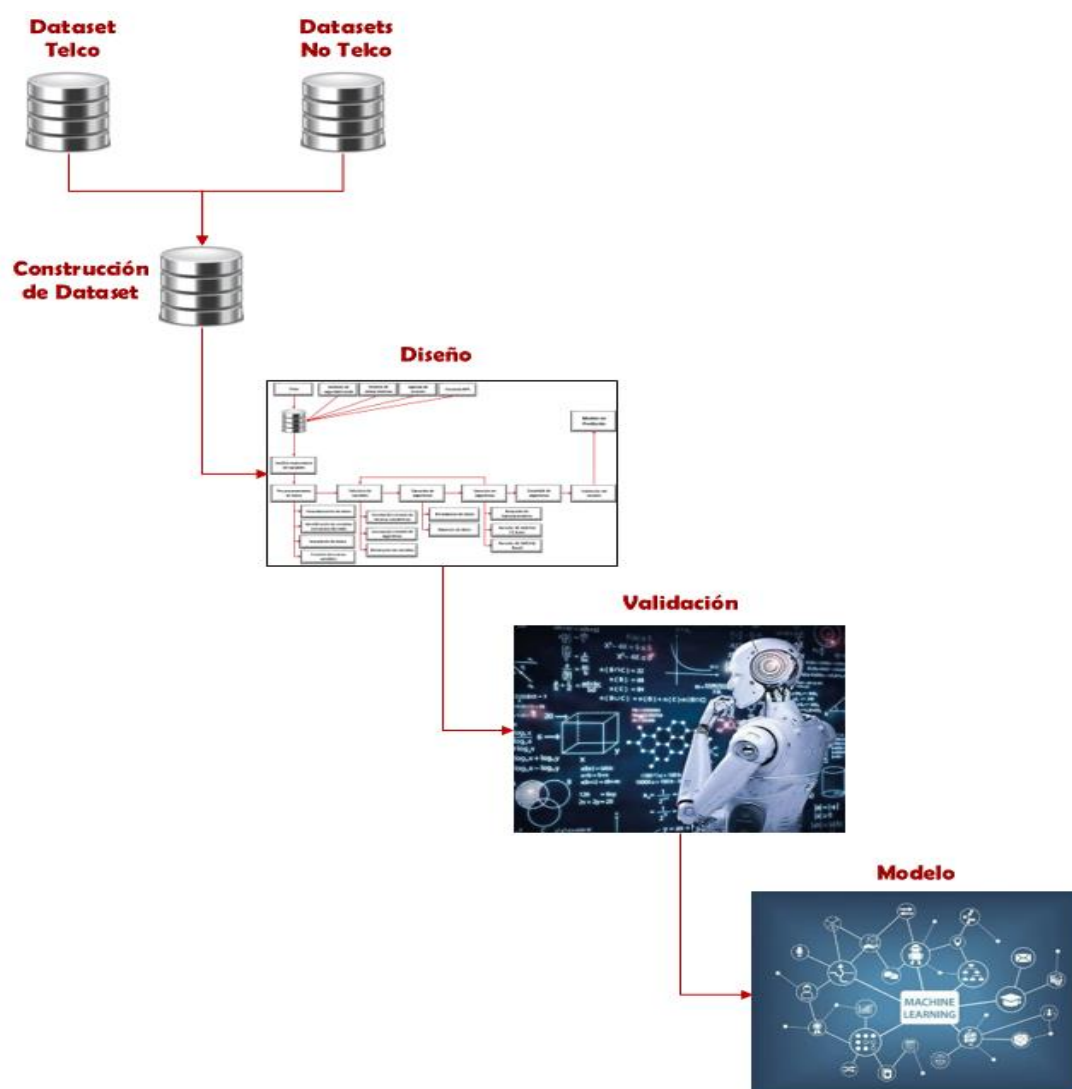


Figura 22. Secuencia de pasos del Modelo de predicción

Para respaldar lo anteriormente expuesto, este estudio empleó dos técnicas estadísticas: la prueba t de Student y el test de McNemar. Los resultados obtenidos mostraron un p-value igual a 0 en la prueba t de Student y un p-value muy cercano a 0 en el test de McNemar.

Es importante destacar que, a diferencia de otros estudios que se han centrado principalmente en la precisión (accuracy), la cual tiende a ser alta al promediar los aciertos tanto en la clase objetiva como en la no objetiva, este estudio se enfoca en el F1 score de la clase objetiva. Esta métrica proporciona una evaluación más equilibrada y precisa del modelo al considerar tanto la precisión como el recall, lo que es especialmente relevante en escenarios donde las clases están desequilibradas. Al priorizar el F1 score, nuestro enfoque asegura una mejor identificación de la deserción de clientes, superando las limitaciones de las evaluaciones basadas únicamente en la precisión general.

5.2.1 Sobre construcción del dataset (Telco y no Telco)

En este estudio se realizaron 3 técnicas estadísticas y 2 de algoritmos de clasificación para determinar que 26 variables del dataset influyen significativamente en el modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones. En estudios previos únicamente se basan en una sola técnica estadística o en el uso de un algoritmo de clasificación, sin embargo, el uso combinado de estas técnicas, fortalecieron el proceso de selección de variables, descartando lo que no aporta al modelo de predicción.

Además, se realizó un proceso de creación de nuevas variables, el cual también paso por el proceso de selección de variables antes mencionado, aportando con 14 variables, es decir con un 35%.

Adicional a esto, se destaca que fueron seleccionadas 8 de las 20 variables pertenecientes a datasets no Telco (40% de estas variables), mostrando que

la incorporación de este tipo de información, influye significativamente en el modelo de predicción de deserción de clientes postpago en empresas de telecomunicaciones.

El estudio se limita a los datasets propuestos, y cabe recalcar que no todas las variables de los datasets no Telco aportan al modelo de predicción.

Futuros estudios podrían explorar la incorporación de otros datasets distintos a los utilizados en la presente investigación, con el fin de explorar el comportamiento de los datos frente a fuentes de información distintas a las expuestas.

5.2.2 Sobre el diseño del modelo de aprendizaje automático

En este estudio se realizó un diagrama detallado de los pasos a seguir, para la creación de un modelo de aprendizaje automático, donde se detallan cada una de las etapas por las que pasa el dataset, y los resultados obtenidos lograron un F1 score de 84% y un recall del 74% sobre la clase objetiva, lo cual demuestra la efectividad para identificar correctamente los casos. Por otro lado, se obtuvo un F1 score de 92% y recall del 87% de manera promediada, lo cual demuestra que el diseño tiene un impacto significativo en la predicción de deserción de clientes postpago en empresas de telecomunicaciones.

Los resultados obtenidos resaltan la importancia de un diseño que guíe paso a paso la ejecución de un modelo, comenzando con el análisis exploratorio y avanzando por distintas etapas, como el preprocesamiento de datos, la selección de variables y algoritmos, el ensamble, y culminando en la validación. Esta metodología podría servir como un marco de referencia para futuras investigaciones en este campo.

5.2.3 Sobre la Validación del modelo de aprendizaje automático

En este estudio se realizó una validación cruzada, con el fin de medir el desempeño del modelo propuesto, obteniendo un F1 Score de 0.839, lo cual demuestra que los resultados obtenidos son generalizables, además de que minimiza el riesgo de sobreajuste (overfitting) y garantiza que el modelo funciona bien con los datos de entrenamiento y por ende, funcionará correctamente con los próximos datos.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Esta investigación presenta un modelo de predicción de deserción de clientes postpago de empresas de telecomunicaciones, para lo cual se obtuvieron las siguientes conclusiones que están relacionadas a las hipótesis.

1. Se logró el objetivo de determinar la contribución del modelo de aprendizaje automático en la predicción de la deserción de clientes postpago en las empresas de telecomunicaciones; evaluando el impacto significativo del modelo propuesto, con la aplicación de las técnicas estadísticas, prueba t de Student y test de McNemar. Para demostrar el cumplimiento del objetivo, se utilizó un algoritmo de referencia (Adaboost) como modelo base y se comparó sus resultados con los del modelo propuesto en este estudio, logrando un p-value igual a 0 en la prueba t de Student y un p-value cercano a 0 en el test de McNemar. Estos resultados indican, con un alto nivel de confianza estadística, que las diferencias observadas entre los dos modelos no son atribuibles al azar. Cabe recalcar que, la prueba t de Student demuestra que el desempeño promedio del modelo propuesto supera significativamente al del modelo base, mientras que el test de McNemar refuerza que el modelo propuesto logra mejoras significativas.

En tal sentido, las evidencias estadísticas respaldan la robustez del modelo propuesto, resaltando su efectividad no solo en términos de precisión global, sino también en escenarios específicos, logrando que este enfoque contribuya a validar la hipótesis y a resaltar el impacto

práctico de los avances metodológicos implementados en la investigación.

Además, el modelo de aprendizaje automático desarrollado no solo mejora la predicción de la deserción de clientes, sino que también proporciona información valiosa para la industria, ya que pueden aplicar estos hallazgos para diseñar estrategias de retención más efectivas, optimizando sus recursos y mejorando la satisfacción del cliente.

2. Se construyó el conjunto de datos, que incorporó las variables Telco y no Telco, las cuales influyeron en el modelo de predicción de deserción de clientes postpago de telefonía móvil en empresas de telecomunicaciones. Además, los resultados obtenidos en la evaluación de las variables del dataset mostraron que 26 de las 84 variables influyen significativamente en el modelo de predicción propuesto, incluyendo variables como edad, antigüedad, tráfico de minutos, tipo de vehículo, estado de contribuyente, entre otras; demostrando la influencia significativa de las variables y respaldando la hipótesis, con un impacto notable en la precisión del modelo predictivo.
3. Se realizó un diseño detallado de los pasos necesarios para desarrollar el modelo de aprendizaje automático enfocado en predecir la deserción de clientes postpago de telefonía móvil en empresas de telecomunicaciones. Este diseño incluyó múltiples etapas, que fueron desde el análisis exploratorio de datos hasta la obtención del modelo predictivo final. Estas etapas incluyeron procesos como la selección de variables, la ejecución de algoritmos y la evaluación de distintos escenarios, tales como el balanceo de datos y la búsqueda de hiperparámetros, permitiendo identificar los mejores algoritmos para ser ensamblados y por ende, la obtención de una mejor precisión y robustez, alcanzando un F1 score del 92%, lo que indica una excelente capacidad predictiva y un equilibrio adecuado entre precisión y

exhaustividad. Este resultado evidencia una alta capacidad predictiva, por lo tanto, se valida la hipótesis planteada, al demostrar que los componentes en el diseño del modelo de aprendizaje automático tienen un impacto significativo en la predicción de deserción de clientes postpago en empresas de telecomunicaciones.

4. Se validó el modelo de aprendizaje automático utilizando métricas de clasificación diseñadas para datasets con clases desequilibradas, valiéndose de la técnica denominada validación cruzada. Este enfoque permitió evaluar la robustez y capacidad de generalización del modelo al dividir el dataset en múltiples subconjuntos para entrenamiento y prueba, garantizando que el modelo no solo se ajusta adecuadamente a los datos de entrenamiento, sino que también mantiene un rendimiento consistente en datos no vistos previamente (datos nuevos). Los resultados obtenidos confirmaron la estabilidad, precisión y aplicabilidad del modelo en escenarios reales, demostrando su capacidad para predecir la deserción de clientes con un alto grado de confianza, respaldando así la hipótesis planteada.

6.2 Recomendaciones

Luego de confirmar la generalización de los resultados obtenidos, la implementación del modelo propuesto puede darse en diferentes contextos, dominios o conjuntos de datos. Para ello se recomienda la aplicación de otras técnicas estadísticas como análisis discriminante, con el objetivo de evaluar como el modelo de aprendizaje automático puede contribuir con una mejor precisión en la predicción de la deserción de clientes en otros tipos de empresas diferentes a las telecomunicaciones, que busquen identificar cuando se presenta una deserción de clientes.

Además, para mejorar la predicción es recomendable incrementar el conjunto de datos incorporando variables de otros datasets sea de empresas

competidoras o de otra índole, con el fin de que se tenga una visión global de los clientes. Adicional se podrían incorporar datasets que contribuyan con variables macroeconómicas y microeconómicas, que complementen la información del cliente con la realidad de cada contexto en el que se desarrolla la investigación.

Es importante profundizar en cada una de las etapas del diseño propuesto, con el fin de incorporar etapas específicas como manejo de datos anómalos o, en su defecto, excluir aquellas que se consideren innecesarias.

También es importante explorar nuevas técnicas de ensamblaje y evaluación de combinaciones. Aunque se evaluaron 7 algoritmos, es necesario ampliar la gama propuesta, ya que nuevas combinaciones podrían ofrecer mejoras adicionales en precisión y exhaustividad.

Para la validación del modelo, se utilizaron métricas específicas para datasets con clases desequilibradas, sin embargo, es recomendable incluir métricas adicionales como el área bajo la curva ROC y el área bajo la curva PR (Precisión – Recall), lo cual ofrecería una visión más integral del rendimiento del modelo, especialmente en escenarios donde la clase minoritaria tiene mayor relevancia.

REFERENCIAS BIBLIOGRÁFICAS

- Abarca Sánchez, Y., Barreto Rivera, U., Barreto Jara, O., & Díaz Ugarte, J. L. (2022). Fidelización y retención de clientes en una empresa líder de telecomunicaciones en Perú. *Revista Venezolana de Gerencia*, 27(28), 729-743. <https://doi.org/10.52080/rvgluz.27.98.22>
- Abdi, F., & Abolmakarem, S. (2019). Customer Behavior Mining Framework (CBMF) using clustering and classification techniques. *Journal of Industrial Engineering International*, 15(S1), 1-18. <https://doi.org/10.1007/s40092-018-0285-3>
- Abdulsalam, S. O., Arowolo, M. O., Saheed, Y. K., & Afolayan, J. O. (2022). Customer Churn Prediction in Telecommunication Industry Using Classification and Regression Trees and Artificial Neural Network Algorithms. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, 10(2), 431-440. <https://doi.org/10.52549/ijeei.v10i2.2985>
- Abualkishik, A. Z., Almajed, R., & Thompson, W. (2023). Intelligent model for customer churn prediction using deep learning optimization algorithms. *Journal of Intelligent Systems and Internet of Things*, 8(1), 43-54. Scopus. <https://doi.org/10.54216/JISIoT.080104>
- Acero-Charaña, C., Osco-Mamani, E., & Ale-Nieto, T. (2021). Model for Predicting Customer Desertion of Telephony Service using Machine Learning. *International Journal of Advanced Computer Science and Applications*, 12(3). <https://doi.org/10.14569/IJACSA.2021.0120320>
- Adhikary, D. D., & Gupta, D. (2021). Applying over 100 classifiers for churn prediction in telecom companies. *Multimedia Tools and Applications*, 80(28-29), 35123-35144. <https://doi.org/10.1007/s11042-020-09658-z>
- Agencia Nacional de tránsito. (2022). <https://datosabiertos.gob.ec/sk/dataset/?organization=antec>
- Ahmad, A. K., Jafar, A., & Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, 6(1), 28. <https://doi.org/10.1186/s40537-019-0191-6>
- Ahmed, M., Afzal, H., Siddiqi, I., Amjad, M. F., & Khurshid, K. (2020). Exploring nested ensemble learners using overproduction and choose approach for

- churn prediction in telecom industry. *Neural Computing and Applications*, 32(8), 3237-3251. <https://doi.org/10.1007/s00521-018-3678-8>
- Ahmed, & Maheswari, D. (2019). An enhanced ensemble classifier for telecom churn prediction using cost based uplift modelling. *International Journal of Information Technology*, 11(2), 381-391. <https://doi.org/10.1007/s41870-018-0248-3>
- Ahn, J., Hwang, J., Kim, D., Choi, H., & Kang, S. (2020). A Survey on Churn Analysis in Various Business Domains. *IEEE Access*, 8, 220816-220839. <https://doi.org/10.1109/ACCESS.2020.3042657>
- Ajitha, P., Sivasangari, A., Gomathi, R. M., & Indira, K. (2020). Prediction of customer plan using churn analysis for telecom industry. *Recent Advances in Computer Science and Communications*, 13(5), 926-929. Scopus. <https://doi.org/10.2174/2213275912666190410114104>
- Akbar, M. D. A., & Chowanda, A. (2022). SGD OPTIMIZER TO REDUCE COST VALUE IN DEEP LEARNING FOR CUSTOMER CHURN PREDICTION. *Journal of Theoretical and Applied Information Technology*, 100(11), 3579-3587.
- Alboukaey, N., Joukhadar, A., & Ghneim, N. (2020). Dynamic behavior based churn prediction in mobile telecom. *Expert Systems with Applications*, 162, 113779. <https://doi.org/10.1016/j.eswa.2020.113779>
- Allam, S., & Srinivas, S. (2019). Churn Prediction using Attention Based Autoencoder Network. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(3), 725-730. <https://doi.org/10.30534/ijatcse/2019/60832019>
- Al-Mashraie, M., Chung, S. H., & Jeon, H. W. (2020). Customer switching behavior analysis in the telecommunication industry via push-pull-mooring framework: A machine learning approach. *Computers & Industrial Engineering*, 144, 106476. <https://doi.org/10.1016/j.cie.2020.106476>
- Almufadi, N., & Qamar, A. M. (2022). Deep Convolutional Neural Network Based Churn Prediction for Telecommunication Industry. *Computer Systems Science and Engineering*, 43(3), 1255-1270. Scopus. <https://doi.org/10.32604/csse.2022.025029>

- Almufadi, N., Qamar, A. M., Khan, R. U., & Othman, M. T. B. (2019). Deep learning-based churn prediction of telecom subscribers. *International Journal of Engineering Research and Technology*, 12(12), 2743-2748.
- AL-Shatnwai, A. M., & Faris, M. (2020). Predicting Customer Retention using XGBoost and Balancing Methods. *International Journal of Advanced Computer Science and Applications*, 11(7).
<https://doi.org/10.14569/IJACSA.2020.0110785>
- Al-Shourbaji, I., Helian, N., Sun, Y., Alshathri, S., & Abd Elaziz, M. (2022). Boosting Ant Colony Optimization with Reptile Search Algorithm for Churn Prediction. *Mathematics*, 10(7), 1031. <https://doi.org/10.3390/math10071031>
- Alzubaidi, A. M. N., & Al-Shamery, E. S. (2019). Churners Prediction Based on Mining the Content of Social Network Taxonomy. *International Journal of Recent Technology and Engineering*, 8(2S10), 341-351.
<https://doi.org/10.35940/ijrte.B1056.0982S1019>
- Alzubaidi, A. M. N., & Al-Shamery, E. S. (2020). Projection pursuit Random Forest using discriminant feature analysis model for churners prediction in telecom industry. *International Journal of Electrical and Computer Engineering*, 10(2), 1406-1421. Scopus. <https://doi.org/10.11591/ijece.v10i2.pp1406-1421>
- Amali, Arunkumar, R., & Mohan, S. (2019). Comparing and evaluating machine learning algorithms for predicting customer churn in telecommunication industry. *Journal of Advanced Research in Dynamical and Control Systems*, 11(6 Special Issue), 170-178.
- Amali, I., & Arunkumar, R. (2020). Particle swarm optimization with kernel support vector machine for churn prediction in telecommunication industry. *International Journal of Scientific and Technology Research*, 9(4), 253-257.
- América Móvil. (2022). *América Móvil—Relación con Inversionistas—Informes financieros—Reportes trimestrales*.
<https://www.americamovil.com/Spanish/relacion-con-inversionistas/informes-financieros/reportes-trimestrales/default.aspx>
- Amin, A., Adnan, A., & Anwar, S. (2023). An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and Naïve Bayes. *Applied Soft Computing*, 137. Scopus. <https://doi.org/10.1016/j.asoc.2023.110103>

- Amin, A., Al-Obeidat, F., Shah, B., Tae, M. A., Khan, C., Durrani, H. U. R., & Anwar, S. (2020). Just-in-time customer churn prediction in the telecommunication sector. *The Journal of Supercomputing*, 76(6), 3924-3948. <https://doi.org/10.1007/s11227-017-2149-9>
- Amin, A., Shah, B., Khattak, A. M., Lopes Moreira, F. J., Ali, G., Rocha, A., & Anwar, S. (2019). Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods. *International Journal of Information Management*, 46, 304-319. <https://doi.org/10.1016/j.ijinfomgt.2018.08.015>
- Arcotel. (2024, septiembre). *Informe de Cuentas internet fijos y moviles_sep-2024*. https://www.arcotel.gob.ec/wp-content/uploads/2024/11/3-1-1-Cuentas-internet-fijos-y-moviles_sep-2024_f.xlsx
- Arora, L., Kapur, A., & Lavanya, K. (2019). Prediction of telecom churns and consumer behaviour using recurrent neural networks. *International Journal of Engineering and Advanced Technology*, 8(4), 353-358.
- Arshad, S., Iqbal, K., Naz, S., Yasmin, S., & Rehman, Z. (2022). A Hybrid System for Customer Churn Prediction and Retention Analysis via Supervised Learning. *Computers, Materials & Continua*, 72(3), 4283-4301. <https://doi.org/10.32604/cmc.2022.025442>
- Awang, M. K., Ismail, M. R., Makhtar, M., Nordin A. Rahman, M., & Mamat, A. R. (2018). Performance comparison of neural network training algorithms for modeling customer churn prediction. *International Journal of Engineering and Technology(UAE)*, 7(2), 35-37. Scopus. <https://doi.org/10.14419/ijet.v7i2.15.11196>
- Awang, M. K., Makhtar, M., Udin, N., & Mansor, N. F. (2021). Improving Customer Churn Classification with Ensemble Stacking Method. *International Journal of Advanced Computer Science and Applications*, 12(11). <https://doi.org/10.14569/IJACSA.2021.0121132>
- Azeem, M., & Usman, M. (2018). A fuzzy based churn prediction and retention model for prepaid customers in telecom industry: *International Journal of Computational Intelligence Systems*, 11(1), 66. <https://doi.org/10.2991/ijcis.11.1.6>

- Babatunde, R., Abdulsalam, S. O., Abdulsalam, O. A., & Arowolo, M. O. (2023). Classification of customer churn prediction model for telecommunication industry using analysis of variance. *IAES International Journal of Artificial Intelligence*, 12(3), 1323-1329. Scopus.
<https://doi.org/10.11591/ijai.v12.i3.pp1323-1329>
- Banco Mundial. (2022). *Suscripciones a telefonía celular móvil (por cada 100 personas) | Data*.
<https://datos.bancomundial.org/indicador/IT.CEL.SETS.P2?end=2020&start=2000>
- Beeharry, Y., & Fokone, R. (2022). Hybrid approach using machine learning algorithms for customers' churn prediction in the telecommunications industry. *Concurrency and Computation: Practice and Experience*, 34(4).
<https://doi.org/10.1002/cpe.6627>
- Bell, D., & Mgbemena, C. (2018). Data-driven agent-based exploration of customer behavior. *SIMULATION*, 94(3), 195-212.
<https://doi.org/10.1177/0037549717743106>
- Beschi, J., & Chenthur, S. (2020). An optimal ensemble classification for predicting Churn in telecommunication. *Journal of Engineering Science and Technology Review*, 13(2), 44-49. Scopus. <https://doi.org/10.25103/jestr.132.07>
- Bhale, & Bedi, H. S. (2020). A study on the impact of engagement with service channels and factors affecting mobile number portability. *International Journal of Scientific and Technology Research*, 9(3), 271-277. Scopus.
- Bhale, U., & Bedi. (2021). *A Study On The Impact Of Engagement With Service Channels And Factors Affecting Mobile Number Portability*.
<https://doi.org/10.13140/RG.2.2.26203.08483>
- Bhattacharyya, J., & Dash, M. K. (2020). Investigation of customer churn insights and intelligence from social media: A netnographic research. *Online Information Review*, 45(1), 174-206. <https://doi.org/10.1108/OIR-02-2020-0048>
- Bilal, S. F., Almazroi, A. A., Bashir, S., Khan, F. H., & Almazroi, A. A. (2022). An ensemble based approach using a combination of clustering and classification algorithms to enhance customer churn prediction in telecom industry. *PeerJ Computer Science*, 8. Scopus. <https://doi.org/10.7717/PEERJ-CS.854>

- Bose, I., & Chen, X. (2015). Detecting the migration of mobile service customers using fuzzy clustering. *Information & Management*, 52(2), 227-238.
<https://doi.org/10.1016/j.im.2014.11.001>
- Brmez, S., & Znidarsic, M. (2019). A Case of Churn Prediction in Telecommunications Industry. *IPSI BGD TRANSACTIONS ON INTERNET RESEARCH*, 15(2).
- Bugajev, A., Kriaucienė, R., Vasilecas, O., & Chadyšas, V. (2022). The Impact of Churn Labelling Rules on Churn Prediction in Telecommunications. *Informatica*, 247-277. <https://doi.org/10.15388/22-INFOR484>
- Chouiekh, A., & El Haj, E. H. I. (2020). Deep Convolutional Neural Networks for Customer Churn Prediction Analysis: *International Journal of Cognitive Informatics and Natural Intelligence*, 14(1), 1-16.
<https://doi.org/10.4018/IJCINI.2020010101>
- Colot, C., Baecke, P., & Linden, I. (2021a). Alternatives for Telco Data Network: The Value of Spatial and Referral Networks for Churn Detection. *Information Systems Management*, 38(3), 218-236.
<https://doi.org/10.1080/10580530.2021.1887981>
- Colot, C., Baecke, P., & Linden, I. (2021b). Leveraging fine-grained mobile data for churn detection through Essence Random Forest. *JOURNAL OF BIG DATA*, 8(1). <https://doi.org/10.1186/s40537-021-00451-9>
- Dalli, A. (2022). Impact of Hyperparameters on Deep Learning Model for Customer Churn Prediction in Telecommunication Sector. *Mathematical Problems in Engineering*, 2022, 1-11. <https://doi.org/10.1155/2022/4720539>
- Dataset—Instituto Ecuatoriano de Seguridad Social*. (2022, diciembre 1).
<https://datosabiertos.gob.ec/sk/dataset/?q=iess&organization=instituto-ecuatoriano-de-seguridad-social>
- Datasets—Servicio de Rentas Internas*. (2022, diciembre 1).
<https://www.sri.gob.ec/datasets#La%20informaci%C3%B3n%20del%20Catastro%20Tributario,%20proporcionar%20informaci%C3%B3n%20a%20grupada%20del%20n%C3%BAmero%20de%20contribuyentes%20registrados%20por%20provincia>,
- Deng, W., Deng, L., Liu, J., & Qi, J. (2019). Sampling method based on improved C4.5 decision tree and its application in prediction of telecom customer

- churn. *International Journal of Information Technology and Management*, 18(1), 93. <https://doi.org/10.1504/IJITM.2019.097887>
- Dolnicar, S., Grün, B., & Leisch, F. (2018). *Market Segmentation Analysis: Understanding It, Doing It, and Making It Useful*. Springer Singapore. <https://doi.org/10.1007/978-981-10-8818-6>
- Edwine, N., Wang, W., Song, W., & Ssebuggwawo, D. (2022). Detecting the Risk of Customer Churn in Telecom Sector: A Comparative Study. *Mathematical Problems in Engineering*, 2022, 1-16. <https://doi.org/10.1155/2022/8534739>
- Eria, K., & Marikannan, B. P. (2019). Significance-Based Feature Extraction for Customer Churn Prediction Data in the Telecom Sector. *Journal of Computational and Theoretical Nanoscience*, 16(8), 3428-3431. <https://doi.org/10.1166/jctn.2019.8303>
- Esteves, G., & Mendes-Moreira, J. (2016). Churn prediction in the telecom business. *2016 Eleventh International Conference on Digital Information Management (ICDIM)*, 254-259. <https://doi.org/10.1109/ICDIM.2016.7829775>
- Ewieda, M., M, E., & Roushdy, M. (2021). Customer Retention: Detecting Churners in Telecoms Industry using Data Mining Techniques. *International Journal of Advanced Computer Science and Applications*, 12(3). <https://doi.org/10.14569/IJACSA.2021.0120326>
- Faris, H. (2018). A Hybrid Swarm Intelligent Neural Network Model for Customer Churn Prediction and Identifying the Influencing Factors. *Information*, 9(11), 288. <https://doi.org/10.3390/info9110288>
- Faritha Banu, J., Neelakandan, S., Geetha, B. T., Selvalakshmi, V., Umadevi, A., & Martinson, E. O. (2022). Artificial Intelligence Based Customer Churn Prediction Model for Business Markets. *Computational Intelligence and Neuroscience*, 2022. Scopus. <https://doi.org/10.1155/2022/1703696>
- Freire, D., Santos Mauricio Sanchez, D., Castillo Sequera, J. L., & Fiallo Moncayo, D. (2024). Factors, Predictability, and Explainability of Mobile Telephony Customer Departure in Telecommunications Companies: A Systematic Review of the Literature. *IEEE Access*, 12, 118968-118980. <https://doi.org/10.1109/ACCESS.2024.3443318>

- Fujo, S. W., Subramanian, S., & Khder, M. A. (2022). Customer Churn Prediction in Telecommunication Industry Using Deep Learning. *Information Sciences Letters*, 11(1), 185-198. <https://doi.org/10.18576/isl/110120>
- Garimella, B., Prasad, G. V. S. N. R. V., & Prasad, M. H. M. K. (2021). Adaptive Optimization-Enabled Neural Networks to Handle the Imbalance Churn Data in Churn Prediction. *International Journal of Computational Intelligence and Applications*, 20(04), 2150025. <https://doi.org/10.1142/S1469026821500255>
- Garimella, B., Prasad, G. V. S. N. R. V., & Prasad, M. H. M. K. (2023). Churn prediction using optimized deep learning classifier on huge telecom data. *Journal of Ambient Intelligence and Humanized Computing*, 14(3), 2007-2028. Scopus. <https://doi.org/10.1007/s12652-021-03413-4>
- Global Overview Report. (2022, enero 26). *Global Overview Report*. DataReportal – Global Digital Insights. <https://datareportal.com/reports/digital-2022-global-overview-report>
- Golubev, A., Anh Tuan, N., Shcherbakov, M., & Van Phu, T. (2017). Clustering helps to determine the changes in telecom subscribers' behavior. *Proceedings of the IV International Research Conference «Information Technologies in Science, Management, Social Sphere and Medicine» (ITSMSSM 2017)*. IV International research conference «Information technologies in Science, Management, Social sphere and Medicine» (ITSMSSM 2017), Tomsk, Russia. <https://doi.org/10.2991/itsmssm-17.2017.70>
- Han, J., Kamber, M., & Pei, J. (2012). Getting to Know Your Data. En *Data Mining* (pp. 39-82). Elsevier. <https://doi.org/10.1016/B978-0-12-381479-1.00002-2>
- Haridasan, V., Muthukumaran, K., & Hariharanath, K. (2023). Arithmetic Optimization with Deep Learning Enabled Churn Prediction Model for Telecommunication Industries. *Intelligent Automation and Soft Computing*, 35(3), 3531-3544. Scopus. <https://doi.org/10.32604/iasc.2023.030628>
- Hemalatha, M., & Mahalakshmi, S. (2020). Customer churns prediction in telecom using adaptive logitboost learning approach. En *International Journal of Scientific and Technology Research* (Vol. 9, Número 2, pp. 5703-5713).
- Höppner, S., Stripling, E., Baesens, B., Broucke, S. vanden, & Verdonck, T. (2020). Profit driven decision trees for churn prediction. *European Journal of*

- Operational Research*, 284(3), 920-933.
<https://doi.org/10.1016/j.ejor.2018.11.072>
- Idris, A., Iftikhar, A., & Rehman, Z. ur. (2019). Intelligent churn prediction for telecom using GP-AdaBoost learning and PSO undersampling. *Cluster Computing*, 22(S3), 7241-7255. <https://doi.org/10.1007/s10586-017-1154-3>
- Jafari-Marandi, R., Denton, J., Idris, A., Smith, B. K., & Keramati, A. (2020). Optimum profit-driven churn decision making: Innovative artificial neural networks in telecom industry. *Neural Computing and Applications*, 32(18), 14929-14962. <https://doi.org/10.1007/s00521-020-04850-6>
- Jain, H., Khunteta, A., & Srivastava, S. (2022). Telecom Churn Prediction Using an Ensemble Approach with Feature Engineering and Importance. *International Journal of Intelligent Systems and Applications in Engineering*, 10(3), 22-33. Scopus.
- Jaisakthi, S. M., Gayathri, N., Uma, K., & Vijayarajan, V. (2018). Customer Churn Prediction Using Stochastic Gradient Boosting Technique. *Journal of Computational and Theoretical Nanoscience*, 15(6), 2410-2414.
<https://doi.org/10.1166/jctn.2018.7479>
- Jamalian, E., & Foukerdi, R. (2018). A Hybrid Data Mining Method for Customer Churn Prediction. *ENGINEERING TECHNOLOGY & APPLIED SCIENCE RESEARCH*, 8(3), 2991-2997.
- Jiang, P., Liu, Z., Zhang, L., & Wang, J. (2023). Hybrid model for profit-driven churn prediction based on cost minimization and return maximization. *Expert Systems with Applications*, 228. Scopus.
<https://doi.org/10.1016/j.eswa.2023.120354>
- Jin, X., & Han, J. (2017). Partitional Clustering. En C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning and Data Mining* (pp. 973-974). Springer US. https://doi.org/10.1007/978-1-4899-7687-1_637
- Jyothi, Ch. R., Boyina, N. B., Lakshmi, S. V., Akhila, B., & Sai, A. R. (2020). A churn detection model in telecommunication using machine learning techniques. *Journal of Advanced Research in Dynamical and Control Systems*, 12. <https://doi.org/10.5373/JARDCS/V12I2/S20201118>
- Karuppaiah, S., & Gopalan, N. (2021). Enhanced Churn Prediction Using Stacked Heuristic Incorporated Ensemble Model. *JOURNAL OF INFORMATION*

TECHNOLOGY RESEARCH, 14(2), 174-186.

<https://doi.org/10.4018/JITR.2021040109>

Kassem, E. A. el, Ali, S., Mostafa, A., & Kamal, F. (2020). Customer Churn Prediction Model and Identifying Features to Increase Customer Retention based on User Generated Content. *International Journal of Advanced Computer Science and Applications*, 11(5).

<https://doi.org/10.14569/IJACSA.2020.0110567>

Khalili-Damghani, K., Abdi, F., & Abolmakarem, S. (2018). Hybrid soft computing approach based on clustering, rule mining, and decision tree analysis for customer segmentation problem: Real case of customer-centric industries. *Applied Soft Computing*, 73, 816-828.

<https://doi.org/10.1016/j.asoc.2018.09.001>

Khan, Y., Shafiq, S., Naeem, A., Ahmed, S., Safwan, N., & Hussain, S. (2019). Customers Churn Prediction using Artificial Neural Networks (ANN) in Telecom Industry. *International Journal of Advanced Computer Science and Applications*, 10(9). <https://doi.org/10.14569/IJACSA.2019.0100918>

Khoh, W. H., Pang, Y. H., Ooi, S. Y., Wang, L.-Y.-K., & Poh, Q. W. (2023). Predictive Churn Modeling for Sustainable Business in the Telecommunication Industry: Optimized Weighted Ensemble Machine Learning. *Sustainability (Switzerland)*, 15(11). Scopus.

<https://doi.org/10.3390/su15118631>

Kim, S., Chang, Y., Wong, S. F., & Park, M. C. (2020). Customer resistance to churn in a mature mobile telecommunications market. *International Journal of Mobile Communications*, 18(1), 41.

<https://doi.org/10.1504/IJMC.2020.104421>

Lalwani, P., Mishra, M. K., Chadha, J. S., & Sethi, P. (2022). Customer churn prediction system: A machine learning approach. *Computing*, 104(2), 271-294. <https://doi.org/10.1007/s00607-021-00908-y>

Lazarov, V., & Capota, M. (2007). Churn Prediction. *Technical University of Munich, Business Analytics Course*, 33, 34.

Lee, P.-H., & Chen, Y. (2023). A Prediction of the Broadband Internet Customer Churn Using Support Vector Machine. *Journal of Quality*, 30(1), 1-12. Scopus. [https://doi.org/10.6220/joq.202302_30\(1\).0001](https://doi.org/10.6220/joq.202302_30(1).0001)

- Lera, E. (2000). Changing relations between manufacturing and service provision in a more competitive telecom environment. *Telecommunications Policy*, 24(5), 413-437. [https://doi.org/10.1016/S0308-5961\(00\)00029-X](https://doi.org/10.1016/S0308-5961(00)00029-X)
- Li, M., Yan, C., Liu, W., & Liu, X. (2021). An early warning model for customer churn prediction in telecommunication sector based on improved bat algorithm to optimize ELM. *International Journal of Intelligent Systems*, 36(7), 3401-3428. <https://doi.org/10.1002/int.22421>
- Li, & Marikannan, B. P. (2019). Hybrid particle swarm optimization-extreme learning machine algorithm for customer churn prediction. *Journal of Computational and Theoretical Nanoscience*, 16(8), 3432-3436. Scopus. <https://doi.org/10.1166/jctn.2019.8304>
- Liu, Ali, S., Bilal, S. F., Sakhawat, Z., Imran, A., Almuhaimeed, A., Alzahrani, A., & Sun, G. (2022). An Intelligent Hybrid Scheme for Customer Churn Prediction Integrating Clustering and Classification Algorithms. *Applied Sciences*, 12(18), 9355. <https://doi.org/10.3390/app12189355>
- Liu, Y., Fan, J., Zhang, J., Yin, X., & Song, Z. (2022). Research on telecom customer churn prediction based on ensemble learning. *Journal of Intelligent Information Systems*. <https://doi.org/10.1007/s10844-022-00739-z>
- Loukili, M., Messaoudi, F., & Ghazi, M. E. (2022). Supervised Learning Algorithms for Predicting Customer Churn with Hyperparameter Optimization. *International Journal of Advances in Soft Computing and its Applications*, 14(3), 49-63. Scopus. <https://doi.org/10.15849/IJASCA.221128.04>
- Mahmood, I. N., & Abdullah, H. S. (2022). Telecom Churn Prediction based on Deep Learning Approach. *Iraqi Journal of Science*, 2667-2675. <https://doi.org/10.24996/ijis.2022.63.6.32>
- Makinde, A. S., Agbeyangi, A. O., & Nwankwo, W. (2021). Predicting Mobile Portability Across Telecommunication Networks Using the Integrated-KLR: *International Journal of Intelligent Information Technologies*, 17(3), 50-62. <https://doi.org/10.4018/IJIT.2021070104>
- Malleswari, M., Maniraj, R., Kumar, P., & Murugan. (2018). Comparative analysis of machine learning techniques to Identify churn for telecom data. *International Journal of Engineering and Technology(UAE)*, 7(3.34 Special Issue 34), 291-295. <https://doi.org/10.14419/ijet.v7i3.2.14422>

- Mandák, J., & Hančlová, J. (2019). Use of logistic regression for understanding and prediction of customer churn in telecommunications. *Statistika*, 99(2), 129-141.
- Mandić, M., & Kraljević, G. (2022). Churn Prediction Model Improvement Using Automated Machine Learning with Social Network Parameters. *Revue d'Intelligence Artificielle*, 36(3), 373-379.
<https://doi.org/10.18280/ria.360304>
- Manjupriya, R., & Poornima, A. (2018). Customer Churn Prediction in the Mobile Telecommunication Industry Using Decision Tree Classification Algorithm. *Journal of Computational and Theoretical Nanoscience*, 15(9), 2789-2793.
<https://doi.org/10.1166/jctn.2018.7540>
- Mirza, O. M., Moses, G. J., Rajender, R., Lydia, E. L., Kadry, S., Me-Ead, C., & Thinnukool, O. (2022). Optimal Deep Canonically Correlated Autoencoder-Enabled Prediction Model for Customer Churn Prediction. *Computers, Materials and Continua*, 73(2), 3757-3769. Scopus.
<https://doi.org/10.32604/cmc.2022.030428>
- Mishachandar, B., & Kumar, K. A. (2018). Predicting customer churn using targeted proactive retention. *International Journal of Engineering and Technology(UAE)*, 7(2.27 Special Issue 27), 69-76.
- Mitrović, S., & De Weerd, J. (2020). Churn modeling with probabilistic meta paths-based representation learning. *Information Processing & Management*, 57(2), 102052. <https://doi.org/10.1016/j.ipm.2019.06.001>
- Mohan, R., Chaudhury, S., & Lall, B. (2021). Temporal Causal Modelling on Large Volume Enterprise Data. *IEEE Transactions on Big Data*, 1-1.
<https://doi.org/10.1109/TBDDATA.2021.3053879>
- Nagaraj, K., Gs, S., & Sridhar, A. (2019). Encrypting and Preserving Sensitive Attributes in Customer Churn Data Using Novel Dragonfly Based Pseudonymizer Approach. *Information*, 10(9), 274.
<https://doi.org/10.3390/info10090274>
- Nguyen, N. N., Duong, A. T., & Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam. (2021). Comparison of Two Main Approaches for Handling Imbalanced Data

- in Churn Prediction Problem. *Journal of Advances in Information Technology*, 12(1), 29-35. <https://doi.org/10.12720/jait.12.1.29-35>
- Nigam, B., Dugar, H., & Niranjnamurthy, M. (2019). Effectual predicting telecom customer churn using deep neural network. En *International Journal of Engineering and Advanced Technology* (Vol. 8, Número 5, pp. 121-127).
- Óskarsdóttir, M., Van Calster, T., Baesens, B., Lemahieu, W., & Vanthienen, J. (2018). Time series for early churn detection: Using similarity based classification for dynamic networks. *Expert Systems with Applications*, 106, 55-65. <https://doi.org/10.1016/j.eswa.2018.04.003>
- Pamina, J., Raja, B., SathyaBama, S., Soundarya, S., Sruthi, M., Kiruthika, S., Aiswaryadevi, V., & Priyanka, G. (2019). An Effective Classifier for Predicting Churn in Telecommunication. *Jour of Adv Research in Dynamical & Control Systems*, 11. <https://ssrn.com/abstract=3399937>
- Poornappriya, T. S., & Durairaj, M. (2019). High relevancy low redundancy vague set based feature selection method for telecom dataset. *Journal of Intelligent & Fuzzy Systems*, 37(5), 6743-6760. <https://doi.org/10.3233/JIFS-190242>
- Praseeda, C. K., & Shivakumar, B. L. (2021). Fuzzy particle swarm optimization (FPSO) based feature selection and hybrid kernel distance based possibilistic fuzzy local information C-means (HKD-PFLICM) clustering for churn prediction in telecom industry. *SN Applied Sciences*, 3(6), 613. <https://doi.org/10.1007/s42452-021-04576-7>
- Pustokhina, I. V., Pustokhin, D. A., Nguyen, P. T., Elhoseny, M., & Shankar, K. (2023). Multi-objective rain optimization algorithm with WELM model for customer churn prediction in telecommunication sector. *Complex and Intelligent Systems*, 9(4), 3473-3485. Scopus. <https://doi.org/10.1007/s40747-021-00353-6>
- Pustokhina, I. V., Pustokhin, D. A., Rh, A., Jayasankar, T., Jeyalakshmi, C., Díaz, V. G., & Shankar, K. (2021). Dynamic customer churn prediction strategy for business intelligence using text analytics with evolutionary optimization algorithms. *Information Processing & Management*, 58(6), 102706. <https://doi.org/10.1016/j.ipm.2021.102706>
- Rabbah, J., Ridouani, M., & Hassouni, L. (2022). A New Churn Prediction Model Based on Deep Insight Features Transformation for Convolution Neural

- Network Architecture and Stacknet: *International Journal of Web-Based Learning and Teaching Technologies*, 17(1), 1-18.
<https://doi.org/10.4018/ijwltt.300342>
- Ramana, A., Rao, A. S., & Kesavulu Reddy, E. (2019). Applications of business intelligence and decision making for the customer behaviour analysis in telecom industry. *International Journal of Recent Technology and Engineering*, 7(6), 688-693.
- Ramesh, P., Jeba Emilyn, J., & Vijayakumar, V. (2022). Hybrid Artificial Neural Networks Using Customer Churn Prediction. *Wireless Personal Communications*, 124(2), 1695-1709. <https://doi.org/10.1007/s11277-021-09427-7>
- Rani, B., & Kant, S. (2020). Semi-Supervised Learning Approach to Improve Machine Learning Algorithms for Churn Analysis in Telecommunication. *International Journal of Computer Information Systems and Industrial Management Applications*, 12, 265-275.
- Reichheld, F. (2003). The one number you need to grow. 2003, 81(12), 46.
- Renuka Devi, V., Bharathi, G., & Prasad, G. V. S. N. R. V. (2019). Prediction of customer churn in telecom sector using clustering technique. *International Journal of Engineering and Advanced Technology*, 8(6 Special Issue 2), 826-832. Scopus. <https://doi.org/10.35940/ijeat.F1207.0886S219>
- Retana, G. F., Forman, C., & Wu, D. J. (2016). Proactive Customer Education, Customer Retention, and Demand for Technology Support: Evidence from a Field Experiment. *Manufacturing & Service Operations Management*, 18(1), 34-50. <https://doi.org/10.1287/msom.2015.0547>
- Riyadh, H. A., Alfaiza, S. A., Sultan, A. A., Faaeq, M. K., & Dananjoyo, R. (2021). An analysis of CRM practices in the telecommunication industry in Iraq. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(1), 453. <https://doi.org/10.11591/ijeecs.v23.i1.pp453-462>
- Sabbeh, S. F. (2018). Machine-Learning Techniques for Customer Retention: A Comparative Study. *International Journal of Advanced Computer Science and Applications*, 9(2). <https://doi.org/10.14569/IJACSA.2018.090238>
- Saha, L., Tripathy, H. K., Gaber, T., El-Gohary, H., & El-kenawy, E.-S. M. (2023). Deep Churn Prediction Method for Telecommunication Industry.

- Sustainability (Switzerland)*, 15(5). Scopus.
<https://doi.org/10.3390/su15054543>
- Salunkhe, U. R., & Mali, S. N. (2018). A Hybrid Approach for Class Imbalance Problem in Customer Churn Prediction: A Novel Extension to Under-sampling. *International Journal of Intelligent Systems and Applications*, 10(5), 71-81. <https://doi.org/10.5815/ijisa.2018.05.08>
- Sandhya, N., Samuel, P., & Chacko, M. (2019). Feature intersection for agent-based customer churn prediction. *Data Technologies and Applications*, 53(3), 318-332. <https://doi.org/10.1108/DTA-03-2019-0043>
- Sari, R. P., Febriyanto, F., & Adi, A. C. (2023). Analysis Implementation of the Ensemble Algorithm in Predicting Customer Churn in Telco Data: A Comparative Study. *Informatica (Slovenia)*, 47(7), 63-70. Scopus.
<https://doi.org/10.31449/inf.v47i7.4797>
- Sharm, S., Saripudi, S., & Sushasukhanya. (2019). Prediction of customer churn in telecom industries. *International Journal of Recent Technology and Engineering*, 8(1), 369-372.
- Shokouhyar, S., Saeidpour, P., & Otarkhani, A. (2018). Predicting Customers' Churn Using Data Mining Technique and its Effect on the Development of Marketing Applications in Value-Added Services in Telecom Industry: *International Journal of Information Systems in the Service Sector*, 10(4), 59-72. <https://doi.org/10.4018/IJISSS.2018100104>
- Sivasankar, E., & Vijaya, J. (2019). Hybrid PPFCM-ANN model: An efficient system for customer churn prediction through probabilistic possibilistic fuzzy clustering and artificial neural network. *Neural Computing and Applications*, 31(11), 7181-7200. <https://doi.org/10.1007/s00521-018-3548-4>
- Sjarif, N. N. A., Yusof, M. R. M., Wong, D. H.-T., Ya'akob, S., Ibrahim, R., & Osman, M. Z. (2019). A customer Churn prediction using Pearson correlation function and K nearest neighbor algorithm for telecommunication industry. *International Journal of Advances in Soft Computing and its Applications*, 11(2), 46-59.
- Soeini, R. A., & Rodpysh, K. V. (2012). *Applying Data Mining to Insurance Customer Churn Management*. 30, 11.

- Sudharsan, R., & Ganesh, E. N. (2022). A Swish RNN based customer churn prediction for the telecom industry with a novel feature selection strategy. *Connection Science*, 34(1), 1855-1876.
<https://doi.org/10.1080/09540091.2022.2083584>
- Suguna, R., Shyamala Devi, M., & Mathew, R. M. (2019). Customer churn predictive analysis by component minimization using machine learning. *International Journal of Innovative Technology and Exploring Engineering*, 8(8), 3229-3233.
- Swetha, P., & Dayananda, R. B. (2020). Improvised_XgBoost Machine learning Algorithm for Customer Churn Prediction. *EAI Endorsed Transactions on Energy Web*, 7(30), 1-7. Scopus. <https://doi.org/10.4108/eai.13-7-2018.164854>
- Swetha, P., & Dayananda, R. B. (2023). A customer churn prediction model in telecom industry using Improved_XGBoost. *International Journal of Cloud Computing*, 12(2-4), 277-294. Scopus.
<https://doi.org/10.1504/IJCC.2023.130903>
- Tang, Q., Xia, G., & Zhang, X. (2020). A hybrid classification model for churn prediction based on customer clustering. *JOURNAL OF INTELLIGENT & FUZZY SYSTEMS*, 39(1), 69-80. <https://doi.org/10.3233/JIFS-190677>
- Telefónica Movistar. (2020). *Telefónica Movistar—Informe de resultados*.
<https://www.telefonica.com/es/wp-content/uploads/sites/4/2021/07/rdos20t4-esp.pdf>
- Tong, L., Wang, Y., Wen, F., & Li, X. (2017). The research of customer loyalty improvement in telecom industry based on NPS data mining. *China Communications*, 14(11), 260-268. <https://doi.org/10.1109/CC.2017.8233665>
- Toor, A. A., Usman, M., Younas, F., & Fong, A. (2020). A Robust Systematic Approach for Ensuring Optimal Telecom Service Delivery. *IEEE Communications Magazine*, 58(8), 49-53. Scopus.
<https://doi.org/10.1109/MCOM.001.2000258>
- Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in

- Telecom Sector. *IEEE Access*, 7, 60134-60149.
<https://doi.org/10.1109/ACCESS.2019.2914999>
- Ullah, I., Rios, A., Gala, V., & McKeever, S. (2022). Explaining deep learning models for tabular data using layer-wise relevance propagation. *Applied Sciences (Switzerland)*, 12(1). Scopus. <https://doi.org/10.3390/app12010136>
- Uner, M. M., Guven, F., & Cavusgil, S. T. (2020). Churn and loyalty behavior of Turkish digital natives: Empirical insights and managerial implications. *Telecommunications Policy*, 44(4), 101901.
<https://doi.org/10.1016/j.telpol.2019.101901>
- Usman, M., Ahmad, W., & Fong, A. (2021). Design and Implementation of a System for Comparative Analysis of Learning Architectures for Churn Prediction. *IEEE Communications Magazine*, 59(9), 86-90.
<https://doi.org/10.1109/MCOM.110.2100145>
- Usman-Hamza, F. E., Balogun, A. O., Capretz, L. F., Mojeed, H. A., Mahamad, S., Salihu, S. A., Akintola, A. G., Basri, S., Amosa, R. T., & Salahdeen, N. K. (2022). Intelligent Decision Forest Models for Customer Churn Prediction. *Applied Sciences*, 12(16), 8270. <https://doi.org/10.3390/app12168270>
- Vijaya, J., Jubilson, E. A., & Sangam, R. S. (2019). A performance study of dimensionality reduction techniques for telecommunication customer retention. *Journal of Advanced Research in Dynamical and Control Systems*, 11(3 Special Issue), 1799-1804.
- Vijaya, J., & Sivasankar, E. (2018). Computing efficient features using rough set theory combined with ensemble classification techniques to improve the customer churn prediction in telecommunication sector. *Computing*, 100(8), 839-860. <https://doi.org/10.1007/s00607-018-0633-6>
- Vijaya, J., & Sivasankar, E. (2019). An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated annealing. *Cluster Computing*, 22(S5), 10757-10768.
<https://doi.org/10.1007/s10586-017-1172-1>
- Wang, Q.-F., Xu, M., & Hussain, A. (2019). Large-scale Ensemble Model for Customer Churn Prediction in Search Ads. *Cognitive Computation*, 11(2), 262-270. <https://doi.org/10.1007/s12559-018-9608-3>

- Wang, S.-T. (2018). Integrating KPSO and C5.0 to analyze the omnichannel solutions for optimizing telecommunication retail. *Decision Support Systems*, 109, 39-49. <https://doi.org/10.1016/j.dss.2017.12.009>
- Wu, S., Yau, W.-C., Ong, T.-S., & Chong, S.-C. (2021). Integrated Churn Prediction and Customer Segmentation Framework for Telco Business. *IEEE Access*, 9, 62118-62136. <https://doi.org/10.1109/ACCESS.2021.3073776>
- Xia, Y., Cui, B., & Duan, Y. (2022). Analysis and Prediction of Telecom Customer Churn based on Machine Learning. *Highlights in Science, Engineering and Technology*, 16, 131-145. <https://doi.org/10.54097/hset.v16i.2495>
- Xu, Liu, J., Yao, T., & Li, Y. (2023). Prediction and Big Data Impact Analysis of Telecom Churn by Backpropagation Neural Network Algorithm from the Perspective of Business Model. *BIG DATA*. <https://doi.org/10.1089/big.2021.0365>
- Xu, T., Ma, Y., & Kim, K. (2021). Telecom Churn Prediction System Based on Ensemble Learning Using Feature Grouping. *Applied Sciences*, 11(11), 4742. <https://doi.org/10.3390/app11114742>
- Yu, R., An, X., Jin, B., Shi, J., Move, O. A., & Liu, Y. (2018). Particle classification optimization-based BP network for telecommunication customer churn prediction. *Neural Computing and Applications*, 29(3), 707-720. <https://doi.org/10.1007/s00521-016-2477-3>
- Zhai, C., Zhang, M., Xia, X., Miao, Y., & Chen, H. (2022). Customer Churn Prediction Model Based on User Behavior Sequences. *Journal of Donghua University (English Edition)*, 39(6), 597-602. Scopus. <https://doi.org/10.19884/j.1672-5220.202202957>
- Zhang, T., Moro, S., & Ramos, R. F. (2022). A Data-Driven Approach to Improve Customer Churn Prediction Based on Telecom Customer Segmentation. *Future Internet*, 14(3), 94. <https://doi.org/10.3390/fi14030094>
- Zhao, Y., Shao, Z., Zhao, W., Han, J., Zheng, Q., & Jing, R. (2023). Combining unsupervised and supervised classification for customer value discovery in the telecom industry: A deep learning approach. *Computing*, 105(7), 1395-1417. Scopus. <https://doi.org/10.1007/s00607-023-01150-4>

Anexo: Código Python

```
pip install pyodbc
```

```
Collecting pyodbc
  Downloading pyodbc-5.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.7 kB)
  Downloading pyodbc-5.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (336 kB)
    336.0/336.0 kB 4.1 MB/s eta 0:00:00
Installing collected packages: pyodbc
Successfully installed pyodbc-5.2.0
```

```
pip install sweetviz
```

```
Collecting sweetviz
  Downloading sweetviz-2.3.1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: pandas<1.0.0,!=1.0.1,!=1.0.2,>=0.25.3 in /usr/local/lib/python3.10/dist-packages (from sweetviz) (2.2.2)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from sweetviz) (1.26.4)
Requirement already satisfied: matplotlib>=3.1.3 in /usr/local/lib/python3.10/dist-packages (from sweetviz) (3.8.0)
Requirement already satisfied: tqdm>=4.43.0 in /usr/local/lib/python3.10/dist-packages (from sweetviz) (4.66.6)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from sweetviz) (1.13.1)
Requirement already satisfied: Jinja2>=2.11.1 in /usr/local/lib/python3.10/dist-packages (from sweetviz) (3.1.4)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.11.1->sweetviz) (3.0.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<1.0.0,!=1.0.1,!=1.0.2,>=0.25.3->sweetviz) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas<1.0.0,!=1.0.1,!=1.0.2,>=0.25.3->sweetviz) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>=3.1.3->sweetviz) (1.16.0)
  Downloading sweetviz-2.3.1-py3-none-any.whl (15.1 MB)
    15.1/15.1 MB 26.2 MB/s eta 0:00:00
Installing collected packages: sweetviz
Successfully installed sweetviz-2.3.1
```

```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.17.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf<4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: requests>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.64.1)
Requirement already satisfied: tensorboard<2.18,>=2.17 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.0)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.26.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0.44.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages (from keras>=3.2.0->tensorflow) (13.9.3)
Requirement already satisfied: nameex in /usr/local/lib/python3.10/dist-packages (from keras>=3.2.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages (from keras>=3.2.0->tensorflow) (0.13.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (2024.8.30)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow) (3.0.6)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.18,>=2.17->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.0 in /usr/local/lib/python3.10/dist-packages (from rich>keras>=3.2.0->tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>keras>=3.2.0->tensorflow) (2.18.0)
Requirement already satisfied: mdurl<=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>keras>=3.2.0->tensorflow) (0.1.2)
```

```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.5.2)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
```

```
pip install catboost
```

```
Collecting catboost
  Downloading catboost-1.2.7-cp310-cp310-manylinux2014_x86_64.whl.metadata (1.2 kB)
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from catboost) (0.20.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from catboost) (3.8.0)
Requirement already satisfied: numpy<2.0,>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from catboost) (1.26.4)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-packages (from catboost) (2.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from catboost) (1.13.1)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from catboost) (5.24.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2024.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (3.2.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly->catboost) (9.0.0)
Downloading catboost-1.2.7-cp310-cp310-manylinux2014_x86_64.whl (98.7 MB)
----- 98.7/98.7 MB 7.8 MB/s eta 0:00:00
Installing collected packages: catboost
Successfully installed catboost-1.2.7
```

```
!pip install sklearn
```

```
Collecting feature_engine
  Downloading feature_engine-1.8.2-py3-none-any.whl.metadata (9.9 kB)
Requirement already satisfied: numpy>=1.18.2 in /usr/local/lib/python3.10/dist-packages (from feature_engine) (1.26.4)
Requirement already satisfied: pandas>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from feature_engine) (2.2.2)
Requirement already satisfied: scikit-learn>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from feature_engine) (1.5.2)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from feature_engine) (1.13.1)
Requirement already satisfied: statsmodels>=0.11.1 in /usr/local/lib/python3.10/dist-packages (from feature_engine) (0.14.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.2.0->feature_engine) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.2.0->feature_engine) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=2.2.0->feature_engine) (2024.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.4.0->feature_engine) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.4.0->feature_engine) (3.5.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.11.1->feature_engine) (0.5.6)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.11.1->feature_engine) (24.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.6->statsmodels>=0.11.1->feature_engine) (1.16.0)
Downloading feature_engine-1.8.2-py3-none-any.whl (374 kB)
----- 375.0/375.0 kB 5.4 MB/s eta 0:00:00
Installing collected packages: feature_engine
Successfully installed feature_engine-1.8.2
```

```
pip install kneed
```

```
Collecting kneed
  Downloading kneed-0.8.5-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: numpy>=1.14.2 in /usr/local/lib/python3.10/dist-packages (from kneed) (1.26.4)
Requirement already satisfied: scipy>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from kneed) (1.13.1)
Downloading kneed-0.8.5-py3-none-any.whl (10 kB)
Installing collected packages: kneed
Successfully installed kneed-0.8.5
```

```
!pip install -U imbalanced-learn
```

```
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.10/dist-packages (0.12.4)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.26.4)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.13.1)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.5.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (3.5.0)
```

IMPORTS

```
import pandas as pd
import numpy as np
import seaborn as sns
import os
import pyodbc
import statistics
import sweetviz as sv
import matplotlib.pyplot as plt

from feature_engine.imputation import MeanMedianImputer
from feature_engine.imputation import CategoricalImputer
from collections import Counter
import collections
from catboost import CatBoostClassifier
import lightgbm as lgb
from xgboost import XGBClassifier

from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import scale
from sklearn.metrics import accuracy_score, recall_score
from sklearn.model_selection import KFold
from sklearn.feature_selection import RFECV
from sklearn.metrics import confusion_matrix, classification_report, f1_score,
precision_score, recall_score, roc_auc_score, roc_curve
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import make_scorer, roc_auc_score
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score, GridSearchCV
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_predict
from sklearn.decomposition import PCA
from sklearn.pipeline import make_pipeline
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import VotingClassifier
from imblearn.ensemble import RUSBoostClassifier, BalancedRandomForestClassifier
from imblearn.over_sampling import SMOTE
from imblearn.ensemble import BalancedBaggingClassifier
from imblearn.over_sampling import SMOTE, ADASYN
from imblearn.under_sampling import TomekLinks
from imblearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE, ADASYN, BorderlineSMOTE
from imblearn.combine import SMOTEENN
import tensorflow as tf
from tensorflow.keras import layers, models
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from lightgbm import LGBMClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix, classification_report
from keras.models import Model
from keras.layers import Input, Dense, concatenate, Flatten
from keras.optimizers import Adam
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from kneed import KneeLocator
import pickle
```

FUNCIONES

```
def clasificar_generacion(edad):
    edad = int(edad)
    if edad >=40 and edad <=55:      # edad >= 1965 and edad <= 1980:
        return "Generacion_X"
    elif edad >=24 and edad <=39:    # edad>= 1981 and edad <= 1996:
        return "Generacion_Y"
    elif edad >=8 and edad <= 23:    # edad>= 1997 and edad <= 2012:
        return "Generacion_Z"
    elif edad >55:                    # edad <= 1964:
        return "Boomers"
    else:
        return "x"

def perform_clustering(data, column, cluster_label):
    # Seleccionar la columna específica del DataFrame
    selected_data = data[[column]]

    # Normalizar los datos
    scaler = StandardScaler()
    normalized_data = scaler.fit_transform(selected_data)

    # Realizar el clustering utilizando K-Means
    kmeans = KMeans(n_clusters=2) # Puedes ajustar el número de clústeres según tus
necesidades
    kmeans.fit(normalized_data)

    # Agregar las etiquetas de clúster al DataFrame
    data[cluster_label] = kmeans.labels

    return data

def perform_clustering_auto(data, column, cluster_label):
    # Seleccionar la columna específica del DataFrame
    selected_data = data[[column]]

    # Normalizar los datos
    scaler = StandardScaler()
    normalized_data = scaler.fit_transform(selected_data)

    # Calcular la inercia para diferentes números de clústeres
    inertias = []
    for i in range(1, 11): # Puedes ajustar el rango según tus necesidades
        kmeans = KMeans(n_clusters=i)
        kmeans.fit(normalized_data)
        inertias.append(kmeans.inertia )

    # Utilizar el método del codo para encontrar el número óptimo de clústeres
    knee_locator = KneeLocator(range(1, 11), inertias, curve='convex',
direction='decreasing')
    optimal_clusters = knee_locator.elbow

    # Realizar el clustering con el número óptimo de clústeres
    kmeans = KMeans(n_clusters=optimal_clusters)
    kmeans.fit(normalized_data)

    # Asignar etiquetas categóricas a los clústeres
    cluster_labels_mapping = {
        label: f'Cluster_{i + 1}' for i, label in
enumerate(sorted(set(kmeans.labels_)))
    }

    # Agregar las etiquetas de clúster al DataFrame
    data[cluster_label] = [cluster_labels_mapping[label] for label in kmeans.labels ]

    # Mostrar la gráfica del codo para ayudar en la elección del número de clústeres
    knee_locator.plot_knee()
    plt.xlabel('Número de Clústeres')
    plt.ylabel('Inercia')
    plt.title('Método del Codo')
    plt.show()

    return data
```

```

def obtiene_valores_0(df):
    result = pd.DataFrame(columns=['Columna', 'Valor 0', 'Valor 1',
'Participacion Porcentual 0', 'Participacion Porcentual 1'])

    for columna in df.columns:
        # Crear una copia del DataFrame para evitar sobrescribir
        df_copia = df.copy()

        # Crear nuevas columnas 'valor 0' y 'valor 1'
        df_copia['valor 0'] = (df_copia[columna] == 0).astype(int)
        df_copia['valor 1'] = (df_copia[columna] != 0).astype(int)

        # Agrupar por 'valor_0' y 'valor_1' y contar la cantidad
        conteo = df_copia.groupby(['valor 0',
'valor 1']).size().reset index(name='Cantidad')

        # Calcular la participación porcentual
        conteo['Participacion Porcentual'] = (conteo['Cantidad'] /
conteo['Cantidad'].sum()) * 100

        # Filtrar solo las filas relevantes para 0 y 1
        conteo_0 = conteo[conteo['valor 0'] == 1] if 1 in conteo['valor 0'].values
    else pd.DataFrame({'valor_0': [0], 'valor_1': [0], 'Cantidad': [0],
'Participacion Porcentual': [0]})
        conteo_1 = conteo[conteo['valor_1'] == 1] if 1 in conteo['valor_1'].values
    else pd.DataFrame({'valor 0': [0], 'valor 1': [0], 'Cantidad': [0],
'Participacion Porcentual': [0]})

        # Agregar información al DataFrame final
        result = pd.concat([result, pd.DataFrame({'Columna': [columna],
'Valor 0':
conteo_0['Cantidad'].values[0],
'Valor 1':
conteo_1['Cantidad'].values[0],
'Participacion Porcentual_0':
conteo_0['Participacion Porcentual'].values[0],
'Participacion Porcentual 1':
conteo_1['Participacion Porcentual'].values[0]})])

    return result

def greedy_stepwise(X_train, X_test, y_train, y_test, model, metric):
    best_features = []
    best_score = 0
    feature_scores = []

    for i in range(len(X_train.columns)):
        candidate_feature = X_train.columns[i]
        candidate_features = best_features + [candidate_feature]

        model.fit(X_train[candidate_features], y_train)
        y_pred = model.predict(X_test[candidate_features])
        score = metric(y_test, y_pred)

        feature_scores.append((candidate_feature, score))

        if score > best_score:
            best_score = score
            best_features = candidate_features

    # Crear un DataFrame para mostrar los resultados
    feature_scores_df = pd.DataFrame(feature_scores, columns=['Variable', 'Score'])
    feature_scores_df = feature_scores_df.sort_values(by='Score', ascending=False)

    print("Mejores características:")
    print(feature_scores_df)

    return best_features, feature_scores_df

def test_diferencias(churn, no_churn):
    # test de shapiro para validar la normalizacion de los datos
    _, p_no_churn = shapiro(no_churn.squeeze())
    _, p_churn = shapiro(churn.squeeze())

    # test de levene para evaluar la igualdad de las varianzas para una variable

```

```

# prueba de Bartlett se utiliza para probar si k muestras provienen de
poblaciones con la misma varianza.
    , p levene = levene(no churn.squeeze(), churn.squeeze())
    , p bartlett = bartlett(no churn.squeeze(), churn.squeeze())

# si la data no esta normalizada o las varianzas no son iguales, usa Whitney
if p_no_churn > 0.05 and p_churn > 0.05 and p_levene > 0.05 and p_bartlett >
0.05:
    stat, p = f oneway(no churn.squeeze(), churn.squeeze())
    test = 'ANOVA'
else:
    stat, p = kruskal(no churn.squeeze(), churn.squeeze())
    test = 'Kruskal-Wallis'

d = {'VARIABLE': no churn.columns, 'PRUEBA': test, 'VALOR P': p, 'ESTADISTICA':
stat}

return pd.DataFrame(d)

def fit_models_sampling(X_train, y_train, models):
    models_trained = []

    for name, model in models:
        # Incorpora técnicas de remuestreo al modelo usando un pipeline
        if 'ADASYN' in name:
            model = Pipeline([
                ('sampling', ADASYN(sampling strategy=0.1, random state=42)),
                ('model', model)
            ])
        elif 'BorderlineSMOTE' in name:
            model = Pipeline([
                ('sampling', BorderlineSMOTE(sampling strategy=0.1,
random state=42)),
                ('model', model)
            ])
        elif 'SMOTEENN' in name:
            model = Pipeline([
                ('sampling', SMOTEENN(sampling strategy=0.1, random state=42)),
                ('model', model)
            ])
        else:
            model = Pipeline([
                ('sampling', SMOTE(sampling strategy=0.1, random state=42)),
                ('model', model)
            ])

        model.fit(X_train, y_train)
        models_trained.append((name, model))
    return models_trained

def fit_models(X_train: pd.DataFrame , y_train: pd.DataFrame, modelos:list) -> list:
    """
    Funcion para entrenar multiples modelos a la par
    :modelos: list of tuples with model's name and function to train. Ej:
    [('CBC', CatboostClassifier())]
    :return: DataFrame of predictions
    """

    models_trained = []

    for name, model in modelos:
        model.fit(X_train, y_train)
        models_trained.append((name, model))
    return models_trained

def predict_models(x_test: np.ndarray, modelos: list, predict_proba: bool = False,
label: int = 1) -> pd.DataFrame:
    """
    Para predecir modelos a la vez
    modelos: modelos ya entrenados
    predict_proba: bool. True devuelve las probabilidades del evento 1 o 0 según se
defina en el parámetro label
    label: 1 los positivos, 0 los negativos
    Devuelve un dataframe con los resultados de la prediccion, una columna por cada
modelo"""

```

```

dfs = []
predictions = pd.DataFrame(index=x_test_p.index) # Nuevo DataFrame para
almacenar las predicciones

for name, model in modelos:
    column = 'PRED_{}'.format(name)

    if predict_proba:
        y_pred = model.predict_proba(x_test)[: , label]
    else:
        y_pred = model.predict(x_test)

    this_df = pd.DataFrame({column: y_pred}, index=x_test_p.index)
    dfs.append(this_df)
    predictions[column] = y_pred # Agregar la predicción al DataFrame principal

final = pd.concat(dfs, axis=1)

return final, predictions

def run_exps_imb(X: pd.DataFrame, y: pd.DataFrame, modelos:list, estimadores:list,
splits:int = 10) -> pd.DataFrame:
    """
    Con esto pruebo algunos modelos a la vez usando StratifiedKfold
    :modelos: modelos a evaluar. Tienen que venir como una lista de tuplas. Ej:
    (['CBC',CatboostClassifier()])
    :estimadores: preprocesamientos a la fuente. Se usará para crear un pipeline.
    Tienen que venir como una lista de tuplas. Ej:
    (['UnderSampling',RandomUndersampling()])
    :return: Devuelve un dataframe con el resultado de las predicciones con la
    métricas default
    """
    from sklearn import model_selection
    from sklearn.metrics import make_scorer, accuracy_score, precision_score,
recall_score, f1_score
    from imblearn.pipeline import Pipeline

    models = modelos
    scoring = {'accuracy': make_scorer(accuracy_score),
               'precision': make_scorer(precision_score),
               'recall': make_scorer(recall_score),
               'f1_score': make_scorer(f1_score),
               'roc_auc': make_scorer(roc_auc_score)}

    dfs = []
    results = []
    names = []

    for name, model in models:
        est = estimadores.copy()
        est.extend([(name, model)])
        pipeline = Pipeline(est)

        kfold = model_selection.StratifiedKFold(n_splits=splits, shuffle=True,
random_state=69)
        cv_results = model_selection.cross_validate(pipeline, X, y, cv=kfold,
scoring=scoring, n_jobs=-1)
        results.append(cv_results)
        names.append(name)
        this_df = pd.DataFrame(cv_results)
        this_df['model'] = name
        dfs.append(this_df)
        print('1')

    final = pd.concat(dfs, ignore_index=True)
    return final

def correlation(df, tipo):
    df_Corr = pd.DataFrame({'Variable':[], 'Correlacion '+tipo:[0]})
    for i in range(1, len(df.columns)):
        var = df['churn'].corr(df[df.columns[i]], method= tipo)
        df_Corr= pd.concat([df_Corr, pd.DataFrame({'Variable': [df.columns[i]],
'Correlacion '+tipo: [var] }) ])
    return df_Corr

def find_best_hyperparameters(X, y, models, param_grids, scoring_metric='roc_auc',
cv=7):
    best_models = {}

```

```

for (name, model), param_grid in zip(models, param_grids):
    print(f"Searching for the best hyperparameters for {name}...")

    # Utiliza roc auc score como la métrica si está disponible
    if 'roc auc' in scoring_metric:
        scoring = make_scorer(roc_auc_score)
    else:
        scoring = scoring_metric

    grid_search = GridSearchCV(model, param_grid, scoring=scoring, cv=cv,
n jobs=-1)
    grid_search.fit(X, y)

    best_models[name] = {
        'best_params': grid_search.best_params,
        'best_score': grid_search.best_score,
        'best_estimator': grid_search.best_estimator
    }

    print(f"Best parameters for {name}: {grid_search.best_params}")
    print(f"Best {scoring} score: {grid_search.best_score}")
    print("\n")

return best_models

def eval_ensemble(X_train, y_train, X_test, y_test, models, param_grids):
    results = []
    for i, (name, model) in enumerate(models):
        print(f"Searching for the best hyperparameters for {name}...")
        # Obtener los parámetros para el modelo actual
        params = param_grids[i]
        # Utilice f1 score y recall score como las métricas de puntuación
        scoring = {'f1': make_scorer(f1_score), 'recall': make_scorer(recall_score)}
        # Ajustar los hiperparámetros del modelo utilizando el conjunto de
        # entrenamiento y la validación cruzada
        grid_search = GridSearchCV(model, params, cv=7, scoring=scoring, refit='f1')
        grid_search.fit(X_train, y_train)
        # Evalúa el rendimiento del modelo en el conjunto de prueba
        y_pred = grid_search.predict(X_test)
        recall = recall_score(y_test, y_pred)
        f1 = f1_score(y_test, y_pred)
        # Agrega los resultados a la lista
        results.append({'Model': name, 'Best Params': grid_search.best_params,
'Recall': recall, 'F1-Score': f1})
        print(f"Best parameters for {name}: {grid_search.best_params}")
        print(f"Best F1 score: {f1}")
        print(f"Best recall score: {recall}")
        print("\n")
    # Cree un dataframe a partir de los resultados
    results_df = pd.DataFrame(results)
    return results_df

def objective(trial):
    # Definir el espacio de búsqueda
    alpha = trial.suggest_float("alpha", 0.001, 100)
    solver = trial.suggest_categorical("solver", ['auto', 'svd', 'cholesky', 'lsqr',
'sparse_cg', 'sag', 'saga'])
    class_weight = trial.suggest_categorical("class_weight", [None, 'balanced'])

    # Crear el modelo con los hiperparámetros sugeridos
    model = RidgeClassifier(
        alpha=alpha,
        solver=solver,
        class_weight=class_weight
    )

    # Calcular el promedio de puntaje F1 usando validación cruzada

    scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy',
n jobs=-1)

    return np.mean(scores)

```

DATASET

```
df = pd.read_csv('dataset tesis filtro2.txt', sep='|', encoding='latin-1').reset_index()
df.reset_index(drop=True, inplace=True)
df.head(3)
```

index	churn	MES	CAT_COMPORTAMIENTO_PAGO	id	NIVEL_SOCIOECONOMICO	renta	DIAS_EXCESO_MAXPAGO	VALOR_PAGADO	cant_lineas_postpago	...	estado_contribuyente	clase_contribuyente	
0	0	0	202303	EXCELENTE	0909156226593999555268	3.0	55.31	-1.0	61.94	1	...	ACTIVO	RIMPE
1	1	0	202303	EXCELENTE	1305237636593991022980	4.0	25.99	-8.0	29.10	1	...	ACTIVO	RIMPE
2	2	0	202305	EXCELENTE	0103662250593999516894	3.0	20.99	-6.0	23.50	1	...	SUSPENDIDO	OTROS

3 rows × 14 columns

```
df.info(10)
```

Data columns (total 88 columns):

#	Column	Non-Null Count	Dtype
0	index	77520 non-null	int64
1	churn	77520 non-null	int64
2	MES	77520 non-null	int64
3	CAT_COMPORTAMIENTO_PAGO	77355 non-null	object
4	id	77520 non-null	object
5	NIVEL_SOCIOECONOMICO	77310 non-null	float64
6	renta	77520 non-null	float64
7	DIAS_EXCESO_MAXPAGO	64331 non-null	float64
8	VALOR_PAGADO	77520 non-null	float64
9	cant_lineas_postpago	77520 non-null	int64
10	cant_lineas_prepago	77520 non-null	int64
11	ACTIVACIONES_PAQUETE	77520 non-null	float64
12	TARIFA_CUPO	77520 non-null	float64
13	CALIFICACION_cliente	77520 non-null	float64
14	DESCRIPCION_FORMA_PAGO	77520 non-null	object
15	TOTAL_MB_2G_M	77520 non-null	float64
16	TOTAL_MB_3G_M	77520 non-null	float64
17	TOTAL_MB_4G_M	77520 non-null	float64
18	ingresos_excedentes_granel	77520 non-null	object
19	ingresos_adicional_paq	77520 non-null	object
20	minutos_salientes	77520 non-null	float64
21	megas_navegados	77520 non-null	object
22	CANT_VISITAS_CENTROS_ATENCION	77520 non-null	int64
23	CANTON	77340 non-null	object
24	PROVINCIA	77340 non-null	object
25	ANTIG_ULTIMO_FICIAMIENTO	9935 non-null	float64
26	PORC_COMPETENCIA	77520 non-null	float64
27	CANTIDAD_PROGR_COMPETENCIA	77520 non-null	int64
28	VISITAS_INTENCION_RENUNCIA	77520 non-null	int64
29	DIAS_SUSPENSION	77520 non-null	int64
30	SUSP_COBRANZA	77520 non-null	int64
31	MODELO	77514 non-null	object
32	MARCA	77512 non-null	object
33	DEVICE_TYPE	77500 non-null	object
34	DIAS_USO_VOZ	77520 non-null	int64
35	DIAS_USO_MB	77520 non-null	int64
36	DIAS_USO_VOZ_Y_MB	77520 non-null	int64
37	ESTADO	77520 non-null	object
38	PRODUCTO	77520 non-null	object
39	EDAD_CLIENTE	77520 non-null	int64
40	ANTIGUEDAD_CLIENTE	77520 non-null	float64
41	CANTIDAD_SITIO	77520 non-null	int64
42	CUPO_PRECALIFICADO	77520 non-null	float64
43	CANT_LLAMADAS_SALIENTES	77520 non-null	int64
44	CANT_LLAMADAS_ENTRANTES	77520 non-null	int64
45	MINS_ENTRANTES	77520 non-null	float64
46	PORC_MIN_COMPETENCIA	77520 non-null	float64
47	CANTIDAD_MIN_INTERNOS_POS	77520 non-null	int64
48	CANTIDAD_MIN_INTERNOS_PPA	77520 non-null	int64
49	CANTIDAD_MIN_INTERNOS_OTRAS_OPE	77520 non-null	int64
50	PORC_MIN_INTERNOS	77520 non-null	float64
51	PORC_MIN_INTERNOS_POST	77520 non-null	float64
52	PORC_MIN_INTERNOS_PPA	77520 non-null	float64
53	VISITAS_FOR RENUNCIA	77520 non-null	int64
54	MODULO_NEGOCIACION	77520 non-null	int64
55	SOLICITUD_PORTABILIDAD_POR_SERVICIO	77520 non-null	int64
56	SOLICITUD_PORTABILIDAD_POR_DNI	77520 non-null	int64
57	LLAMADAS_CALL_CENTER_SERVICIO	77520 non-null	int64
58	LLAMADAS_CALL_CENTER_DNI	77520 non-null	int64
59	INBOUND_SERVICIO	77520 non-null	int64
60	INBOUND_DNI	77520 non-null	int64
61	SDA_SERVICIO	77520 non-null	int64

```

62 SDA_IDENTIFICACION          77520 non-null   int64
63 PORC_RECAUDACION           77520 non-null   float64
64 CANT_LLAM_CALLCENTER        77520 non-null   int64
65 MORA_CTA                     77499 non-null   object
66 MORA_FIJO                   6519 non-null    object
67 FACTURACION_FIJO            77520 non-null   float64
68 facturacion                  77520 non-null   float64
69 NPS_CAC_CATEG                537 non-null     object
70 NPS_CAC                     77520 non-null   int64
71 NPS_ASESOR_CATEG            528 non-null     object
72 NPS_ASESOR                   77520 non-null   int64
73 clase_vehiculo              77520 non-null   object
74 tipo_vehiculo                77520 non-null   object
75 marca_vehiculo              77520 non-null   object
76 anio_vehiculo                77520 non-null   int64
77 avaluo_vehiculo             77520 non-null   int64
78 estado_contribuyente        77517 non-null   object
79 clase_contribuyente         77517 non-null   object
80 fecha_ini_actividad         77517 non-null   object
81 fecha_suspension            77517 non-null   object
82 numero_establecimiento      77517 non-null   float64
83 codigo_ciiu                  77487 non-null   object
84 tipo_empresa_laboral        77194 non-null   object
85 fecha_ing_afiliado          77502 non-null   object
86 ocupacion                    77502 non-null   object
87 sueldo                       77502 non-null   float64
dtypes: float64(25), int64(35), object(28)

```

```

# Ejecutar estas lineas de codigo en caso de que tenga problemas de formateo de datos
df['MES'] = df['MES'].astype('object')
df['NIVEL SOCIOECONOMICO'] = df['NIVEL SOCIOECONOMICO'].astype('object')
df['renta'] = df['renta'].astype('float')
df['DIAS_EXCESO_MAXPAGO'] = df['DIAS_EXCESO_MAXPAGO'].astype('object')
df['cant lineas postpago'] = df['cant lineas postpago'].astype('int')
df['CANT VISITAS CENTROS ATENCION'] =
df['CANT VISITAS CENTROS ATENCION'].astype('int')
df['CANTIDAD PROGR COMPETENCIA'] = df['CANTIDAD PROGR COMPETENCIA'].astype('int')
df['VISITAS_INTENCION_RENUNCIA'] = df['VISITAS_INTENCION_RENUNCIA'].astype('int')
df['DIAS_SUSPENSION'] = df['DIAS_SUSPENSION'].astype('int')
df['SUSP COBRANZA'] = df['SUSP COBRANZA'].astype('int')
df['SOLICITUD PORTABILIDAD POR SERVICIO'] =
df['SOLICITUD PORTABILIDAD POR SERVICIO'].astype('category')
df['SOLICITUD_PORTABILIDAD_POR_DNI'] =
df['SOLICITUD_PORTABILIDAD_POR_DNI'].astype('category')

```

```

# Se observa la distribucion de la variable objetivo
df["churn"].value_counts()

```

```

count

churn
0    76265
1     1255

```

Se realiza un análisis exploratorio inicial, embebido en el file SWEETVIZ_REPORT.html, el cual permite verificar como está distribuida la data en cada una de las variables, y que valores posee.

Una vez realizado el análisis exploratorio inicial, se procede a procesar las variables categóricas. Para que después se realice el encoding y estas variables aporten

```

#Una vez realizado el Analisis exploratorio, se debe eliminar las variables
categoricas con multiples valores
data = {'PROVINCIA': ['GUAYAS', 'PICHINCHA', 'MANABI', 'EL ORO', 'AZUAY', 'LOS
RIOS', 'TUNGURAHUA', 'LOJA', 'SANTO DOMINGO DE LOS TSACHILAS', 'CHIMBORAZO',
'ESMERALDAS', 'IMBABURA', 'SUCUMBIOS', 'SANTA ELENA', 'ORELLANA', 'COTOPAXI',
'CAÑAR', 'BOLIVAR', 'ZAMORA CHINCHIPE', 'CARCHI', 'MORONA SANTIAGO', 'NAPO', 'GALAPAGOS',
'PASTAZA', 'NO DELIMITADO'],
'REGION':
['COSTA', 'SIERRA', 'COSTA', 'COSTA', 'SIERRA', 'COSTA', 'SIERRA', 'SIERRA', 'COSTA',
'SIERRA', 'COSTA', 'SIERRA', 'ORIENTE', 'COSTA', 'ORIENTE', 'SIERRA', 'SIERRA', 'SIERRA', 'ORI
ENTE', 'SIERRA', 'ORIENTE', 'ORIENTE', 'INSULAR', 'ORIENTE', 'COSTA']}

df_prov = pd.DataFrame(data)
df = pd.merge(df, df_prov, how='left', on='PROVINCIA')
df['REGION'] = df['REGION'].fillna('COSTA')
df.head(3)

```

renta	DIAS_EXCESO_MAXPAGO	VALOR_PAGADO	cant_lineas_postpago	...	clase_contribuyente	fecha_ini_actividad	fecha_suspension	numero_establecimiento	codigo_ciiu	tipo_empresa_laboral	fecha_ing_afiliado	ocupacion	suelo	REGION
55.31	-1.0	61.94	1	...	RIMPE	2018-09-04	1900-01-01	1.0	M702002	3-EMPRESA UNIPERSONAL/ PEQUE A INDUSTRIA	2016-08-01	ASISTENTE / AYUDANTE / AUXILIAR ADMINIST	433.24	COSTA
25.99	-8.0	28.10	1	...	RIMPE	2002-05-27	1900-01-01	1.0	A032102	3-EMPRESA UNIPERSONAL/ PEQUE A INDUSTRIA	2010-11-01	GERENTES Y/O AFINES	444.19	COSTA
20.99	-6.0	23.50	1	...	OTROS	2019-04-10	2019-04-10	1.0	V030000	12-ENTIDADES PUBLICAS CON AUTONOMIA	2011-01-01	INGENIERO DE PROYECTOS	1632.00	SIERRA

```

data = {'tipo empresa laboral': ['2-EMPRESA PRIVADA -SOCIEDADES / COMPANIA', '3-EMPRESA UNIPERSONAL / PEQUE A INDUSTRIA', '14-FUNCION EJECUTIVA, LEGISL, JUDICIAL', '12-ENTIDADES PUBLICAS CON AUTONOMIA', '10-ORGANISMOS DEL REGIMEN SECCIONAL', '16-ENTIDADES P+BLICAS DE EDUCACION SUPER', '32-AFILIACION VOLUNTARIA(TIPEM-32)', '8-ARTESANAL', '1-INSTITUCION FINANCIERA PRIVADA, BANCO', '6-CONSTRUCCION', '31- MICRO EMPRESA (NO ARTESANAL)', '24-MIEMBROS DE SINDICATOS Y COOP DE T', 'NULL', '13-ENTIDADES FINANCIERAS', '27-COOPERA NO CONTROLADAS POR LA SUPERI', '34-RISE-R+GIMEN IMPOSITIVO SIMPLIFICADO', '26-COOP AHORRO CRED. VIVIENDA ETC.CONTR', '25-EMPLEADOR DOMESTICO', '7-AGRICOLA LABORES DE CAMPO', '4-AGRICOLA SEGURO GENERAL', '30-TRABAJADORES AUTONOMOS Y SIN RELACI', '9-NOTARIAS, (LOS REGITRADORES SON P+BLI', '99-POR CLASIFICAR', '5-CURIA DIOSESANA', '29-EMBAJADAS, MISIONES DIPL., CONSULARES', '28-OTRAS FINANC. PRIVAD, MUTUALISTAS ETC', '22-ARTISTAS PROFESIONALES'],
'CLASIFICACION EMPRESA': ['EMPRESA PRIVADA', 'INDEPENDIENTE/OTROS', 'EMPRESA PUBLICA', 'EMPRESA PUBLICA', 'EMPRESA PUBLICA', 'EMPRESA PUBLICA', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'EMPRESA PRIVADA', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'EMPRESA PRIVADA', 'EMPRESA PRIVADA', 'INDEPENDIENTE/OTROS', 'EMPRESA PRIVADA', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'EMPRESA PUBLICA', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'INDEPENDIENTE/OTROS', 'EMPRESA PRIVADA', 'INDEPENDIENTE/OTROS']}
}
df_tip_emp = pd.DataFrame(data)
df = pd.merge(df, df_tip_emp, how='left', on='tipo_empresa_laboral')
df['CLASIFICACION_EMPRESA'] =
df['CLASIFICACION_EMPRESA'].fillna('INDEPENDIENTE/OTROS')
df.head(3)

```

renta	DIAS_EXCESO_MAXPAGO	VALOR_PAGADO	cant_lineas_postpago	...	fecha_ini_actividad	fecha_suspension	numero_establecimiento	codigo_ciiu	tipo_empresa_laboral	fecha_ing_afiliado	ocupacion	suelo	REGION	CLASIFICACION_EMPRESA
55.31	-1.0	61.94	1	...	2018-09-04	1900-01-01	1.0	M702002	3-EMPRESA UNIPERSONAL/ PEQUE A INDUSTRIA	2016-08-01	ASISTENTE / AYUDANTE / AUXILIAR ADMINIST	433.24	COSTA	INDEPENDIENTE/OTROS
25.99	-8.0	28.10	1	...	2002-05-27	1900-01-01	1.0	A032102	3-EMPRESA UNIPERSONAL/ PEQUE A INDUSTRIA	2010-11-01	GERENTES Y/O AFINES	444.19	COSTA	INDEPENDIENTE/OTROS
20.99	-6.0	23.50	1	...	2019-04-10	2019-04-10	1.0	V030000	12-ENTIDADES PUBLICAS CON AUTONOMIA	2011-01-01	INGENIERO DE PROYECTOS	1632.00	SIERRA	EMPRESA PUBLICA

```

data = {'clase_vehiculo':
['AUTOBUS', 'AUTOBUS', 'AUTOBUS', 'AUTOBUS', 'AUTOBUS', 'AUTOBUS', 'AUTOBUS', 'AUTOBUS', 'AUTOBUS', 'AUT
OBUS', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMO
VIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'AUTOMOVIL', 'BUS/CAM
ION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'B
US/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMI
ON', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION', 'BUS/CAMION',
'BUS/CAMION', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA',
'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'CAMIONETA', 'JEEP', 'JEEP',
'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA',
'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA', 'MOTOCICLETA',
'MOTOCICLETA', 'NO ESPECIFICADO', 'OMNIBUS', 'OMNIBUS', 'OMNIBUS', 'TRAILER', 'TRAILER',
'UNIDAD DE CARGA Y REMOLQUE', 'UNIDAD DE CARGA Y REMOLQUE', 'VEH UTILITARIO', 'VEHICULO
ESP', 'VEHICULO ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO
ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO
ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO ESPECIAL', 'VEHICULO
ESPECIAL', 'VEHICULO UTILITARIO', 'VEHICULO UTILITARIO', 'VEHICULO UTILITARIO', 'VEHICULO
UTILITARIO', 'VEHICULO UTILITARIO', 'VEHICULO UTILITARIO', 'VEHICULO UTILITARIO', 'VEHICULO
UTILITARIO', 'VEHICULO UTILITARIO', 'VOLQUETA'],
'tipo_vehiculo': ['BUS TIPO COSTA', 'BUSETA', 'MICROB  SS', 'BUS ESCOLAR', 'BUS/CAMION',
'MICROB  S', 'MINIB  S', 'OMNIBUS', 'MINIB  S', 'SEDAN/HIBRIDO', 'CUPI  ',
'CUP  ', 'HIBRIDO-A', 'DEPORTIVO', 'STATION WAGON', 'CONVERTIBLE', 'HIBRIDO', 'HATCHBACK',
'VAN O FAMILIAR', 'CUPE', 'STATION-ORTOPEDICO', 'BLINDADO', 'MIXTO', 'CAJ  N', 'CAMION DE
REPARTO', 'PLATAFORMA', 'CABEZAL-T', 'PLATAFORMA C', 'CONTENEDOR', 'TANDEM', 'VOLQUETA',
'FURGON', 'PASEO', 'REPARTO', 'CARROCERIAS DE ESTACAS', 'BUS/CAMION', 'CAMION
PEQUE  O', 'GANADERO', 'CAJON-C', 'FURG  N', 'FURGI  N', 'FURGON-C', 'FURGONETA',
'FURGON', 'UTILITY', 'M.CARRO CAB.FURGON', 'CAJ  N', 'CABINA SIMPLE', 'FURGONETA',
'CAMIONETA DE REPARTO', 'CABINA DOBLE', 'CAMION DE REPARTO', 'DOBLE CABINA', 'PICK-
UP', 'HIBRIDO-J', 'JEEP', 'TRAIL', 'TRICAR', 'MOTOCICLETA', 'CROSS', 'CUADRON', 'DEPORTIVA',
'DOS RUEDAS', 'TRES RUEDAS', 'DOS RUEDAS, DEPORTIVA, CROSS', 'CUATRO RUEDAS', 'PASEO',
'MOTOCICLETA', 'MINIB  S', 'BUS ESCOLAR', 'BUS/CAMION', 'CABEZAL', 'CABEZAL-
T', 'SEMIREMOLQUE', 'REMOLQUE', 'JEEP', 'VOLQUETA', 'CISTERNA', 'PLATAFORMA', 'VOLQUETA', 'G
R  A', 'CANASTILLA', 'BLINDADO', 'AMBULANCIA', 'JEEP', 'TANQUERO', 'BASURERO', 'GR  SA', 'HOR

```

VALOR_PAGADO	cant_lineas_postpago	...	fecha_suspension	numero_establecimiento	codigo_ciu	tipo_empresa_laboral	fecha_ing_afiliado	ocupacion	suelo	REGION	CLASIFICACION_EMPRESA	CLASIF_VEHICULO
61.94	1	...	1900-01-01	1.0	M702002	3-EMPRESA UNIPERSONAL / PEQUE A INDUSTR	2016-08-01	ASISTENTE / AYUDANTE / AUXILIAR ADMINIST	433.24	COSTA	INDEPENDIENTE/OTROS	OTROS
29.10	1	...	1900-01-01	1.0	A032102	3-EMPRESA UNIPERSONAL / PEQUE A INDUSTR	2010-11-01	GERENTES Y/O AFINES	444.19	COSTA	INDEPENDIENTE/OTROS	CAMIONETA/JEEP
23.50	1	...	2019-04-10	1.0	V030000	12-ENTIDADES PUBLICAS CON AUTONOMIA	2011-01-01	INGENIERO DE PROYECTOS	1632.00	SIERRA	EMPRESA PUBLICA	SEDAN/HIBRIDO

[illegible]

VALOR_PAGADO	cant_lineas_postago	...	codigo_ciiu	tipo_empresa_laboral	fecha_ing_afiliado	ocupacion	sueldo	REGION	CLASIFICACION_EMPRESA	CLASIF_VEHICULO	generacion	GAMA_ASSIGNADA_DISPPOSITIVO
61.94	1	...	M702002	3-EMPRESA UNIPERSONAL / PEQUE A INDUSTR	2016-06-01	ASISTENTE / AYUDANTE / AUXILIAR ADMINIST	433.24	COSTA	INDEPENDIENTE/OTROS	OTROS	Boomers	ALTA
29.10	1	...	A032102	3-EMPRESA UNIPERSONAL / PEQUE A INDUSTR	2010-11-01	GERENTES Y/O AFINES	444.19	COSTA	INDEPENDIENTE/OTROS	CAMIONETA/JEEP	Boomers	BASICA
23.50	1	...	V030000	12-ENTIDADES PUBLICAS CON AUTONOMIA	2011-01-01	INGENIERO DE PROYECTOS	1632.00	SIERRA	EMPRESA PUBLICA	SEDAN/HIBRIDO	Generacion_X	ALTA

```
# Definir los límites de los intervalos
bins = [0, 24, 48, 72, 400]

# Definir las etiquetas para cada intervalo
labels = ['Hasta 2 Anios', 'De 2 a 4 Anios', 'De 4 a 6 Anios', 'Mas de 6 Anios']

# Aplicar la función cut para agrupar la variable
df['ANTIGUEDAD_CLTE'] = pd.cut(df['ANTIGUEDAD_CLIENTE'], bins=bins, labels=labels,
right=False)

df['ANTIGUEDAD_CLTE'] = df['ANTIGUEDAD_CLTE'].fillna('Hasta 2 Anios')
df.groupby(['ANTIGUEDAD_CLTE']).size().reset index()
```

	ANTIGUEDAD_CLTE	0
0	Hasta 2 Anios	2576
1	De 2 a 4 Anios	12217
2	De 4 a 6 Anios	19811
3	Mas de 6 Anios	42916

```
# Definir los límites de los intervalos
bins = [0, 80, 90, 120]

# Definir las etiquetas para cada intervalo
labels = ['Riesgo Alto', 'Riesgo Medio', 'Riesgo Bajo']

# Aplicar la función cut para agrupar la variable
df['CALIFICACION_CLTE'] = pd.cut(df['CALIFICACION cliente'], bins=bins,
labels=labels, right=False)
df['CALIFICACION_CLTE'] = df['CALIFICACION_CLTE'].fillna('Riesgo Medio')
df.groupby(['CALIFICACION_CLTE']).size().reset_index()
```

	CALIFICACION_CLTE	0
0	Riesgo Alto	4299
1	Riesgo Medio	23532
2	Riesgo Bajo	49689

```
# Definir los límites de los intervalos
bins = [0, 5, 10, 30]
# Definir las etiquetas para cada intervalo
labels = ['LLama a menos de 5 Personas', 'Llama de 6 a 10 personas', 'Llama a mas de 10 personas']
# Aplicar la función cut para agrupar la variable
df['CANT PERSONAS MISMA OPE'] = pd.cut(df['CANTIDAD MIN INTERNOS POS'], bins=bins,
labels=labels, right=False)
df['CANT PERSONAS MISMA OPE'] = df['CANT PERSONAS MISMA OPE'].fillna('LLama a menos de 5 Personas')

df.groupby(['CANT_PERSONAS_MISMA_OPE']).size().reset_index()
```

	CANT_PERSONAS_MISMA_OPE	0
0	LLama a menos de 5 Personas	31881
1	Llama de 6 a 10 personas	23356
2	Llama a mas de 10 personas	22283

```
# Definir los límites de los intervalos
bins = [0, 5, 10, 30]
# Definir las etiquetas para cada intervalo
labels = ['LLama a menos de 5 Personas', 'Llama de 6 a 10 personas', 'Llama a mas de 10 personas']
# Aplicar la función cut para agrupar la variable
df['CANT PERSONAS OTRA OPE'] = pd.cut(df['CANTIDAD MIN INTERNOS OTRAS OPE'],
bins=bins, labels=labels, right=False)
df['CANT_PERSONAS_OTRA_OPE'] = df['CANT_PERSONAS_OTRA_OPE'].fillna('LLama a menos de 5 Personas')

df.groupby(['CANT_PERSONAS_OTRA_OPE']).size().reset_index()
```

	CANT_PERSONAS_OTRA_OPE	0
0	LLama a menos de 5 Personas	38263
1	Llama de 6 a 10 personas	22335
2	Llama a mas de 10 personas	16922

```
# Definir los límites de los intervalos
bins = [0, 5, 10, 30]
# Definir las etiquetas para cada intervalo
labels = ['LLama a menos de 5 Personas', 'Llama de 6 a 10 personas', 'Llama a mas de 10 personas']
# Aplicar la función cut para agrupar la variable
```

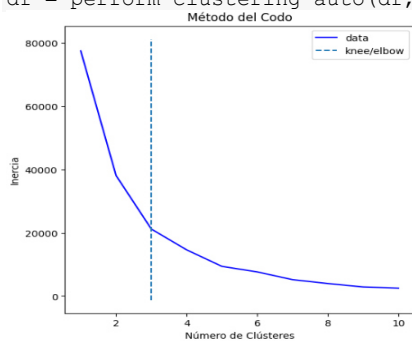
```
df['CANT_PERSONAS_PPA'] = pd.cut(df['CANTIDAD_MIN_INTERNOS_PPA'], bins=bins,
labels=labels, right=False)
df['CANT PERSONAS PPA'] = df['CANT PERSONAS PPA'].fillna('LLama a menos de 5
Personas')
df.groupby(['CANT PERSONAS PPA']).size().reset index()
```

	CANT_PERSONAS_PPA	0
0	LLama a menos de 5 Personas	54422
1	Llama de 6 a 10 personas	16504
2	Llama a mas de 10 personas	6594

```
# Definir los límites de los intervalos
bins = [-30, 0, 60]
# Definir las etiquetas para cada intervalo
labels = ['Paga a tiempo', 'No paga a tiempo']
# Aplicar la función cut para agrupar la variable
df['DIAS_EXC_MAXPAGO'] = pd.cut(df['DIAS_EXCESO_MAXPAGO'], bins=bins, labels=labels,
right=False)
df['DIAS_EXC_MAXPAGO'] = df['DIAS_EXC_MAXPAGO'].fillna('Paga a tiempo')
df.groupby(['DIAS_EXC_MAXPAGO']).size().reset_index()
```

	DIAS_EXC_MAXPAGO	0
0	Paga a tiempo	76159
1	No paga a tiempo	1361

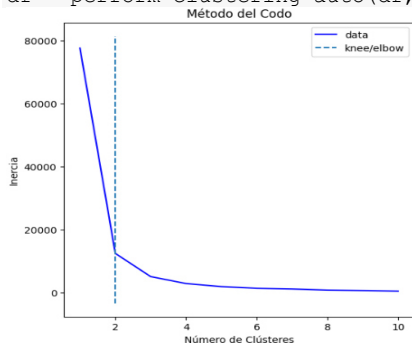
```
df = perform clustering auto(df, 'facturacion', 'FACT')
```



```
df.groupby(['FACT']).size().reset_index()
```

	FACT	0
0	Cluster_1	2836
1	Cluster_2	37817
2	Cluster_3	36867

```
df = perform clustering auto(df, 'FACTURACION_FIJO', 'FACT_FIJO')
```



```
df.groupby(['FACT_FIJO']).size().reset_index()
```

	FACT_FIJO	0
0	Cluster_1	3802
1	Cluster_2	73718

```
df['TOTAL_MB'] = df['TOTAL_MB_4G_M'] + df['TOTAL_MB_3G_M'] + df['TOTAL_MB_2G_M']
df['TOTAL_MB'] = df['TOTAL_MB'].fillna(0)
```

```
# Definir los límites de los intervalos
bins = [0, 10000, 20000, 90000]

# Definir las etiquetas para cada intervalo
labels = ['Bajo consumo', 'Consumo Medio', 'Alto consumo']

# Aplicar la función cut para agrupar la variable
df['CLASIFICACION_MB'] = pd.cut(df['TOTAL_MB'], bins=bins, labels=labels,
right=False)
df['CLASIFICACION_MB'] = df['CLASIFICACION_MB'].fillna('Bajo consumo')

df.groupby(['CLASIFICACION_MB']).size().reset_index()
```

	CLASIFICACION_MB	0
0	Bajo consumo	35975
1	Consumo_Medio	22292
2	Alto consumo	19253

```
# Definir los límites de los intervalos
bins = [0, 500, 1000, 1500, 3000, 50000]

# Definir las etiquetas para cada intervalo
labels = ['Basico', 'Medio Basico', 'Medio', 'Medio Alto', 'Alto']

# Aplicar la función cut para agrupar la variable
df['SALARIO'] = pd.cut(df['sueldo'], bins=bins, labels=labels, right=False)
df['SALARIO'] = df['SALARIO'].fillna('Basico')
```

```
# Definir los límites de los intervalos
bins = [0, 8000, 15000, 28000, 80000]

# Definir las etiquetas para cada intervalo
labels = ['Bajo', 'Medio Bajo', 'Medio', 'Alto']

# Aplicar la función cut para agrupar la variable
df['GAMA_VEHICULO'] = pd.cut(df['avaluo_vehiculo'], bins=bins, labels=labels,
right=False)
df['GAMA_VEHICULO'] = df['GAMA_VEHICULO'].fillna('Bajo')
```

```
# Definir los límites de los intervalos
bins = [0, 40, 70, 120, 500]

# Definir las etiquetas para cada intervalo
labels = ['Bajo', 'Medio Bajo', 'Medio', 'Alto']

# Aplicar la función cut para agrupar la variable
df['CANT_SITIO'] = pd.cut(df['CANTIDAD_SITIO'], bins=bins, labels=labels,
right=False)
df['CANT_SITIO'] = df['CANT_SITIO'].fillna('Bajo')
```

```
# Definir los límites de los intervalos
bins = [0, 15, 30, 31]

# Definir las etiquetas para cada intervalo
labels = ['Usa pocos dias', 'Usa mas de la mitad del periodo', 'periodo completo']

# Aplicar la función cut para agrupar la variable
df['USO_VOZ_Y_MB'] = pd.cut(df['DIAS_USO_VOZ_Y_MB'], bins=bins, labels=labels,
right=False)
df['USO_VOZ_Y_MB'] = df['USO_VOZ_Y_MB'].fillna('Usa pocos dias')
```

```
# Definir los límites de los intervalos
bins = [0, 15, 30, 31]

# Definir las etiquetas para cada intervalo
labels = ['Usa pocos dias', 'Usa mas de la mitad del periodo', 'periodo completo']

# Aplicar la función cut para agrupar la variable
df['USO_MB'] = pd.cut(df['DIAS_USO_MB'], bins=bins, labels=labels, right=False)
df['USO_MB'] = df['USO_MB'].fillna('Usa pocos dias')
```

```
# Definir los límites de los intervalos
bins = [0, 1, 2, 50]

# Definir las etiquetas para cada intervalo
labels = ['No tiene', 'Tiene 1 linea', 'Tiene mas de 1 linea']

# Aplicar la función cut para agrupar la variable
df['LINEAS_PPA'] = pd.cut(df['cant lineas prepago'], bins=bins, labels=labels,
right=False)
df['LINEAS_PPA'] = df['LINEAS_PPA'].fillna('No tiene')

# Definir los límites de los intervalos
bins = [0, 2, 50]
# Definir las etiquetas para cada intervalo
labels = ['No tiene o Tiene 1', 'Tiene varios']
# Aplicar la función cut para agrupar la variable
df['ESTABLECIMIENTO'] = pd.cut(df['numero_establecimiento'], bins=bins,
labels=labels, right=False)

df['ESTABLECIMIENTO'] = df['ESTABLECIMIENTO'].fillna('No tiene o Tiene 1')
df.groupby(['ESTABLECIMIENTO']).size().reset_index()
```

	ESTABLECIMIENTO	0
0	No tiene o Tiene 1	61788
1	Tiene varios	15732

```
# Definir los límites de los intervalos
bins = [12, 16, 21, 26, 60]

# Definir las etiquetas para cada intervalo
labels = ['Bajo', 'Medio Bajo', 'Medio', 'Alto']

# Aplicar la función cut para agrupar la variable
df['TARIFA'] = pd.cut(df['TARIFA CUPO'], bins=bins, labels=labels, right=False)
df['TARIFA'] = df['TARIFA'].fillna('Bajo')
```

FEATURE SELECTION

```
# Se eliminan las variables reemplazadas
df = df.drop('clase_vehiculo', axis=1)
df = df.drop('tipo_vehiculo', axis=1)
df = df.drop('tipo_empresa_laboral', axis=1)
df = df.drop('PROVINCIA', axis=1)
df = df.drop('CANTON', axis=1)
df = df.drop('MARCA', axis=1)
df = df.drop('MODELO', axis=1)
df = df.drop('EDAD_CLIENTE', axis=1)
df = df.drop('ANTIGUEDAD_CLIENTE', axis=1)
df = df.drop('CALIFICACION_cliente', axis=1)
df = df.drop('CANTIDAD_MIN_INTERNOS_POS', axis=1)
df = df.drop('CANTIDAD_MIN_INTERNOS_OTRAS_OPE', axis=1)
df = df.drop('CANTIDAD_MIN_INTERNOS_PPA', axis=1)
df = df.drop('sueldo', axis=1)
df = df.drop('avaluo_vehiculo', axis=1)
df = df.drop('CANTIDAD_SITIO', axis=1)
df = df.drop('DIAS_USO_VOZ_Y_MB', axis=1)
df = df.drop('DIAS_USO_MB', axis=1)
df = df.drop('cant_lineas_prepago', axis=1)
df = df.drop('numero_establecimiento', axis=1)
df = df.drop('TARIFA CUPO', axis=1)
df = df.drop('TOTAL_MB_2G_M', axis=1)
df = df.drop('TOTAL_MB_3G_M', axis=1)
df = df.drop('TOTAL_MB_4G_M', axis=1)
```

```
null_counts= df.isna().sum()
null_counts = null_counts[null_counts > 0].sort_values(ascending=False)
null_counts
```

	0
NPS_ASESOR_CATEG	76992
NPS_CAC_CATEG	76983
MORA_FIJO	71001
ANTIG_ULTIMO_FICIAMIENTO	67585
DIAS_EXCESO_MAXPAGO	13189
NIVEL_SOCIOECONOMICO	210
CAT_COMPORTAMIENTO_PAGO	165
codigo_ciu	33
MORA_CTA	21
DEVICE_TYPE	20
fecha_ing_afiliado	18
ocupacion	18
estado_contribuyente	3
clase_contribuyente	3
fecha_ini_actividad	3
fecha_suspension	3

```
# Se eliminan las variables con mayor cantidad de registros nulos
df = df.drop(['NPS_ASESOR_CATEG'], axis=1)
df = df.drop(['NPS_CAC_CATEG'], axis=1)
df = df.drop(['MORA_FIJO'], axis=1)
df = df.drop(['DIAS_EXCESO_MAXPAGO'], axis=1)
df = df.drop(['ANTIG_ULTIMO_FICIAMIENTO'], axis=1)
df_num = df.select_dtypes(include=['number'])
df_str = df.select_dtypes(exclude=['number']).iloc[:,1:]
```

Correlacion estadistica

```
df_Corr_spearman= correlation(df_num, 'spearman')
df_Corr_pearson= correlation(df_num, 'pearson')
df_Corr_kendall= correlation(df_num, 'kendall')

df_Corr = pd.concat([df_Corr_spearman, df_Corr_pearson['Correlacion pearson'],
df_Corr_kendall['Correlacion kendall'] ], axis=1)
df_Corr
```

Variable	Correlacion spearman	Correlacion pearson	Correlacion kendall
	0.000000	0.000000	0.000000
churn	1.000000	1.000000	1.000000
renta	0.010671	0.005960	0.008807
VALOR_PAGADO	-0.013480	-0.015582	-0.011216
cant_lineas_postpago	NaN	NaN	NaN
ACTIVACIONES_PAQUETE	0.008817	0.016776	0.008518
minutos_salientes	-0.050514	-0.031841	-0.042071
CANT_VISITAS_CENTROS_ATENCION	0.464844	0.449615	0.464731
PORC_COMPETENCIA	-0.024596	-0.016845	-0.020309
CANTIDAD_PROGR_COMPETENCIA	-0.000714	-0.001024	-0.000714
VISITAS_INTENCION_RENUNCIA	0.533288	0.518847	0.533228
DIAS_SUSPENSION	-0.001596	-0.001078	-0.001596
SUSP_COBRANZA	0.019999	0.020156	0.019998
DIAS_USO_VOZ	-0.070292	-0.104870	-0.059236
CUPO_PRECALIFICADO	-0.022120	-0.018495	-0.018146
CANT_LLAMADAS_SALIENTES	-0.052267	-0.033132	-0.043618
CANT_LLAMADAS_ENTRANTES	-0.055729	-0.037625	-0.046499
MINS_ENTRANTES	-0.053815	-0.031960	-0.044799
PORC_MIN_COMPETENCIA	-0.000695	0.001070	-0.000693
PORC_MIN_INTERNOS	-0.052615	-0.071548	-0.043421
PORC_MIN_INTERNOS_POST	-0.085086	-0.106671	-0.070982
PORC_MIN_INTERNOS_PPA	-0.005718	0.007988	-0.004797
VISITAS_POR_RENUNCIA	0.531714	0.517164	0.531655

```
# Se dropea la(s) variable(s) que no tienen correlacion
df = df.drop('cant_lineas_postpago', axis=1)
df_num = df_num.drop('cant_lineas_postpago', axis=1)
```

```
# Se eliminan las variables con baja correlacion
nombre columna = 'Variable'
columnas a convertir = df Corr.columns[df Corr.columns != nombre columna]
df Corr[columnas a convertir] = df Corr[columnas a convertir].apply(pd.to numeric,
errors='coerce')

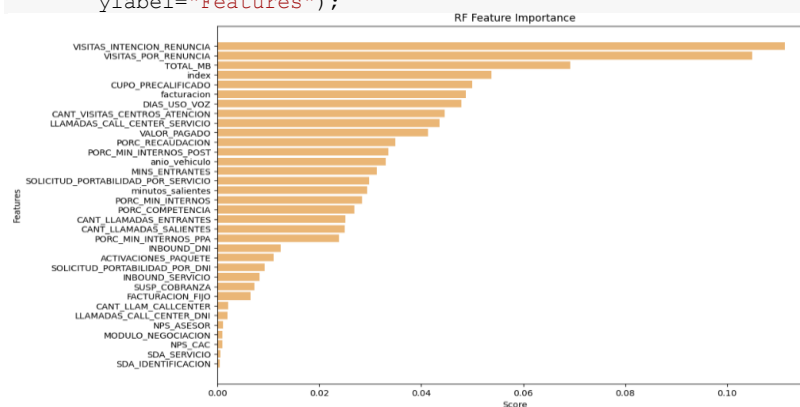
df_filtered =
df Corr.loc[df Corr.drop(columns=['Variable']).abs().lt(0.002).sum(axis=1) >= 2]
df_filtered
df = df.drop(['CANTIDAD PROGR COMPETENCIA'], axis=1)
df = df.drop(['DIAS SUSPENSION'], axis=1)
df = df.drop(['PORC_MIN_COMPETENCIA'], axis=1)
df_num = df_num.drop(['CANTIDAD_PROGR_COMPETENCIA'], axis=1)
df_num = df_num.drop(['DIAS SUSPENSION'], axis=1)
df_num = df_num.drop(['PORC_MIN_COMPETENCIA'], axis=1)
```

Correlaciones usando algoritmos

```
# Obtenemos nuestras variables independientes y dependientes
X = df_num.drop(columns=["churn"], axis=1)
y = df_num["churn"]

# Realizamos el split de datos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Realizamos el Feature Selection usando el algoritmo de RandomForestClassifier
rf = RandomForestClassifier()
# Realizamos el entrenamiento con los datos de entrenamiento
rf.fit(X_train, y_train)
f_i = list(zip(X_train.columns, rf.feature_importances_))
f_i.sort(key = lambda x : x[1])
fig, ax = plt.subplots(figsize=(12,8))
ax.barh([x[0] for x in f_i], [x[1] for x in f_i], color="#eab676");
ax.set(title="RF Feature Importance",
       xlabel="Score",
       ylabel="Features");
```

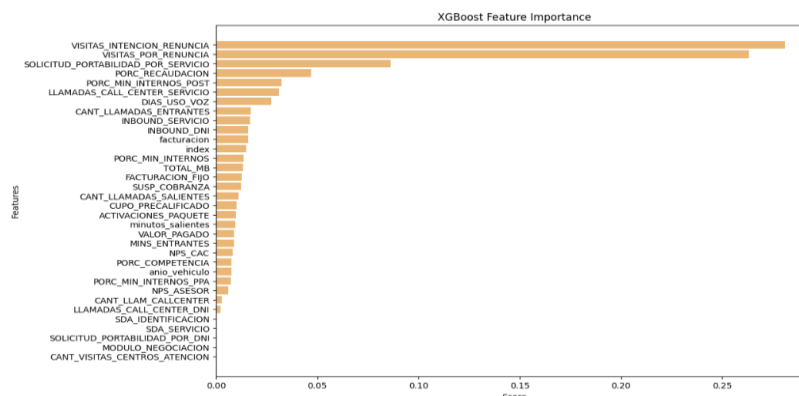


```
# Crear un DataFrame con las características importantes
feature_importance_rf = pd.DataFrame(f_i, columns=['Variable', 'Valor'])

# Ordenar el DataFrame por valores
feature_importance_rf = feature_importance_rf.sort_values(by='Valor',
ascending=False)

# Mostrar el DataFrame
feature_importance_rf

model_xgboost = XGBClassifier()
model_xgboost.fit(X_train, y_train)
# Realizamos un plot de las características más importantes obtenidas
f_i_xgb = list(zip(X_train.columns, model_xgboost.feature_importances_))
f_i_xgb.sort(key = lambda x : x[1])
fig, ax = plt.subplots(figsize=(12,8));
#x_train_labels = X_train.columns.tolist()
ax.barh([x[0] for x in f_i_xgb], [x[1] for x in f_i_xgb], color="#eab676");
ax.set(title="XGBoost Feature Importance",
       xlabel="Score",
       ylabel="Features");
```



```
# Crear un DataFrame con las características importantes
feature_importance_xgb = pd.DataFrame(f i xgb, columns=['Variable', 'Valor'])

# Ordenar el DataFrame por valores
feature_importance_xgb = feature_importance_xgb.sort_values(by='Valor',
ascending=False)

df_merged = pd.merge(feature_importance_rf, feature_importance_xgb, on='Variable',
how='inner')
df_filtrado = df_merged[(df_merged['Valor_xgb'] < 0.001) | (df_merged['Valor_rf'] <
0.002)]
df_filtrado
```

Variable	Valor_x	Valor_y
CANT_VISITAS_CENTROS_ATENCION	0.044590	0.000000
SOLICITUD_PORTABILIDAD_POR_DNI	0.009278	0.000000
MODULO_NEGOCIACION	0.001008	0.000000
NPS_CAC	0.000987	0.008348
SDA_SERVICIO	0.000582	0.000000
SDA_IDENTIFICACION	0.000463	0.000000

```
df = df.drop(['CANT_VISITAS_CENTROS_ATENCION'], axis=1)
df = df.drop(['SOLICITUD_PORTABILIDAD_POR_DNI'], axis=1)
df = df.drop(['NPS_CAC'], axis=1)
df = df.drop(['SDA_SERVICIO'], axis=1)
df = df.drop(['SDA_IDENTIFICACION'], axis=1)
df = df.drop(['MODULO_NEGOCIACION'], axis=1)
```

Greedy Stepwise: para realizar la búsqueda hacia adelante

```
# Crear un modelo de clasificación (por ejemplo, RandomForestClassifier)
classifier_model = RandomForestClassifier(random state=42)

# Seleccionar características utilizando búsqueda hacia adelante
selected_features, feature_scores = greedy_stepwise(X_train, X_test, y_train, y_test,
classifier_model, accuracy score)
```

Mejores características:

	Variable	Score
32	ano_vehiculo	0.989680
29	facturacion	0.989551
21	LLAMADAS_CALL_CENTER_DNI	0.989508
20	LLAMADAS_CALL_CENTER_SERVICIO	0.989465
33	TOTAL_MB	0.989422
30	NPS_CAC	0.989336
22	INBOUND_SERVICIO	0.989336
31	NPS_ASESOR	0.989293
26	PORC_RECAUDACION	0.989250
28	FACTURACION_FIJO	0.989207
18	SOLICITUD_PORTABILIDAD_POR_SERVICIO	0.989164
10	CANT_LLAMADAS_SALIENTES	0.989164
24	SDA_SERVICIO	0.989121
17	MODULO_NEGOCIACION	0.989121
27	CANT_LLAM_CALLCENTER	0.989078
19	SOLICITUD_PORTABILIDAD_POR_DNI	0.989078
16	VISITAS_POR_RENUNCIA	0.989035
13	PORC_MIN_INTERNOS	0.989035
9	CUPO_PRECALIFICADO	0.989035
15	PORC_MIN_INTERNOS_PPA	0.989035
25	SDA_IDENTIFICACION	0.989035
12	MINS_ENTRANTES	0.988992
11	CANT_LLAMADAS_ENTRANTES	0.988992
23	INBOUND_DNI	0.988992
14	PORC_MIN_INTERNOS_POST	0.988906
7	SUSP_COBRANZA	0.988605
8	DIAS_USO_VOZ	0.988605
6	VISITAS_INTENCION_RENUNCIA	0.988433
5	PORC_COMPETENCIA	0.986842
4	CANT_VISITAS_CENTROS_ATENCION	0.986670
3	minutos_salientes	0.983617
1	VALOR_PAGADO	0.982155
2	ACTIVACIONES_PAQUETE	0.981940
0	index	0.974329

DATASET FINAL

#Se eligen las variables que pasaron el proceso de eliminacion y se adicionan las variables que fueron procesadas de manera independiente

```
df_entrena=
df[['churn', 'VALOR PAGADO', 'ACTIVACIONES PAQUETE', 'minutos salientes', 'PORC COMPETENCIA', 'VISITAS INTENCION RENUNCIA', 'SUSP COBRANZA', 'DIAS USO VOZ', 'CUPO PRECALIFICADO', 'CANT LLAMADAS SALIENTES', 'CANT LLAMADAS ENTRANTES', 'MINS ENTRANTES', 'PORC MIN INTERNOS', 'PORC MIN INTERNOS POST', 'PORC MIN INTERNOS PPA', 'VISITAS POR RENUNCIA', 'SOLICITUD PORTABILIDAD POR SERVICIO', 'LLAMADAS CALL CENTER SERVICIO', 'LLAMADAS CALL CENTER DNI', 'INBOUND SERVICIO', 'INBOUND DNI', 'PORC RECAUDACION', 'CANT LLAM CALLCENTER', 'FACTURACION FIJO', 'facturacion', 'NPS ASESOR', 'anio vehiculo', 'GAMA ASIGNADA DISPOSITIVO', 'CLASIF VEHICULO', 'ANTIGUEDAD CLTE', 'CALIFICACION CLTE', 'CANT PERSONAS MISMA OPE', 'CANT PERSONAS OTRA OPE', 'CANT PERSONAS PPA', 'SALARIO', 'GAMA VEHICULO', 'USO MB', 'LINEAS PPA', 'TARIFA', 'ESTABLECIMIENTO', 'CLASIFICACION EMPRESA']]
df_entrena.dropna()
```

```
df_entrena["churn"].value counts()
#76265 /1255
```

```
# Crear un objeto LabelEncoder
le = LabelEncoder()
```

```
#df_entrena=pd.get_dummies(df_entrena, columns=['DESCRIPCION FORMA PAGO'], prefix='Forma_pago')
df_entrena=pd.get_dummies(df_entrena, columns=['GAMA ASIGNADA DISPOSITIVO'], prefix='tipo_disposit') # puede cambiarse
df_entrena=pd.get_dummies(df_entrena, columns=['CLASIF VEHICULO'], prefix='Clasif vehiculo')
df_entrena=pd.get_dummies(df_entrena, columns=['CLASIFICACION EMPRESA'], prefix='Clasif_empresa')

df_entrena=pd.get_dummies(df_entrena, columns=['ANTIGUEDAD CLTE'], prefix='antig')
df_entrena[['CALIFICACION CLTE']] = df_entrena[['CALIFICACION CLTE']].apply(le.fit transform)
df_entrena[['CANT PERSONAS MISMA OPE']] = df_entrena[['CANT PERSONAS MISMA OPE']].apply(le.fit transform)
df_entrena[['CANT PERSONAS OTRA OPE']] = df_entrena[['CANT PERSONAS OTRA OPE']].apply(le.fit transform)
df_entrena[['CANT PERSONAS PPA']] = df_entrena[['CANT PERSONAS PPA']].apply(le.fit transform)
df_entrena[['SALARIO']] = df_entrena[['SALARIO']].apply(le.fit transform)
df_entrena=pd.get_dummies(df_entrena, columns=['GAMA VEHICULO'], prefix='GAMA VEHICULO')
df_entrena[['USO MB']] = df_entrena[['USO MB']].apply(le.fit transform)
df_entrena[['LINEAS PPA']] = df_entrena[['LINEAS PPA']].apply(le.fit transform)
df_entrena=pd.get_dummies(df_entrena, columns=['ESTABLECIMIENTO'], prefix='ESTABLECIMIENTO')
df_entrena[['TARIFA']] = df_entrena[['TARIFA']].apply(le.fit transform)
```

```
df_entrena.head(3)
```

churn	VALOR PAGADO	ACTIVACIONES PAQUETE	minutos salientes	PORC COMPETENCIA	VISITAS_INTENCION_RENUNCIA	SUSP_COBRANZA	DIAS_USO_VOZ	CUPO_PRECALIFICADO	CANT_LLAMADAS_SALIENTES	...	antig_Hasta 2 Años	antig_De 2 a 4 Años	antig_De 4 a 6 Años	antig_Mas de 6 Años	GAMA_VEHICU
0	61.94	22.2	76.333332	0.240000	0	1	31	200.0	115	...	False	False	False	True	
0	29.10	0.0	245.666664	0.080000	0	0	31	116.4	144	...	False	False	False	True	
0	23.50	0.0	0.000000	0.545455	0	0	26	94.0	0	...	False	False	False	True	

us x 57 columns


```
resultados = pd.concat(dfs, ignore_index=True)
resultados
```

	MODELO	GINI	ROC_AUC	PRECISION	F1_SCORE	RECALL	TRUE_POSITIVE	FALSE_POSITIVE	TRUE_NEGATIVE	FALSE_NEGATIVE
	PRED_DT	0.677982	0.838991	0.607774	0.644195	0.685259	172	111	15142	79
	PRED_RFC	0.597282	0.798641	0.967742	0.738916	0.597610	150	5	15248	101
	PRED_LGR	0.453528	0.726764	0.919355	0.608000	0.454183	114	10	15243	137
	PRED_XGB	0.716804	0.858402	0.972973	0.825688	0.717131	180	5	15248	71
	PRED_CVC	0.692637	0.846319	0.950820	0.801843	0.693227	174	9	15244	77
	PRED_LGBM	0.712623	0.856311	0.957219	0.817352	0.713147	179	8	15245	72
	PRED_Logistic Regression	0.726960	0.863480	0.145428	0.246341	0.804781	202	1187	14066	49
	PRED_Random Forest	0.561425	0.780713	0.965753	0.710327	0.561753	141	5	15248	110
	PRED_BRRF	0.811387	0.905694	0.223602	0.354971	0.860558	216	750	14503	35
	PRED_AdaBoost	0.425311	0.712656	0.877049	0.573727	0.426295	107	15	15238	144
	PRED_CatBoost	0.724641	0.862320	0.962963	0.827273	0.725100	182	7	15246	69

```
resultados_filtrados = resultados[resultados['F1_SCORE'] > 0.8]
resultados_filtrados
```

BALANCEO

```
sc = StandardScaler()
sc = sc.fit(X_train)
```

```
models = [
    ('SMOTE_XGB', XGBClassifier(learning_rate
=0.1,gamma=0.25,use label encoder=False, n jobs=-1)),
    ('SMOTE_CatBoost', CatBoostClassifier(verbose=False, depth=7, 12 leaf reg=1,
iterations=250, learning_rate=0.1)),
    ('SMOTE_LGBM', lgb.LGBMClassifier()),
    ('ADASYN_XGB', XGBClassifier(learning_rate
=0.1,gamma=0.25,use label encoder=False, n jobs=-1)),
    ('ADASYN_CatBoost', CatBoostClassifier(verbose=False, depth=7, 12 leaf reg=1,
iterations=250, learning_rate=0.1)),
    ('ADASYN_LGBM', lgb.LGBMClassifier()),
    ('BorderlinesMOTE_XGB', XGBClassifier(learning_rate
=0.1,gamma=0.25,use label encoder=False, n jobs=-1)),
    ('BorderlinesMOTE_CatBoost', CatBoostClassifier(verbose=False, depth=7,
12_leaf_reg=1, iterations=250, learning_rate=0.1)),
    ('BorderlinesMOTE_LGBM', lgb.LGBMClassifier()),
    ('SMOTEENN_XGB', XGBClassifier(learning_rate
=0.1,gamma=0.25,use label encoder=False, n jobs=-1)),
    ('SMOTEENN_CatBoost', CatBoostClassifier(verbose=False, depth=7, 12 leaf reg=1,
iterations=250, learning_rate=0.1)),
    ('SMOTEENN_LGBM', lgb.LGBMClassifier())
]
```

```
# Entrena y predice con los modelos
modelos_11 = fit_models_sampling(X_train, y_train, models)
final = predict_models(X_test, modelos_11)
probs = predict_models(pd.DataFrame(X_test), modelos_11, predict_proba=True, label=1)
probs_0 = predict_models(pd.DataFrame(X_test), modelos_11, predict_proba=True,
label=0)
```

```
dfs = []
```

```
for model name, model in modelos_11:
    # Realizar predicciones sobre el conjunto de datos de prueba
    predictions = model.predict(X_test)

    # Calcular métricas de evaluación
    gini = 2 * roc_auc_score(y_test, predictions) - 1
    roc_auc = roc_auc_score(y_test, predictions)
    precision = precision_score(y_test, predictions)
    f1 = f1_score(y_test, predictions)
    recall = recall_score(y_test, predictions)
    cm = confusion_matrix(y_test, predictions)
    true_positive = cm[1][1]
    false_positive = cm[0][1]
    true_negative = cm[0][0]
    false_negative = cm[1][0]
```

```
# Crear un DataFrame con las métricas de evaluación
this df = pd.DataFrame(data={'MODELO': model name,
                             'GINI': gini,
                             'ROC AUC': roc_auc,
                             'PRECISION': precision,
                             'F1_SCORE': f1,
                             'RECALL': recall,
                             'TRUE POSITIVE': true positive,
                             'FALSE POSITIVE': false positive,
                             'TRUE NEGATIVE': true negative,
                             'FALSE NEGATIVE': false negative
                             }, index=[0])

dfs.append(this df)

resultados = pd.concat(dfs, ignore index=True)
resultados
```

MODELO	GINI	ROC_AUC	PRECISION	F1_SCORE	RECALL	TRUE_POSITIVE	FALSE_POSITIVE	TRUE_NEGATIVE	FALSE_NEGATIVE
SMOTE_XGB	0.684538	0.842269	0.939891	0.792627	0.685259	172	11	15242	79
SMOTE_CatBoost	0.716607	0.858303	0.957447	0.820046	0.717131	180	8	15245	71
SMOTE_LGBM	0.700605	0.850303	0.951351	0.807339	0.701195	176	9	15244	75
ADASYN_XGB	0.688719	0.844359	0.955801	0.800926	0.689243	173	8	15245	78
ADASYN_CatBoost	0.692834	0.846417	0.966667	0.807425	0.693227	174	6	15247	77
ADASYN_LGBM	0.708377	0.854188	0.936842	0.807256	0.709163	178	12	15241	73
BorderlineSMOTE_XGB	0.700671	0.850335	0.956522	0.809195	0.701195	176	8	15245	75
BorderlineSMOTE_CatBoost	0.696687	0.848343	0.956284	0.806452	0.697211	175	8	15245	76
BorderlineSMOTE_LGBM	0.716476	0.858238	0.947368	0.816327	0.717131	180	10	15243	71
SMOTEENN_XGB	0.715624	0.857812	0.886700	0.792952	0.717131	180	23	15230	71
SMOTEENN_CatBoost	0.736003	0.868001	0.920398	0.818584	0.737052	185	16	15237	66
SMOTEENN_LGBM	0.747562	0.873781	0.895238	0.815618	0.749004	188	22	15231	63

```
resultados['TIPO_BALANCEADOR'] = resultados['MODELO'].apply(lambda x:
x.split(' ')[0])
resultados['MODELO1'] = resultados['MODELO'].apply(lambda x: x.split('_')[1])
```

```
# Pivotar el DataFrame
pivot_resultados = resultados.pivot_table(index='MODELO1',
columns='TIPO_BALANCEADOR', values='F1_SCORE', aggfunc='first')

# Reordenar las columnas según el orden específico
column_order = ['ADASYN', 'BorderlineSMOTE', 'SMOTEENN', 'SMOTE']
pivot_resultados = pivot_resultados[column_order]

# Reordenar las filas según el orden específico
row_order = ['CatBoost', 'LGBM', 'XGB']
pivot_resultados = pivot_resultados.loc[row_order]

# Imprimir el DataFrame resultante
pivot_resultados
```

TIPO_BALANCEADOR	ADASYN	BorderlineSMOTE	SMOTEENN	SMOTE
MODELO1				
CatBoost	0.807425	0.806452	0.818584	0.820046
LGBM	0.807256	0.816327	0.815618	0.807339
XGB	0.800926	0.809195	0.792952	0.792627

BUSQUEDA DE HIPERPARAMETROS

```
models1 = [
    ('XGB', XGBClassifier())
]

param_grids = [
    {'learning_rate': [0.1, 0.01, 0.001], 'max_depth': [3, 5, 7, 9], 'gamma': [0.1,
0.25, 0.5], 'scale_pos_weight': [1, 5, 7.85, 10]
    , 'subsample': [0.8, 0.9, 1.0], 'colsample_bytree': [0.8, 0.9,
1.0]}
    # XGB
]
```

```
results = eval_ensemble(X_train, y_train, X_test, y_test, models1, param_grids)
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_columns', None)
results
```

Model	Best Params	Recall	F1-Score
XGB	{'colsample_bytree': 1.0, 'gamma': 0.5, 'learning_rate': 0.1, 'max_depth': 9, 'scale_pos_weight': 7.85, 'subsample': 0.9}	0.756972	0.842572

```
models1 = [
    ('LGBM', lgb.LGBMClassifier())
]

param_grids = [
    {'num_leaves': [31, 63, 127], 'min_child_samples': [10, 20, 30], 'subsample':
    [0.8, 0.9, 1.0], 'colsample_bytree': [0.8, 0.9, 1.0]} # LGBM
]

results = eval_ensemble(X_train, y_train, X_test, y_test, models1, param_grids)
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_columns', None)
results
```

Model	Best Params	Recall	F1-Score
LGBM	{'colsample_bytree': 0.9, 'min_child_samples': 10, 'num_leaves': 127, 'subsample': 0.8}	0.729084	0.833713

```
models1 = [
    ('CatBoost', CatBoostClassifier())
]

param_grids = [
    {'border_count': [15, 25, 35], 'depth': [5, 7, 9], 'l2_leaf_reg': [1, 3],
    'iterations': [350, 400, 415], 'learning_rate': [0.15, 0.9]} # CatBoost
]
```

```
results = eval_ensemble(X_train, y_train, X_test, y_test, models1, param_grids)
```

```
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_columns', None)
results
```

Model	Best Params	Recall	F1-Score
CatBoost	{'border_count': 35, 'depth': 5, 'iterations': 415, 'l2_leaf_reg': 3, 'learning_rate': 0.15}	0.709163	0.809091

ENSEMBLE

Sin hiperparámetros

```
ensemble_model = VotingClassifier(estimators=[
    ('XGB', XGBClassifier(learning_rate=0.1, gamma=0.25, use_label_encoder=False,
n_jobs=-1))
    , ('Catboost', CatBoostClassifier())
    , ('LGBM', lgb.LGBMClassifier())
], voting='hard')
```

```
# Entrena el modelo de votación en tus datos de entrenamiento (asegúrate de tener
X_train y y_train)
ensemble_model.fit(X_train, y_train)
```

```
# Realiza predicciones con el modelo de votación
y_pred_ensemble = ensemble_model.predict(X_test)
```

```
# Calcula la precisión del modelo de votación
accuracy_ensemble = accuracy_score(y_test, y_pred_ensemble)
```

```
# Otras métricas
precision_ensemble = precision_score(y_test, y_pred_ensemble)
recall_ensemble = recall_score(y_test, y_pred_ensemble)
f1_ensemble = f1_score(y_test, y_pred_ensemble)
```

```
# Matriz de confusión
conf_matrix_ensemble = confusion_matrix(y_test, y_pred_ensemble)
```

```
# Reporte de clasificación
classification_report_ensemble = classification_report(y_test, y_pred_ensemble)

# Muestra las métricas
print(f"Ensemble Model Accuracy: {accuracy_ensemble}")
print(f"Ensemble Model Precision: {precision_ensemble}")
print(f"Ensemble Model Recall: {recall_ensemble}")
print(f"Ensemble Model F1-Score: {f1_ensemble}")
print(conf_matrix_ensemble)
```

Confusion Matrix:

```
[[15245   8]
 [   71 180]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	15253
1	0.96	0.72	0.82	251
accuracy			0.99	15504
macro avg	0.98	0.86	0.91	15504
weighted avg	0.99	0.99	0.99	15504

Con hiperparámetros

```
ensemble_model = VotingClassifier(estimators=[
    ('XGB', XGBClassifier(colsample_bytree= 1.0, gamma= 0.5, learning_rate= 0.1,
max_depth= 9, scale_pos_weight= 7.85, subsample= 0.9 )),
    ('Catboost', CatBoostClassifier( border count= 35, depth= 5, iterations= 415,
12_leaf_reg= 3, learning_rate= 0.16 ))
# , ('Catboost', CatBoostClassifier( border count= 35, depth= 5, iterations= 415,
12_leaf_reg= 3, learning_rate= 0.16 ))
    ('LGBM', lgb.LGBMClassifier(colsample_bytree= 0.9, min_child_samples= 10,
num_leaves= 127, subsample= 0.8 ))
], voting='soft')
```

```
# Entrena el modelo de votación en tus datos de entrenamiento (asegúrate de tener
X_train y y_train)
ensemble_model.fit(X_train, y_train)
```

```
# Realiza predicciones con el modelo de votación
y_pred_ensemble = ensemble_model.predict(X_test)
label = 1
c = ensemble_model.predict_proba(X_test)[:, label]
```

```
# Calcula la precisión del modelo de votación
accuracy_ensemble = accuracy_score(y_test, y_pred_ensemble)
```

```
# Otras métricas
precision_ensemble = precision_score(y_test, y_pred_ensemble)
recall_ensemble = recall_score(y_test, y_pred_ensemble)
f1_ensemble = f1_score(y_test, y_pred_ensemble)
```

```
# Matriz de confusión
conf_matrix_ensemble = confusion_matrix(y_test, y_pred_ensemble)
```

```
# Reporte de clasificación
classification_report_ensemble = classification_report(y_test, y_pred_ensemble)
```

```
# Muestra las métricas
print(f"Ensemble Model Accuracy: {accuracy_ensemble}")
print(f"Ensemble Model Precision: {precision_ensemble}")
print(f"Ensemble Model Recall: {recall_ensemble}")
print(f"Ensemble Model F1-Score: {f1_ensemble}")
print(conf_matrix_ensemble)
```

Confusion Matrix:

```
[[15243   10]
 [   64 187]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	15253
1	0.95	0.75	0.83	251
accuracy			1.00	15504
macro avg	0.97	0.87	0.92	15504
weighted avg	1.00	1.00	0.99	15504

CREACIÓN DE PKL

```
# Crear pkl
with open('ensamble_dfr.pkl', 'wb') as file:
    pickle.dump(ensemble_model, file)

# abrir pkl
with open('ensamble_dfr.pkl', 'rb') as f:
    modelo_propuesto = pickle.load(f)

models = [
    ('AdaBoost', AdaBoostClassifier() )
]

#DEFINES EL SCALAMIENTO DE LOS DATOS
modelo_1 = fit models(X train, y train, models)
final,predictions = predict models(X test, modelo_1)
probs = predict models(pd.DataFrame(X test), modelo_1, predict_proba=True, label=1 )
probs_0 = predict_models(pd.DataFrame(X_test), modelo_1, predict_proba=True, label=0)

dfs=[]

for i in final.columns:
    this_df = pd.DataFrame(data={'MODELO':i
                                , 'GINI': 2 * roc_auc_score(y_test, final[i]) - 1
                                , 'ROC AUC':roc_auc_score(y_test, final[i])
                                , 'PRECISION':precision_score(y_test, final[i])
                                , 'F1_SCORE':f1_score(y_test, final[i])
                                , 'RECALL':recall_score(y_test, final[i])
                                }, index=[i])

    dfs.append(this_df)

resultados = pd.concat(dfs, ignore_index=True)
resultados

# Crear pkl
with open('modeloDT.pkl', 'wb') as file:
    pickle.dump(modelo_1[0][1], file)

# abrir pkl
with open('modeloDT.pkl', 'rb') as f:
    modelo_comparativo = pickle.load(f)

modelo_comparativo.predict_proba(X_test)
modelo_propuesto.predict_proba(X_test)

# Prueba t para comparar ambas "precisiones"
from scipy.stats import ttest_ind
t_stat, p_val = ttest_ind(modelo_propuesto.predict_proba(X_test),
                           modelo_comparativo.predict_proba(X_test))

print(f"Prueba t de Student: t-statistic = {t_stat}, p-value = {p_val}")

Prueba t de Student: t-statistic = [ 535.797125 -535.79712501], p-value = [0. 0.]

df = pd.DataFrame({
    'actual': y_test,
    'ia': modelo_propuesto.predict_proba(X_test)[: ,0],
    'basic': modelo_comparativo.predict_proba(X_test)[: ,0]
})

# Crear umbral para binarizar las predicciones
binarizer = Binarizer(threshold=0.5)

# Binarizar las predicciones
df = pd.DataFrame({
    'actual': y_test,
    'ia': binarizer.fit_transform(modelo_propuesto.predict_proba(X_test)[: ,
0].reshape(-1, 1)).flatten(),
    'basic': binarizer.fit_transform(modelo_comparativo.predict_proba(X_test)[: ,
0].reshape(-1, 1)).flatten()
})
```

```

# Prueba t de Student
t_stat, p_val = ttest_ind(modelo_propuesto.predict_proba(X_test)[: , 0],
                           modelo_comparativo.predict_proba(X_test)[: , 0])
print(f"Prueba t de Student: t-statistic = {t_stat}, p-value = {p_val}")

# Crear tabla de contingencia
confusion_matrix = pd.crosstab(df['ia'], df['basic'],
                               rownames=['IA'], colnames=['Basic'])

# Test de McNemar
result = mcnemar(confusion_matrix, exact=True)
print(f"Test de McNemar: estadístico = {result.statistic}, p-valor = {result.pvalue}")

# Mostrar matriz de confusión (usando pandas)
print("Matriz de Confusión (IA vs Basic):")
print(confusion_matrix)

```

```

Prueba t de Student: t-statistic = 535.7971249963781, p-value = 0.0
Test de McNemar: estadístico = 12.0, p-valor = 3.369757074208766e-15
Matriz de Confusión (IA vs Basic):
Basic  0.0    1.0
IA
0.0    110     87
1.0     12 15295

```

VALIDACIÓN CRUZADA

```

# Definir los modelos base
xgb_model = XGBClassifier()
catboost_model = CatBoostClassifier(verbose=False)
lgbm_model = LGBMClassifier()
# Crear el ensemble
ensemble = [('XGB', xgb_model), ('CatBoost', catboost_model), ('LGBM', lgbm_model)]

# Definir el espacio de búsqueda de hiperparámetros
param_dist = {
    'XGB_n_estimators': randint(50, 500),
    'XGB_max_depth': randint(3, 10),
    'XGB_learning_rate': uniform(0.001, 0.9),
    'XGB_colsample_bytree': uniform(0.8, 0.2),
    'XGB_gamma': uniform(0.1, 0.9),
    'XGB_scale_pos_weight': randint(1, 10),
    'XGB_subsample': uniform(0.8, 0.2),

    'CatBoost_iterations': randint(100, 800),
    'CatBoost_learning_rate': uniform(0.01, 1),
    'CatBoost_depth': randint(1, 10),
    'CatBoost_l2_leaf_reg': randint(1, 9),
    'CatBoost_border_count': randint(32, 255), # Valores razonables para CatBoost

    'LGBM_num_leaves': randint(31, 127),
    'LGBM_min_child_samples': randint(10, 30),
    'LGBM_subsample': uniform(0.8, 0.2),
    'LGBM_colsample_bytree': uniform(0.8, 0.2)
}

# Definir la métrica de evaluación (puedes ajustar según tu problema)
scorer = make_scorer(f1_score, greater_is_better=True)

# Crear el modelo RandomizedSearchCV para la búsqueda de hiperparámetros
search = RandomizedSearchCV(
    VotingClassifier(estimators=ensemble, voting='soft', n_jobs=-1),
    param_distributions=param_dist,
    scoring=scorer,
    n_iter=100, # Número de combinaciones a probar
    cv=10, # Número de particiones en la validación cruzada
    random_state=42,
    n_jobs=-1 # Usar todos los núcleos disponibles
)

# Ajustar el modelo a tus datos
search.fit(X_train, y_train)

```

```

# Obtener los mejores hiperparámetros
best params = search.best params

# Obtener el modelo con los mejores hiperparámetros
best model = search.best estimator

# Hacer predicciones en el conjunto de prueba
y pred = best model.predict(X test)

# Calcular el F1-score
f1 = f1 score(y test, y pred)

# Crear un DataFrame con los resultados
result df = pd.DataFrame([
    "Modelo": "VotingClassifier",
    "Mejores Hiperparámetros": best params,
    "Mejor F1 Score": f1
])

print(result df)

```

Modelo	Mejores Hiperparámetros	Mejor F1 Score
VotingClassifier	("CatBoost__border_count": 204, "CatBoost__depth": 5, "CatBoost__iterations": 425, "CatBoost__l2_leaf_reg": 2, "CatBoost__learning_rate": 0.3336792364042437, "LGBM__colsample_bytree": 0.8850872877232834, "LGBM__min_child_samples": 10, "LGBM__num_leaves": 50, "LGBM__subsample": 0.8484819464830161, "XGB__colsample_bytree": 0.8229673649478407, "XGB__gamma": 0.6495580381974694, "XGB__learning_rate": 0.2507674979162302, "XGB__max_depth": 7, "XGB__n_estimators": 431, "XGB__scale_pos_weight": 4, "XGB__subsample": 0.8641284384319029)	0.839729