Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Data Analytics in Industrial and Business Environments

Machine Learning para predecir cancelaciones y mejorar la retención en seguros

---

Autor: Luis Carlos Ocaña Hoeber

Tutor: Jorge Segura Gisbert

Profesora: Susana Acedo Nadal

Barcelona, 6 de junio de 2025

Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	Machine Learning para predecir cancelaciones y mejorar la retención en seguros
Nombre del autor:	Luis Carlos Ocaña Hoeber
Nombre del colaborador docente:	Jorge Segura Gisbert
Nombre del PRA:	Susana Acedo Nadal
Fecha de entrega:	06/2025
Titulación:	Máster Universitario en Ciencia de Datos
Área del Trabajo Final:	Data Analytics in Industrial and Business Environments
Idioma del trabajo:	Español
Palabras clave	Machine Learning, Renovación de pólizas, Modelado predictivo, Predicción de abandono de clientes

Resumen

Este trabajo de fin de máster se enfoca en el desarrollo y análisis de modelos de machine learning para predecir la renovación de pólizas de seguros en función de diversas variables explicativas. El objetivo principal es evaluar cómo diferentes algoritmos de machine learning pueden predecir la renovación o no de una póliza de seguro, utilizando características como el monto anual de la póliza, antigüedad del cliente o nivel de ingresos, entre otras variables explicativas. Para ello, se realizará un análisis exploratorio de datos, con el fin de comprender las relaciones entre las variables, identificar patrones y limpiar los datos. Posteriormente, se entrenarán y evaluarán diferentes modelos de machine learning para determinar cuál ofrece el mejor rendimiento predictivo. Este trabajo busca proporcionar una visión práctica sobre cómo el machine learning puede ser aplicado en la industria de seguros para la toma de decisiones sobre la renovación de pólizas.

Palabras clave: Machine Learning, Renovación de pólizas, Modelado predictivo, Predicción de abandono de clientes

Abstract

This master's thesis focuses on the development and analysis of machine learning models to predict the renewal of insurance policies based on various explanatory variables. The main objective is to evaluate how different machine learning algorithms can predict whether or not an insurance policy will be renewed, using features such as the annual policy amount, customer loyalty duration, income level, and other explanatory variables. To achieve this, an exploratory data analysis will be conducted to understand the relationships between variables, identify patterns, and clean the data. Subsequently, various machine learning models will be trained and evaluated to determine which one provides the best predictive performance. This thesis aims to provide practical insights into how machine learning can be applied in the insurance industry for decision-making regarding policy renewals.

Keywords: Machine Learning, Policy Renewal, Predictive Modeling, Customer Churn Predictions.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1. Contexto y motivación

Con el aumento de la digitalización y el uso de datos a gran escala, las aseguradoras buscan nuevas formas de comprender y predecir el comportamiento del cliente con mayor precisión. Uno de los problemas más importantes en este ámbito es la cancelación de contratos, que puede generar pérdidas económicas significativas para las aseguradoras y reducir su cuota de mercado. La capacidad de anticipar estas situaciones permitiría optimizar la gestión del riesgo y mejorar la retención de clientes.

Considero que las técnicas de Machine Learning pueden desempeñar un papel clave al identificar patrones ocultos en los datos y generar modelos predictivos precisos. La automatización de este proceso no solo mejoraría la eficiencia operativa y reduciría costes para las aseguradoras, sino que también ofrecería oportunidades para desarrollar estrategias proactivas centradas en el cliente. Por ejemplo, al anticipar el riesgo de cancelación, se podrían aplicar acciones dirigidas como ajustes en la oferta, comunicación personalizada o mejoras en el servicio, contribuyendo así a una mayor fidelización.

Mi interés en este proyecto surge de la intersección entre economía y Machine Learning. Aplicar estas técnicas al análisis de cancelaciones de pólizas no solo permite mejorar la precisión de las predicciones, sino que también ayuda a comprender mejor el comportamiento de los clientes y a desarrollar estrategias más efectivas para su retención.

El objetivo de este trabajo es desarrollar y comparar distintos modelos de Machine Learning para predecir la anulación de contratos en seguros. Se espera obtener un modelo eficiente que permita anticipar qué clientes tienen mayor probabilidad de cancelar su póliza y, a partir de ello, diseñar estrategias de retención más efectivas, mejorando tanto la sostenibilidad financiera de las aseguradoras como la satisfacción de sus clientes.

2. Sostenibilidad, diversidad y desafíos ético/sociales

El desarrollo de modelos de machine learning para predecir la renovación de pólizas de seguros tiene un impacto significativo tanto desde el punto de vista ético como social [1]. A continuación, se presentan algunas reflexiones sobre los posibles sesgos y el impacto social que este proyecto podría generar.

- **Sesgos en los datos**

  1. **Sesgos socioeconómicos**: Las variables como el nivel de ingresos (`income`), el valor de mercado de la vivienda (`home_market_value`), y la posesión de un título universitario (`college_degree`) podrían introducir sesgos socioeconómicos. Por ejemplo, si los datos muestran que las personas con menores ingresos tienen una mayor probabilidad de cancelar sus pólizas, esto podría llevar a decisiones sesgadas que afecten negativamente a este grupo demográfico.
  2. **Sesgos demográficos**: Variables como la edad (`age_in_years`), el estado civil (`marital_status`), y la presencia de hijos (`has_children`) podrían introducir sesgos demográficos. Por ejemplo, si el modelo predice que las personas mayores tienen una mayor probabilidad de cancelar sus pólizas, esto podría llevar a prácticas discriminatorias en la fijación de precios o en la oferta de servicios.
  3. **Sesgos geográficos**: Las variables de ubicación (`city`, `state`, `county`) podrían introducir sesgos geográficos. Por ejemplo, si el modelo predice que las personas que viven en ciertas áreas tienen una mayor probabilidad de cancelar sus pólizas, esto podría llevar a una discriminación basada en la ubicación.

- **Impacto social**

  1. **Desigualdad en el acceso a seguros**: Si los modelos predictivos no se gestionan adecuadamente, podrían perpetuar o incluso exacerbar las desigualdades existentes en el acceso a los seguros. Por ejemplo, si las personas con menores ingresos o de ciertas áreas geográficas son sistemáticamente identificadas como de alto riesgo, podrían enfrentar primas más altas o ser excluidas de ciertos servicios.
  2. **Transparencia y justicia**: Es crucial que los modelos de machine learning sean transparentes y que las decisiones basadas en ellos sean justas y explicables. Los clientes tienen derecho a saber cómo se toman las decisiones que les afectan y a tener la oportunidad de apelar o corregir decisiones que consideren injustas.
  3. **Responsabilidad social corporativa**: Las aseguradoras tienen una responsabilidad social de garantizar que sus prácticas no contribuyan a la exclusión social o

económica. Esto incluye la implementación de políticas que mitiguen los sesgos en los modelos predictivos y la promoción de prácticas justas y equitativas.

3. Objetivos

Objetivo principal

Desarrollar modelos de Machine Learning para predecir la cancelación de pólizas de seguros, optimizando su rendimiento y proporcionando interpretaciones claras sobre los factores que influyen en dichas predicciones.

Objetivos secundarios

- **Desarrollo de scripts propios para el análisis de datos**: Crear y personalizar scripts en Python para limpiar, procesar y analizar los datos relacionados con las pólizas de seguros y sus características. Implementar un flujo de trabajo automatizado para la limpieza y preparación de los datos, incluyendo la gestión de valores nulos, duplicados y outliers.

- **Análisis exploratorio de datos (EDA) y detección de patrones**: Realizar un análisis exploratorio de los datos para identificar distribuciones, correlaciones y patrones subyacentes en las variables que podrían influir en la cancelación de pólizas. Crear visualizaciones (gráficos de distribución, diagramas de caja, matrices de correlación) para explorar las relaciones entre variables clave.

- **Desarrollo de modelos de Machine Learning para la predicción de cancelaciones de pólizas**: Implementar y entrenar varios modelos de Machine Learning (como regresión logística, SVM, Random Forest, entre otros) para predecir la cancelación de pólizas. Realizar una validación cruzada para evaluar el desempeño de los modelos en datos no vistos.

- **Evaluación y comparación de modelos**: Comparar los modelos de Machine Learning utilizando métricas de rendimiento como precisión, recall, AUC, F1-score, etc. Realizar un análisis comparativo entre los modelos base y los modelos mejorados mediante técnicas de ensamblaje como stacking o boosting.

- **Optimización de los modelos a través del ajuste de hiperparámetros**: Mejorar el rendimiento de los modelos mediante técnicas de optimización de hiperparámetros, utilizando herramientas como GridSearchCV. Evaluar el impacto de los ajustes de hiperparámetros en las métricas de rendimiento.

- **Interpretación de los resultados y explicación de las predicciones**: Utilizar herramientas como SHAP (SHapley Additive exPlanations) para interpretar las predicciones de los modelos y comprender los factores clave que influyen en la cancelación de pólizas. Identificar los elementos más importantes que contribuyen a las decisiones de los modelos y elaborar recomendaciones para la toma de decisiones en la industria de seguros.

- **Visualización y presentación de resultados**: Crear visualizaciones claras y comprensibles para ilustrar el desempeño de los modelos, como matrices de confusión, curvas ROC y otras representaciones gráficas. Preparar un informe detallado y una presentación que resuman los objetivos, la metodología, los resultados obtenidos y las conclusiones del análisis.

Objetivos intermedios

- **Evaluación continua del rendimiento de los modelos**: Durante el proceso de desarrollo, se realizará una evaluación continua del rendimiento de los modelos, permitiendo ajustar la estrategia de modelado conforme se vayan obteniendo resultados.

- **Análisis detallado de los errores**: Se llevará a cabo un análisis detallado de los errores cometidos por los modelos para identificar áreas de mejora y ajustar las técnicas de preprocesamiento o las características utilizadas.

- **Implementación de iteraciones de mejora de los modelos**: Se realizarán iteraciones de mejora de los modelos según las métricas de rendimiento obtenidas, con el fin de optimizar el desempeño antes de la evaluación final.

4. Metodología

Para llevar a cabo este estudio, se utilizará la metodología CRISP-DM (Cross Industry Standard Process for Data Mining) [2], un enfoque ampliamente utilizado en proyectos de ciencia de datos que permite estructurar y organizar el proceso de análisis y modelado de datos en seis fases principales. A continuación, se describen en detalle cada una de estas fases y su aplicación en este trabajo:

1. **Comprensión del negocio**

En esta primera fase, se busca comprender el contexto del problema y los objetivos del proyecto desde una perspectiva de negocio. En este caso, el objetivo es predecir si una póliza de seguro será renovada o no, basándose en diversas variables explicativas. Para ello, se analizarán aspectos clave como:

- Impacto financiero y estratégico de la retención de clientes.
- Definición clara del problema y los criterios de éxito del modelo.

2. **Comprensión de los datos (estudio y exploración de los datos)**

Una vez definido el problema de negocio, se procede a analizar los datos disponibles. Esto incluye la exploración y limpieza inicial para comprender su estructura y calidad.

Se llevarán a cabo las siguientes actividades: Esta fase implica la exploración de los datos disponibles para conocer su calidad, estructura y posibles problemas. Se llevarán a cabo las siguientes actividades:

- Análisis exploratorio de datos (EDA): se examinarán estadísticas descriptivas, distribuciones de variables y relaciones entre ellas.
- Detección de valores atípicos y datos faltantes: identificación de inconsistencias que puedan afectar el rendimiento del modelo.
- Análisis de correlación: estudio de la relación entre variables para seleccionar aquellas más relevantes.

3. **Preparación de los datos**

Una vez comprendidos los datos, se procede a su preparación para que sean adecuados para el modelado. Esta fase incluye:

- Limpieza de datos: tratamiento de valores nulos, eliminación de duplicados y corrección de errores.
- Codificación de variables categóricas y normalización de variables numéricas.
- División del conjunto de datos en conjuntos de entrenamiento y prueba.

4. **Modelado**

En esta etapa se entrenan y prueban distintos algoritmos de Machine Learning para encontrar el modelo con mejor rendimiento. Se consideran las siguientes técnicas:

- Selección de modelos: se evaluarán algoritmos como regresión logística, árboles de decisión, random forest, gradient boosting y redes neuronales.
- Optimización de hiperparámetros: se aplicará validación cruzada y optimización de hiperparámetros para mejorar el desempeño del modelo.
- Los modelos serán comparados con métricas como precisión, recall, F1-score y AUC-ROC para determinar cuál ofrece mejores predicciones.

5. **Evaluación**

Una vez seleccionado el modelo más eficiente, se validará su desempeño en un conjunto de datos no visto para garantizar su fiabilidad,

- Interpretación de los resultados: análisis de la importancia de las variables para entender qué factores influyen en la cancelación de pólizas.
- Validación del modelo: evaluación del modelo con datos de prueba para verificar su estabilidad y capacidad de generalización.
- Comparación con modelos base: el modelo será comparado con enfoques más simples.

6. **Despliegue**

Si el modelo demuestra ser efectivo, se puede proceder a su implementación en un entorno de producción. Las actividades incluyen:

- Documentación y replicabilidad: almacenamiento del código en GitHub con instrucciones detalladas.
- Informe final: entrega de un informe con la metodología, resultados y conclusiones obtenidas, junto con recomendaciones para su aplicación en la empresa.

5. Planificación

Semana 1-2 (24 de febrero - 9 de marzo): Comprensión del negocio y estudio previo.

- Definir los objetivos del proyecto y el impacto en la industria de seguros.
  - Reunión inicial con el tutor para definir objetivos específicos.
  - Análisis de la problemática de cancelación de pólizas en seguros.
- Revisión de bibliografía y estudios previos sobre modelos de predicción de renovación de pólizas.
  - Revisión de artículos científicos y libros sobre técnicas de predicción.
  - Revisión de casos de uso en la industria de seguros.

Semana 3-4 (11 - 24 de marzo): Comprensión de los datos.

- Exploración inicial del conjunto de datos disponible.

  - Análisis inicial del tamaño y la estructura de los datos.
  - Identificación de tipos de variables y distribución de los datos.
- Identificación de valores nulos, duplicados y outliers.
  - Comprobación de valores nulos en las variables.
  - Identificación y eliminación de duplicados y outliers.
- Análisis de distribuciones de variables y relaciones entre ellas.
  - Creación de gráficos de distribución (histogramas, boxplots).
  - Análisis de correlación entre variables.

Semana 5-6 (25 de marzo - 7 de abril): Preparación de los datos.

- Manejo de valores nulos, duplicados y outliers.
  - Imputación de valores nulos utilizando técnicas adecuadas (si las hubiera).
  - Eliminar o transformar outliers si es necesario.
- Transformación de variables categóricas (One-Hot Encoding, Label Encoding, etc.).
  - Identificación de variables categóricas.
  - Implementación de técnicas de codificación de variables.
- Guardar dataset limpio y documentar los cambios realizados.
  - Crear un dataset limpio.
  - Documentar todos los cambios y decisiones tomadas en la limpieza de datos.
- Análisis de patrones de pago y cancelaciones.
  - Crear un resumen de las principales conclusiones del análisis.

Semana 7-8 (8 - 21 de abril): Modelado inicial.

- Entrenamiento de modelos base.
  - Selección de modelos base: regresión logística, SVM, KNN, etc.
  - Entrenar los modelos con el conjunto de datos preparado.
- Comparación de métricas iniciales para identificar enfoques más prometedores.

- Evaluación utilizando métricas como precisión, recall, AUC, etc.
- Análisis de los resultados para seleccionar los modelos más prometedores.

Semana 9-10 (22 de abril - 5 de mayo): Combinación y mejora de modelos.

- Utilizar Random Forest, AdaBoost y Gradient Boosting como ensambladores.
  - Entrenar modelos con Random Forest, AdaBoost y Gradient Boosting.
- Ajuste de hiperparámetros y optimización del desempeño.
  - Realizar búsqueda en cuadrícula o aleatoria de hiperparámetros.
  - Evaluar mejoras en el desempeño.
- Comparar desempeño entre modelos.
  - Análisis comparativo entre los modelos base y los modelos combinados.
  - Selección del modelo con mejor desempeño.

Semana 11 (6 - 12 de mayo): Evaluación y visualización de resultados.

- Análisis de métricas finales (precisión, recall, AUC, etc.).
  - Evaluación final de los modelos seleccionados.
  - Análisis de los resultados en profundidad.
- Creación de visualizaciones para explicar los resultados (matriz de confusión, curvas ROC, SHAP, etc.).
  - Crear gráficos visuales para ilustrar el desempeño de los modelos.
  - Generar explicaciones interpretables con SHAP.
- Comparación con modelos previos y justificación de la elección final.
  - Comparar los resultados con modelos anteriores.
  - Justificación de la selección del modelo final.

Semana 12 (13 - 17 de mayo): Documentación y presentación de resultados.

- Elaborar la memoria del trabajo.
  - Redacción de la introducción, metodología, resultados y conclusiones.

    - Incluir detalles sobre la implementación de los modelos.

- Revisión ortográfica y formateo del documento.
    - Revisión exhaustiva de errores gramaticales y de estilo.
    - Formateo final del documento.
- Preparación de la presentación final del trabajo.
    - Crear presentación.
    - Preparación de los puntos clave a presentar.

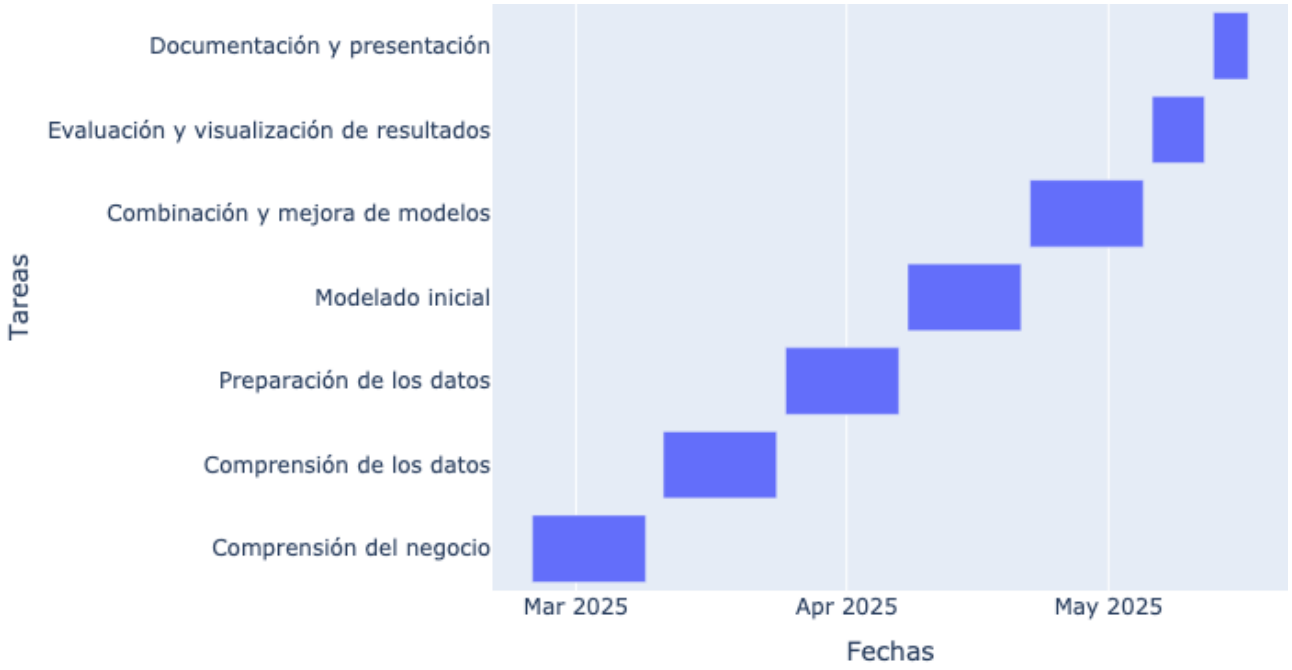


Figura 1.1: Cronología de las tareas del proyecto

Capítulo 2

Estado del arte

El aprendizaje automático ha transformado la industria de los seguros al permitir a las compañías optimizar procesos, reducir costos, mejorar la experiencia del cliente y gestionar riesgos con mayor precisión. Gracias al machine learning y al análisis de datos, las aseguradoras pueden tomar decisiones más informadas, anticipar tendencias y adaptarse a un mercado en constante cambio. Esto no solo mejora la eficiencia operativa, sino que también permite ofrecer productos más personalizados y justos para los asegurados.

Algunas de las aplicaciones del aprendizaje automático en la industria de los seguros son:

- **Detección y prevención del fraude**:

  Gracias al machine learning, las aseguradoras pueden identificar patrones sospechosos y predecir comportamientos fraudulentos con mayor precisión. En España, la tasa de fraude ha aumentado ligeramente al 1,97 %, cuatro décimas más que el año anterior [3]. Aunque el número de siniestros declarados se ha mantenido estable, los casos detectados han aumentado de 15.000 en 2012 a más de 23.000 en 2024, evitando pagos fraudulentos por un total de casi 87 millones de euros en 2024.

- **Optimización de precios**:

  En lugar de aplicar un aumento uniforme a las primas, las aseguradoras utilizan el machine learning para ajustar los precios de forma individualizada. Esto permite penalizar a aquellos clientes con mayor riesgo de siniestralidad, manteniendo precios competitivos y sin afectar a los asegurados de bajo riesgo.

- **Identificación de clientes en riesgo de churn**: Los modelos predictivos son capaces de detectar señales tempranas de fuga, es decir, identificar a los clientes en riesgo de abandonar la aseguradora. Esto brinda la oportunidad de tomar medidas preventivas que mejoren la retención y fidelización antes de que el cliente decida irse.

El presente trabajo se centra en aplicar estos modelos predictivos para intervenir de manera temprana y evitar la pérdida de clientes.

1. Customer churn y machine learning

El término churn [4] hace referencia a la pérdida de clientes en un determinado periodo de tiempo. En el sector asegurador, el customer churn ocurre cuando un asegurado decide cancelar o no renovar su póliza. Es un fenómeno crítico en industrias con modelos de ingresos recurrentes, como telecomunicaciones, banca, servicios de suscripción y seguros, ya que implica no solo la pérdida de ingresos, sino también costos adicionales asociados con la adquisición de nuevos clientes para reemplazar a los que se han marchado.

Existen dos tipos principales de churn:

- **Churn voluntario**: cuando el cliente decide abandonar la empresa por insatisfacción, mejores ofertas de la competencia o cambios en sus necesidades.
- **Churn involuntario**: cuando la empresa pierde un cliente por razones externas, como problemas de pago, cambios en las condiciones del servicio o factores ajenos a la experiencia del cliente.

Su gestión es de vital importancia para las aseguradoras, ya que permite reducir costos, mejorar la rentabilidad y fortalecer su posición en el mercado. Retener a un cliente existente resulta significativamente más económico que adquirir uno nuevo, lo que optimiza los recursos y minimiza el impacto financiero de la cancelación de pólizas. De hecho existen estudios que indican que, dependiendo de la industria, adquirir un nuevo cliente puede costar entre cinco y siete veces más que retener a uno antiguo [5].

El uso de machine learning (ML) en la predicción del customer churn ha cobrado una importancia creciente en los últimos años. Tradicionalmente, los modelos de regresión logística han sido el enfoque más común en estudios académicos y aplicaciones empresariales. Sin embargo, la creciente disponibilidad de datos y los avances en técnicas de aprendizaje automático han abierto nuevas posibilidades para mejorar la precisión de estas predicciones.

El sector de telecomunicaciones ha sido pionero en la aplicación de modelos analíticos para la retención de clientes desde hace más de una década [6]. No obstante, en la actualidad, otros sectores como el de seguros también están adoptando enfoques basados en ML. Ejemplo de ello son estudios como Customer Churn Prediction System: A Machine Learning Approach [7] o Machine-Learning Techniques for Customer Retention: A Comparative Study [8], que exploran el potencial de diferentes algoritmos para anticipar la fuga de clientes.

A medida que las empresas buscan optimizar sus estrategias de retención, el ML se está convirtiendo en una herramienta clave. Algoritmos como árboles de decisión, redes neuronales y modelos de boosting están demostrando ser más efectivos que los enfoques tradicionales. Esto permite no solo prever con mayor precisión qué clientes podrían abandonar, sino también diseñar estrategias personalizadas para retenerlos.

2. Modelos de machine learning

Regresión logística

La Regresión Logística [9] es un modelo estadístico utilizado para predecir la probabilidad de un evento binario, es decir, cuando el resultado solo puede pertenecer a una de dos categorías posibles. A diferencia de la regresión lineal, que predice valores continuos, la regresión logística utiliza una función logística (o sigmoide) para limitar la salida a un rango entre 0 y 1, interpretado como una probabilidad. Este modelo es ampliamente utilizado en clasificación binaria, donde se busca asignar una observación a una de dos clases basadas en características o variables independientes.

Matemáticamente, la regresión logística predice la probabilidad de que una observación pertenezca a una clase particular (por ejemplo, clase 1) dado un vector de características x. Esta probabilidad se modela mediante una función logística, que se expresa como:

$$E(Y|X = x) = \pi(x) = \frac{\exp(x^t\beta)}{1 + \exp(x^t\beta)} = \frac{1}{1 + \exp(-x^t\beta)} \tag{2.1}$$

Donde x es el vector de características, β es el vector de coeficientes estimados y exp representa la función exponencial. El valor de $\pi(x)$ es la probabilidad de que la observación pertenezca a la clase 1, y $1 - \pi(x)$ es la probabilidad de que pertenezca a la clase 0.

A pesar de su simplicidad, la regresión logística puede ser muy eficaz en escenarios donde las relaciones entre las variables y la clase objetivo son lineales. Sin embargo, su rendimiento puede verse afectado en problemas más complejos donde existen relaciones no lineales entre las variables.

Naïve Bayes

Naïve Bayes [10] es una familia de algoritmos de clasificación basados en el teorema de Bayes, utilizada para modelar la probabilidad de que una observación pertenezca a una determinada categoría. Su principio fundamental radica en la suposición de independencia condicional entre las características, lo que simplifica los cálculos y permite una clasificación eficiente incluso con

grandes volúmenes de datos.

Matemáticamente, el Teorema de Bayes se expresa como:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.2}$$

Donde $P(A|B)$ es la probabilidad de que ocurra el evento A dado que se ha observado el evento B, $P(B|A)$ es la probabilidad de observar B dado que A ha ocurrido, $P(A)$ es la probabilidad de A y $P(B)$ es la probabilidad de B.

El término "Naïve"(ingenuo) proviene de la suposición clave del algoritmo: asume que todas las características son independientes entre sí, lo cual rara vez ocurre en la práctica. Sin embargo, esta simplificación permite que los cálculos sean más rápidos y eficientes, lo que hace que Naïve Bayes siga siendo una opción popular en múltiples aplicaciones de clasificación.

SVM

Las Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés) [11] son un conjunto de algoritmos de clasificación supervisada utilizados para encontrar el hiperplano óptimo que separa las clases dentro de un espacio de datos. Su objetivo principal es maximizar el margen entre los puntos de datos de distintas categorías, lo que permite mejorar la capacidad de generalización del modelo. SVM es particularmente útil en problemas de clasificación binaria y funciona bien en espacios de alta dimensionalidad, incluso cuando los datos no son linealmente separables.

Matemáticamente, el objetivo de SVM es encontrar el hiperplano que separa las clases de manera óptima. Este hiperplano se define mediante la ecuación:

$$w^T x + b = 0 \tag{2.3}$$

Donde w es el vector normal al hiperplano, x es el vector de características y b es el sesgo. El modelo de SVM intenta encontrar los valores de w y b que maximizan el margen, es decir, la distancia entre el hiperplano y los puntos más cercanos de cada clase, conocidos como los vectores de soporte.

La fortaleza de SVM radica en su capacidad para encontrar una frontera de decisión óptima, incluso en casos donde otras técnicas fallan. Sin embargo, su principal desafío radica en la selección del kernel adecuado y en el ajuste de hiperparámetros como C y gamma, que afectan directamente su rendimiento.

KNN

El algoritmo K-Nearest Neighbors (KNN) [12] es un método de clasificación supervisada basado en la similitud entre observaciones. Su funcionamiento se basa en la idea de que una nueva instancia será clasificada según la categoría mayoritaria de sus K vecinos más cercanos en el espacio de características. La proximidad entre los datos se mide generalmente mediante métricas como la distancia euclidiana, manhattan o coseno, dependiendo de la naturaleza del problema.

Aunque KNN es intuitivo y fácil de interpretar, su principal desventaja es su alto costo computacional en problemas con grandes volúmenes de datos, ya que cada predicción requiere calcular distancias con todas las observaciones de entrenamiento.

Árbol de decisión

El árbol de decisión [13] es un algoritmo de clasificación supervisada basado en la segmentación recursiva de los datos. Su estructura jerárquica se compone de nodos de decisión, donde se realizan divisiones basadas en ciertas características, y hojas, que representan la clasificación final. La construcción del árbol se realiza seleccionando en cada nodo la característica que mejor separa las clases, utilizando métricas como la entropía (para el criterio de Ganancia de Información) o el índice de Gini.

Una de las principales ventajas de los árboles de decisión es su interpretabilidad, ya que permiten visualizar el proceso de toma de decisiones de manera intuitiva. Además, son eficientes en términos computacionales para conjuntos de datos de tamaño moderado. No obstante, en escenarios con alta variabilidad en los datos o muchas dimensiones, los árboles individuales pueden ser inestables.

Redes neuronales

Las redes neuronales [14] son un conjunto de algoritmos inspirados en el funcionamiento del cerebro humano, diseñados para reconocer patrones y resolver problemas complejos de clasificación y predicción. Una red neuronal está compuesta por múltiples neuronas organizadas en capas: la capa de entrada, las capas ocultas y la capa de salida. Cada neurona en una capa está conectada a las neuronas de la capa siguiente, y cada conexión tiene un peso que ajusta la intensidad de la señal transmitida. Las redes neuronales aprenden ajustando estos pesos durante el proceso de entrenamiento mediante un algoritmo de retropropagación y el uso de una función de activación que determina si una neurona debe activarse en función de la entrada recibida.

A pesar de su flexibilidad y capacidad de adaptación, las redes neuronales presentan varios desafíos. Uno de los principales problemas es el sobreajuste, especialmente cuando la red es muy profunda y el número de parámetros es elevado. Para mitigar este riesgo, se utilizan técnicas como regularización y dropout, que ayudan a generalizar mejor el modelo.

Random Forest Classifier

El Random Forest Classifier [15] es un algoritmo de aprendizaje supervisado basado en la construcción de múltiples árboles de decisión para mejorar la precisión y estabilidad del modelo. Funciona mediante un enfoque de bagging (Bootstrap Aggregating), donde se crean varios subconjuntos de datos de entrenamiento y se generan múltiples árboles de decisión independientes. La predicción final se obtiene a partir del voto mayoritario de los árboles, lo que reduce la posibilidad de sobreajuste y mejora la capacidad de generalización del modelo.

Una de las principales ventajas de Random Forest es su robustez y precisión, ya que al combinar múltiples árboles de decisión se reduce la varianza y se mejora la estabilidad de las predicciones. Sin embargo, su principal limitación radica en su mayor demanda computacional en comparación con modelos más simples, especialmente cuando se incrementa el número de árboles en el bosque.

AdaBoost

El Adaptive Boosting (AdaBoost) [16] es un algoritmo de aprendizaje supervisado basado en la combinación de múltiples clasificadores débiles para formar un modelo robusto y preciso. Su funcionamiento se basa en un enfoque iterativo donde se entrenan varios modelos simples, generalmente árboles de decisión de profundidad 1, y en cada iteración se ajustan los pesos de las observaciones para mejorar la clasificación de los casos más difíciles. De esta manera, AdaBoost pone más énfasis en los errores cometidos en iteraciones anteriores, fortaleciendo progresivamente el modelo final.

Una de las principales ventajas de AdaBoost es su capacidad de mejorar modelos débiles de manera iterativa, obteniendo predicciones más precisas sin necesidad de un aumento significativo en la complejidad computacional.

XGBoost

XGBoost *(Extreme Gradient Boosting)* [17] es un algoritmo de aprendizaje supervisado basado en árboles de decisión y en la técnica de gradient boosting, optimizado para eficiencia y rendimiento. Construye modelos de forma secuencial, donde cada árbol intenta corregir los

errores del conjunto anterior. Su fortaleza radica en su regularización, manejo automático de valores faltantes, y capacidad de paralelización.

El objetivo de XGBoost es minimizar una función de pérdida penalizada:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.4}$$

donde $l(yi, \hat{y}_i)$ es la función de pérdida y $\Omega(f_k)$ penaliza la complejidad del modelo:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2 \tag{2.5}$$

XGBoost es especialmente útil en tareas de clasificación binaria y por su capacidad para modelar relaciones no lineales. Su principal desafío está en el ajuste de hiperparámetros como la profundidad de los árboles, el learning rate y el número de iteraciones.

3. Métricas de rendimiento

En problemas de clasificación binaria, como el que se aborda en este trabajo, evaluar correctamente el rendimiento de los modelos es crucial para garantizar la calidad y utilidad práctica de las predicciones [18] [19].

Matriz de confusión

La matriz de confusión es una herramienta visual y numérica que permite observar el desempeño del modelo en cada una de las clases. Su estructura permite analizar los aciertos y errores de clasificación dividiendo las predicciones en verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). Esta matriz es especialmente útil para calcular métricas adicionales y entender en detalle el comportamiento del modelo.

Cuadro 2.1: Matriz de confusión

Clase		**Clase Real**	
		1	**0**
Clase Predicha	**1**	Verdadero Positivo (TP)	Falso Negativo (FN)
	0	Falso Positivo (FP)	Verdadero Negativo (TN)

Exactitud

La *accuracy* o exactitud es la proporción de predicciones correctas sobre el total de predicciones realizadas. Se define como:

$$\text{Accuarcy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

Aunque es una métrica intuitiva y útil cuando las clases están balanceadas, en contextos desbalanceados puede ser poco representativa. Por ejemplo, si el 90 % de los clientes renuevan la póliza, un modelo que siempre predice "renovación" tendrá un 90 % de accuracy, pero no identificará a los clientes que realmente no renovarán, que suelen ser el objetivo de interés.

Precisión

La *precision* o precisión mide la proporción de instancias clasificadas como positivas que realmente pertenecen a la clase positiva. Es decir, evalúa cuán confiables son las predicciones positivas del modelo:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.7}$$

Sensibilidad

El *recall* o sensibilidad mide la capacidad del modelo para identificar correctamente todas las instancias reales de la clase positiva:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.8}$$

En el contexto del churn, un alto recall implica que el modelo logra identificar correctamente a la mayoría de los clientes que no renovarán la póliza. Esta métrica es crítica.

F1-score

La *F1-score* es la media armónica entre precision y recall, y ofrece un equilibrio entre ambas métricas. Es especialmente útil cuando se busca un balance entre la capacidad de detección y la precisión de las predicciones:

$$\text{F1} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.9}$$

4. Desbalance de clases

El desequilibrio de clases se define formalmente como una situación en la que el número de instancias pertenecientes a una clase (la clase mayoritaria, que en el contexto de churn suele ser la de los clientes que no abandonan) es sustancialmente mayor que el número de instancias en la otra clase (la clase minoritaria, los clientes que sí hacen churn). Los conjuntos de datos utilizados para predecir la fuga de clientes son intrínsecamente propensos a este problema debido a la naturaleza misma del evento de churn.

Esta disparidad en la representación de las clases puede tener un impacto significativo en el rendimiento y la precisión de los modelos de predicción de churn [20]. Los modelos entrenados con conjuntos de datos desequilibrados tienden a desarrollar un sesgo hacia la clase mayoritaria (no churn), lo que resulta en una alta precisión general pero un rendimiento deficiente en la identificación de la clase minoritaria (clientes que hacen churn).

En el contexto de la predicción de churn, el objetivo principal es identificar correctamente a la clase minoritaria, es decir, a los clientes potenciales que abandonarán el servicio. Por lo tanto, un modelo que no logra identificar a los clientes que harán churn (lo que se conoce como recall 2.8) puede llevar a la pérdida de clientes valiosos y a la disminución de los ingresos.

Sobremuestreo

El sobremuestreo, también conocido como *oversampling* [**?**] es una estrategia consiste en aumentar artificialmente el número de instancias pertenecientes a la clase minoritaria, mediante la replicación de ejemplos existentes o la generación de nuevas instancias sintéticas (cuando se combina con otras técnicas más avanzadas).

El objetivo principal del sobremuestreo es proporcionar al modelo una mayor cantidad de ejemplos de la clase minoritaria para que pueda aprender sus patrones de manera efectiva, y así reducir el sesgo natural hacia la clase mayoritaria que se produce durante el entrenamiento de muchos algoritmos de clasificación.

La replicación directa de ejemplos puede conducir al sobreajuste del modelo, especialmente si el número de instancias originales de la clase minoritaria es muy bajo.

Submuestreo

El submuestreo, también conocido como *undersampling* [**?**], es una técnica complementaria al sobremuestreo que consiste en reducir el número de instancias de la clase mayoritaria para equilibrar la proporción entre clases. Esto se logra eliminando ejemplos de la clase más frecuente, con el objetivo de evitar que el modelo aprenda un sesgo excesivo hacia ella.

Esta estrategia es útil cuando el conjunto de datos es muy grande y contiene redundancia en la clase mayoritaria, ya que permite disminuir la complejidad del modelo y reducir los recursos computacionales necesarios para el entrenamiento.

Existe un riesgo alto de eliminar ejemplos valiosos o representativos de la clase mayoritaria, lo que puede reducir la capacidad del modelo para generalizar correctamente.

Generación de datos sintéticos

Una alternativa más avanzada al sobremuestreo tradicional es la generación de datos sintéticos, en la cual se crean nuevas instancias artificiales de la clase minoritaria en lugar de replicar ejemplos existentes.

- SMOTENC (Synthetic Minority Over-sampling Technique for Nominal and Continuous features) [21] es una extensión del algoritmo SMOTE, diseñada específicamente para manejar conjuntos de datos que contienen tanto características nominales (categóricas) como continuas. Esta técnica es particularmente valiosa en aplicaciones del mundo real, donde los datos suelen presentar ambos tipos de atributos.

  El algoritmo SMOTENC fue desarrollado para abordar una limitación crítica del SMOTE original, su incapacidad para manejar adecuadamente atributos nominales. En conjuntos

de datos del mundo real, donde las clases minoritarias están significativamente subrepresentadas, es común encontrar una mezcla de variables numéricas y categóricas.

SMOTENC genera nuevas instancias sintéticas siguiendo un enfoque híbrido que trata de manera diferente los atributos continuos y nominales.

1. Para cada instancia x_i de la clase minoritaria, se identifican sus K vecinos más cercanos utilizando una medida de distancia mixta que penaliza las discrepancias en las variables categóricas.
2. Se selecciona aleatoriamente uno de los vecinos $\hat{x}_i$ entre estos K vecinos.
3. Para las variables continuas, se genera un nuevo valor mediante interpolación lineal:

$$x_{\text{new}}^{(\text{cont})} = x_i^{(\text{cont})} + \delta \cdot (\hat{x}_i^{(\text{cont})} - x_i^{(\text{cont})}), \quad \delta \sim \mathcal{U}(0,1)$$

4. Para las variables categóricas, se asigna el valor que aparece con mayor frecuencia entre x_i y $\hat{x}_i$:

$$x_{\text{new}}^{(\text{cat})} = \text{moda}(x_i^{(\text{cat})}, \hat{x}_i^{(\text{cat})})$$

Este enfoque permite generar ejemplos sintéticos realistas en entornos mixtos (numérico y categórico), manteniendo la coherencia semántica de las variables categóricas y la continuidad del espacio de características.

5. Optimización de parámetros

GridSearchCV

GridSearchCV [22] es una técnica de optimización de hiperparámetros que busca la mejor combinación de parámetros para un modelo de aprendizaje automático mediante la evaluación exhaustiva de un conjunto predefinido de valores. Su funcionamiento se basa en la exploración de un espacio de hiperparámetros especificado por el usuario, evaluando cada combinación posible de estos valores utilizando una técnica de validación cruzada para estimar su desempeño en datos no vistos. Este método es especialmente útil cuando se desea probar una serie de combinaciones de parámetros para encontrar la configuración que maximice el rendimiento del modelo.

Optuna

Optuna [23] es una biblioteca de optimización automática de hiperparámetros que se utiliza para mejorar el rendimiento de modelos de redes neuronales y otros algoritmos de aprendizaje

automático. El principal objetivo de Optuna es facilitar la búsqueda de la mejor configuración de hiperparámetros de manera eficiente, utilizando técnicas avanzadas como la optimización bayesiana. A través de un proceso iterativo, Optuna ajusta los hiperparámetros del modelo de forma inteligente, explorando el espacio de búsqueda para encontrar las configuraciones que maximicen el rendimiento del modelo en una tarea específica.

A diferencia de las búsquedas de hiperparámetros tradicionales, como la búsqueda en cuadrícula o la búsqueda aleatoria, Optuna utiliza un enfoque basado en la optimización de tipo "Bayesian Optimization". Esta técnica evalúa la relación entre los parámetros y los resultados del modelo, lo que permite a la herramienta hacer ajustes más precisos y rápidos, incluso en espacios de búsqueda muy grandes. Esto hace que la optimización sea más eficiente, reduciendo la cantidad de evaluaciones necesarias para encontrar la configuración óptima.

6. Interpretabilidad de modelos

Además de alcanzar un buen rendimiento predictivo, en problemas reales como la renovación de pólizas de seguros es crucial entender por qué un modelo de machine learning clasifica a un cliente como propenso a no renovar. Los modelos de machine learning más precisos suelen ser complejos y difíciles de interpretar directamente. Esta interpretabilidad no solo permite validar que el modelo toma decisiones coherentes, sino que también proporciona a las aseguradoras información valiosa para diseñar estrategias proactivas de retención. Esta necesidad de comprender el comportamiento del modelo resulta especialmente importante en el contexto de customer churn, como se discute en los artículos [24] y [25], donde se destaca que una adecuada explicación de las predicciones no solo mejora la confianza en el modelo, sino que también permite aplicar acciones personalizadas y efectivas para reducir la pérdida de clientes.

SHAP

SHAP *(Shapley Additive Explanations)* [26] utiliza los valores de Shapley, originarios de la teoría de juegos, para asignar a cada característica una contribución cuantitativa en la predicción realizada por el modelo. Intuitivamente, cada variable se considera un 'jugador' en un juego cooperativo cuyo objetivo es predecir un resultado. La contribución de cada variable se calcula como el promedio de su aporte marginal en todas las combinaciones posibles de características (coaliciones).

SHAP ofrece explicaciones coherentes tanto a nivel local como global, y garantiza ciertas propiedades deseables como la equidad (características con mayor impacto tienen mayor peso) y la consistencia.

Capítulo 3

Materiales y métodos

1. Obtención de datos

Los datos utilizados en este trabajo se han obtenido a través de la plataforma Kaggle [27]. La base de datos seleccionada se titula "Auto Insurance Churn Analysis Dataset" y está compuesta por cuatro archivos de texto con formato CSV, los cuales contienen datos tabulares. En total, el conjunto de datos cuenta con 1.680.909 observaciones, que están conectadas entre los diferentes archivos mediante los identificadores Address Id e Individual Id, donde este último es único para cada individuo.

Es importante destacar que los datos proporcionados en esta base de datos son sintéticos, y toda la información de los clientes es ficticia.

Las cuatro fuentes de datos incluidas en el conjunto son las siguientes:

- **Address**: Contiene información geográfica detallada sobre cada uno de los clientes, proporcionando datos relevantes para el análisis de la localización y su posible influencia en el comportamiento de cancelación de pólizas.

- **Customer**: Incluye información personal de los clientes, como el tiempo que llevan en la compañía aseguradora, el gasto que realizan y su edad.

- **Demographic**: Contiene información demográfica y financiera de los clientes, incluyendo variables como ingresos estimados, estado civil, propiedad de vivienda, nivel educativo, historial crediticio y otros datos relevantes, como la presencia de hijos en el hogar y la antigüedad en su residencia actual.

- **Termination**: Recoge un listado de aquellos clientes que cancelaron su póliza, especificando la fecha en la que lo hicieron.

Si bien la base de datos completa, resultado de la unión de los cuatro archivos mencionados, también está disponible, la primera tarea realizada en este estudio ha sido la combinación de toda esta información en una única base de datos. Esta fusión permite garantizar la consistencia de los datos y comprobar que la información contenida en los archivos individuales coincide con la de la base de datos unificada.

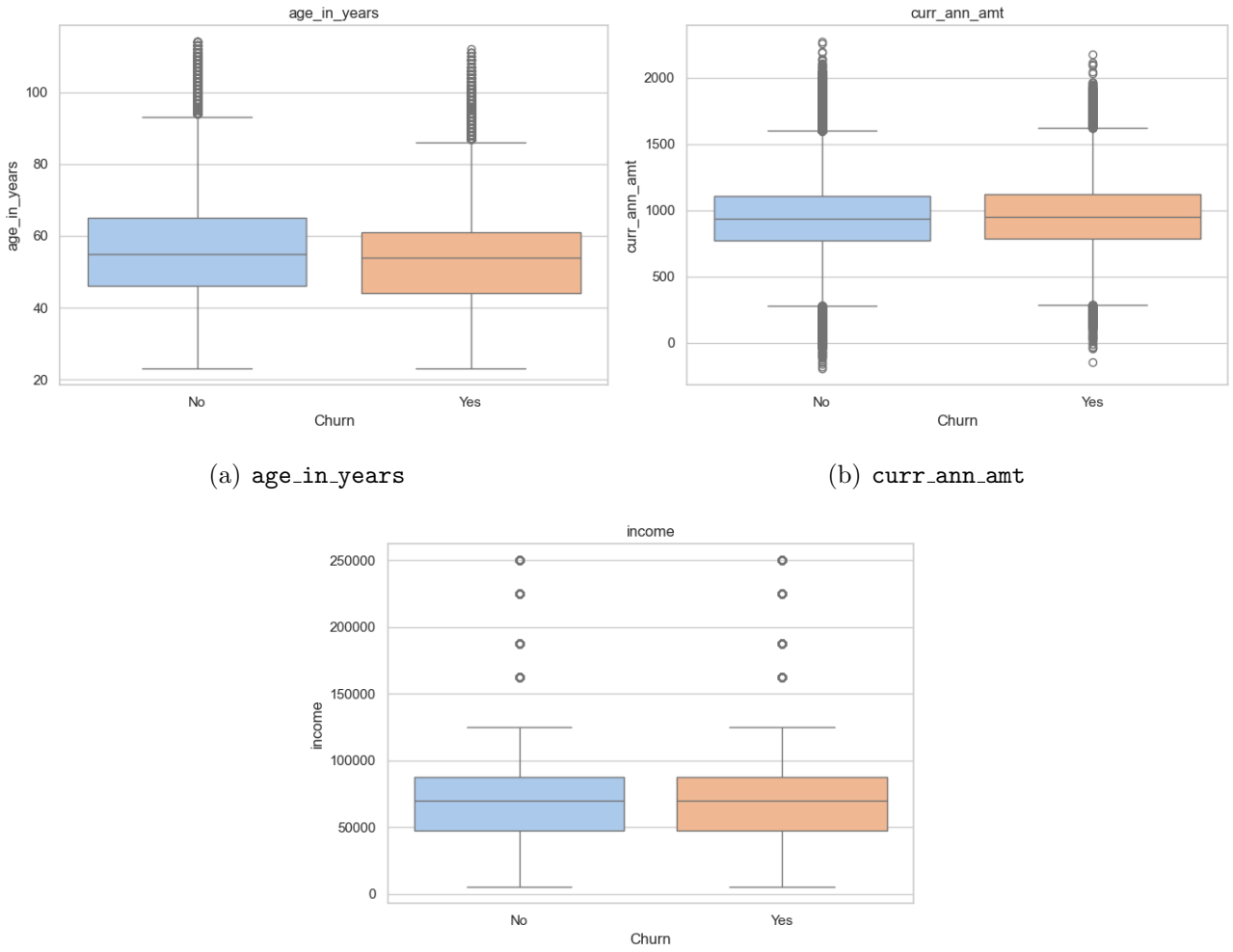
Para realizar esta fusión de manera eficiente, se ha empleado una operación de join utilizando los identificadores individuales y de dirección.

Variable	Descripción
ADDRESS_ID	ID único para una dirección específica
LATITUDE	Latitud de la dirección
LONGITUDE	Longitud de la dirección
STREET_ADDRESS	Dirección postal
CITY	Ciudad
STATE	Estado
COUNTY	Condado
INDIVIDUAL_ID	ID único para un cliente de seguros específico
CURR_ANN_AMT	Monto anual en dólares pagado por el cliente (no el monto de la póliza, sino el total pagado en el año anterior)
DAYS_TENURE	Número de días que el individuo ha sido cliente de la aseguradora
CUST_ORIG_DATE	Fecha en que el individuo se convirtió en cliente
AGE_IN_YEARS	Edad del individuo en años
DATE_OF_BIRTH	Fecha de nacimiento del individuo
SOCIAL_SECURITY_NUMBER	Número de Seguridad Social (los dos dígitos del medio son "XX" para protección de datos)
INCOME	Ingreso estimado del hogar asociado con el individuo
HAS_CHILDREN	Indicador (1 si tiene hijos en el hogar, 0 en caso contrario)
LENGTH_OF_RESIDENCE	Número estimado de años que el individuo ha vivido en su hogar actual
MARITAL_STATUS	Estado civil estimado (Casado o Soltero)
HOME_MARKET_VALUE	Valor estimado de la vivienda
HOME_OWNER	Indicador (1 si el individuo es propietario de su vivienda principal, 0 en caso contrario)
COLLEGE_DEGREE	Indicador (1 si el individuo tiene un título universitario o superior, 0 en caso contrario)
GOOD_CREDIT	Indicador (1 si el individuo tiene un puntaje FICO mayor a 630, 0 en caso contrario)
ACCT_SUSPD_DATE	Fecha de suspensión o cancelación de la cuenta

Cuadro 3.1: Descripción de las variables en el conjunto de datos.

2. Preprocesado

Detección de valores atípicos



(a) `age_in_years` (b) `curr_ann_amt`

(c) `income`

Figura 3.1: Valores atípicos de las variables

En cuanto a la calidad de los datos, se realizó una exploración preliminar para detectar la presencia de valores atípicos en las variables numéricas. Se identificaron valores atípicos en varias de ellas:

- **curr_ann_amt**: Se detectaron valores negativos, lo cual es inconsistente con la definición de la variable, ya que representa el monto anual pagado por el cliente durante el año previo. Al tratarse de pagos realizados, no debería existir valores negativos. Por esta razón, se eliminan estos registros.

- **age_in_years**: Se observaron valores extremadamente altos, superiores a 93 años, lo cual resulta poco probable para titulares de seguros de coche. Estos valores podrían corresponder a errores en la captura de datos o inconsistencias, por lo que se eliminan dichos registros.

- **income**: Aunque también se detectaron valores atípicos elevados, se decidió no eliminarlos, dado que los ingresos suelen presentar una distribución sesgada con una cola larga hacia valores altos, especialmente en un contexto como el de la distribución de la riqueza en Estados Unidos. Estos valores altos pueden representar clientes con ingresos legítimamente elevados y forman parte del comportamiento real de la población.

Valores nulos

Se identificaron algunas variables que presentan valores nulos. Entre ellas, destacan las variables asociadas a la localización geográfica (como `latitude`, `longitude`, `city` y `county`), que presentan una cantidad considerable de datos faltantes. No obstante, esto no representa un inconveniente para el desarrollo del modelo predictivo, ya que estas variables no serán utilizadas en dicho proceso. La decisión de excluirlas se basa en la intención de evitar posibles sesgos geográficos que podrían influir negativamente en la generalización del modelo, especialmente si ciertas zonas están sobrerrepresentadas o presentan características particulares.

En cuanto a la variable `home_market_value`, que indica el valor estimado del hogar, también se detectaron valores nulos. Inicialmente, se exploró la posibilidad de interpolar esta variable. Para ello, se creó una nueva variable denominada `home_value_ordinal`, que representa los intervalos de precio como una escala ordinal. A continuación, se llevó a cabo un análisis de correlación utilizando el coeficiente de Spearman, con el objetivo de identificar posibles variables correlacionadas que pudieran servir como base para la imputación.

Por último, la variable `acct_suspd_date`, que corresponde a la fecha de suspensión de cuenta, contiene un gran número de valores nulos. Este hecho es completamente esperable, ya que dicha información solo está disponible para los clientes que han realizado *churn*. Por tanto, la ausencia de datos en esta variable para clientes activos no constituye un problema de calidad, sino que refleja de manera coherente el estado de la relación con el servicio.

División del conjunto de datos

En primer lugar, el conjunto de datos se dividió en dos subconjuntos: uno de entrenamiento, que representa el 70 % de los datos totales, y otro de prueba, con el 30 %. Esta división se realizó de manera aleatoria, garantizando que se mantuviera la proporción de la variable objetivo

(churn) en ambos subconjuntos. Este enfoque permite evaluar de forma objetiva el rendimiento de los modelos, utilizando datos no vistos durante el entrenamiento.

Estandarización de variables numéricas

Posteriormente, se procedió a la estandarización de las variables numéricas para homogeneizar su escala y facilitar el proceso de aprendizaje de los modelos. Las variables estandarizadas fueron: `days_tenure`, `curr_ann_amt`, `age_in_years`, `income` y `length_of_residence`. La estandarización se llevó a cabo utilizando la técnica de z-score, transformando cada valor para que la variable resultante tenga media cero y desviación estándar uno. Esto es especialmente importante en algoritmos sensibles a la escala, como regresión logística.

Balanceo de la clase Churn

Dado que la clase objetivo no estaba balanceada, se aplicaron diversas técnicas de balanceo. Se utilizó *oversampling* para aumentar la cantidad de instancias de la clase minoritaria, *undersampling* para reducir las instancias de la clase mayoritaria, y finalmente se empleó *SMOTENC* para generar ejemplos sintéticos de la clase minoritaria. Estas técnicas permitieron equilibrar las clases y mejorar el rendimiento de los modelos.

Tratamiento de variables categóricas

En cuanto al tratamiento de variables categóricas, se realizó una recodificación de la variable `marital_status` para convertirla en binaria. Específicamente, se codificó como 1 si el estado civil era "Married", y 0 en cualquier otro caso.

Adicionalmente, se aplicó la técnica de one-hot encoding a la variable `home_market_value`, con el objetivo de convertir sus categorías en variables binarias independientes. Esta estrategia permite representar adecuadamente variables categóricas no ordinales sin introducir supuestos de orden entre sus niveles.

3. Selección de variables

Se ha decidido excluir una serie de variables que no aportan valor predictivo relevante o cuya inclusión podría introducir ruido o sesgo en los modelos.

En primer lugar, variables como `individual_id` y `address_id` son identificadores. Estas variables no contienen información útil para la predicción del abandono, por lo que se han eliminado del análisis.

Asimismo, se han descartado variables de tipo fecha como `cust_orig_date`, `date_of_birth` y `acct_suspd_date`. En el caso de `cust_orig_date` y `date_of_birth`, la información relevante que contienen ya se encuentra representada por las variables `days_tenure` y `age_in_years`, respectivamente, las cuales capturan de forma directa la antigüedad del cliente y su edad. En cuanto a `acct_suspd_date`, esta variable sólo está disponible para clientes que han realizado *churn*.

También se han excluido las variables geográficas como `street_adress`, `latitude`, `longitude`, `city`, `state` y `county`. Estas variables podrían inducir sesgos geográficos no deseados en el modelo.

4. Modelado

El trabajo se ha realizado utilizando el lenguaje de programación Python. En particular, se han utilizado librerías como Keras y Scikit-learn para la construcción y evaluación de los modelos de aprendizaje automático. Además, se ha empleado pandas y numpy para la manipulación de datos, así como matplotlib y seaborn para la visualización de los resultados.

El código se ha ejecutado en la plataforma de Google Colab, que ha proporcionado los recursos de computación necesarios para ejecutar los modelos de manera eficiente y sin restricciones de hardware.

Con el fin de garantizar la transparencia y la reproducibilidad del trabajo realizado, los notebooks resultantes se publicará en GitHub (luiocahoe), donde estarán disponibles para su consulta y posterior análisis.

Capítulo 4

Resultados

1. EDA

El conjunto de datos original estaba compuesto por 1,680,909 registros y 23 variables, con información proveniente del estado de Texas, Estados Unidos, específicamente de los condados de Kaufman, Dallas, Tarrant, Denton, Collin, Parker, Ellis, Navarro, Hunt, Johnson, Rockwall, Cooke, Grayson y Hill. Algunos registros presentaban valores nulos en el campo del condado. Tras eliminar identificadores, variables geográficas (para evitar sesgos), variables redundantes y aquellas con una alta proporción de valores nulos no imputables, se obtuvo un subconjunto con 1,588,623 registros y 12 variables. Este conjunto reducido será la base sobre la cual se desarrollará el Análisis Exploratorio de Datos (EDA), con el fin de identificar patrones y relaciones.

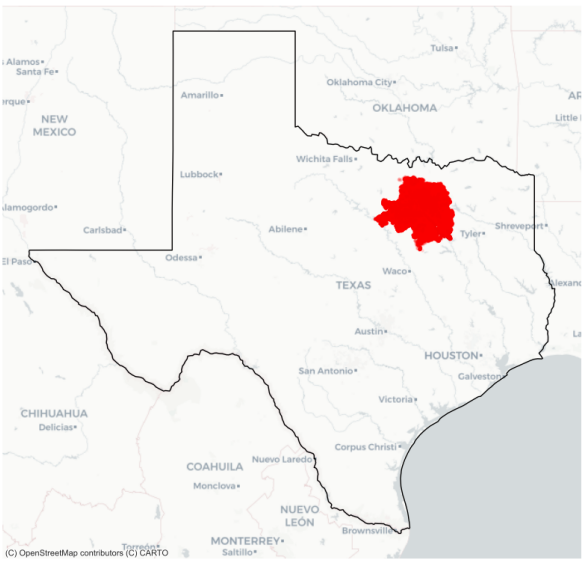


Figura 4.1: Distribución geográfica de los individuos

La representación gráfica de la distribución geográfica muestra una alta concentración de individuos en la región noreste del estado.

Es importante analizar la distribución de la variable `Churn` para determinar si las clases están balanceadas.

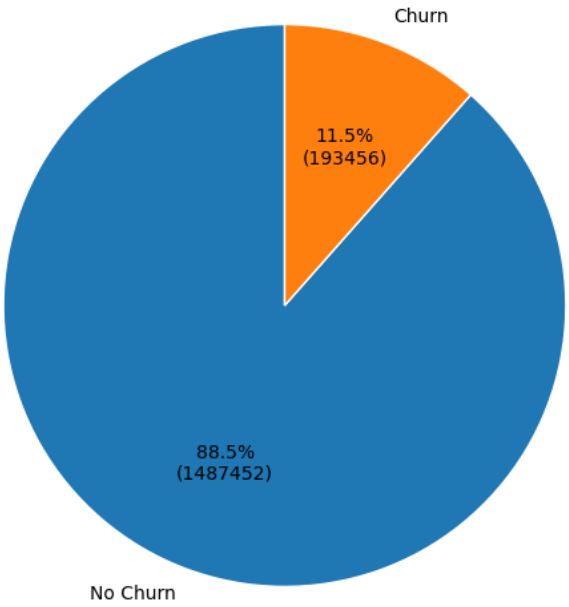


Figura 4.2: Distribución de la variable Churn

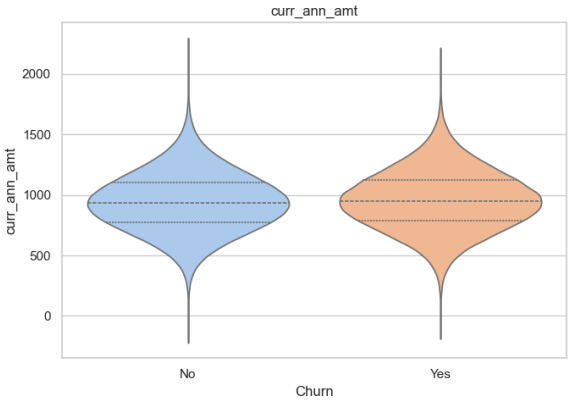
Para comprender mejor las características de los individuos en función de la variable `Churn`, se presenta a continuación un resumen estadístico de algunas variables relevantes. La tabla 1 compara los valores promedio de diversas características sociodemográficas y de comportamiento entre los usuarios que permanecen en el servicio y aquellos que lo han abandonado. Esta comparación permite identificar posibles diferencias significativas entre ambos grupos que podrían ser útiles en el análisis predictivo del churn.

Variable	No churn	Churn
Age (years)	56.34	53.80
Annual Amount	942.33	956.66
Income	81,966.06	81,025.45
Length of Residence	7.97	7.43
Days Tenure	3791.13	2271.65
Has Children	51.73 %	55.10 %
Marital Status (Married)	62.82 %	62.17 %
College Degree	35.63 %	33.07 %
Good Credit	84.63 %	83.86 %

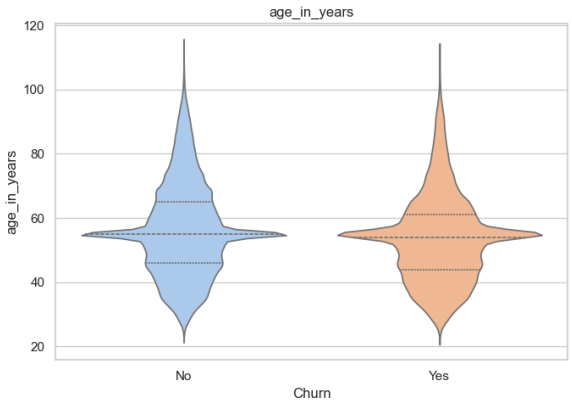
Cuadro 4.1: Promedio de variables por grupo de Churn

Asimismo, se han elaborado representaciones gráficas con el objetivo de visualizar la distribución de los datos y facilitar la detección de patrones.

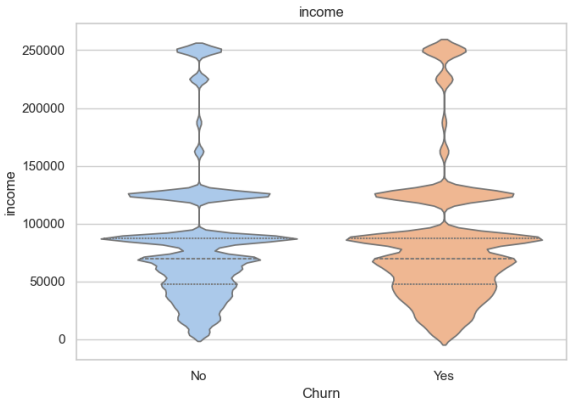
(a) `days_tenure`



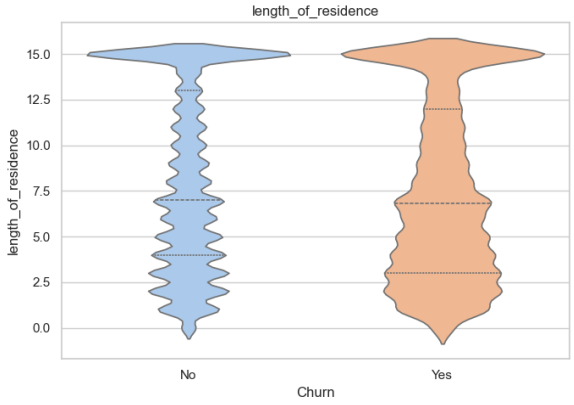
(b) `curr_ann_amt`



(c) `age_in_years`

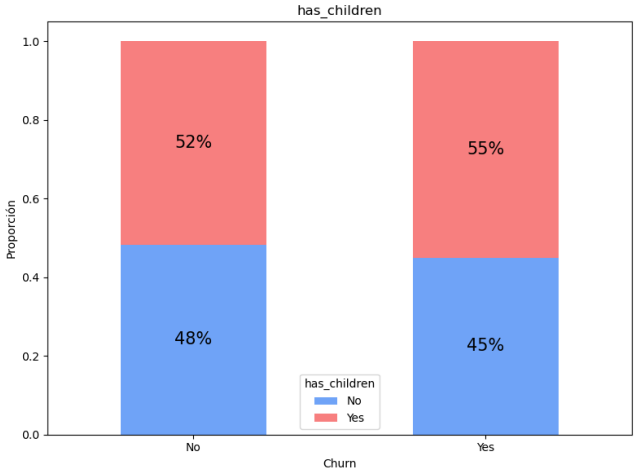


(d) `income`

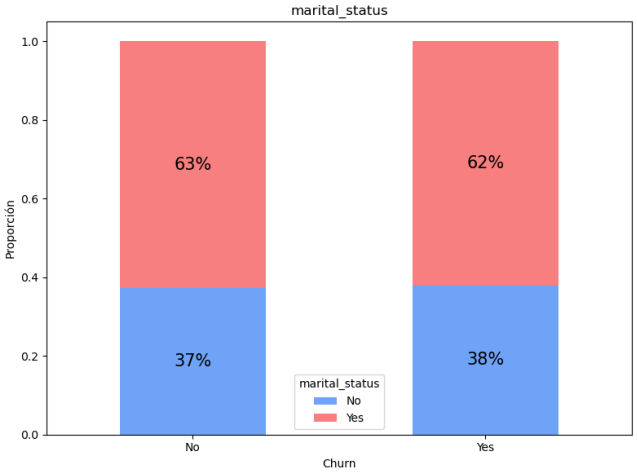


(e) `length_of_residence`

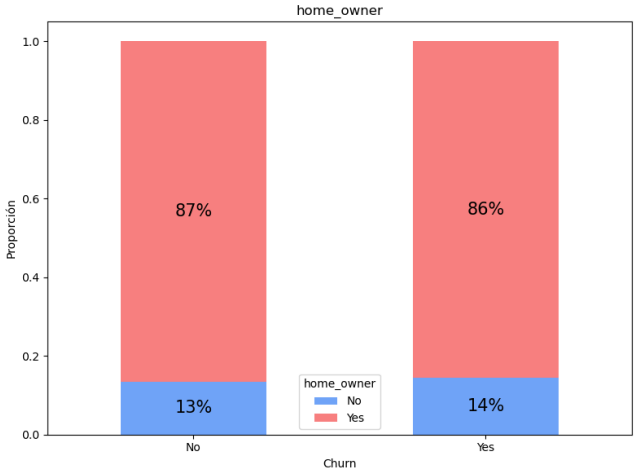
Figura 4.3: Distribución de variables numéricas según el estado de `Churn`



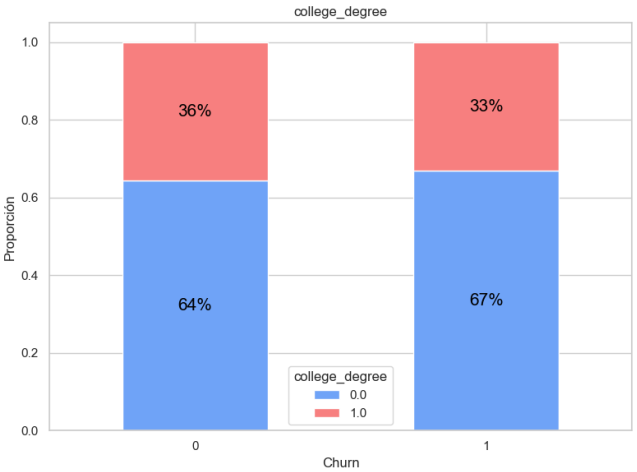
(a) `has_children`



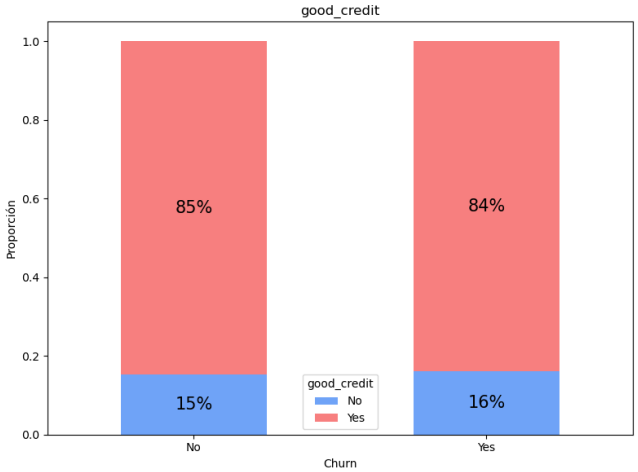
(b) `marital_status`



(c) `home_owner`



(d) `college_degree`



(e) `good_credit`

Figura 4.4: Distribución de variables binarias según el estado de `Churn`

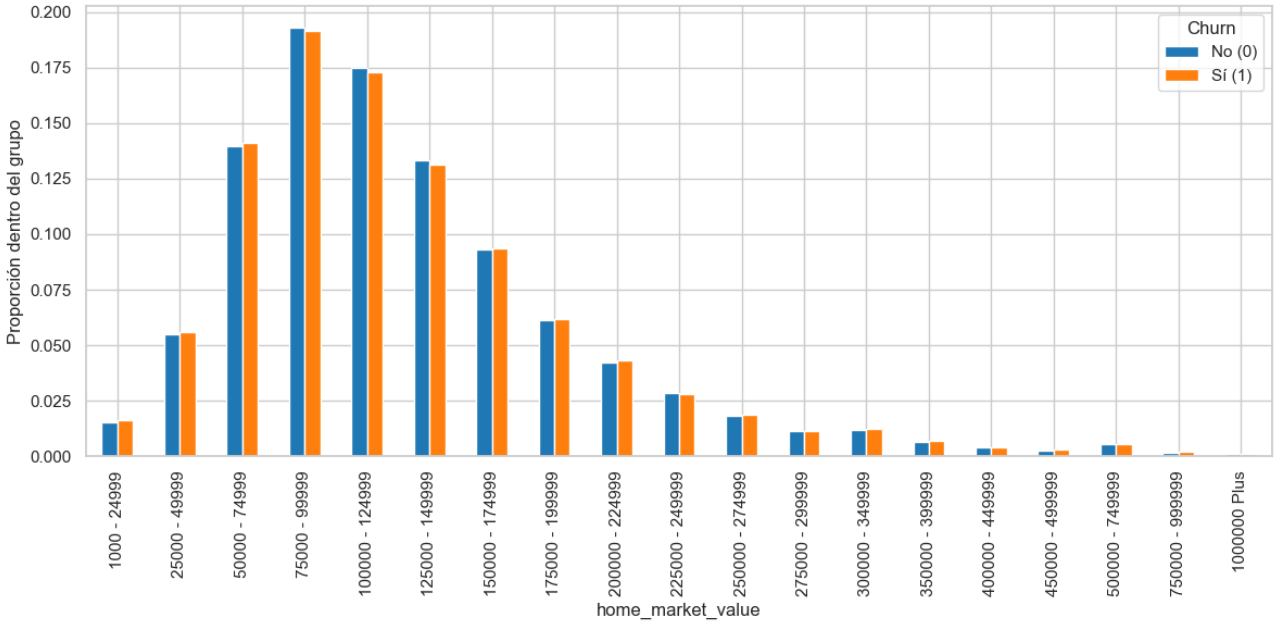


Figura 4.5: Distribución proporcional de `home market value` según `Churn`

Al analizar las variables segregadas por *churn*, se observa que la mayoría presentan distribuciones muy similares entre quienes hacen *churn* y quienes no, sin diferencias sustanciales. Sin embargo, hay algunas variables que muestran ligeras variaciones:

- **Tiempo de residencia (`length_of_residence`):** Los clientes que no hacen *churn* tienden a tener una mayor permanencia, mientras que quienes hacen *churn* se concentran en valores más bajos de residencia, aunque ambos grupos presentan casos con residencias prolongadas.

- **Edad (`age_in_years`):** Las distribuciones de edad son bastante parecidas entre ambos grupos. Existen pequeñas diferencias en los percentiles, pero no resultan especialmente significativas.

- **Nivel educativo (`college_degree`):** Entre los clientes que no hacen *churn*, el 64 % no tiene título universitario y el 36 % sí. En el grupo que sí hace *churn*, el 67 % no tiene título y el 33 % sí. Esto indica una ligera mayor proporción de personas sin título universitario entre quienes hacen *churn*.

- **Tener hijos (`has_children`):** En el grupo que no hace *churn*, el 52 % tiene hijos y el 48 % no. En el grupo que sí hace *churn*, el 55 % tiene hijos y el 45 % no, mostrando una pequeña mayor proporción de personas con hijos entre quienes hacen *churn*.

- **Antigüedad (`days_tenure`):** Los clientes que han hecho *churn* tienden a tener una antigüedad mucho menor, con un pico muy pronunciado cerca de 0 días. En cambio, los clientes que no han hecho *churn* muestran una distribución bimodal: un grupo pequeño con baja antigüedad y un grupo grande con alta antigüedad (cerca de 6500 días). Esto sugiere que el *churn* es más común en clientes nuevos, mientras que los clientes antiguos tienden a permanecer.

En resumen, aunque hay algunas diferencias puntuales, la mayoría de las variables no muestran patrones claramente distintos entre ambos grupos.

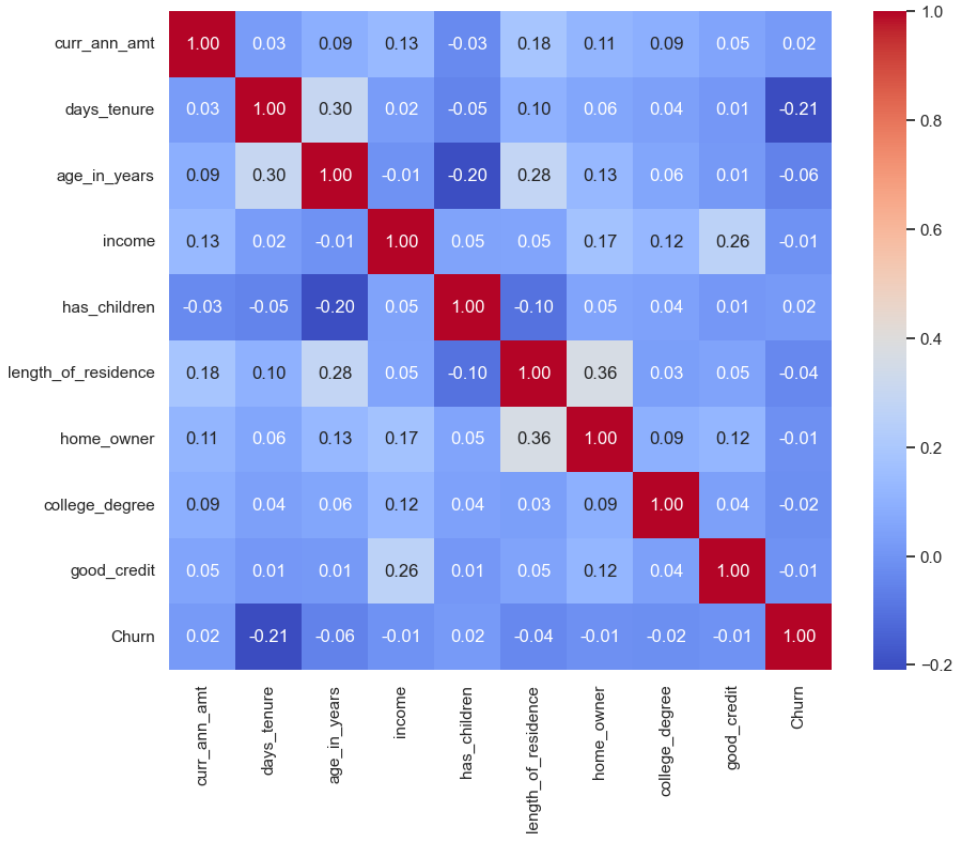


Figura 4.6: Matriz de correlación entre variables numéricas

Para analizar las relaciones entre las distintas variables del estudio, se ha elaborado una matriz de correlación empleando métodos estadísticos adaptados a la naturaleza de cada par de variables. En concreto, se ha utilizado el coeficiente de correlación de Pearson para medir la asociación entre variables continuas, así como para pares de variables binarias (interpretado como coeficiente Phi). En los casos que involucran variables ordinales o combinaciones de ordinales con continuas o binarias, se ha empleado el coeficiente de correlación de Spearman, que permite

captar relaciones monótonas. Finalmente, cuando se han relacionado variables continuas con binarias, se ha utilizado el coeficiente de correlación biserial puntual.

Notamos que ninguna variable muestra una correlación fuerte con el abandono de clientes. La variable más correlacionada negativamente es `days_tenure` (-0.21), como habíamos podido observar gráficamente.

Otras variables como `age_in_years` (-0.06), `length_of_residence` (-0.04) y `home_owner` (-0.01) tienen correlaciones muy bajas (cercanas a cero) con `Churn`, lo que indica que su relación con la variable objetivo es muy débil, al menos desde una perspectiva lineal.

Por otro lado, se observan algunas correlaciones interesantes entre variables independientes, como por ejemplo entre `length_of_residence` y `home_owner` (0.36), lo que tiene sentido ya que quienes han residido más tiempo en un lugar tienen más probabilidad de ser propietarios. También destaca una correlación moderada entre `age_in_years` y `length_of_residence` (0.28), coherente con el hecho de que las personas mayores tienden a tener más estabilidad residencial.

Es importante destacar que una baja correlación lineal no implica necesariamente que la variable carezca de valor predictivo. La ausencia de una relación lineal no descarta la existencia de relaciones no lineales o interacciones complejas con otras variables, las cuales pueden ser capturadas por modelos más sofisticados como árboles de decisión, bosques aleatorios o redes neuronales.

2. Modelado inicial

En esta primera etapa, se construyeron los diversos modelos predictivos empleando sus configuraciones por defecto, sin la introducción de ajustes específicos en los hiperparámetros. El objetivo principal de esta fase fue obtener una línea base de comparación que permitiera evaluar el desempeño inicial de cada algoritmo antes de aplicar técnicas más avanzadas de optimización y ajuste.

En el caso del modelo de red neuronal, se diseñó una arquitectura secuencial compuesta por distintas capas con funciones específicas:

- **Capas densas**: son responsables de aprender representaciones complejas a partir de las características de entrada. La primera capa densa recibe el vector de características y aplica una transformación no lineal, mediante la función de activación `ReLU`. Una segunda capa densa permite refinar dichas representaciones y capturar interacciones de mayor nivel entre las variables.

- **Capas de `dropout`**: se intercalan entre las capas densas y cumplen la función de regularizar el modelo, desactivando aleatoriamente una fracción de las neuronas durante el

entrenamiento. Esto previene el sobreajuste y promueve una generalización más robusta del modelo.

- **Capa de salida**: utiliza una función de activación `sigmoid` en tareas de clasificación binaria, devolviendo una probabilidad asociada a la clase positiva. Esta probabilidad se convierte posteriormente en una predicción binaria utilizando un umbral determinado.

Para mejorar la capacidad de generalización y evitar el sobreajuste, se incorporaron dos mecanismos de regularización dinámica durante el entrenamiento:

- `EarlyStopping`: detiene el proceso de entrenamiento automáticamente cuando el desempeño del modelo deja de mejorar sobre el conjunto de validación durante un número determinado de épocas consecutivas. Esto permite evitar un entrenamiento excesivo que podría llevar al modelo a ajustarse demasiado a los datos de entrenamiento.

- `ReduceLROnPlateau`: ajusta dinámicamente la tasa de aprendizaje del optimizador. Cuando la métrica de validación no mejora durante varias épocas, esta técnica reduce la tasa de aprendizaje, lo que facilita que el modelo realice ajustes más finos y estabilice su convergencia en las últimas etapas del entrenamiento.

Para entrenar este modelo, fue necesario dividir el conjunto de datos en tres subconjuntos: entrenamiento, validación y prueba. El conjunto original de datos se dividió inicialmente en un 80 % para entrenamiento y un 20 % para prueba. Luego, el subconjunto de entrenamiento fue subdividido nuevamente en un 80 % para entrenamiento y un 20 % para validación. Como resultado, la distribución final de los datos fue de aproximadamente un 56 % para entrenamiento, un 14 % para validación y un 30 % para prueba. Esta división permitió ajustar los hiperparámetros del modelo de red neuronal y monitorear su desempeño en un conjunto de validación independiente, lo cual fue clave para prevenir el sobreajuste y seleccionar la mejor configuración.

A continuación, se muestra una tabla con los resultados obtenidos por cada modelo bajo distintas técnicas de balanceo, utilizando sus configuraciones por defecto. Se presentan las métricas AUC y F1-score, enfocadas en la clase minoritaria, con el objetivo de establecer una línea base para comparar el impacto de futuros ajustes y mejoras.

Modelo	RandomOverSampler		RandomUnderSampler		SMOTENC	
	AUC	**F1-score**	**AUC**	**F1-score**	**AUC**	**F1-score**
Regresión Logística	0.6875	0.2735	0.6876	0.2734	0.6807	0.2729
Naive Bayes	0.6701	0.2795	0.6651	0.2814	0.6506	0.2036
KNN	0.6220	0.2654	0.6490	0.2692	0.6255	0.2633
Árbol de Decisión	0.5739	0.2448	0.5739	0.2487	0.5775	0.2492
Bosque Aleatorio	0.6917	0.3727	0.6943	0.3571	0.6889	0.3961
AdaBoost	0.6917	0.4113	0.6943	0.4110	0.6889	0.3814
XGBoost	0.6917	0.4513	0.6943	0.4427	0.6889	0.4410
Red Neuronal	0.6961	0.4516	0.6958	0.4394	0.6934	0.4051

Cuadro 4.2: Resultados de modelos con diferentes técnicas de balanceo (AUC y F1-score)

3. Optimización de parámetros

Se realizó un ajuste de hiperparámetros (hipertuning) para cada uno de los modelos evaluados, con el objetivo de optimizar su rendimiento en la predicción de la clase minoritaria. Para ello, se utilizó previamente la técnica de balanceo que mostró los mejores resultados preliminares para cada modelo, asegurando así que el ajuste de hiperparámetros se llevara a cabo bajo las condiciones más favorables para cada caso.

Para optimizar los hiperparámetros de los modelos, se empleó la técnica de GridSearchCV con un valor de cv=2. Esta elección se realizó considerando las limitaciones computacionales disponibles, ya que un número mayor de particiones en la validación cruzada podría haber resultado en un tiempo de cómputo excesivo. A pesar de esta restricción, el proceso permitió identificar los mejores hiperparámetros para cada modelo, enfocándose en mejorar la capacidad de predicción de la clase minoritaria. Cabe destacar que, en el caso de la red neuronal, se utilizó la biblioteca Optuna para llevar a cabo una optimización más eficiente y flexible, dada la complejidad y mayor número de hiperparámetros involucrados.

Naive Bayes

En el caso de Naive Bayes, dado que es un modelo de baja complejidad, se optó por un ajuste muy enfocado: únicamente se exploraron diferentes valores del parámetro `var_smoothing`, clave para la estabilidad numérica del modelo. Se probaron los siguientes valores: 1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3. Esto implicó un total de 14 ejecuciones, considerando las dos particiones.

KNN

Para K-Nearest Neighbors, el enfoque fue explorar combinaciones entre el número de vecinos considerados (`n_neighbors`) y la forma en que se ponderan las distancias (`weights`). Se probaron los siguientes valores para `n_neighbors`: 3, 5, 7, y para `weights`: uniform, distance. Se realizaron 12 pruebas en total, combinando estos valores para evaluar distintos tamaños de vecindario y estrategias de ponderación.

Árbol de decisión

El ajuste del árbol de decisión fue un poco más complejo. Aquí se evaluaron interacciones entre la función para calcular la impureza (`criterion`), la profundidad máxima del árbol (`max_depth`) y el número mínimo de muestras para dividir un nodo (`min_samples_split`). Se probaron los siguientes valores: `criterion`: gini, entropy, `max_depth`: 3, 5, 7, `min_samples_split`: 2, 5, 10. Se exploraron 18 combinaciones distintas, lo que resultó en un total de 36 ejecuciones.

Bosque aleatorio

El bosque aleatorio, a pesar de ser más robusto frente al sobreajuste, también requirió un ajuste cuidadoso. Se trabajó con distintos valores para la cantidad de árboles (`n_estimators`), la profundidad máxima, y los parámetros que definen cuándo dividir o detener un nodo (`min_samples_split` y `min_samples_leaf`). Los valores probados fueron los siguientes: `n_estimators`: 50, 100, 150, `max_depth`: 5, 10, 15, `min_samples_split`: 2, 5, `min_samples_leaf`: 1, 2. Esto resultó en 36 configuraciones, sumando un total de 72 ejecuciones.

AdaBoost

En el caso de AdaBoost, se enfocó en dos aspectos fundamentales: la cantidad de estimadores y la tasa de aprendizaje (`learning_rate`). Los valores probados fueron: `n_estimators`: 50, 100, 150, `learning_rate`: 0.01, 0.1, 0.5, 1. La combinación entre estos dos factores es clave, ya que un `learning_rate` mal ajustado puede anular los beneficios de contar con más iteraciones. En total se exploraron 12 combinaciones, con un total de 24 pruebas.

XGBoost

Por último, el modelo de XGBoost fue el más exigente en términos computacionales. Se ajustaron parámetros fundamentales como la cantidad de árboles, la tasa de aprendizaje, la profundidad máxima de cada árbol y el parámetro `gamma`, que regula la complejidad del modelo. Los valores probados fueron: `n_estimators`: 50, 100, 150, `learning_rate`: 0.01, 0.1, 0.2, 0.3, `max_depth`: 3, 6, 9, `gamma`: 0, 0.1, 0.3. Dado el número de combinaciones posibles, se realizaron 216 ejecuciones en total.

Red neuronal

El modelo de red neuronal se optimizó utilizando la biblioteca `Optuna`, que permite una búsqueda eficiente de hiperparámetros mediante algoritmos de optimización bayesiana. Se ajustaron parámetros clave que influyen directamente en la arquitectura y capacidad de generalización del modelo, tales como el número de neuronas en las primeras capas densas, las tasas de `dropout` y la tasa de aprendizaje del optimizador `Adam`. Los hiperparámetros evaluados fueron:

- `n_neurons_1`: número de neuronas en la primera capa densa, en el rango [32, 128].
- `n_neurons_2`: número de neuronas en la segunda capa densa, en el rango [16, 64].
- `dropout_1`: tasa de `dropout` en la primera capa, entre 0.1 y 0.5.

- `dropout_2`: tasa de `dropout` en la segunda capa, entre 0.1 y 0.5.
- `learning_rate`: tasa de aprendizaje del optimizador, en escala logarítmica entre 10^{-5} y 10^{-2}.

Se realizaron 10 ejecuciones de prueba, explorando diferentes combinaciones de hiperparámetros. Cada configuración seleccionada fue entrenada durante un máximo de 100 épocas, utilizando un batch_size de 256 muestras.

El entrenamiento se realizó con `EarlyStopping` y `ReduceLROnPlateau` como estrategias de regularización dinámica para evitar el sobreajuste:

- `EarlyStopping`: detiene el entrenamiento anticipadamente si la métrica de validación no mejora después de un número determinado de épocas consecutivas, previniendo el sobreajuste.
- `ReduceLROnPlateau`: reduce la tasa de aprendizaje automáticamente cuando la métrica de validación se estanca, facilitando una convergencia más fina del modelo.

La métrica utilizada para guiar el proceso de optimización fue el *F1-score*, evaluado sobre un conjunto de validación.

A continuación, se presentan los resultados de las métricas obtenidas específicamente sobre la clase minoritaria:

Modelo	**AUC**	**F1-score**	**Precision**	**Recall**
Naive Bayes	0.6780	0.2787	0.18	0.63
KNN	0.6798	0.2994	0.20	0.58
Árbol de decisión	0.6524	0.3963	0.46	0.35
Bosque Aleatorio	0.6937	0.4533	0.46	0.45
AdaBoost	0.6946	0.4114	0.36	0.48
XGBoost	0.6879	0.3667	0.46	0.30
Red Neuronal	0.6957	0.4391	0.42	0.46

Cuadro 4.3: Resultados de modelos de clasificación optimizados

4. Modelo resultante

El modelo de Random Forest con SMOTENC y ajuste de hiperparámetros es la opción más destacada para predecir el churn.

Si bien el recall del modelo de Random Forest para la clase minoritaria es de solo 0.45, su elección como modelo final respondió a un conjunto de criterios más amplios que van más allá de esa métrica aislada. En primer lugar, Random Forest logró el F1-score más alto (0.4533) entre todos los modelos evaluados, lo que sugiere un buen equilibrio global entre precisión y recall, lo cual es crucial en escenarios con clases desbalanceadas.

Además, aunque modelos como AdaBoost o la Red Neuronal mostraron recall ligeramente superior (0.48 y 0.46, respectivamente), también tuvieron menor precisión o menor F1-score, lo que indica un mayor número de falsos positivos o menor robustez en el equilibrio de métricas. Por ejemplo, AdaBoost tiene mejor recall pero una precisión considerablemente más baja (0.36), lo cual puede traducirse en intervenciones ineficientes si se usara en un entorno real.

Otro aspecto importante fue la eficiencia computacional y la estabilidad de Random Forest. Frente a modelos más complejos como XGBoost o redes neuronales, Random Forest ofreció tiempos de entrenamiento e inferencia más reducidos.

Por tanto, la elección de Random Forest no se basó exclusivamente en el recall, sino en un balance entre rendimiento, robustez y eficiencia, haciendo de él una opción práctica y sólida para el problema de churn.

Curva ROC

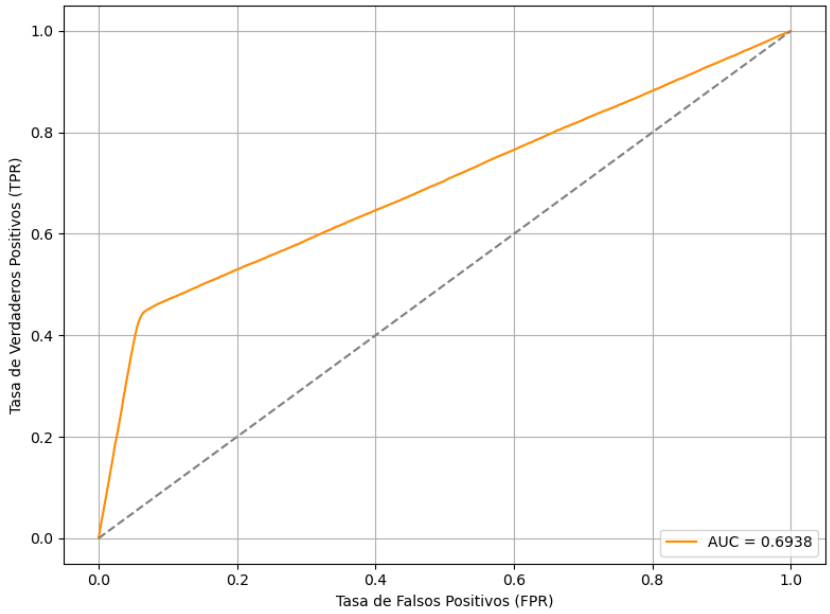


Figura 4.7: Curva ROC

En esta curva, se compara la tasa de verdaderos positivos (TPR) en el eje vertical con la tasa de falsos positivos (FPR) en el eje horizontal. En este caso:

- El AUC (Área Bajo la Curva) es 0.6938, lo que indica que el modelo tiene un rendimiento moderado.
- La línea diagonal punteada representa el rendimiento de un clasificador aleatorio (AUC = 0.5).
- La curva naranja muestra cómo el modelo supera al clasificador aleatorio.

El modelo presenta un aumento inicial pronunciado en la TPR, alcanzando aproximadamente 0.45, con una FPR muy baja al principio, lo que sugiere que está identificando correctamente muchos casos positivos sin cometer demasiados errores al inicio.

El valor de AUC de 0.6938 indica que el modelo tiene una capacidad discriminativa superior al azar, aunque no es excelente. Generalmente, un AUC superior a 0.8 se considera un buen desempeño, mientras que un valor superior a 0.9 sería excelente.

Matriz de confusión

Figura 4.8: Matriz de confusión

Donde:

- **Verdaderos Negativos**: 393,100
  Clientes que realmente no iban a abandonar y el modelo predijo correctamente que no abandonarían.
- **Falsos Positivos**: 29,079
  Clientes que realmente no iban a abandonar, pero el modelo predijo que sí abandonarían.
- **Falsos Negativos**: 29,971
  Clientes que realmente iban a abandonar, pero el modelo predijo que no lo harían.
- **Verdaderos Positivos**: 24,437
  Clientes que realmente iban a abandonar y el modelo lo predijo correctamente.

Métricas de clasificación

A continuación se presenta el *classification report* para el modelo, con las métricas de precisión, recall y F1-score para las clases de abandono y no abandono:

Clase	Precisión	Recall	F1-score	Soporte
0	0.9292	0.9311	0.9301	422179
1	0.4566	0.4491	0.4529	54408
Exactitud		0.8761		476587
Macro avg	0.6929	0.6901	0.6915	476587
Promedio ponderado	0.8752	0.8761	0.8757	476587

Cuadro 4.4: Reporte de clasificación

No Churn:

- **Precisión:** 0.9292: De todas las instancias clasificadas como No Churn, el 92.92 % realmente pertenecen a la clase No Churn.
- **Recall:** 0.9311: De todas las instancias reales de la clase No Churn, el 93.11 % fueron correctamente identificadas como No Churn.
- **F1-score:** 0.9301: Esta es la media armónica entre precisión y recall, lo que da una buena medida del rendimiento general en la clase No Churn.
- **Soporte:** 422179: Número total de instancias reales de la clase No Churn en el conjunto de datos (422,179).

Churn:

- **Precisión:** 0.4566: De todas las instancias clasificadas como Churn, solo el 45.66 % realmente pertenecen a la clase Churn.
- **Recall:** 0.4491: De todas las instancias reales de la clase Churn, solo el 44.91 % fueron correctamente identificadas como Churn.
- **F1-score:** 0.4529: El F1-score es de 0.4529, lo que indica que el modelo tiene un desempeño moderado en términos de balance entre precisión y recall para la clase Churn, aunque es significativamente más bajo que para la clase No Churn.
- **Soporte:** 54408: Número total de instancias reales de la clase Churn en el conjunto de datos (54,408).

El modelo identifica correctamente alrededor del 45 % de los clientes que realmente harán churn. Esto significa que más de la mitad de los que abandonarán no son detectados. Para un problema de retención, este es un punto débil importante, ya que limita la capacidad de la empresa para tomar acciones preventivas eficaces.

Además, de todos los clientes que el modelo predice que harán churn, solo alrededor del 46 % realmente lo hacen. Esto implica un número considerable de falsos positivos, es decir, clientes marcados para intervención que en realidad no iban a abandonar, lo que podría traducirse en un uso ineficiente de recursos de retención.

Métricas globales

Exactitud (Accuracy): 0.8761: El modelo tiene una exactitud global del 87.61 %, lo que significa que, en general, el modelo acierta en aproximadamente el 87.61 % de las predicciones, sumando las predicciones correctas tanto de la clase mayoritaria (Churn) como de la minoritaria (No Churn).

Promedio Macro (Macro avg): Las métricas de promedio macro se calculan como el promedio simple de las métricas para cada clase:

- **Precisión:** 0.6929
- **Recall:** 0.6901
- **F1-score:** 0.6915

Importancia de variables

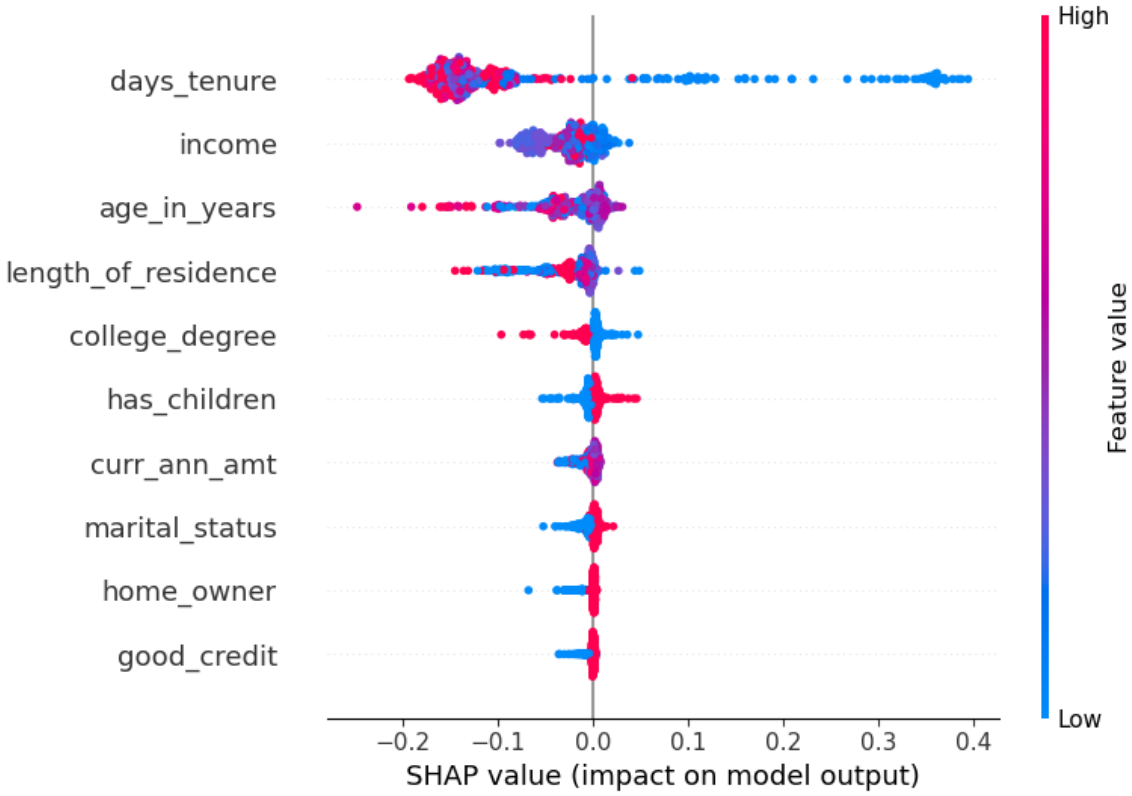


Figura 4.9: Gráfico resumen de valores SHAP

La imagen presentada muestra un *summary plot* de valores SHAP. Este tipo de visualización es especialmente útil en contextos donde se busca entender cómo y por qué un modelo toma determinadas decisiones.

Debido a las limitaciones computacionales asociadas al cálculo de los valores SHAP, que resulta intensivo en términos de recursos, especialmente en conjuntos de datos de gran tamaño como el empleado en este estudio, se optó por trabajar con una muestra aleatoria de 500 clientes del total disponible.

El gráfico SHAP resulta particularmente relevante, ya que contribuye a responder una de las preguntas clave del análisis: ¿qué factores influyen realmente en la probabilidad de que un cliente cancele su póliza?

Cada punto del gráfico representa un cliente individual, y cada fila corresponde a una de las variables utilizadas por el modelo. El eje horizontal muestra los valores SHAP, que cuantifican el efecto individual de cada variable sobre la predicción de cancelación (positiva o negativa) para cada cliente. Los colores indican el valor que toma la variable: rojo para valores altos y azul para valores bajos.

El gráfico revela que la variable más influyente es, con diferencia, `days_tenure` (antigüedad del cliente en la compañía). Los clientes con menor antigüedad (puntos azules hacia la derecha del gráfico) tienden a mostrar una mayor probabilidad de cancelación, mientras que aquellos con mayor permanencia (puntos rojos hacia la izquierda) tienden a permanecer con la aseguradora. Este hallazgo concuerda con lo observado en el análisis exploratorio previo: los clientes nuevos son más propensos al *churn*, mientras que los de larga trayectoria son más fieles.

Otras variables con una contribución significativa, aunque menor, son `income`, `age_in_years` y `length_of_residence`. En general, valores bajos en estas variables están asociados a un riesgo ligeramente mayor de cancelación. Por ejemplo, clientes más jóvenes, con menor estabilidad residencial o con ingresos más bajos presentan una mayor propensión al *churn*, aunque el impacto de estos factores es más sutil en comparación con la antigüedad.

En cambio, el resto de variables analizadas, como `has_children`, `annual_premium`, `marital_status`, `owns_house`, `has_university_education` o `good_credit_history`, se ubican en la parte inferior del gráfico, indicando un impacto mucho menor o prácticamente neutro sobre la decisión del modelo para la mayoría de los clientes.

Capítulo 5

Conclusiones y trabajo futuro

Este trabajo abordó de forma integral el problema del customer churn en el sector asegurador, identificando al modelo Random Forest combinado con SMOTENC como la mejor solución para predecir la cancelación de pólizas. Tras un proceso riguroso de preprocesamiento, selección de variables y evaluación comparativa de diversos algoritmos, este modelo alcanzó un F1-score de 0.4533, un AUC de 0.6937 y una exactitud global de 87.61 %, mostrando un desempeño equilibrado en la detección de clientes que abandonan, un reto clave en escenarios con clases desbalanceadas.

El análisis de interpretabilidad mediante valores SHAP indicó que los factores más influyentes en la predicción son la antigüedad del cliente, los ingresos y la edad.

La selección del modelo se justifica por su rendimiento superior en F1-score frente a otros modelos evaluados, como AdaBoost (0.4114), redes neuronales (0.4391) y XGBoost (0.3667). Aunque AdaBoost y redes neuronales obtuvieron un recall ligeramente mayor (0.48 y 0.46 respectivamente), lo hicieron sacrificando significativamente la precisión (0.36 y 0.42), generando un mayor número de falsos positivos. Random Forest logró un equilibrio más favorable, con precisión de 0.46 y recall de 0.45.

Asimismo, el análisis de interpretabilidad mediante valores SHAP permitió identificar que la antigüedad del cliente (`days tenure`) es el factor más influyente en la predicción del *churn*, seguido de otras variables como ingresos (`income`) y edad (`age_in_years`). Esta información resulta clave para la toma de decisiones estratégicas por parte de las aseguradoras, permitiéndoles enfocar sus esfuerzos de retención en los segmentos más propensos al abandono.

No obstante, es importante destacar las limitaciones del rendimiento del modelo. Un recall del 45 % para la clase churn implica que el modelo aún falla en identificar a más de la mitad de los clientes que efectivamente cancelarán su póliza. Si bien esto supera lo que se obtendría al azar y proporciona cierta capacidad de discriminación útil, para una aplicación industrial real se requeriría un rendimiento más alto, o bien una estrategia de intervención que acepte un

elevado número de falsos positivos (considerando que la precisión también es baja/moderada). El modelo actual, si se implementara como base para acciones de retención, implicaría contactar a muchos clientes que no se iban a ir (lo que genera un coste operativo), y aun así se perdería a más de la mitad de los clientes que sí abandonan. Esta limitación debe ser considerada cuidadosamente al diseñar políticas de retención basadas en las predicciones del modelo.

En este sentido, se proponen varias estrategias para mitigar el riesgo de pérdida de clientes. Entre ellas, destacan programas que refuercen la relación durante los primeros 12 a 18 meses, período en el que el riesgo de abandono es más elevado. También se recomienda una comunicación más frecuente y personalizada, incluyendo explicaciones detalladas sobre los beneficios y coberturas del producto.

Además, se sugiere implementar incentivos específicos para la primera renovación, como descuentos progresivos o beneficios adicionales que premien la continuidad. El desarrollo de productos con opciones de pago más flexibles, planes modulares que se adapten a las necesidades cambiantes del cliente y programas de fidelización con beneficios tangibles a corto plazo también pueden ser efectivos.

Finalmente, es importante adaptar las estrategias según el perfil del cliente: para los clientes jóvenes, se recomienda una comunicación digital, integración con aplicaciones móviles y programas que recompensen comportamientos seguros y responsables; mientras que para los clientes de mayor edad, se prioriza una atención más personalizada, simplicidad en los procesos y énfasis en la confianza y continuidad de la relación.

Cabe destacar que no se utilizó el modelo de *Support Vector Machines* (SVM), ya que no logró converger adecuadamente durante el entrenamiento, lo cual imposibilitó su evaluación en condiciones comparables con el resto de modelos.

A partir de los resultados obtenidos en este estudio, se abren varias líneas de investigación y desarrollo que podrían enriquecer el trabajo realizado y ampliar su aplicación práctica dentro del sector asegurador.

Aunque el conjunto de datos utilizado ya ofrece una base informativa valiosa sobre los clientes, existe un margen significativo de mejora al incorporar nuevas fuentes de información. Datos sobre interacciones con el servicio de atención al cliente, historial de siniestros y reclamaciones, o la contratación de productos complementarios, podrían enriquecer el contexto del cliente y aportar señales adicionales de comportamiento, incrementando así la capacidad predictiva del modelo.

Además, el enfoque actual se basa en una fotografía estática del cliente, es decir, un análisis transversal. Una evolución natural y necesaria sería adoptar una perspectiva longitudinal, integrando variables que capturen cómo evoluciona el comportamiento del cliente a lo largo del tiempo. Aspectos como cambios en los patrones de pago, modificaciones en el valor de la prima

o fluctuaciones en el historial de reclamaciones pueden ofrecer señales tempranas de deterioro en la relación, permitiendo anticiparse mejor a posibles cancelaciones. Este enfoque tiene un alto potencial para mejorar el recall, ya que permite identificar dinámicas de riesgo que los modelos estáticos no capturan.

Por otro lado, para que el modelo no solo prediga con precisión sino que también genere valor estratégico, es clave complementarlo con un análisis del impacto económico de las acciones de retención. Esto incluiría estimaciones del valor del cliente a lo largo de su ciclo de vida, evaluación de costos de intervención y proyecciones del retorno esperado de distintas estrategias. Así, las aseguradoras podrían priorizar acciones donde el beneficio neto sea mayor.

De cara a superar la limitación del recall, la incorporación de un enfoque longitudinal parece ser la línea con mayor potencial, ya que permite captar señales de abandono antes de que se manifiesten completamente. Sin embargo, su impacto sería aún mayor si se combina con fuentes de datos adicionales y un análisis económico que guíe la toma de decisiones de manera rentable y dirigida.

Bibliografía

[1] B. C. Stahl, *Artificial Intelligence for a Better Future: An Ecosystem Perspective on the Ethics of AI and Emerging Digital Technologies*, ser. SpringerBriefs in Research and Innovation Governance. Springer International Publishing. [Online]. Available: http://link.springer.com/10.1007/978-3-030-69978-9

[2] N. Hotz. What is CRISP DM? [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/

[3] La inteligencia artificial, gran aliada en nuestra lucha contra el fraude. [Online]. Available: https://www.axa.es/gl/-/la-inteligencia-artificial-gran-aliada-en-nuestra-lucha-contra-el-fraude

[4] Keith OBrien and Amanda Downie. What is customer churn? [Online]. Available: https://www.ibm.com/think/topics/customer-churn

[5] S. Kumar. Customer retention versus customer acquisition. Section: Small Business. [Online]. Available: https://www.forbes.com/councils/forbesbusinesscouncil/2022/12/12/customer-retention-versus-customer-acquisition/

[6] S.-Y. Hung, D. C. Yen, and H.-Y. Wang, "Applying data mining to telecom churn management," vol. 31, no. 3, pp. 515–524. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417405002654

[7] P. Lalwani, M. K. Mishra, J. S. Chadha, and P. Sethi, "Customer churn prediction system: a machine learning approach," vol. 104, no. 2, pp. 271–294. [Online]. Available: https://doi.org/10.1007/s00607-021-00908-y

[8] Sahar F. Sabbeh, "Machine-learning techniques for customer retention: A comparative study." [Online]. Available: https://www.researchgate.net/publication/323536762_Machine-Learning_Techniques_for_Customer_Retention_A_Comparative_Study

[9] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 5th ed., ser. Wiley Series in Probability and Statistics. Wiley.

[10] Y. Huang and L. Li, "Naive bayes classification algorithm based on small sample set," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, pp. 34–39, ISSN: 2376-595X. [Online]. Available: https://ieeexplore.ieee.org/document/6045027/?arnumber=6045027

[11] G.-e. Xia and W.-d. Jin, "Model of customer churn prediction on support vector machine," vol. 28, no. 1, pp. 71–77. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187486510960003X

[12] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN classification," vol. 8, no. 3, pp. 43:1–43:19. [Online]. Available: https://dl.acm.org/doi/10.1145/2990508

[13] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," in *2011 IEEE Control and System Graduate Research Colloquium*, pp. 37–42. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5991826

[14] Y.-c. Wu and J.-w. Feng, "Development and application of artificial neural network," vol. 102, no. 2, pp. 1645–1656. [Online]. Available: https://doi.org/10.1007/s11277-017-5224-x

[15] A. Parmar, R. Katariya, and V. Patel, "A review on random forest: An ensemble classifier," in *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, J. Hemanth, X. Fernando, P. Lafata, and Z. Baig, Eds. Springer International Publishing, pp. 758–763.

[16] F. Wang, Z. Li, F. He, R. Wang, W. Yu, and F. Nie, "Feature learning viewpoint of adaboost and a new algorithm," vol. 7, pp. 149 890–149 899, conference Name: IEEE Access. [Online]. Available: https://ieeexplore.ieee.org/document/8868178/?arnumber=8868178

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. Association for Computing Machinery, pp. 785–794. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[18] G. Weiss, "Mining with rarity: A unifying framework," vol. 6, pp. 7–19.

[19] H. Jain, A. Khunteta, and S. Srivastava, "Telecom churn prediction and used techniques, datasets and performance measures: a review," vol. 76, no. 4, pp. 613–630. [Online]. Available: https://doi.org/10.1007/s11235-020-00727-0

[20] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," vol. 36, pp. 4626–4636.

[21] Imbalanced-learn. Over-sampling. [Online]. Available: https://imbalanced-learn.org/stable/over_sampling.html#smote-adasyn

[22] 3.2. tuning the hyper-parameters of an estimator. [Online]. Available: https://scikit-learn/stable/modules/grid_search.html

[23] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework." [Online]. Available: http://arxiv.org/abs/1907.10902

[24] G. Marín Díaz, J. J. Galán, and R. A. Carrasco, "XAI for churn prediction in b2b models: A use case in an enterprise software company," vol. 10, no. 20, p. 3896, number: 20 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2227-7390/10/20/3896

[25] C. K. Leung, A. G. Pazdor, and J. Souza, "Explainable artificial intelligence for data science on customer churn," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9564166

[26] H. Nguyen, H. Cao, V. Nguyen, and D. Pham, "Evaluation of explainable artificial intelligence: SHAP, LIME, and CAM."

[27] Merishna Singh Suwal. Auto insurance churn analysis dataset. [Online]. Available: https://www.kaggle.com/datasets/merishnasuwal/auto-insurance-churn-analysis-dataset

[28] J. P. Nielsen, A. Asimit, and I. Kyriakou, *Machine Learning in Insurance*. MDPI - Multidisciplinary Digital Publishing Institute.

[29] J. Klaas, *Machine learning for finance: the practical guide to using data-driven algorithms in banking, insurance, and investments*, 1st ed. Packt Publishing.

[30] Y. He, Y. Xiong, and Y. Tsai, "Machine learning based approaches to predict customer churn for an insurance company," in *2020 Systems and Information Engineering Design*

Symposium (SIEDS), pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9106691

[31] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," vol. 55, pp. 1–9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X15000386

[32] N. Lu, H. Lin, J. Lu, and G. Zhang, "A customer churn prediction model in telecom industry using boosting," vol. 10, no. 2, pp. 1659–1665, conference Name: IEEE Transactions on Industrial Informatics. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6329952?casa_token=5boSVTpelXsAAAAA:htGoGjpC5BOqB3a5uizaXHHdQPfld9HiH87T1HSf7CL2mj3YHJ6S3crAUp-HFoXQjop3dIbVEA

[33] S. Khodabandehlou and M. Z. Rahman, "Comparison of supervised machine learning techniques for customer churn prediction based on analysis of customer behavior," vol. 19, no. 1, pp. 65–93, publisher: Emerald Publishing Limited. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/jsit-10-2016-0061/full/html

[34] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," vol. 6, no. 1, p. 28. [Online]. Available: https://doi.org/10.1186/s40537-019-0191-6