Modelos de Clasificación Basados en Ingeniería de Características Robusta para la Predicción de Deserción de Clientes en el Sector Financiero

Jefferson Adolfo Quiroz Fino
C.C. 1065815404
jaquirozf@eafit.edu.co

Director
Santiago Ortiz
sortiza2@eafit.edu.co

Maestría en Ciencias de los Datos y Analítica
Escuela de Ciencias Aplicadas e Ingeniería,
Universidad EAFIT, Medellín, Colombia

Resumen

La deserción temprana de clientes es un desafío crucial para las entidades financieras, afectando su rentabilidad y estabilidad. Este trabajo desarrolla y compara modelos de clasificación para predecir la deserción temprana de clientes en base a una entidad financiera que ofrece productos de consumo y movilidad, utilizando los modelos Support Vector Machines (SVM) y Elastic-Net. El SVM emplea eliminación de variables correlacionadas y detección de outliers mediante distancia de Mahalanobis robusta y bootstrap, y se entrena con los kernels Linear y RBF (Radial Basis Function). Elastic-Net utiliza una metodología similar para la ingeniería de características, sin eliminación de variables redundantes, y se entrena con regularización de sus parámetros Lambda y Alpha. Ambos modelos se evaluaron utilizando las métricas Accuracy, F1 Score, Precision y Recall. Los resultados muestran que el SVM con kernel RBF es el más efectivo, superando en todas las métricas a los otros modelos. Este enfoque proporciona una herramienta robusta para identificar clientes propensos a desertar tempranamente, permitiendo a las entidades financieras implementar estrategias preventivas y mejorar la retención de clientes.

Palabras Clave: Ingeniería de características, detección de atípicos, deserción de clientes, retención de clientes, Distancia de Mahalanobis, Bootstrap, Regularización.

1. Introducción

La deserción de clientes es un desafío crítico para las entidades que ofrecen productos crediticios. La competencia en el sector financiero es intensa, y la retención de clientes se ha convertido en una prioridad para muchas instituciones. Los clientes que abandonan una entidad financiera representan una pérdida significativa no solo en términos de ingresos, sino también en costos adicionales relacionados con la adquisición de nuevos clientes. La deserción puede estar influenciada por varios factores, incluidos la calidad del servicio al cliente, la competitividad de los productos ofrecidos y la percepción general del cliente sobre la entidad financiera (Amuda and Adeyemo, 2019). Además, la personalización en la oferta de productos y servicios es fundamental para mantener la lealtad del cliente (Ashraf, 2024). La integración de tecnologías mas recientes de análisis de datos, como los modelos de aprendizaje automático, puede mejorar significativamente la precisión en la predicción de la deserción, permitiendo acciones preventivas más efectivas (Li and Zhang, 2024).

En el contexto de una entidad que financia productos de movilidad y consumo, el problema de la deserción de clientes puede ser aún más pronunciado debido a la naturaleza específica y competitiva de estos mercados. Las entidades financieras deben abordar no solo la retención de clientes, sino también mejorar continuamente sus ofertas para satisfacer las demandas cambiantes de los consumidores. La satisfacción y la calidad del servicio son fundamentales para reducir la deserción (Rajola, 2019). Estudios recientes sugieren que la adopción de análisis predictivos y técnicas recientes de minería de datos puede ayudar a identificar patrones de deserción y desarrollar estrategias efectivas de retención (De Caigny, 2020). Además, la identificación temprana de clientes propensos a desertar permite a las instituciones financieras tomar medidas proactivas para retener a estos clientes, ofreciendo incentivos personalizados y mejorando la experiencia del cliente (Tran et al., 2022).

La investigación sobre la deserción de clientes en el sector financiero ha evolucionado considerablemente en las últimas décadas. Inicialmente, los estudios se centraron en identificar las razones fundamentales de la deserción y en el desarrollo de modelos predictivos básicos. Un estudio pionero en este campo utilizó técnicas de minería de datos para predecir la deserción de clientes en bancos europeos (Burez and Van den Poel, 2009). Posteriormente, otros investigadores ampliaron estos modelos utilizando algoritmos más recientes de aprendizaje automático, como los árboles de decisión y las redes neuronales, para mejorar la precisión de las predicciones (Verbeke, 2012). Estos avances permitieron una mejor comprensión de los factores que influyen en la deserción y ayudaron a desarrollar estrategias más efectivas para retener a los clientes.

En años más recientes, la atención se ha desplazado hacia la integración de tecnologías emergentes y el análisis de grandes volúmenes de datos. Por ejemplo, se desarrollaron modelos que combinan datos demográficos, transaccionales y comportamentales para proporcionar una visión más holística del cliente (Baesens et al., 2015). Además, se introdujo el uso de técnicas de machine learning para mejorar aún más las estrategias de retención, destacando la importancia de la personalización en la oferta de productos y servicios financieros (Lemmens and Croux, 2016). Por otro lado también se ha explorado cómo el riesgo

crediticio afecta la deserción de clientes, y se ha abordado el problema de la deserción en el contexto de las tarjetas de crédito utilizando modelos predictivos conocidos (Ashraf, 2024; Li and Zhang, 2024). Estos estudios han demostrado que la implementación de modelos predictivos y la personalización de servicios pueden ser altamente efectivas para reducir la deserción y mejorar la lealtad del cliente.

La aplicación de técnicas de machine learning se ha demostrado como una herramienta poderosa para abordar la deserción de clientes, permitiendo a las entidades financieras predecir comportamientos futuros y tomar decisiones proactivas. La adopción de modelos predictivos no solo ayuda a identificar a los clientes en riesgo de desertar, sino que también permite personalizar estrategias de retención, mejorando la satisfacción del cliente y optimizando los recursos de la entidad. Esta capacidad de anticipar y responder a la deserción de manera efectiva puede marcar una diferencia significativa en la competitividad y sostenibilidad para una institución que financia productos de movilidad y consumo.

El documento se desarrolla de la siguiente manera: en la Sección 2 se presenta la metodología utilizada, incluyendo técnicas de ingeniería de características y modelos de clasificación. En la Sección 3 se presentan los resultados obtenidos del modelo propuesto en términos de resolución de la problemática del caso de estudio, incluyendo una discusión. Finalmente, en la Sección 4 se muestran las conclusiones y comentarios finales sobre este trabajo.

2. Metodología

El enfoque principal de este trabajo se centra en la ingeniería de características robusta para la detección de outliers y la evaluación de su impacto en modelos básicos de clasificación, específicamente Support Vector Machines (SVM) y Elastic-Net. La robustez en este trabajo se refiere a la capacidad que tendrá proceso de ingeniería de características para manejar los datos atípicos o outliers y así reducir la influencia en el entrenamiento de los modelos. La metodología se divide en varios pasos clave que garantizan la integridad y validez del análisis, destacando la eliminación de variables correlacionadas, la detección de outliers mediante técnicas robustas, y la comparación de modelos de clasificación sin ajustes de hiperparámetros.

2.1. Ingeniería de Características

La ingeniería de características es un proceso esencial en el desarrollo de modelos de aprendizaje automático que implica la creación, transformación y selección de variables relevantes a partir de datos brutos para mejorar el rendimiento y la precisión del modelo Guyon and Elisseeff (2003). Este proceso abarca desde la selección de características más significativas utilizando métodos como el análisis de correlación y la importancia de características basada en modelos, hasta la transformación de datos mediante técnicas como la normalización, estandarización y codificación de variables categóricas. También incluye la creación de nuevas características derivadas que capturen mejor las relaciones y patrones

subyacentes en los datos. Un aspecto crítico de la ingeniería de características es el manejo de datos atípicos y la reducción de la multicolinealidad, eliminando outliers y variables redundantes que pueden afectar negativamente el modelo Zheng and Casari (2018). En conjunto, estas técnicas ayudan a construir modelos más robustos, precisos y fáciles de interpretar .

2.1.1. Eliminación de Variables Redundantes

Las variables redundantes o altamente correlacionadas son aquellas que contienen información similar o incluso duplicada en un conjunto de datos. Estas variables pueden resultar problemáticas en el entrenamiento de modelos de clasificación, ya que pueden llevar a un sobre ajuste o un mal condicionamiento, donde el modelo se ajusta demasiado a los datos de entrenamiento y pierde capacidad de generalización a nuevos datos. Además, pueden aumentar la complejidad del modelo innecesariamente, haciendo que los algoritmos de aprendizaje automático sean menos eficientes y más difíciles de interpretar (Toloşi and Lengauer, 2011). Una de las técnicas mas comunes al momento de remover variables correlacionadas es mediante el coeficiente de correlación, la cual implica calcular la correlación entre pares de variables en un conjunto de datos. Si dos variables tienen una alta correlación (positiva o negativa), indica que contienen información similar (Yu and Liu, 2004). En este caso, una de las variables redundantes debería eliminarse .

2.1.2. Correlación de Spearman

La correlación de Spearman (ρ_{sp}) es una medida no paramétrica que evalúa la relación entre dos variables al considerar los rangos en lugar de los valores originales Puth et al. (2015). La fórmula para calcular la correlación de Spearman entre dos variables X e Y es:

$$\rho_{sp}(X, Y) = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Donde d_i es la diferencia entre los rangos de cada par de observaciones y n es el número de observaciones.

2.1.3. Eliminación de Variables Correlacionadas por el Método de Spearman

Uno de los pasos fundamentales antes de proceder con la detección y eliminación de datos atípicos es la eliminación de variables correlacionadas. Este proceso se realiza utilizando la matriz de correlación de Spearman, que es particularmente adecuada para evaluar relaciones monotónicas entre variables, sin suponer que estas relaciones sean lineales (Spearman, 1904). Una vez calculada la matriz de correlación de Spearman para todas las variables del conjunto de datos, se identifican aquellos pares de variables cuya correlación absoluta sea mayor a 0.8. Esta umbral se elige basándose en la literatura que sugiere que una correlación superior a 0.8 indica una relación fuerte, lo que puede causar problemas de multicolinealidad en análisis posteriores (Sakshi et al., 2024). Por lo cual, de cada par de variables altamente correlacionadas, se elimina una para evitar redundancias y simplificar el modelo.

La elección de cuál variable eliminar puede depender de su relevancia en el contexto del análisis o de su interpretabilidad. Para ello se debe llevar a cabo el siguiente proceso:

1. Calcular la matriz de correlación de Spearman al conjunto de datos.
2. Identificar aquellos pares de variables con $|\rho_{sp}| > 0{,}8$
3. Eliminar aleatoriamente una variable.

Este enfoque garantiza que las variables incluidas en el análisis sean lo menos incorreladas entre sí, reduciendo la multicolinealidad y mejorando la eficiencia de los análisis subsiguientes.

2.1.4. Eliminación de Datos Atípicos por Distancia de Mahalanobis Robusta

La remoción de outliers es un proceso crucial en el análisis de datos y en la construcción de modelos de aprendizaje automático, ya que pueden distorsionar las inferencias estadísticas y afectar negativamente el rendimiento de los modelos. Los outliers son observaciones que se desvían significativamente de otras observaciones en el conjunto de datos, pudiendo surgir por errores de medición, variaciones extremas o condiciones inusuales. La remoción de outliers es esencial para mejorar la calidad de los datos, reducir el sesgo y aumentar la precisión y generalización de los modelos predictivos. Sin embargo, es importante realizar este proceso con cuidado para no eliminar datos valiosos que puedan contener información relevante (Nyitrai and Virág, 2019).

La distancia de Mahalanobis es una medida multivariada de distancia que toma en cuenta la correlación entre las variables de un conjunto de datos. Fue introducida por Mahalanobis (1936) y es ampliamente utilizada en análisis de datos multivariados para identificar y analizar patrones en los datos, así como para la detección de outliers. La distancia de Mahalanobis D_M) para una observación $\mathbf{x}$ con respecto a una distribución con media $\boldsymbol{\mu}$ y matriz de covarianza $\boldsymbol{\Sigma}$ es:

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

A diferencia de la distancia euclidiana, que trata a todas las variables de forma independiente y no considera la correlación entre ellas, la distancia de Mahalanobis ajusta la distancia en función de la estructura de la covarianza de los datos. Esto significa que es capaz de identificar correctamente la dirección y magnitud de las relaciones entre variables, lo que permite una evaluación más precisa de las distancias en el espacio multivariado. La distancia de Mahalanobis, bajo el supuesto de una distribución normal multivariada de los datos, sigue una distribución $\chi^2_{(k)}$, donde k es el número de variables. Esto permite establecer umbrales para identificar outliers. Para un nivel de significancia α, cualquier observación cuya distancia de Mahalanobis exceda el valor crítico $\chi^2_{(1-\alpha,k)}$ puede considerarse un outlier.

2.1.5. Método MCD para Estimación de la Matriz de Covarianza

El Método Minimum Covariance Determinant (MCD) (Rousseeuw and Van Driessen, 1999) es una técnica robusta para estimar la matriz de covarianza. Este método busca el subconjunto de datos de tamaño h (con h cercano a $n/2$) cuyo determinante de la matriz de covarianza es mínimo. La estimación robusta de la matriz de covarianza es crucial en muchas aplicaciones, ya que la presencia de outliers puede distorsionar significativamente las estimaciones basadas en métodos tradicionales. El método MCD es altamente eficaz para manejar outliers en los datos multivariantes. Croux and Haesbroeck (2000) destacan que este método minimiza el determinante de la matriz de covarianza en presencia de datos contaminados, tiene alto punto de ruptura y propiedades asintóticas deseables. Para estimar la matriz de covarianza por el método MCD es necesario realizar el siguiente procedimiento.

1. Se seleccionan aleatoriamente varios subconjuntos de tamaño h de los datos originales, donde h es el número de muestras usadas para la estimación. Para un conjunto de datos de n observaciones y p variables, se elige $h = \lfloor n/2 \rfloor + \lfloor (n + p + 1)/2 \rfloor$.

2. Para cada subconjunto, se calcula la matriz de covarianza y su determinante. El método para computar el determinante de una matriz de covarianza Σ se define como el producto de los autovalores de la matriz estimada (Hubert et al., 2008).

3. Se elige el subconjunto que minimiza el determinante de la matriz de covarianza.

4. Una vez seleccionado el mejor subconjunto, se reponderan las observaciones para calcular la matriz de covarianza final. Este paso ayuda a reducir la influencia de cualquier outlier que no haya sido completamente eliminado.

5. Finalmente, se calculan las medias y las matrices de covarianza robustas para el conjunto de datos completo, utilizando las observaciones reponderadas.

2.1.6. Regularización de Matrices por Método de Ledoit & Wolf

El Método de encogimiento de (Ledoit and Wolf, 2004) es una técnica de regularización para la estimación de matrices de covarianza, especialmente útil en situaciones donde el número de variables p es grande en comparación con el número de observaciones n. Este método se basa en la idea de encoger (shrinkage) la matriz de covarianza muestral hacia una matriz de covarianza estructurada o mejor condicionada, lo que mejora la estabilidad y precisión de las estimaciones. Esta técnica es altamente eficaz para manejar matrices de covarianza de alta dimensión. Ledoit and Wolf (2004) destacan que este método minimiza el error cuadrático medio esperado entre la matriz de covarianza verdadera y el estimador. A su vez, proporciona una extensión no lineal de la estimación de shrinkage para matrices de covarianza de alta dimensión (Wolf and Ledoit, 2011).

1. Dada una matriz de datos X de dimensión $n \times p$, donde n es el número de observaciones y p es el número de variables, la matriz de covarianza muestral S se calcula como:

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})(X_i - \bar{X})^T$$

Donde $\bar{X}$ es el vector de medias muestrales.

2. La matriz target T es típicamente una matriz estructurada como la matriz identidad escalada por el promedio de las varianzas muestrales:

$$T = \frac{\text{trace}(S)}{p} I$$

Donde I es la matriz identidad.

3. El estimador shrinkage combina S y T de la siguiente manera:

$$\Sigma_{LW} = \rho T + (1 - \rho) S$$

Donde ρ es el parámetro de shrinkage que se elige para minimizar el error cuadrático medio esperado entre la matriz de covarianza target y la matriz estimada.

4. Ledoit and Wolf (2004) proponen una ecuacuón específica para calcular ρ que minimiza el error cuadrático medio. Este parámetro se estima como:

$$\rho = \frac{\sum_{i \neq j} \text{Var}(s_{ij})}{\sum_{i \neq j} (s_{ij} - t_{ij})^2}$$

Donde s_{ij} y t_{ij} son los elementos de S y T, respectivamente.

2.1.7. Distancia de Mahalanobis con Medianas y Matriz de Covarianza Regularizada

Como se describió anteriormente, la distancia de Mahalanobis es una medida multivariante que toma en cuenta las correlaciones entre las variables. En este método, se quiere robustecer las medidas de localización y dispersión, que generalmente utilizan al calcular la distancia de Mahalanobis. Por lo cual, utilizamos el vector de medianas en lugar del vector de medias para robustecer la medida de localización y una matriz de covarianza robusta obtenida mediante el método MCD, la cual es luego regularizada por el método de Ledoit & Wolf, con el fin de robustecer la medida de dispersión. Para ello, es esencial seguir a detalle los siguientes pasos:

1. Dado un conjunto de datos X de dimensión $n \times p$, donde n es el número de observaciones y p es el número de variables, el vector de medianas $\tilde{\mathbf{x}}$ se calcula como:

$$\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p]$$

donde $\tilde{x}_i$ es la mediana de la i-ésima variable.

2. La matriz de covarianza robusta $\mathbf{S}_{\text{MCD}}$ se obtiene mediante el método MCD (Rousseeuw and Van Driessen, 1999).

3. La matriz de covarianza $\mathbf{S}_{\text{LW}}$ se calcula combinando la matriz de covarianza robusta $\mathbf{S}_{\text{MCD}}$ con una matriz de covarianza target $\mathbf{T}$ mediante un estimador shrinkage (Ledoit and Wolf, 2003):
$$\mathbf{S}_{\text{LW}} = \rho\mathbf{T} + (1-\rho)\mathbf{S}_{\text{MCD}}$$
donde ρ es el parámetro de shrinkage estimado.

4. La distancia de Mahalanobis para una observación $\mathbf{x}_i$ se calcula como:
$$D_M(\mathbf{x}_i) = \sqrt{(\mathbf{x}_i - \tilde{\mathbf{x}})^T \mathbf{S}_{\text{LW}}^{-1} (\mathbf{x}_i - \tilde{\mathbf{x}})}$$

Este método es particularmente útil para la detección robusta de outliers debido a su capacidad para resistir la influencia de valores atípicos. El uso de medianas en lugar de medias y la matriz de covarianza robusta regularizada garantiza que la medida de distancia no se vea distorsionada por la presencia de outliers, proporcionando una detección más precisa y fiable de valores atípicos.

2.1.8. Remoción de Datos Outliers por Bootstrapping

Este método combina la distancia de Mahalanobis robusta, utilizando vectores de medianas y la matriz de covarianza obtenida por MCD y regularizada por Ledoit & Wolf visto en la Sección 2.1.7, con el metodología bootstrap (Efron, 1979) para calcular el percentil 90 de las distancias calculadas, es decir, a un nivel de significancia del 10 %. Este procedimiento es requerido, debido a que generalmente la distancia de Mahalanobis sigue una distribución $\chi^2_{(k)}$, no obstante, al momento de robustecer las medidas de localización y dispersión que usualmente utiliza dicha distancia, no podemos garantizar que estas nuevas distancias, provengan de dicha distribución. Por lo cual, la metodología bootstrap nos ayuda a establecer una estimación empírica de la distribución para establecer un valor confiable del umbral de corte de las distancias calculadas. Para ello, es fundamental llevar a cabo el siguiente proceso:

1. Se generan $B = 1000$ muestras bootstrap $\mathbf{X}_b^*$ de la muestra original $\mathbf{X}$.
2. Para cada muestra bootstrap $\mathbf{X}_b^*$ se calcula la distancia de Mahalanobis $D_M^*(\mathbf{x}_{\mathbf{b}i})$.
3. La distribución empírica de las distancias D_M^* se utiliza para calcular el percentil 90 $\hat{Q}_{0,9}(D_M^*)$.

El método combinado permite detectar de manera robusta datos atípicos. Debido a que la distribución de la distancia de Mahalanobis robusta puede no ser conocida, el uso del bootstrap con muchas iteraciones permite inferir esta distribución y calcular el percentil con mayor precisión, identificando así los datos más alejados del centro multivariado.

2.2. Modelos de Clasificación

Un modelo de clasificación es una técnica de aprendizaje automático utilizada para asignar instancias a categorías predefinidas basándose en patrones y características observadas en los datos. Dentro de los modelos de clasificación, los modelos de clasificación

binaria son una categoría específica que se enfoca en predecir una de dos posibles clases. Estos modelos son ampliamente utilizados en aplicaciones donde la decisión es dicotómica, como la detección de fraudes, diagnósticos médicos (presencia o ausencia de una enfermedad) y clasificación de imágenes (por ejemplo, si una imagen contiene un objeto específico) (Roncaglioni et al., 2008).

2.2.1. Modelo de Clasificación Support Vector Machines

Un modelo de clasificación Support Vector Machine (SVM) es una técnica supervisada de aprendizaje automático utilizada para la clasificación de datos. SVM encuentra el hiperplano óptimo que separa las distintas clases en un espacio de características, maximizando el margen entre los puntos de datos de cada clase (Cortes and Vapnik, 1995). El modelo de clasificación SVM puede describirse de la siguiente manera:

1. Función de decisión:
$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$
Donde $\mathbf{w}$ es el vector de pesos, $\mathbf{x}$ es el vector de características de la muestra y b es el término de sesgo (Cortes and Vapnik, 1995).

2. Margen de separación:
$$\text{Margen} = \frac{2}{\|\mathbf{w}\|}$$

3. Problema de Optimización
$$\underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{sujeto a} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i$$
Donde y_i es la etiqueta de clase de la muestra i Boser et al. (1992).

4. SVM con Kernel
$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$
Donde $K(\cdot)$ es la función kernel que permite a SVM operar en un espacio de características de mayor dimensión (Cortes and Vapnik, 1995).

2.2.2. Kernels Linear y RBF en el Clasificador SVM

Lo dos tipos de kernels comúnmente utilizados en SVM al implementar modelos de clasificación, son el kernel linear y del kernel de función de base radial (RBF). A continuación se describirá a detalle el funcionamiento de dichos kernels en base a sus respectivas fórmulas matemáticas. Dando así, un entendimiento mas claro de su funcionalidad.

1. El kernel linear es una función utilizada en SVM que proyecta los datos de entrada en un espacio de mayor dimensión de forma lineal. Es útil cuando los datos son linealmente separables, es decir, cuando se puede encontrar una línea recta (o hiperplano

en dimensiones superiores) que divida las diferentes clases de datos (Abdullah et al., 2021; Joshi, 2022).

$$K(x, x') = x \cdot x'$$

Donde: x y x' son vectores de características en el espacio de entrada, $K(x, x')$ es el producto punto entre los vectores x y x', que representa la similitud lineal entre ellos.

2. El kernel RBF es una función utilizada en SVM que proyecta los datos de entrada en un espacio de mayor dimensión de manera no lineal. Es útil para problemas donde los datos no son linealmente separables, ya que puede modelar relaciones más complejas.

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

Donde γ es un parámetro que define la influencia de un solo punto de entrenamiento. Un valor pequeño de γ significa una mayor influencia y viceversa (Parisi et al., 2021; Dogaru and Dogaru, 2018).

2.2.3. Modelo Logístico Elastic-Net

Un modelo de clasificación Elastic-Net (Zou and Hastie, 2005) es una técnica supervisada de aprendizaje automático que combina las penalizaciones de LASSO (Tibshirani, 1996) y Ridge (Hoerl and Kennard, 1970) para manejar problemas de multicolinealidad y seleccionar características relevantes. Este modelo es especialmente útil cuando hay muchas variables correlacionadas y se requiere una regularización eficiente (Hastie et al., 2020). El modelo de clasificación se basa en la solución del siguiente problema de optimización :

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \left[\alpha \|\beta\|_1 + \frac{1-\alpha}{2} \|\beta\|_2^2 \right] \right\}$$

Donde λ es el parámetro de regularización que controla la magnitud de la penalización, α controla el balance entre LASSO ($\|\beta\|_1$) y Ridge ($\|\beta\|_2^2$). Cuando $\alpha = 1$, Elastic-Net se convierte en LASSO, y cuando $\alpha = 0$, se convierte en Ridge. $\|\beta\|_1$ es la norma ℓ_1 de los coeficientes β, que induce sparsity o selección de características y finalmente $\|\beta\|_2^2$ es la norma ℓ_2 de los coeficientes β, que induce estabilidad en los coeficientes. Sabiendo esto, para un correcto entrenamiento de un modelo de clasificación Elastic-Net se sugiere realizar los siguientes pasos:

1. Se divide el conjunto de datos en conjuntos de entrenamiento y prueba.
2. Se normalizan las características del conjunto de entrenamiento.
3. Se ajusta el modelo Elastic-Net al conjunto de entrenamiento utilizando validación cruzada para seleccionar los mejores valores de λ y α.
4. Se evalúa el rendimiento del modelo en el conjunto de prueba utilizando métricas como precisión, sensibilidad, especificidad y F1.

2.3. Modelos Propuestos

En esta sección se presentan tres modelos de aprendizaje supervisado para la clasificación de datos, cada uno optimizado mediante técnicas de ingeniería de características y preprocesamiento de datos específicas. Los modelos incluyen un SVM con kernel linear, un SVM con kernel RBF, y un modelo Elastic-Net. Se detallan los pasos de preprocesamiento y la justificación de la elección de cada modelo.

2.3.1. Modelo 1: SVM con Kernel linear e Ingeniería de Características Robusta

El primer modelo propuesto utiliza un SVM con kernel linear. Inicialmente, se lleva a cabo la eliminación de variables redundantes utilizando el método de correlación de Spearman, lo que permite identificar y eliminar aquellas variables con una correlación absoluta mayor a 0.8, reduciendo así la multicolinealidad. Posteriormente, se procede a la eliminación de datos atípicos mediante la distancia de Mahalanobis robusta, utilizando el vector de medianas y la matriz de covarianza obtenida a través del método MCD y regularizada por el método de Ledoit & Wolf. Este proceso se complementa con el método bootstrap para determinar el percentil 90 de las distancias de Mahalanobis robustas y eliminar el 10 % de los datos atípicos.

Una vez preprocesados los datos, se entrena un modelo SVM con kernel linear. Este modelo busca encontrar el hiperplano que maximiza el margen entre las clases, es decir, la distancia más amplia posible entre los datos de diferentes clases. La elección del kernel linear es adecuada cuando los datos son aproximadamente linealmente separables en el espacio de características, permitiendo una separación clara y sencilla entre las clases. El objetivo principal de este modelo es lograr una clasificación precisa y eficiente, minimizando el riesgo de sobreajuste y mejorando la capacidad de generalización del modelo a nuevos datos.

2.3.2. Modelo 2: SVM con Kernel RBF e Ingeniería de Características Robusta

El segundo modelo es similar al primero en cuanto a los pasos de preprocesamiento, donde se realiza la eliminación de variables redundantes mediante la correlación de Spearman y la detección de datos atípicos utilizando la distancia de Mahalanobis robusta y el método bootstrap. La diferencia principal radica en la elección del kernel: en lugar de un kernel linear, se utiliza un kernel RBF.

El kernel RBF permite al SVM manejar relaciones no lineales entre las variables. Este kernel proyecta los datos en un espacio de mayor dimensión donde las clases pueden ser separadas por un hiperplano. La función RBF mide la similitud entre dos puntos en el espacio de características y es especialmente útil para capturar relaciones complejas en los datos. El parámetro gamma del kernel RBF controla la influencia de un solo punto de entrenamiento, permitiendo ajustar la flexibilidad del modelo. Este ajuste es crítico para equilibrar la capacidad del modelo de adaptarse a los datos de entrenamiento y su

generalización a nuevos datos. La capacidad del kernel RBF para capturar relaciones no lineales mejora significativamente el rendimiento del modelo en conjuntos de datos donde las clases no son linealmente separables, proporcionando una clasificación más precisa y robusta.

2.3.3. Modelo 3: Elastic-Net con Eliminación de Datos Atípicos Robusta

El tercer modelo propuesto es un Elastic-Net. A diferencia de los dos modelos anteriores, este modelo no elimina variables redundantes durante la fase de preprocesamiento. En su lugar, se enfoca en la regularización de los parámetros λ y α para manejar la multicolinealidad y seleccionar características relevantes. La eliminación de datos atípicos se realiza de la misma manera que en los modelos anteriores, utilizando la distancia de Mahalanobis robusta y el método bootstrap.

Elastic-Net combina las penalizaciones de LASSO y Ridge, lo que permite seleccionar características y mantener la estabilidad del modelo. La regularización LASSO tiende a eliminar variables irrelevantes, produciendo un modelo más sencillo y fácil de interpretar. Por otro lado, la regularización Ridge penaliza la magnitud de los coeficientes para evitar el sobreajuste, especialmente útil cuando existen muchas variables correlacionadas. La combinación de ambas técnicas en Elastic-Net permite aprovechar las ventajas de LASSO y Ridge, mejorando la capacidad del modelo para manejar conjuntos de datos con alta dimensionalidad y multicolinealidad. El objetivo final del modelo Elastic-Net es lograr un equilibrio óptimo entre precisión y simplicidad, proporcionando un modelo robusto y eficiente para la clasificación de datos complejos.

3. Resultados

3.1. Aplicación para Datos de una Entidad Financiera

El análisis se centra en una entidad que financia bienes de consumo y movilidad, la cual requiere enfrentar el reto de prevenir la deserción temprana de clientes, dado que esta afecta negativamente el valor de la marca y provoca pérdidas financieras. Para mitigar este impacto, la entidad desea emplear estrategias basadas en machine learning. Por lo cual, el objetivo es desarrollar un modelo predictivo que permita a la entidad implementar estrategias preventivas, con el fin de ayudar a mejorar la retención de clientes y a reducir las pérdidas financieras. Para ello, se utilizó un conjunto de datos anonimizado de créditos cancelados en los últimos seis meses, con 94,644 registros y 64 variables predictoras que incluyen información financiera, socio-demográfica y crediticia. La variable objetivo es binaria, representando la deserción temprana (1) y la cancelación normal del crédito (0). Los datos se mantuvieron dentro del ecosistema analítico de la entidad para asegurar su protección e integridad. La descripción de las variables se detalla en los Cuadros 1, 2 y 3.

Cuadro 1: Descripción de las variables parte 1

Nombre Variable	Descripción
numero_cuota	Número de la cuota correspondiente a un plan de pagos o préstamo.
fact_cap	Monto total del capital pagado.
fact_int_corrient	Monto total de interés corriente pagado.
fact_int_caus	Monto total de interés causado.
val_desembolso	Monto total de dinero desembolsado.
val_saldo	Monto restante o saldo del préstamo.
plazo	Plazo del préstamo en meses.
dias_permanencia	Días de permanencia que el cliente lleva en la institución financiera.
valor_tasa_efectiva	Valor de la tasa efectiva del préstamo.
valor_tasa_nominal	Valor de la tasa nominal del préstamo.
edad	Edad del cliente.
ing_mes	Ingresos mensuales del cliente.
otros_ing	Otros ingresos del cliente.
share_of_wallet	Porcentaje de cartera compartida con otras entidades.
cant_mantenimientos	Cantidad de mantenimientos realizados al crédito del cliente.
cant_requerimientos	Cantidad de requerimientos realizados por el cliente.
n_polizas	Cantidad de pólizas de seguro activas.
sva	Valor económico agregado por el cliente.

Cuadro 2: Descripción de las variables parte 2

Nombre Variable	Descripción
cant_llamadas_saldo	Cantidad de llamadas realizadas por el cliente para solicitar su saldo.
is_calif_cartera__A	Indica si la calificación de la cartera es A (1 si es, 0 si no).
is_calif_cartera__B	Indica si la calificación de la cartera es B (1 si es, 0 si no).
is_calif_cartera__C	Indica si la calificación de la cartera es C (1 si es, 0 si no).
is_calif_cartera__D	Indica si la calificación de la cartera es D (1 si es, 0 si no).
is_calif_cartera__E	Indica si la calificación de la cartera es E (1 si es, 0 si no).
is_marca_producto__BICIC	Indica si la categoría del crédito es bicicleta (1 si es, 0 si no).
is_marca_producto__CELUL	Indica si la categoría del crédito es celular (1 si es, 0 si no).
is_marca_producto__COMPU	Indica si la categoría del crédito es computadora (1 si es, 0 si no).

is_marca_producto__CONSU	Indica si la categoría del crédito es consumo (1 si es, 0 si no).
is_marca_producto__ENTRE	Indica si la categoría del crédito es entretenimiento (1 si es, 0 si no).
is_marca_producto__HOGAR	Indica si la categoría del crédito es hogar (1 si es, 0 si no).
is_marca_producto__MOTO	Indica si la categoría del crédito es moto (1 si es, 0 si no).
is_marca_producto__REEST	Indica si hubo una reestructuración en el crédito (1 si es, 0 si no).
is_marca_producto__SALUD	Indica si la marca del producto es salud (1 si es, 0 si no).
is_ocupacion__EMPLE	Indica si la ocupación es empleado (1 si es, 0 si no).
is_ocupacion__EMPRE	Indica si la ocupación es emprendedor (1 si es, 0 si no).
is_ocupacion__ESTUD	Indica si la ocupación es estudiante (1 si es, 0 si no).
is_ocupacion__HOGAR	Indica si la ocupación es hogar (1 si es, 0 si no).
is_ocupacion__INDEP	Indica si la ocupación es independiente (1 si es, 0 si no).
is_ocupacion__PENSI	Indica si la ocupación es pensionado (1 si es, 0 si no).
is_ocupacion__RENTI	Indica si la ocupación es rentista (1 si es, 0 si no).
is_ocupacion__TRANS	Indica si la ocupación es transportista (1 si es, 0 si no).
is_estado_civil__CASAD	Indica si el estado civil es casado (1 si es, 0 si no).
is_estado_civil__DIVOR	Indica si el estado civil es divorciado (1 si es, 0 si no).
is_estado_civil__RELIG	Indica si el estado civil es religioso (1 si es, 0 si no).

Cuadro 3: Descripción de las variables parte 3

Nombre Variable	**Descripción**
is_estado_civil__SEPAR	Indica si el estado civil es separado (1 si es, 0 si no).
is_estado_civil__SOLTE	Indica si el estado civil es soltero (1 si es, 0 si no).
is_estado_civil__UNION	Indica si el estado civil es unión libre (1 si es, 0 si no).
is_estado_civil__VIUDO	Indica si el estado civil es viudo (1 si es, 0 si no).
is_genero__FEMEN	Indica si el género es femenino (1 si es, 0 si no).
is_genero__MASCU	Indica si el género es masculino (1 si es, 0 si no).
is_regional__ANTIO	Indica si la región es Antioquia (1 si es, 0 si no).
is_regional__BOGOT	Indica si la región es Bogotá (1 si es, 0 si no).
is_regional__CARIB	Indica si la región es Caribe (1 si es, 0 si no).
is_regional__CENTR	Indica si la región es Centro (1 si es, 0 si no).
is_regional__SUR	Indica si la región es Sur (1 si es, 0 si no).
is_grupo_consolidado_EDU	Indica si el grupo consolidado al que pertenece el crédito es educativo (1 si es, 0 si no).
is_grupo_consolidado_LIBRE	Indica si el grupo consolidado al que pertenece el crédito es libre inversión (1 si es, 0 si no).
is_segmento_banco__INDEP	Indica si el segmento del banco es independiente (1 si es, 0 si no).

is_segmento_banco__PERSO	Indica si el segmento del banco es personal (1 si es, 0 si no).
is_segmento_banco__PLUS	Indica si el segmento del banco es plus (1 si es, 0 si no).
is_segmento_banco__PREFE	Indica si el segmento del banco es preferencial (1 si es, 0 si no).
is_segmento_banco__PYMES	Indica si el segmento del banco es PYMES (1 si es, 0 si no).
is_segmento_banco__SOCIA	Indica si el segmento del banco es social (1 si es, 0 si no).
is_declarante	Indica si es declarante (1 si es, 0 si no).
DC	Variable objetivo.

3.2. Simulaciones

Para equilibrar los datos y entrenar los modelos de SVM y Elastic-Net, se realizaron simulaciones detalladas partiendo de los datos reales. Inicialmente, se eliminaron los outliers utilizando el metodo de deteccion de datos atipicos por medio de la distancia de mahalanobis robusta y el método bootstrap documentado en la sección 2.1.8, lo que resultó en un conjunto de datos de 85,328 registros, de los cuales 5,726 pertenecían a la clase minoritaria (1) y 79,602 a la clase mayoritaria (0). Con el fin de evitar sesgos hacia la clase mayoritaria, se seleccionaron aleatoriamente 6,000 registros de esta clase y se iteraron 100 veces, creando así un conjunto de datos balanceado por cada simulación.

Como se describió anteriormente, se seleccionaron 100 conjuntos de datos balanceados, esto con el fin de que cada modelo se entrenara 100 veces por cada hiper-parámetro modificado, para asegurar la validez y robustez de cada modelo. En cada una de estas 100 iteraciones se aplicó la técnica de validación cruzada con 5 folds (Yadav and Shukla, 2016), esto significa que para cada iteración, el conjunto de datos balanceado se dividió en 5 partes, y cada modelo se entrenó 5 veces con diferentes particiones de los datos. Este proceso resultó en un total de 500 simulaciones por hiper-parámetro. El objetivo principal es entrenar los modelos con datos balanceados, lo que permite que los modelos aprendan de manera efectiva y reduzcan el sesgo hacia la clase mayoritaria. Este enfoque busca obtener predicciones más precisas y útiles, asegurando que ambos modelos tengan una oportunidad justa de aprender de cada clase.

Para evaluar el desempeño de cada una de las 500 simulaciones, se utilizaron las métricas de Accuracy, F1 Score, Precision y Recall (Behera et al., 2019). Estas métricas proporcionan una visión completa del rendimiento de los modelos, permitiendo identificar no solo qué tan precisas son las predicciones, sino también cómo se manejan los falsos positivos y los falsos negativos. Estas simulaciones proporcionaron una base robusta para evaluar el rendimiento de los modelos de SVM y Elastic-Net en un entorno equilibrado, permitiendo una mejor comparación y selección del modelo más adecuado para predecir la deserción temprana de clientes en la entidad financiera mecionada en la sección 3.1. Este enfoque meticuloso no solo mejora la capacidad de predicción de los modelos, sino que también asegura que las decisiones basadas en estos modelos sean coherentes conforme a la realidad.

3.3. Resultados de los Modelos Entrenados

En esta sección se presentarán los resultados obtenidos de la evaluación de los tres modelos de clasificación: SVM con kernel linear, SVM con kernel RBF y Elastic-Net. Cada subsección incluirá un breve resumen de la metodología de evaluación del modelo correspondiente, una tabla con el promedio de las métricas de rendimiento (accuracy, F1 score, precision y recall) y un resumen de los resultados obtenidos. Además, se discutirá el rango de parámetros evaluados para el modelo Elastic-Net y se presentarán mapas de calor que ilustran cómo varían las métricas con respecto a los parámetros lambda y alpha. Esta estructura permitirá una comparación clara y detallada del desempeño de cada modelo, facilitando la identificación del modelo más adecuado para el caso de uso propuesto.

3.3.1. Modelo SVM con Kernel linear

Como se describió con anterioridad, este primer modelo, SVM con kernel linear, fue evaluado utilizando técnicas robustas de ingeniería de características y eliminación de outliers. La eliminación de variables correlacionadas se realizó mediante la correlación de Spearman, y los outliers fueron identificados usando la distancia de Mahalanobis robusta y el método bootstrap. Posteriormente, se entrenó el modelo SVM con kernel linear en un conjunto de datos balanceado utilizando validación cruzada con 5 folds, resultando en 500 simulaciones.

Cuadro 4: Promedio del resultado de las simulaciones para el modelo SVM con Kerne Linear

Métrica	**Promedio**	**Mínimo**	**Máximo**
Accuracy	0.923482	0.899727149	0.951568895
F1 Score	0.920594	0.896354539	0.949964764
Precision	0.932937	0.89904502	0.958748222
Recall	0.908687	0.872727273	0.94972067

El modelo SVM con kernel linear mostró un buen desempeño en promedio, en base a las 500 simulaciones realizadas, como se puede observar en el Cuadro 4. La métrica de accuracy promedio fue de 0.9235. El F1 score promedio fue de 0.9206. La precisión promedio fue de 0.9329, y el recall promedio fue de 0.9087. Dadas las características que nos brida el Kernel linear y la calificación obtenida en base a cada métrica, se puede inferir que este modelo es adecuado para aplicaciones donde se necesita un modelo simple y eficiente, especialmente cuando los datos son linealmente separables.

3.3.2. Modelo SVM con Kernel RBF

El segundo modelo, SVM con kernel RBF, siguió el mismo procedimiento de preprocesamiento que el primer modelo. Después de eliminar variables redundantes y outliers, se entrenó el modelo SVM con kernel RBF en el conjunto de datos balanceado, utilizando validación cruzada con 5 folds y realizando 500 simulaciones.

Cuadro 5: Promedio del resultado de las simulaciones para el modelo SVM con Kerne RBF

Métrica	**Promedio**	**Mínimo**	**Máximo**
Accuracy	0.966683	0.950170648	0.979536153
F1 Score	0.966754	0.950575491	0.979367263
Precision	0.943280	0.918709677	0.964769648
Recall	0.991466	0.981818182	0.998603352

El modelo SVM con kernel RBF presentó un rendimiento superior en comparación con el modelo SVM con kernel linear, como podemos observar mediante el Cuadro 5, el cual nos muestra el valor promedio de calificación de cada métrica, en base a las 500 simulaciones realizadas. La accuracy promedio fue de 0.9667. El F1 score promedio fue de 0.9668. La precisión promedio fue de 0.9433, y el recall promedio fue de 0.9915. A continuación, se presentan los mapas de calor que muestran cómo varían las métricas de accuracy, F1 score, precision y recall conforme se modifican los valores de lambda y alpha.

3.3.3. Modelo Logístico Elastic-Net

Como anteriormente se mencionó, al tercer modelo, Elastic-Net, no realizó la eliminación de variables redundantes durante el preprocesamiento. En cambio, se enfocó en la regularización de los parámetros lambda y alpha para manejar la multicolinealidad y seleccionar características relevantes. La eliminación de outliers se realizó utilizando la distancia de Mahalanobis robusta y el método bootstrap. Se realizaron 500 simulaciones para cada combinación de lambda y alpha, resultando en un total de 12500 simulaciones.

Cuadro 6: Promedio del resultado de las simulaciones para el modelo Elastict-Net con Lamda = 0.1 y Alpha = 0.1

Métrica	**Promedio**	**Mínimo**	**Máximo**
Accuracy	0.909552	0.903121004	0.915316621
F1 Score	0.904513	0.897972411	0.910255161
Precision	0.933459	0.922092591	0.944945663
Recall	0.877386	0.873033677	0.882116799

Para esta ocasión en particular, se mostrará los resultados obtenidos de la mejor combinación o regularización de los parámetros Lambda y Alpha, la cual fue $\lambda = 0{,}1$ y $\alpha = 0{,}1$ observados en el Cuadro 6. Este resultado fue obtenido gracias a el entrenamiento del modelo en todas combinaciones posibles de Lambda y alfa, las cuales variaron entre los valores 0.1, 0.25, 0.5, 0.75 y 1 para Lambda y 0.1, 0.5, 0.7, 0.9 y 1, para los posibles valores de Alpha. Por ello, se obtuvo una accuracy promedio de 0.9096. El F1 score promedio fue de 0.9045, mientras que la precisión promedio alcanzó 0.9335, y el recall promedio fue de 0.8774. A continuación, en la Figura 1 se presentan los mapas de calor que muestran cómo varían las métricas de accuracy, F1 score, precision y recall conforme se modifican los valores de los parámetros lambda y alpha .

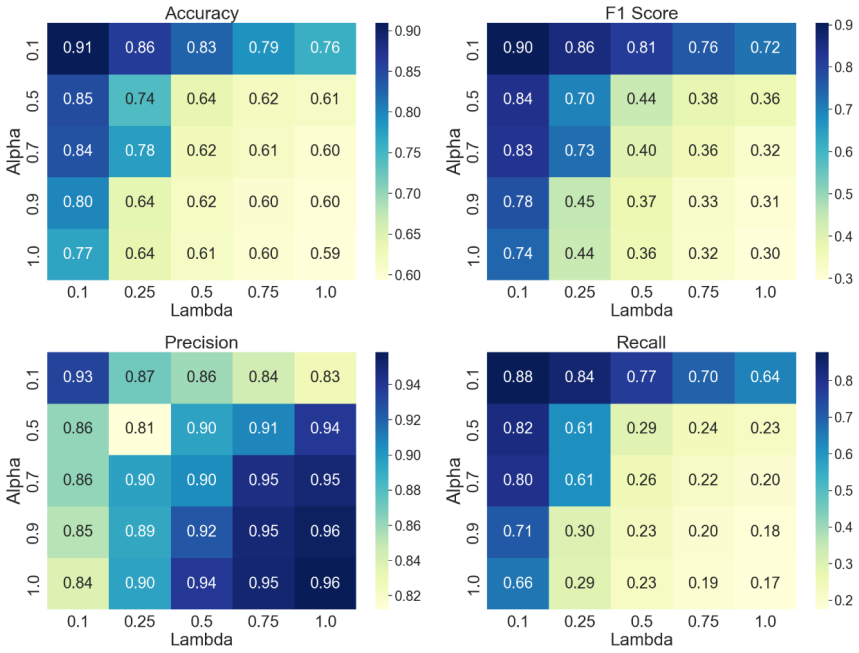

Figura 1: Heatmap de Métricas del Modelo Elastic-Net.

3.4. Discusión

En esta sección, se compararán los tres modelos (SVM con kernel linear, SVM con kernel RBF y Elastic-Net) en base a cada una de las métricas (accuracy, F1 score, precision y recall). Se presentarán gráficos boxplot para cada métrica, mostrando los resultados de las simulaciones realizadas. El análisis se enfocará en los beneficios y falencias de cada métrica al comparar los resultados de los tres modelos.

El accuracy mide la proporción de predicciones correctas realizadas por el modelo en comparación con el total de predicciones. Es una métrica sencilla y ampliamente utilizada, pero puede ser engañosa en conjuntos de datos desbalanceados, donde una alta accuracy no necesariamente indica un buen rendimiento del modelo en identificar correctamente todas las clases. El gráfico boxplot de accuracy que se muestra en la Figura 2, nos dice que el modelo SVM con kernel RBF tiene la mediana más alta y una menor dispersión en comparación con los modelos SVM con kernel linear y Elastic-Net. Esto indica que el modelo SVM con kernel RBF es más consistente y preciso en sus predicciones. Sin embargo, la alta accuracy puede ser parcialmente atribuida a la capacidad del modelo RBF de manejar relaciones no lineales complejas, lo que puede no ser siempre necesario o deseable en escenarios más simples. El modelo SVM con kernel linear también muestra un rendimiento sólido con una mediana elevada, aunque con una dispersión mayor que el RBF, lo que sugiere una variabilidad ligeramente mayor en sus predicciones.

El modelo Elastic-Net, aunque tiene una menor accuracy promedio, muestra una dispersión más amplia, lo que sugiere que sus resultados pueden variar más entre diferentes simulaciones. Esta variabilidad puede ser una desventaja en aplicaciones donde la consis-

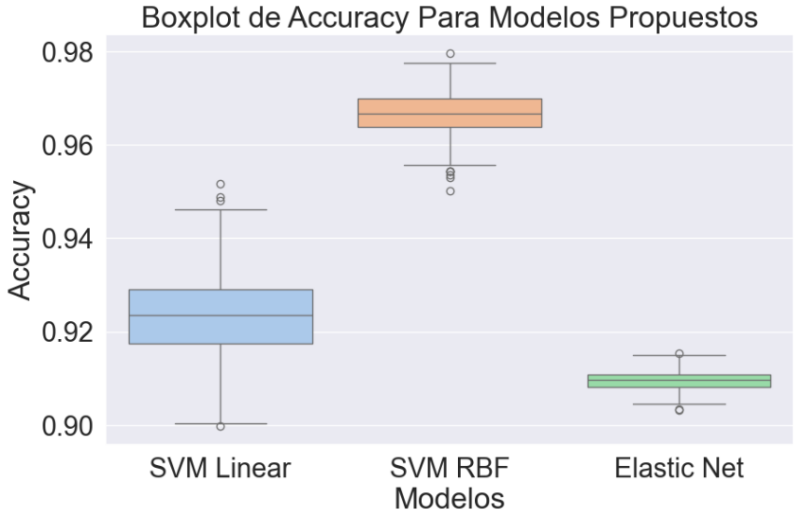

Figura 2: Boxplot de Métrica Accuracy Para Modelos Propuestos.

tencia es crucial. Sin embargo, la capacidad del Elastic-Net para manejar multicolinealidad y seleccionar características relevantes lo hace útil en contextos específicos, aunque no tan robusto como los modelos SVM en este caso particular.

El F1 score es una métrica que combina la precisión y el recall en una sola medida, balanceando ambos aspectos. Es especialmente útil en situaciones donde existe un desbalance entre las clases, ya que proporciona una medida más equilibrada del rendimiento del modelo. El gráfico boxplot de F1 score de la Figura 3, indica que el modelo SVM con kernel RBF nuevamente presenta la mediana más alta, reflejando su capacidad superior para balancear precisión y recall. Esta alta F1 score es particularmente beneficiosa en el contexto de deserción de clientes, donde es crucial identificar correctamente tanto los casos positivos como minimizar los falsos positivos. La consistencia del modelo RBF se demuestra por su baja dispersión, lo que sugiere que el modelo puede ser confiable en diferentes escenarios.

El modelo SVM con kernel linear muestra una dispersión menor en comparación con Elastic-Net, sugiriendo una consistencia razonable en sus predicciones. Sin embargo, aunque tiene una buena mediana de F1 score, es superado por el RBF en términos de rendimiento global. Elastic Net, aunque con una mediana menor, muestra una mayor variabilidad en sus resultados, lo que podría impactar en su fiabilidad. Esta variabilidad sugiere que, aunque Elastic-Net puede ser útil en ciertos contextos, su rendimiento es menos predecible en comparación con los modelos SVM.

La precisión mide la proporción de verdaderos positivos entre las predicciones positivas realizadas por el modelo. Es especialmente importante en situaciones donde el costo de los falsos positivos es alto.El gráfico boxplot de precisión de la Figura 4 muestra que el modelo SVM con kernel linear tiene una mediana de precisión ligeramente superior a la del modelo Elastic-Net, aunque el SVM con kernel RBF tiene una menor dispersión y una mediana menor en comparación con los otros dos modelos. Esto sugiere que el modelo SVM

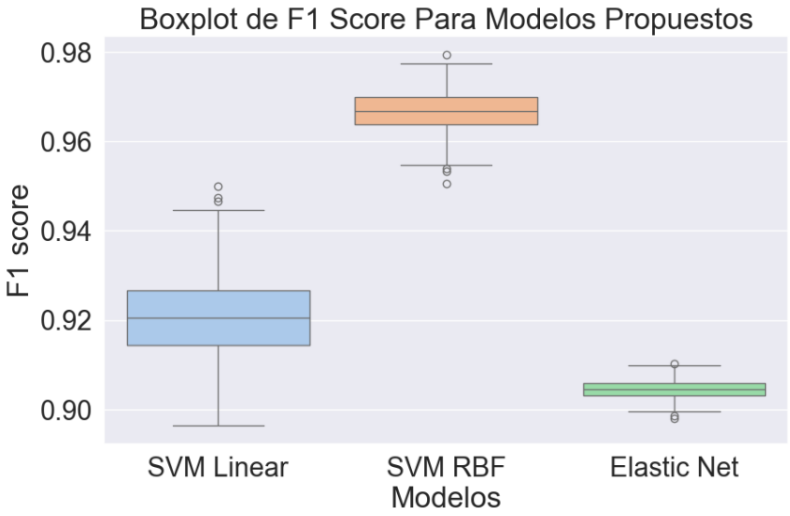

Figura 3: Boxplot de Métrica F1 Score Para Modelos Propuestos.

con kernel linear es más confiable en términos de precisión en comparación con Elastic-Net y RBF. La alta precisión del modelo SVM linear es beneficiosa en aplicaciones donde es crucial minimizar los falsos positivos, aunque esto puede venir a costa de un menor recall.

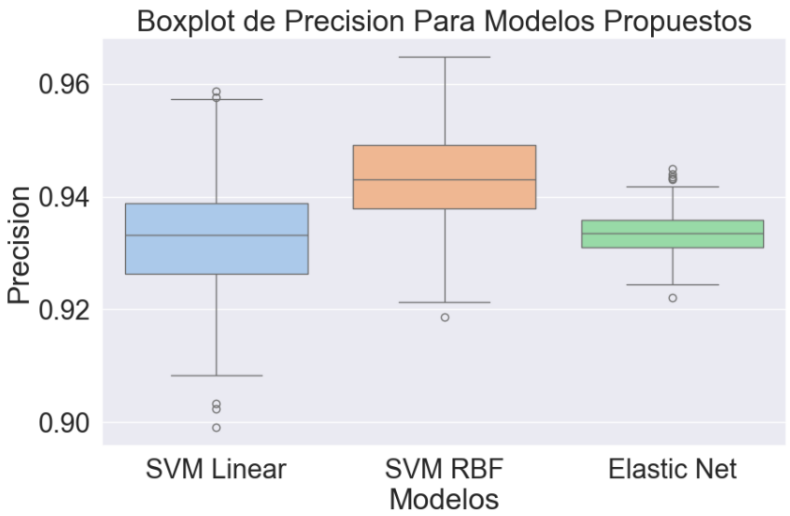

Figura 4: Boxplot de Métrica Precision Para Modelos Propuestos.

El Elastic-Net muestra una mayor variabilidad en precisión, lo que puede ser una desventaja en escenarios donde se requiere consistencia. La precisión del modelo RBF, aunque no tan alta como la del SVM linear, sigue siendo competitiva y su baja dispersión sugiere que es más confiable en general. Esta consistencia es particularmente importante en aplicaciones financieras donde los costos de falsos positivos pueden ser altos, y un modelo preciso puede ayudar a reducir estos costos al minimizar las predicciones incorrectas de deserción.

El recall mide la proporción de verdaderos positivos identificados correctamente entre todos los positivos reales. Es crucial en situaciones donde es importante identificar todos los casos positivos, incluso si eso significa tener más falsos positivos. El gráfico boxplot de recall observado en la Figura 5, nos dice que el modelo SVM con kernel RBF tiene la mediana más alta y una dispersión muy baja, indicando una capacidad superior para identificar correctamente los casos positivos. Esto es crucial en el contexto de la deserción de clientes, donde es importante identificar a todos los clientes en riesgo para poder tomar acciones preventivas. La alta mediana de recall del modelo RBF sugiere que es muy efectivo para minimizar los falsos negativos, asegurando que la mayoría de los clientes en riesgo sean correctamente identificados.

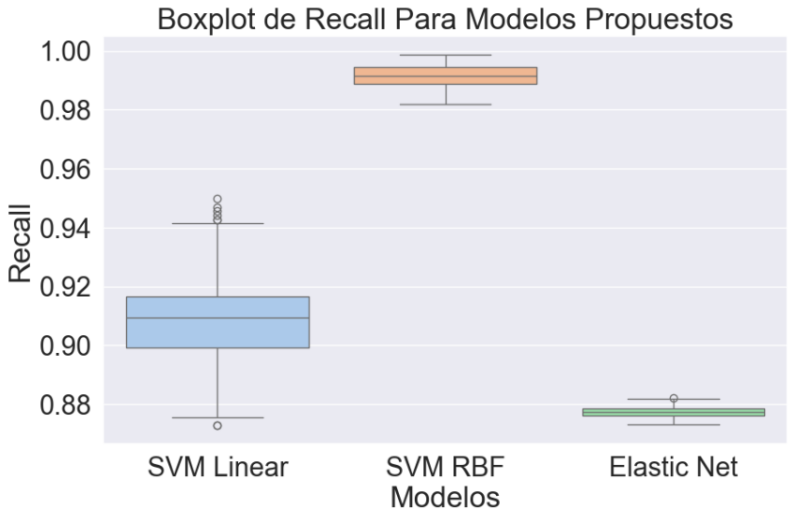

Figura 5: Boxplot de Métrica Recall Para Modelos Propuestos.

El modelo SVM con kernel linear presenta una mediana ligeramente inferior pero aún muestra una buena consistencia, lo que indica un rendimiento razonablemente sólido en términos de recall. Sin embargo, su mayor dispersión en comparación con el RBF sugiere que podría ser menos fiable en algunos escenarios. El modelo Elastic-Net tiene una mayor variabilidad en los resultados de recall, sugiriendo que es menos fiable en comparación con los otros dos modelos. Esta variabilidad es una desventaja significativa en aplicaciones donde es crucial minimizar los falsos negativos, como en la prevención de la deserción de clientes en una entidad financiera.

4. Conclusión

El análisis de los resultados obtenidos de los tres modelos de clasificación, aplicados en el contexto de una entidad financiera que busca prevenir la deserción temprana de clientes, demuestra claramente que el modelo SVM con kernel RBF es el más efectivo en términos de rendimiento global. Este modelo no solo alcanza una alta accuracy promedio de 0.9667, sino que también sobresale en el F1 score, precision y recall. La capacidad del SVM con

kernel RBF para manejar relaciones no lineales entre las variables es crucial para su éxito, ya que permite una mejor separación de las clases en un espacio de mayor dimensión. Esto resulta en una mediana alta y una baja dispersión en las simulaciones, lo que subraya la consistencia y precisión de este modelo. En comparación, el SVM con kernel linear, aunque efectivo, no alcanza los mismos niveles de rendimiento y muestra una mayor variabilidad en sus resultados.

El modelo SVM con kernel linear, con una accuracy promedio de 0.9235, también ofrece un rendimiento sólido, especialmente en términos de precisión, donde su mediana es ligeramente superior. Sin embargo, en métricas clave como el F1 score y el recall, es superado por el modelo RBF. Esto sugiere que, mientras que el SVM linear es una opción viable, especialmente en escenarios donde las relaciones entre variables son más simples, su capacidad para generalizar y manejar complejidades en los datos es limitada en comparación con el RBF. El modelo Elastic-Net, por otro lado, aunque presenta una precisión competitiva de 0.9335, su variabilidad y menor rendimiento en recall lo hacen menos confiable. La regularización mediante Elastic-Net es beneficiosa para la selección de características y el manejo de la multicolinealidad, pero en este caso, la variabilidad en los resultados podría afectar negativamente su aplicabilidad en escenarios donde la consistencia es crucial.

La entidad financiera se beneficiaría significativamente de la implementación del modelo SVM con kernel RBF para la detección de deserción temprana de clientes. Este modelo no solo proporciona la mayor precisión en la identificación de clientes en riesgo de desertar, sino que también mantiene una alta sensibilidad, asegurando que la mayoría de los clientes en riesgo sean identificados correctamente. La capacidad del modelo RBF para manejar relaciones complejas entre las variables predictoras es particularmente valiosa en el contexto financiero, donde múltiples factores socio-demográficos, financieros y crediticios pueden interactuar de maneras no lineales para influir en la deserción. Implementar este modelo permitirá a la entidad financiera diseñar estrategias de retención más efectivas y proactivas, reduciendo así las pérdidas financieras y mejorando el valor de la marca.

Consecuentemente, el modelo SVM con kernel RBF no solo identifica correctamente los verdaderos positivos, sino que también es el mejor en identificar los falsos negativos. Esto es crucial para este caso de uso, ya que es importante minimizar la cantidad de falsos negativos, es decir, asegurar que el modelo no se equivoque al predecir que un cliente no va a desertar cuando en realidad sí lo hará. Este tipo de error es el más crítico para la entidad financiera, ya que la falta de identificación de clientes en riesgo puede llevar a pérdidas realmente significativas. Por otro lado, cabe resaltar que los modelos Elastic Net y SVM linear tiene una ventaja sobre el modelo RBF, debido a su capacidad para identificar las variables relevantes, lo que puede ser muy útil para comprender mejor los factores que contribuyen a la deserción y para diseñar estrategias de retención más informadas.

Para mejorar aún más la capacidad predictiva de los modelos, se recomienda explorar combinaciones adicionales de hiperparámetros y considerar la implementación de técnicas de deep learning que puedan capturar patrones más complejos en los datos. Por otro lado, aunque el rendimiento es ligeramente inferior al de los modelos SVM, el Elastic-Net

no debería descartarse por completo, ya que es una opción valiosa cuando se necesita regularización para manejar datos complejos y prevenir el sobreajuste. Por lo cual, es una herramienta poderosa para tareas de clasificación, ofreciendo una combinación efectiva de precisión y capacidad de detección de verdaderos positivos, con la flexibilidad adicional de ajustar los parámetros de regularización para adaptarse a diferentes necesidades y condiciones de los datos, teniendo la capacidad de brindarnos claridad sobre la contribución marginal que tiene cada variable al momento de predecir la deserción de un cliente.

Por otra parte, también sería beneficioso establecer un sistema de monitoreo continuo para evaluar y ajustar los modelos, asegurando que mantengan su eficacia a medida que evolucionan los datos y las condiciones del mercado. Investigaciones futuras también deberían centrarse en la mejora de los métodos de ingeniería de características y técnicas de preprocesamiento, incluyendo la experimentación con nuevos métodos de detección y manejo de outliers. Finalmente, la implementación de enfoques de ensamblado que combinan múltiples modelos podría ofrecer un rendimiento superior al aprovechar las fortalezas individuales de cada técnica.

Referencias

Abdullah, A. S., Akash, K., and ShaminThres, J. (2021), *Sentiment analysis of movie reviews using support vector machine classifier with linear kernel function*, Springer.

Amuda, K. and Adeyemo, A. (2019), "Customers' churn prediction in financial institutions using artificial neural network," *Journal of Retailing and Consumer Services*, 47, 275–285.

Ashraf, R. (2024), "Bank Customer Churn Prediction Using Machine Learning Framework," *Journal of Applied Finance & Banking*, 14, 1–14.

Baesens, B., Roesch, D., and Scheule, H. (2015), *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS*, Wiley.

Behera, B., Kumaravelan, G., and Kumar, P. (2019), "Performance evaluation of deep learning algorithms in biomedical document classification," in *2019 11th international conference on advanced computing (ICoAC)*, IEEE.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992), "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*.

Burez, J. and Van den Poel, D. (2009), "Handling class imbalance in customer churn prediction," *Expert Systems with Applications*, 36, 4626–4636.

Cortes, C. and Vapnik, V. (1995), "Support-vector networks," *Machine learning*, 20, 273–297.

Croux, C. and Haesbroeck, G. (2000), "Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies," *Biometrika*, 87, 603–618.

De Caigny, A., C. K. D. B. K. (2020), "Bank Customer Churn Prediction," *International Journal of Bank Marketing*, 14, 23–29.

Dogaru, R. and Dogaru, I. (2018), "Optimized Super Fast Support Vector Classifiers Using Python and Acceleration of RBF Computations," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE.

Efron, B. (1979), "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, 7, 1–26.

Guyon, I. and Elisseeff, A. (2003), "An introduction to variable and feature selection," *Journal of machine learning research*, 3, 1157–1182.

Hastie, T., Tibshirani, R., and Wainwright, M. (2020), *Statistical Learning with Sparsity: The Lasso and Generalizations*, Boca Raton, FL: CRC Press.

Hoerl, A. E. and Kennard, R. W. (1970), "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, 12, 55–67.

Hubert, M., Rousseeuw, P. J., and Verdonck, T. (2008), "High-breakdown robust multivariate methods," *Statistical Science*, 23, 92–119.

Joshi, A. V. (2022), *Support vector machines*, Springer.

Ledoit, O. and Wolf, M. (2003), "Improved estimation of the covariance matrix of stock returns with an application to portfolio selection," *Journal of Empirical Finance*, 10, 603–621.

— (2004), "Honey, I shrunk the sample covariance matrix," *The Journal of Portfolio Management*, 30, 110–119.

Lemmens, A. and Croux, C. (2016), "Bagging and boosting classification trees to predict churn," *Journal of Marketing Research*, 43, 276–286.

Li, Y. and Zhang, Q. (2024), "Customer churn prediction in banking industry: A case study of Chinese commercial banks," *Journal of Business Research*, 90, 129–138.

Mahalanobis, P. C. (1936), "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences of India*, 2, 49–55.

Nyitrai, T. and Virág, M. (2019), "The effects of handling outliers on the performance of bankruptcy prediction models," *Socio-Economic Planning Sciences*, 67, 34–42.

Parisi, L., Ma, R., Zaernia, A., and Youseffi, M. (2021), "M-ark-support vector machine for early detection of Parkinson's disease from speech signals," *International Journal of Mathematical and Computational Simulation*.

Puth, M.-T., Neuhäuser, M., and Ruxton, G. D. (2015), "Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits," *Animal Behaviour*, 102, 77–84.

Rajola, F. (2019), *Customer relationship management in the financial industry organizational processes and technology innovation*, Springer.

Roncaglioni, A., Piclin, N., Pintore, M., and Benfenati, E. (2008), "Binary classification models for endocrine disrupter effects mediated through the estrogen receptor," *SAR and QSAR in Environmental Research*, 19, 697–733.

Rousseeuw, P. J. and Van Driessen, K. (1999), "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, 41, 212–223.

Sakshi, A., Benifa, J., and Seba, P. (2024), "Supervised Learning Models for Diagnosing Severity of Cirrhosis Disease," *Handbook of AI-Based Models in Healthcare*, 1, 45–59.

Spearman, C. (1904), "The proof and measurement of association between two things," *The American Journal of Psychology*, 15, 72–101.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, 58, 267–288.

Toloşi, L. and Lengauer, T. (2011), "Classification with correlated features: unreliability of feature ranking and solutions," *Bioinformatics*, 27, 1986–1994.

Tran, H. D., Le, N., and Nguyen, V.-H. (2022), "Customer Churn Prediction in the Banking Sector Using Machine Learning-Based Classification Models," *Interdisciplinary Journal of Information, Knowledge, and Management*, 17, 65–80.

Verbeke, W., M. D. M. C. B. B. (2012), "Building comprehensible customer churn prediction models with advanced rule induction techniques," *Expert Systems with Applications*, 38, 2354–2364.

Wolf, M. and Ledoit, O. (2011), "Nonlinear shrinkage estimation of large-dimensional covariance matrices," *The Annals of Statistics*, 39, 365–411.

Yadav, S. and Shukla, S. (2016), "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International conference on advanced computing (IACC)*, IEEE.

Yu, L. and Liu, H. (2004), "Efficient feature selection via analysis of relevance and redundancy," *The Journal of Machine Learning Research*, 5, 1205–1224.

Zheng, A. and Casari, A. (2018), *Feature engineering for machine learning: principles and techniques for data scientists*, .°'Reilly Media, Inc.".

Zou, H. and Hastie, T. (2005), "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 301–320.