Optimización de la detección de fraudes de contracargos con Tarjetas de Crédito: un enfoque de calibración de probabilidades en esquemas combinatorios

Juan David Tamayo Giraldo
Mónica Sadday Carmona Higuita

Tesis de maestría presentada para optar al título de Magíster en Métodos Cuantitativos para Economía y Finanzas

Director
Camilo Restrepo-Estrada, Doctor (PhD) en Ciencias

Universidad de Antioquia
Facultad de Ciencias Económicas
Maestría en Métodos Cuantitativos para Economía y Finanzas
Medellín, Antioquia, Colombia
2024

Cita	(Tamayo Giraldo & Carmona Higuita, 2024)
Referencia **Estilo APA 7 (2020)**	Tamayo Giraldo, J. D., & Carmona Higuita, M. S. (2024). *Optimización de la detección de fraudes de contracargos con Tarjetas de Crédito: un enfoque de calibración de probabilidades en esquemas combinatorios* [Tesis de maestría]. Universidad de Antioquia, Medellín, Colombia.

creative commons CC BY NC SA

Maestría en Métodos Cuantitativos para Economía y Finanzas, Cohorte III.

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Biblioteca Carlos Gaviria Díaz

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Tabla de contenido

Resumen....8

Abstract....9

Introducción....10

1 Marco teórico....13

1.1 Fraude de contracargo....13

1.2. Modelos de detección basados en datos....14

1.3. Desafíos de la detección de fraude....15

1.4. Esquemas de combinación de clasificadores....16

1.5. Calibración de probabilidades....20

2 Metodología....22

2.1 Datos....23

2.2 Preprocesamiento de datos....24

2.2.1 Ingeniería de características....25

2.2.2 Selección de características....25

2.3 Entrenamiento de clasificadores individuales....26

2.3.1 Calibración de probabilidades....27

2.4 Combinación de clasificadores....28

2.4.1 Stacking....28

2.4.2 Hard voting....28

2.4.3 Soft voting....29

3 Resultados....31

3.1. Calibración de modelos....31

3.2 Desempeño modelos de clasificación y esquemas de combinación....33

4 Discusión....39

5 Conclusiones y recomendaciones 42

Referencias 44

Anexos 51

Lista de tablas

Tabla 1 Descripción de las variables 24

Tabla 2 Lista de modelos e hiperparámetros 26

Tabla 3 Resultados Clasificadores..... 33

Tabla 4 *Train-Test Validation Suite* ..... **Error! Bookmark not defined.**

Tabla 5 Métricas de evaluación para clasificadores y métodos combinados..... 56

Lista de figuras

Figura 1 Proceso metodológico 22

Figura 2 Diagramas de confiabilidad para modelos sin calibrar vs modelos calibrados 32

Figura 3 Comparación de F1 Score entre los modelos calibrados y no calibrados 36

Figura 4 Comparación de la precisión entre los modelos calibrados y no calibrados 37

Figura 5 Comparación de la sensibilidad entre los modelos calibrados y no calibrados 38

Figura 6 Matriz de confusión – Probabilidades no calibradas 57

Figura 7 Matriz de confusión – Probabilidades calibradas 58

Siglas, acrónimos y abreviaturas

AB	AdaBoost
GNB	Gaussian Naive Bayes
GRU	Gated Recurrent Unit
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LR	Logistic Regression
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
NN	Neural Net
PCA	Principal Component Analysis
PIN	Personal Identification Number
RF	Random Forest
SMOTE	Synthetic Minority Oversampling Technique
SMOTE-ENN	Synthetic Minority Oversampling Technique with Edited Nearest Neighbor
SVM	Support Vector Machine
TPE	Tree-structured Parzen Estimator
WOE	Weight of Evidence
XGB	Extreme Gradient Boosting
XGBOOST	Extreme Gradient Boosting

Resumen

Las tarjetas de crédito se han consolidado como uno de los métodos de pago más utilizados en transacciones de comercio electrónico. Este fenómeno ha impulsado un aumento en las disputas contra las entidades financieras, dirigidas a obtener reembolsos de manera fraudulenta. Como contramedida, se han desarrollado sistemas de detección de fraudes que aprovechan la capacidad de los modelos de aprendizaje estadístico para identificar relaciones complejas en los datos. Los esquemas de combinación de clasificadores de fraude superan los modelos individuales de clasificación mediante una construcción de predicciones sólidas a partir de las predicciones más débiles de dos o más modelos base. En este contexto, se han propuesto diversos diseños metodológicos, destacando el *stacked generalization* (*stacking*), *majority voting* (*hard voting*) y *weighted average probability* (*soft voting*). A pesar de estos avances, es crucial abordar la calibración de probabilidades para evaluar la certeza asociada a las predicciones de los modelos base. Este estudio analiza el impacto de la calibración de probabilidades en el rendimiento de los tres esquemas de combinación de clasificadores mencionados. Los resultados revelan que el *stacking* exhibe el mejor desempeño en términos del *F1 Score*. El *soft voting* y *hard voting* aumentan la sensibilidad mientras disminuyen la precisión consiguiendo un aumento en el *F1 Score* del 5,29% y 6,02%, respectivamente. Este trabajo contribuye a la comprensión de la importancia de la calibración de probabilidades en los esquemas de combinación de clasificadores, destacando la eficacia del *stacking* como la opción más sobresaliente en este contexto.

Palabras clave: aprendizaje automático, calibración de clasificadores, desbalance de clases, detección de fraude, métodos combinatorios

Abstract

Credit cards have become one of the most widely used payment methods in online transactions. This fueled an increase in disputes against financial institutions, aiming to obtain refunds through fraud. In response, fraud detection systems have been developed, leveraging statistical learning models' capabilities to untangle complex patterns in data. Fraud classifier combinations surpass individual classification models by weaving robust predictions from the weaker forecast of two or more base models. In this context, several methodological designs have been proposed, including notable stacked generalization (stacking), majority voting (hard voting), and weighted average probability (soft voting). Despite these advancements, it is crucial to address probability calibration to assess the certainty tied to the base of model's predictions. This study examines the impact of probability calibration on the efficacy of the afore mentioned classifier combination schemes. The results underscore that stacking exhibits the best performance in terms of F1 Score. Both soft voting and hard voting increase sensitivity while slight decrease in precision achieving an increase in F1 score of 5.29% and 6.02%, respectively. This research enriches our understanding of the important role that probability calibration plays in classifier combination schemes, highlighting the effectiveness of stacking as the most outstanding performer in this context.

*Keywords***:** machine learning, classifier calibration, class imbalance, fraud detection, combinatorial methods

Introducción

La transformación digital, acelerada por la crisis sanitaria del COVID-19, ha impulsado el desarrollo y la adopción de nuevas tecnologías digitales generando cambios que abarcan desde los modelos de negocio y las formas de trabajo, hasta los sistemas de pagos y los comportamientos de los consumidores (Amankwah-Amoah et al., 2021; De' et al., 2020). Sin embargo, este crecimiento de la demanda digital conlleva desafíos, especialmente en la gestión de la seguridad y el control de las transacciones en línea. La necesidad de medidas más estrictas para prevenir y detectar el fraude se ha vuelto imperativa, dado que, a pesar de ventajas como la disminución de los costos operativos y de personal, las innovaciones digitales también han incrementado el riesgo de vulnerar la seguridad y el control de las transacciones (Bajada & Trayler, 2015). Las empresas han debido ajustar sus operaciones para manejar el alto flujo de transacciones digitales, mientras que los consumidores demandan interacciones más ágiles y confiables. En el caso de las tarjetas de crédito, por ejemplo, cuyo uso se ha masificado con el crecimiento de los *e-commerce*, se calcula que las perdidas por fraude, en escenarios que aún distan de ser completamente efectivos, son multimillonarias (Randhawa et al., 2018).

Para desarrollar métodos para la detección de fraude en línea con tarjetas de crédito, se han empleado diversas tecnologías basadas en el aprendizaje automático, el aprendizaje profundo, el *big data* y la inteligencia computacional (Carta et al., 2019), en donde se analizan patrones de gasto y uso que permiten detectar inconsistencias o desviaciones significativas con respecto a los patrones habituales de los consumidores (Srivastava et al., 2008). Sin embargo, estas técnicas presentan desafíos y limitaciones comunes como la latencia de verificación, el desbalance y la deriva de los datos.

En las investigaciones recientes se exploran métodos más avanzados como las técnicas semi-supervisadas (Fajardo et al., 2021; Lebichot et al., 2021) para mitigar los problemas descritos. También se proponen métodos que combinan las predicciones de dos o más modelos de aprendizaje estadístico, ya que, se ha demostrado que alcanzan un mejor desempeño en comparación con los modelos individuales (Baker et al., 2022; Karthik et al., 2022; Mienye & Sun, 2023; Olowookere & Adewale, 2020; Paldino et al., 2022; Rakhshaninejad et al., 2022; Soleymanzadeh et al., 2022). Además, se han utilizado diferentes estrategias para hacer frente al problema del desbalance de las clases. Entre éstas se encuentra el submuestreo de la clase mayoritaria (Rakhshaninejad et al., 2022;

Soleymanzadeh et al., 2022), el sobremuestreo de la clase minoritaria (Karthik et al., 2022), el sobremuestreo de la clase minoritaria utilizando SMOTE (Baker et al., 2022), métodos híbridos como el SMOTE-ENN (Esenogho et al., 2022; Mienye & Sun, 2023) y la aplicación de costos mayores a las malas clasificaciones sobre la clase minoritaria (Olowookere & Adewale, 2020). Una particularidad que se encuentra en estos trabajos es que los métodos combinatorios sólo se comparan con los clasificadores individuales y no entre esquemas de combinación; por lo que no se determina el esquema más adecuado para cada conjunto de datos.

Otro aspecto que ha recibido poca atención, pero que es fundamental en aplicaciones prácticas, es el costo diferencial de los errores de clasificación y la asignación óptima de recursos. El fraude con tarjetas de crédito es un problema asimétrico, pues el error de clasificar una transacción fraudulenta como legítima tiene un costo mucho mayor que hacer lo contrario (Bahnsen et al., 2013). Ignorar el riesgo o la gravedad del evento y tratar de controlar el flujo completo de casos es más ineficiente que seleccionar los casos con mayor probabilidad de ser fraudulentos (Brant & Hobæk Haff, 2023)[1], lo que puede generar altos costos para las instituciones, especialmente si el sistema produce muchos falsos positivos, es decir, identifica erróneamente transacciones legítimas como fraudulentas (Makki et al., 2019). La gestión adecuada del riesgo se vuelve crucial para optimizar la eficiencia de los sistemas de prevención y detección de fraudes, minimizando al mismo tiempo los costos asociados.

La presente investigación propone evaluar el rendimiento de tres métodos de combinación de clasificadores, *stacked generalization* (*stacking*), *majority voting* (*hard voting*) y *weighted average probability* (*soft voting*), incorporando una noción de criticidad entendida como la mayor probabilidad de ocurrencia al aplicar la calibración sobre los modelos. Para ello, se emplea un método de conjunto que combina las predicciones de los distintos clasificadores e incorpora una política basada en criterios probabilísticos y no sólo de ocurrencia del evento. Este método se aplica en un marco conceptual que integra prácticas de desarrollo eficaces, puesto que adopta una metodología estándar a la industria, como CRISP-DM (Wirth & Hipp, 2000), especialmente en problemas relacionados con la minería de datos. Además, incluye etapas de preprocesamiento que

[1] Brant y Hobæk Haff (2023) proponen que el objetivo cuando hay restricciones de entrada, no es sólo seleccionar las k instancias más relevantes, sino también clasificarlas lo mejor posible. En el contexto del fraude, esto corresponde a seleccionar k casos de modo que el mayor número posible de casos positivos sean positivos, y que, si se ordenan según su probabilidad prevista de fraude, el mayor número posible de casos positivos seleccionados se clasifican por encima de los casos negativos seleccionados.

han demostrado optimizar el aprendizaje y mejorar el desempeño de los clasificadores (Cai et al., 2018; C. Zhang et al., 2018).

Las principales contribuciones de este trabajo se resumen en un *pipeline* de ingeniería reproducible y eficiente para datos tabulares, el uso de un método informado para optimizar los hiperparámetros y el umbral de significancia (para evitar usar decisiones arbitrarias, e.g., el estándar 0.5); y la aplicación de métricas apropiadas para escenarios reales donde la clase positiva requiere mayor peso y son basadas en la probabilidad, no en la frecuencia. Con esta aproximación, se busca ofrecer un marco de trabajo que ayude a reducir el filtrado de casos para atención y análisis en escenarios de fraude con datos altamente desbalanceados.

Este es un trabajo cuantitativo basado en datos abiertos de una empresa de comercio electrónico y la metodología se resumirá en tres secciones: preprocesamiento de datos, entrenamiento y calibración de clasificadores. Al final, se realiza una evaluación del método de conjunto.

El documento se organiza de la siguiente manera: en la sección 1 se presenta una revisión del estado del arte dentro del marco teórico del fraude de contracargo; en la sección 2 se describe la metodología propuesta para desarrollar el método de conjunto basado en la calibración; en la sección 3 se muestran los resultados de la aplicación del método como un estudio cuantitativo con datos abiertos; en la sección 4 y 5 se ofrecen las discusiones, conclusiones y recomendaciones derivadas del trabajo realizado, y por último, en las secciones finales se relaciona la bibliografía junto con los anexos correspondientes.

1 Marco teórico

El fraude, entendido como el acto deliberado de engañar para obtener beneficios económicos o personales ilícitos, presenta varias modalidades como el fraude electrónico, el fraude bancario, el fraude de seguros, el fraude en tarjetas de crédito y el fraude de identidad. Según el informe del "2023 Global Ecommerce Payments & Fraud Survey", las tarjetas se han convertido en uno de los medios de pago más utilizados por los consumidores al realizar compras a través de plataformas de comercio electrónico a nivel global (Cybersource et al., 2023). Sin embargo, este aumento en la popularidad de transacciones con tarjeta se ha visto acompañado por un incremento en los ataques de fraude asociados a este método de pago. Entre las modalidades más utilizadas se encuentran: ingeniería social (*phishing, pharming* y *whaling*), contracargos (*chargeback*), prueba de tarjetas y robo de identidad. Los delincuentes aprovechan las transacciones en línea para realizar estas modalidades de fraude, desde la clonación de tarjetas hasta la realización de compras fraudulentas. El fraude de contracargo se ha destacado como una problemática significativa que afecta tanto a los comerciantes como a los consumidores (Cybersource et al., 2023). En este trabajo se centrará la atención en el fraude de contracargo.

1.1 Fraude de contracargo

El fraude de devolución de cargo ocurre cuando un cliente compra algo en línea usando su tarjeta de crédito o una plataforma de pago en línea y luego pide un reembolso al banco o a la plataforma sin devolver el producto o servicio (Liu & Lee, 2022). Saporta y Maraney (2022) identifican cinco tipos distintos de fraude de contracargo, considerando la intención y motivación inicial del comprador o pagador: el derivado de errores comunes, también conocido como *genuine mistake*, por ejemplo, sí hay discrepancia entre el nombre del comercio en la información de facturación y el del sitio web es posible que el consumidor crea que la compra fue fraudulenta; el *family fraud* que es el fraude cometido por familiares cuando realizan compras en línea sin el consentimiento del titular; el *buyer's remorse* que surge cuando el consumidor se arrepiente de una compra genuina, ya sea por falta de capacidad de pago o por darse cuenta más tarde de que no necesita los productos; el *customer resentment*, el cual se relaciona con el descontento del consumidor cuando el producto no se entrega en condiciones adecuadas o se retrasa

significativamente; y finalmente el *mens rea* (*guilty mind*) que implica intenciones maliciosas del consumidor desde el momento de la compra para obtener productos del comercio de manera gratuita (Saporta & Maraney, 2022).

Para detectar y prevenir el fraude en tarjetas se han creado sistemas de detección que son conformados por distintos niveles de evaluación. Pozzolo y colaboradores (2018) expusieron un sistema de detección compuesto por cinco niveles de control que se ejecutan de manera secuencial, donde los cuatro primeros suceden en tiempo real y definen si se debe aprobar o no un intento de pago y, el último, son tareas de análisis sobre las alertas que haya generado el sistema (Pozzolo et al., 2018). El primer control corresponde a la terminal, donde se realizan validaciones básicas de seguridad, por ejemplo, sobre el *Personal Identification Number* (PIN), el número de intentos de compra, el estado de la tarjeta (activa o bloqueada), el balance disponible o el límite de gasto. El segundo son las reglas de bloqueo transaccional, donde se evalúan condiciones lógicas construidas a partir de la información proporcionada en el contexto de la operación, por ejemplo, "si el monto es mayor que USD 200, entonces rechace la operación". El tercer control incluye las reglas de puntuación, las cuales son condiciones lógicas que hacen uso de información histórica de comportamiento del consumidor (número de transacciones realizadas en la última hora) para emitir un *score* donde, entre mayor sea su valor, mayor es la probabilidad de que se trate de un fraude. El cuarto son los modelos basados en datos, donde se hacen uso de gran cantidad de datos históricos para construir modelos estadísticos que estimen la probabilidad de fraude de la operación[2]. Finalmente, los investigadores, que se encargan de construir las reglas de bloqueo transaccional y las reglas de puntuación del sistema analizan las alertas emitidas por las reglas de puntuación y los modelos estadísticos para determinar si estas corresponden a verdaderos casos de fraude o, por el contrario, representan una falsa alarma.

1.2. Modelos de detección basados en datos

Con respecto a los modelos basados en datos, se han propuesto un sinnúmero de metodologías que hacen uso de técnicas de aprendizaje estadístico supervisado y no supervisado. El enfoque supervisado implica construir un modelo estadístico para predecir o estimar un

[2] El valor estimado de la probabilidad es comparado contra un umbral y, en caso de la probabilidad estimada sea mayor o igual al umbral se rechaza automáticamente el intento de pago.

resultado basado en una o más entradas (James et al., 2013). Lo anterior, sugiere que por cada observación de un vector de medidas x_i con $i = 1,2,\dots n$ hay una variable de respuesta asociada y_i[3]. Con base en esto, se construye un modelo que capture la relación funcional entre la respuesta y los predictores, ya sea para predecir un valor de la variable respuesta con base en una observación futura de los predictores o para entender mejor la relación entre ellos, es decir, hacer inferencia estadística. Bajo este enfoque, se busca etiquetar un intento de pago como fraudulento o no fraudulento. Así, la detección de fraude se ha modelado como un problema de clasificación donde la variable de respuesta y_i toma dos valores: 1 para indicar fraude y 0, no fraude.

Por otra parte, en el enfoque no supervisado se dispone únicamente de un vector de medidas x_i sin contar con la variable de respuesta. En este contexto, el objetivo principal es comprender o descubrir las relaciones existentes entre las observaciones de los predictores (James et al., 2013). En este, las propuestas metodológicas se han orientado principalmente en modelar la detección de fraude como un problema de detección de anomalías (Chakraborty et al., 2022; Cherif et al., 2023). Hilal y colaboradores (2022) reconocen que en la literatura no hay un consenso en torno a la definición de anomalía o valor atípico, por el contrario, esta depende de la estructura propia de los datos y de la aplicación en consideración (Hilal et al., 2022). No obstante, señalan que la definición más ampliamente reconocida es la de Hawkins (1980), la cual sostiene que una anomalía o valor atípico corresponde a una observación que se desvía significativamente de las demás, indicando que posiblemente fue generada por un mecanismo diferente (Hawkins, 1980).

1.3. Desafíos de la detección de fraude

La literatura ha reconocido cuatro grandes desafíos: desbalance de clases, falta de datos reales, no estacionariedad en los datos (*data drift* o *data shift*) y sobreposición de datos (Cherif et al., 2023; Paldino et al., 2022). El primero, consiste en que no se cuenta con el mismo número de observaciones en las clases conocidas dentro de la base de datos. Esto es una característica natural de las bases de datos de fraude dado que el número de observaciones fraudulentas es muy pequeño en comparación con el número de transacciones genuinas. Esto tiene un impacto directo en el desempeño de la mayoría de los algoritmos de aprendizaje estadístico dado que estos asumen

[3]Al vector de medidas x_i se conoce también como vector de características, predictores, variables explicativas o variables independientes.

un mismo costo de clasificación errónea entre las observaciones de la clase mayoritaria y minoritaria, resultando en un clasificador sesgado hacia la clase mayoritaria (Kim & Hwang, 2022). La segunda, alude a la disponibilidad de la etiqueta o a la latencia de verificación, dado que cuando ocurre un fraude pueden pasar varios días para que la víctima se entere e inicie un contracargo contra la entidad financiera, ocasionando que no se cuente con información suficiente u oportuna para la modelación del fraude (Paldino et al., 2022). La tercera, i.e., no estacionariedad en los datos, corresponde a la pérdida de desempeño de un algoritmo de aprendizaje estadístico a lo largo del tiempo por causa de cambios en las distribuciones de los datos. Esto ocurre en la práctica por cambios en los patrones de comportamiento de los usuarios o cambios en las estrategias que utilizan los defraudadores para atacar a sus víctimas (Muaz et al., 2020). Por último, la sobreposición de datos se presenta cuando observaciones de distintas clases se encuentran en un mismo punto dentro del espacio de características haciendo que el clasificador no pueda distinguirlas entre sí. Esto es típico del problema de detección de fraude dado que los defraudadores intentan imitar el comportamiento de usuarios genuinos con el objetivo de sobrepasar los controles de los sistemas de detección (Cherif et al., 2023).

1.4. Esquemas de combinación de clasificadores

Dentro de las propuestas metodológicas que han surgido en el enfoque supervisado, se ha demostrado que los esquemas de combinación de clasificadores obtienen un mejor desempeño en comparación de modelos individuales en la tarea de detección de fraude (Baker et al., 2022; Mienye & Sun, 2023; Rakhshaninejad et al., 2022). Dichos esquemas fusionan las predicciones de dos o más clasificadores individuales para obtener una predicción final. Según la literatura, la combinación de clasificadores puede ser vista como un proceso de clasificación multietapa donde primero, se estiman probabilidades posteriores por un conjunto determinado de clasificadores individuales y, segundo, se implementa un esquema de combinación utilizando las probabilidades posteriores como variables de entrada (Asefi et al., 2023; Kittler, 1998; Sartor et al., 2024; Tavana et al., 2023). Para el caso de detección de fraude en compras con tarjetas de crédito, sean han utilizado principalmente tres esquemas: *stacked generalization* (o *stacking*), *soft voting* y *hard voting*.

El esquema de *stacking* consiste en entrenar un nuevo clasificador, denominado meta-clasificador, utilizando las probabilidades posteriores estimadas por los clasificadores individuales (Wolper, 1992). Haciendo uso de la data de libre acceso "European cardholders" disponible en la plataforma de *Machine learning* "Kaggle" (Kaggle, 2021), Mienye y Sun (2023) proponen un esquema de *stacking* para clasificar los pagos fraudulentos. Los autores hacen uso de tres modelos de red neuronal: *Long Short Term Memory* (LSTM) y *Gated Recurrent Unit* (GRU) como clasificadores individuales y un *Multilayer Perceptron* (MLP) como meta-clasificador (Mienye & Sun, 2023). Además, realizan el tratamiento del desbalance de las clases utilizando una técnica de generación de datos denominada SMOTE-ENN (*Synthetic Minority Oversampling Technique with Edited Neareast Neighbor*). El proceso de entrenamiento lo realizan utilizando validación cruzada de 10 particiones (*10-fold cross validation*) y los resultados sobre la base de prueba son comparados con clasificadores como *AdaBoost* (AB), *Random Forest* (RF), MLP, LSTM y GRU. Los resultados indican que el esquema de combinación propuesto alcanza el mejor desempeño.

Por otro lado, Soleymanzadeh y colaboradores (2022), construyen dos esquemas de *stacking* para tres bases de datos diferentes: el *Ensamble Stacking Poor* y el *Ensamble Stacking Strong*. Para cada base de datos, se hace una selección de los mejores clasificadores base entre *K-Nearest Neighbor* (KNN), DT, *Gaussian Naive Bayes* (GNB), *Support Vector Machine* (SVM), AB y GB, y un meta-clasificador entre MLP, *Extreme Gradient Boosting* (XGB) y GB con la ayuda de métricas de validación cruzada sobre la base de entrenamiento[4]. Se aplica submuestreo sobre la clase mayoritaria para abordar el desbalance de clases y se entrenan los esquemas utilizando *10-fold cross validation*. Los resultados revelan que los esquemas de *stacking* superan en desempeño a los clasificadores individuales en las tres bases de datos. Adicionalmente, no se obtuvieron diferencias significativas entre el desempeño de *stacking Strong* versus *stacking Poor* y la principal diferencia radica en el tiempo de ejecución, siendo más rápido el *stacking Poor* (Soleymanzadeh et al., 2022).

En el trabajo de Olowookere y Olumide (2020) se propone un esquema de *stacking* que implementa un costo mayor para las malas clasificaciones del meta-clasificador sobre la clase minoritaria en la base de datos "European cardholders". Se utilizan tres modelos (KNN, DT y

[4] Estos esquemas se caracterizaron por tener diferentes modelos como clasificadores base y meta-clasificador. Esto sugiere que la elección de los modelos del esquema de combinación presenta una dependencia con el conjunto de datos donde va a ser utilizado.

MLP) como clasificadores base y LR como meta-clasificador. El *stacking* se prueba sobre seis particiones diferentes de la base de datos, con distintas proporciones de transacciones marcadas como fraudulentas. Dichas particiones consideraron todos los casos de fraude y se realizó submuestreo sobre los casos genuinos para obtener diferentes *ratios* (proporciones): 10%, 5%, 2%, 1%, 0.75% y 0.37%. Los resultados indican que el *stacking* propuesto supera el desempeño de los clasificadores individuales en las seis particiones de datos realizadas, siendo una propuesta extensible a distintos conjuntos de datos con diferentes proporciones de fraude y, especialmente, en casos donde se tenga desbalance de clases (Olowookere & Adewale, 2020).

El esquema *hard voting* hace referencia a la regla de combinación *majority voting*, donde la predicción de clase es calculada con base a la clase más frecuente entre las predicciones de clase realizadas por los clasificadores individuales (Kittler, 1998). Para la obtención de las predicciones de clase de los clasificadores individuales se utiliza un valor umbral que toma valores en el intervalo continuo $[0, 1]$. Así, todo valor de probabilidad predicho que sea mayor o igual al umbral tomará el valor de 1 y en el otro caso, el valor de 0. Esta estrategia es utilizada por Baker et al. (2022) para detectar casos de fraude sobre una base de datos tomada de "Kaggle" (Kaggle, 2021). Para esto, hacen uso de siete clasificadores base: *bagging*, DT, RF, AB, LR, NB y SVM. Con estos, construyen cuatro experimentos para probar el uso de tres métodos de reducción de dimensión: un análisis de componentes principales (*Principal Component Analysis* - PCA), un análisis linear discriminante (*Linear Discriminant Analysis* - LDA), un *Autoencoder* y una técnica de sobremuestreo (SMOTE). El mejor resultado se logra utilizando reducción de dimensión con los componentes principales y sin el sobremuestreo para balancear las clases (Baker et al., 2022).

Karthik y colaboradores (2022) proponen un método híbrido que contempla un método selección de características con *AdaBoost* y otro de combinación de clasificadores basado en árboles de decisión (RF y *Extra Tree*). Adicionalmente, los autores realizan tratamiento del desbalance de las clases con la estrategia de sobremuestreo. Finalmente, obtienen una mejora en el desempeño comparando con modelos como LR y GB (Karthik et al., 2022).

Por otra parte, Esenogho y colaboradores (2022) utilizan *Adaptive Boosting* para combinar probabilidades predichas por modelos de red neuronal recurrente LSTM. Esta implementación constituye un proceso iterativo: primero, se entrenan un modelo LSTM con base en los datos iniciales y se calcula los pesos de las observaciones donde se da un mayor valor a las que hayan quedado mal clasificadas. Segundo, los pesos calculados en la primera etapa son pasados a un otro

modelo LSTM para su entrenamiento y se vuelven a calcular los pesos con base en este nuevo modelo. El proceso sigue hasta que se hayan entrenado el número indicado de clasificadores base o hasta que no se tengan errores de clasificación en los datos de entrenamiento. Además, utilizan SMOTE-ENN para corregir el desbalance de las clases. Finalmente, comparando con modelos como SVM, DT, MLP, LSTM y AB demuestran que la metodología propuesta obtiene el mejor desempeño (Esenogho et al., 2022).

Finalmente, el esquema *soft voting* aplica la regla de combinación *weighted average probabilities* donde para cada clase se obtiene un promedio ponderado a partir de las probabilidades posteriores estimadas por los clasificadores individuales y un vector de pesos, siendo la predicción de clase final el argumento máximo entre los dos promedios ponderados (Kittler, 1998)[5]. Rakhshaninejad y colaboradores (2022), implementan esta estrategia utilizando un conjunto de siete clasificadores base, un proceso de ingeniería de características y un cálculo específico para los pesos. Además, utilizan submuestreo sobre las observaciones no fraudulentas para obtener tres particiones de datos, cada una con un *ratio* de desbalance de clases diferente (1:1, 1:4 y 1:9)[6]. El *ratio* 1:1, significa que se tienen una cantidad igual de observaciones para cada clase; el *ratio* 1:4, que se cuenta con cuatro observaciones genuinas por cada fraude y, el *ratio* 1:9, que se cuenta con nueve observaciones genuinas por cada fraude. Dentro de los modelos utilizados están: *Neaural Net* (NN), DT, NB, LR, SVM, AB y GB.

Para calcular el peso de cada modelo base, primero obtienen la exactitud, precisión, sensibilidad, *F1-Score* y *F2-Score*. Segundo, realizan un producto simple entre las cinco métricas. Por último, normalizan dividiendo cada resultado del producto entre la suma de los siete productos y construyen una medida relativa del desempeño de cada modelo individual. Se construyeron 12 particiones de entrenamiento con base en las particiones generadas con los tres *ratios* y el proceso de ingeniería de características. En cada partición, los siete modelos fueron entrenados y los pesos calculados. Finalmente, realizaron un análisis comparativo para determinar la mejor configuración de combinación *soft voting*. Los resultados sugieren el esquema de combinación obtiene el mejor

[5] Estos esquemas se caracterizaron por tener diferentes modelos como clasificadores base y meta-clasificador. Esto sugiere que la elección de los modelos del esquema de combinación presenta una dependencia con el conjunto de datos donde va a ser utilizado.

[6] El *ratio* 1:1, significa que se tienen una cantidad igual de observaciones para cada clase; el *ratio* 1:4, que se cuenta con cuatro observaciones genuinas por cada fraude y, el *ratio* 1:9, que se cuenta con nueve observaciones genuinas por cada fraude.

desempeño en los doce conjuntos de entrenamiento y, particularmente, la configuración décima es la de mejor desempeño (Rakhshaninejad et al., 2022).

Con todo lo anterior, se observa que no existe un consenso en torno al número de clasificadores base que se deberían tener en cuenta para aplicar un esquema de combinación y, para el caso del *stacking*, qué modelo utilizar como meta-clasificador. Además, los trabajos se concentraron en realizar propuestas metodológicas para mejorar el desempeño con base en la precisión, la sensibilidad, el *F1-Score*, el AUC y la exactitud, pero no en definir la estrategia de combinación más adecuada para determinando conjunto de datos y si las probabilidades predichas por los clasificadores base están bien calibradas.

1.5. Calibración de probabilidades

El concepto de calibración se refiere al grado de concordancia entre la evaluación de probabilidad del pronosticador en $C = 1$ y la frecuencia real con la que $C = 1$ se presenta. En otras palabras, la calibración es 0 cuando $p(c|t)$ es igual a t para cualquier valor de t. Cohen y Goldszmitdt (2004) presentan esta definición general, mientras que Nikolaou y colaboradores (2016) la adaptan al caso específico de los clasificadores estableciendo que estos se consideran bien calibrados si la puntuación o *score* que asignan coincide con la probabilidad real. Esto significa que la probabilidad empírica de que un ejemplo con puntuación $s(x) = s$ pertenezca a la clase positiva, se aproxima al valor de puntuación s, cuando el número de ejemplos N tiende a infinito, para cualquier s (Cohen & Goldszmidt, 2004; Nikolaou et al., 2016).

Sin embargo, también se advierte que la puntuación por sí sola no es suficiente para elegir la mejor opción según el costo, lo que es esencial en aplicaciones de clasificadores sensibles a esta variable como es el caso del fraude, tanto en la toma de decisiones, como dentro de un sistema autónomo. Para ello, se requiere una estimación calibrada de la probabilidad que pueda ser utilizada, por ejemplo, con una regla de decisión óptima, ya que implica que el clasificador cuantifica adecuadamente el nivel de incertidumbre o confianza asociado a sus predicciones (Silva Filho et al., 2023). Wang y colaboradores (2021) complementan esta idea, argumentando que la calibración es necesaria cuando hay datos desequilibrados porque algunos algoritmos como *Support Vector Machine* (SVM) o los métodos basados en árboles de decisiones no se basan en un marco probabilístico para entrenar, y además, porque debido al desequilibrio de clases de los

problemas supervisados binarios, es común en que se subestimen sistemáticamente las probabilidades de las clases minoritarias (Wang et al., 2021).

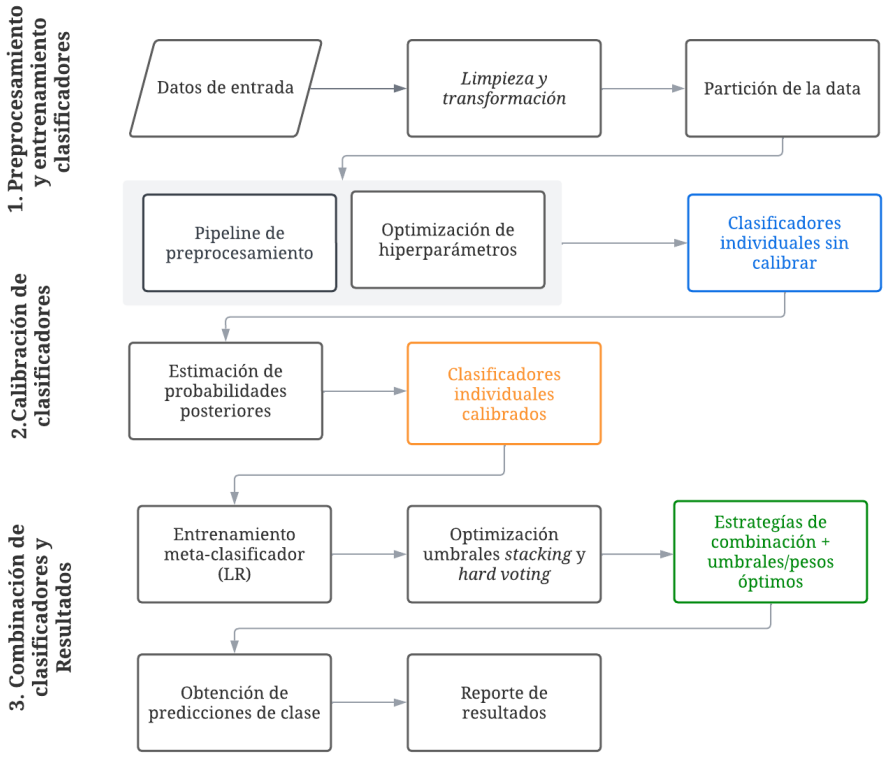
Existen dos estrategias frecuentes para obtener clasificadores calibrados. En una se incorpora la calibración durante el entrenamiento, de modo que el modelo resultante está intrínsecamente bien calibrado (ver Kendall & Gal, 2017; Kumar et al., 2018; Lakshminarayanan et al., 2016; Tran et al., 2018; G. Zhang et al., 2017). La otra estrategia se basa en aplicar una corrección de calibración como un paso de post-procesamiento después de obtener las predicciones del modelo (ver Niculescu-Mizil & Caruana, 2005; Platt, 1999; Zadrozny & Elkan, 2001), y esta última corresponde al método de corrección más utilizado en el caso de la clasificación binaria. Tanto la escala de Platt (1999) que usa una transformación sigmoide para la corrección de la calibración, como los métodos no paramétricos de agrupación de histogramas de igual frecuencia (Zadrozny & Elkan, 2001) y el modelo de calibración isotónica basada en regresión (IsoRegC) (Zadrozny & Elkan, 2002), se pueden interpretar como funciones que transforman las predicciones del modelo en probabilidades mejor calibradas.

En la detección de fraude, la necesidad de obtener modelos con buena calibración de probabilidades se ha centrado en las ventajas que ofrecen para mejorar la gestión de riesgos y la toma de decisiones en situaciones críticas. Por ejemplo, al obtener modelos mejor calibrados se permite la aplicación directa de la clasificación de costos mínimos esperados que optimiza las estrategias para minimizar las pérdidas financieras asociadas al fraude. Algunos autores han propuesto métodos para mejorar la calibración de probabilidades en ese sentido, como la combinación del *Weight of Evidence* (WOE) con un *AdaBoost* (Viaene et al., 2004), la selección de la complejidad del modelo mediante una función de pérdida específica (Brant & Hobæk Haff, 2023) u OP (Bahnsen et al., 2014). Sin embargo, no se han explorado los efectos de un método combinatorio con técnicas de calibración. En este aspecto, la investigación pretende contribuir al conocimiento, al explorar los efectos de un enfoque combinatorio que integra la calibración con los métodos ensamble, para obtener modelos con probabilidades correspondientes a las frecuencias observadas, sin afectar el rendimiento de la predicción del fraude.

2 Metodología

La metodología propuesta consta de tres etapas (Error! Reference source not found.). En la primera etapa se realiza tanto el preprocesamiento de los datos, que implica unificar las fuentes, perfilarlas para identificar sus características y propiedades, limpiarlas, transformarlas, enriquecerlas y seleccionar las variables más relevantes para la clasificación; así como también, la optimización y entrenamiento de los clasificadores individuales. En la segunda, una vez son obtenidos los clasificadores, estos se calibran con un conjunto de datos separado, para eliminar posibles sesgos. Tras estimar las probabilidades posteriores calibradas, se realiza su combinación usando diferentes reglas de agregación, para obtener el método de conjunto. Por último, se evalúan los métodos de conjunto, sobre un subconjunto de prueba y se contrastan los resultados con los obtenidos con los clasificadores individuales como referencia.

Figura 1
Proceso metodológico



2.1 Datos

El conjunto de datos empleado es "IEEE-CIS *Fraud Detection*" proporcionado a la plataforma "Kaggle" por *Vesta Corporation* (Howard et al., 2019). Este conjunto de datos incluye 286,140 transacciones realizadas con tarjetas de crédito y débito en una plataforma de comercio electrónico, con 433 variables que se detallan en la **Tabla 1**. Originalmente, los datos fueron particionados en secciones iguales de entrenamiento y prueba. Para este trabajo, se utiliza sólo la partición de entrenamiento del conjunto de datos original, que contiene 144,233 registros con 132.915 (92,15%) transacciones legítimas y 11,318 (7,85%) transacciones fraudulentas. La partición de prueba no se considera en este análisis porque carece de la variable *isFraud*, que indica el tipo de transacción.

La información contiene detalles tanto de la transacción, como datos asociados a la identificación del usuario, junto con una etiqueta que permite saber si la operación fue identificada como fraudulenta o no.

Tabla 1

Descripción de las variables

Ámbito	Variable	Definición
Datos transaccionales	TransactionID	ID de la transacción
	TransactionDT	Tiempo transcurrido desde una fecha de referencia
	TransactionAmt	Monto de la transacción en dólares
	ProductCD	Código del producto comprado
	card1 - card6	Información asociada a la tarjeta de crédito
	addr1, addr2	Direcciones asociadas al usuario
	dist1, dist2	Información de distancias
	P_emaildomain	Dominio del correo electrónico del comprador
	R_emaildomain	Dominio del correo electrónico del vendedor
	M1 – M9	Información de coincidencia
	C1-C14	Variables de conteos (agregados de comportamiento)
	D1-D15	Variables de diferencia en días
	VXXX	Variables construidas por *Vesta Corporation*
	isFraud	Toma el valor de 1 si es fraude y 0 si es genuina la operación
Datos de la identidad del usuario	DeviceInfo	Información asociada al dispositivo
	id_12 – id_38	Información relacionada con el tipo de conexión
	DeviceType	Tipo de dispositivo usado en la operación

2.2 Preprocesamiento de datos

Los datos son preprocesados a partir de las alertas identificadas en la etapa de perfilamiento (ver **Anexo *1*. Proceso metodológico en extenso** y ***Anexo 2. Pipeline de limpieza***) y divididos en cinco subconjuntos según su finalidad: entrenamiento de clasificadores (56%), calibración de probabilidades (8%), entrenamiento del *stacking* (12%), cálculo de umbrales (4%) y conjunto de

prueba (20%)[7] a partir de un muestreo aleatorio estratificado proporcional que busca conservar el mismo *ratio* de transacciones legítimas y fraudulentas en cada subconjunto.

2.2.1 Ingeniería de características

Después de aplicados los procedimientos de limpieza, se diseña el siguiente *pipeline* de preprocesamiento para el conjunto de entrenamiento: (i) se imputan los valores faltantes con la mediana para las variables numéricas y con una categoría nueva para las variables categóricas, evitando así inconsistencias entre las etiquetas de los conjuntos de entrenamiento y prueba; (ii) se aplica PCA a las variables creadas por *Vesta* a partir de conteos, rankings y otras relaciones de entidades, con el objetivo de eliminar la multicolinealidad y reducir el costo computacional, sin afectar la precisión en la detección del fraude y (iii) se clasifican los atributos cualitativos según su nivel de cardinalidad[8], usando *one-hot encoding* para los de baja cardinalidad y *binary encoding* para los de alta cardinalidad, ya que *one-hot encoding* puede afectar el desempeño y la eficiencia del modelo ante vectores muy largos (Dahouda & Joe, 2021).

2.2.2 Selección de características

Para mejorar el rendimiento y la precisión del modelo de clasificación y debido a la gran cantidad de variables creadas a partir de los procesos aplicados en la sección anterior, se utilizaron dos técnicas de selección de variables descritas en el **Anexo *1*. Proceso metodológico en extenso** Una es un método de filtrado basado en la varianza muestral, que elimina las variables con valores casi constantes y que no aportan información relevante. La otra es un método *embedded* que usa un clasificador *Random Forest* para estimar la importancia de las variables. Para obtener un conjunto de datos más reducido y eficiente para el entrenamiento y la validación del modelo se aplica *SelectFromModel* de la librería "scikit-learn" (versión 1.3.1; scikit-learn, 2013), el cual es

[7] Para la definición de las particiones se dividen originalmente los datos en 80% y 20%; el 80% es a su vez subdividido en un 70% para entrenamiento, un 10% para la calibración, un 5% para la optimización del umbral y finalmente un 15% para el entrenamiento del ensamble. El proceso de muestreo es validado a través de la *suite train-test validation de deepcheck*, comparando que las distribuciones entre las diferentes divisiones de prueba y entrenamiento fuera representativas y conservaran la distribución original (ver **Anexo *3*. Validación del muestreo**).

[8] Se seleccionan los componentes que explicaban al menos el 90% de la varianza.

un meta-transformador que reduce la dimensionalidad de los datos al seleccionar las variables más relevantes para la predicción (Gada et al., 2021) y es configurado con el estimador *Random Forest*[9].

2.3 Entrenamiento de clasificadores individuales

En términos de los clasificadores se emplean cuatro algoritmos supervisados: *Support Vector Machine* (SVM) (Cortes & Vapnik, 1995), *Logistic Regression* (LR) (Cox, 1958), *Random Forest* (RF) (Breiman, 2001) y *Extreme Gradient Boosting* (XGBOOST) (Chen & Guestrin, 2016). Estos modelos se seleccionan por su amplia aplicación y eficacia en la resolución de problemas de clasificación binaria aplicada a fraude en comercios electrónicos (Albashrawi, 2021; Ashtiani & Raahemi, 2022). En particular, se espera que los métodos basados en árboles de decisión, como RF y XGBOOST, ofrezcan un mejor rendimiento en el tratamiento del desbalanceo de clases que los métodos lineales, como SVM y LR, pero con diferencias derivadas del proceso de aprendizaje y la regularización de cada algoritmo. En la **Error! Reference source not found.** se muestran los hiperparámetros considerados para cada clasificador.

Tabla 2

Lista de modelos e hiperparámetros

Modelo	Hiperparámetros
SVM	*C, Kernel, threshold*
LR	*C, solver, penalty, l1_ratio, threshold*
RF	*n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features, threshold*
XGBOOST	*Objective, n_estimators, learning_rate, subsample, colsample_bytree, max_depth, threshold*

La arquitectura del flujo de entrenamiento se muestra en el

[9] Este clasificador se optimizó mediante una búsqueda en cuadrícula y validación cruzada con *Stratified K-Fold*, variando el número de estimadores entre 50, 100 y 500 y el criterio entre: *gini*, *entropía* y *log-loss*.

Anexo 4. Arquitectura de entrenamiento clasificadores[10]. Después de transformar y procesar los datos, se optimizan los hiperparámetros de los cuatro clasificadores mediante el algoritmo *Tree-structured Parzen Estimator* (TPE) (Bergstra et al., 2011, 2013), un método basado en procesos gaussianos que minimiza el número de evaluaciones de la función objetivo. Para reducir el tiempo de optimización, TPE es aplicado por medio de un software de código abierto denominado "Optuna", el cual aprende continuamente de optimizaciones anteriores a través de algoritmo eficiente de muestreo y un algoritmo de poda, permitiendo construir dinámicamente el espacio de búsqueda (Akiba et al., 2019). Con este proceso se espera encontrar los hiperparámetros que maximizan el *F1-Score*[11] y encontrar el umbral óptimo de decisión a partir del cual se pueden clasificar los eventos como "Fraude" o "No Fraude". Los clasificadores son entrenados con *stratified 5-fold cross* y se evalúan con las métricas relacionadas en el **Anexo 5. Medidas de desempeño** sobre el conjunto de validación.

2.3.1 Calibración de probabilidades

Dado que muchos clasificadores no estiman correctamente la probabilidad de clase (Niculescu-Mizil & Caruana, 2005), se calibran los modelos entrenados con los datos reservados para este propósito, para obtener la inferencia en forma de probabilidad usando la regresión isotónica en la mayoría de los modelos, y obtener así los clasificadores individuales calibrados. De esta forma, se espera que los clasificadores generen valores de confianza más fiables cuando se aplique *predict_proba* de "scikit-learn" (Pedregosa et al., 2011). La regresión isotónica es un método no paramétrico que garantiza que la probabilidad calibrada sea una función monótona creciente del *score* (Zadrozny & Elkan, 2001, 2002), ha demostrado buen desempeño en muestras grandes y robustez ante diferentes problemas de clasificación (Nyberg & Klami, 2021), especialmente cuando los datos presentan desbalanceo entre las clases (Huang et al., 2020).

Existen dos excepciones al proceso anterior: en la regresión logística se emplea la escala de Platt (1999) para calibrar las probabilidades predichas siguiendo las recomendaciones de

[10] A grandes rasgos, este esquema es el que sigue el resto de los procesos, con pequeñas variaciones como la cantidad de subconjuntos en la validación cruzada. Todos los métodos son evaluados con las mismas métricas y sobre el mismo conjunto de validación, para dar consistencia y reproducibilidad al ejercicio, buscando que los resultados además puedan ser comparables.

[11] Se elige cómo medida de optimización el *F1 Score* debido a que el interés sobre el caso de prevención de fraude es tener un balance entre la precisión y la sensibilidad.

Niculescu-Mizil y Caruana (2005). Esta decisión se basa en la observación de la gráfica de distorsión, que muestra una forma sigmoidea de las probabilidades antes de la calibración (ver la **Figura 2** de la sección **3.1 Calibración de modelos**). El XGBOOST, por su parte, no requiere ningún método de calibración adicional, ya que las probabilidades estimadas por el modelo son las que mejor desempeño y calidad muestran, de acuerdo con las métricas AUC-ROC, puntuación *Brier y log-loss* en el conjunto de pruebas. Esto se confirma con la experimentación realizada con varias combinaciones de función objetivo y de algoritmos de calibración, cuyos resultados se presentan en la **Tabla 3** de la sección **3.1 Calibración de modelos**.

2.4 Combinación de clasificadores

Para cada una de las tres estrategias, se compararon las métricas de rendimiento sobre la partición de prueba considerando los escenarios planteados: clasificadores individuales sin calibrar probabilidades y clasificadores individuales con probabilidades calibradas.

2.4.1 Stacking

Se eligió cómo meta-clasificador un LR dado que las probabilidades posteriores estimadas son bien calibradas y es de fácil implementación (Niculescu-Mizil & Caruana, 2005). Para cumplir con los escenarios de estudio, no fue posible utilizar las clases construidas en el proyecto de "scikit-learn" (versión 1.3.1) relacionadas a los métodos de ensamble utilizados en este trabajo (Pedregosa et al., 2011; scikit-learn, 2013)[12,13]. Así, se generaron clases propias para consumir las probabilidades posteriores calibradas y no calibradas que entregan los clasificadores individuales en el entrenamiento del LR. Esto último fue realizado utilizando *stratified 10-fold cross validation* para obtener el LR con hiperparámetro optimizado[14] utilizando como métrica de validación el *F1-Score*.

[12] *Sklearn.ensemble.VotingClassifier, sklearn.ensemble.StackingClassifier.*
[13] Por un lado, los objetos instanciados para obtener las probabilidades calibradas requieren como variables de entrada las probabilidades posteriores no calibradas y, por el otro, no cuentan con el método de clase *predict_proba* requerido.
[14] Inverso del parámetro de regularización. A medida que *Cs* tiende a cero se realiza una mayor regularización

2.4.2 Hard voting

En la estrategia de *hard voting* fue necesario transformar las probabilidades posteriores estimadas en predicciones de clase. Esto se logra estableciendo un valor umbral que determina si un pago es fraudulento (1) cuando la probabilidad posterior estimada es mayor o igual al umbral, o no fraudulento (0) cuando es menor[15].

Usualmente, en la literatura no se indica un método para encontrar un valor del umbral y se asigna la clase positiva a las instancias con una probabilidad positiva ≥ 0.5 y la clase negativa a las demás, lo que puede no ser óptimo cuando la distribución de probabilidad está sesgada hacia la clase mayoritaria (King & Zeng, 2001). Sin embargo, la elección de un valor para el umbral es importante para el rendimiento de los modelos, ya que tiene un impacto directo sobre las métricas de clasificación. Por esto, en este trabajo no se asumió un valor fijo para el umbral, sino que fue hallado a través de un proceso de optimización, similar al aplicado en los clasificadores individuales (utilizando la partición 1-3), donde el umbral toma el valor que maximice la métrica del *F1 Score*.

A partir de lo anterior, se obtuvieron las predicciones de clase para los cuatro clasificadores individuales las cuales fueron utilizadas para la aplicación de la regla *majority voting*.

2.4.3 Soft voting

En esta estrategia se utilizan las probabilidades posteriores estimadas por los clasificadores individuales y un vector de pesos $\omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ para calcular el promedio ponderado de cada clase. El objetivo del vector de pesos es asignar una mayor ponderación a las probabilidades posteriores predichas por los clasificadores con mejor desempeño. Como la métrica *F1-Score* corresponde a una media armónica entre la precisión y la sensibilidad con el mismo nivel de importancia, se decidió utilizarla para construir el vector de pesos. En particular, el cálculo se resume en la siguiente ecuación:

$$\omega_i = \frac{\text{F1 Score}_{\text{clf}_i}}{\sum_{i=1}^{4} \text{F1 Score}_{\text{clf}_i}}$$

[15] Corresponde a un valor del intervalo continuo [0, 1].

De esta manera, los clasificadores que hayan obtenido un mayor *F1-Score* tendrán más influencia en el cálculo de la probabilidad promedio ponderada de cada clase.

3 Resultados

Este trabajo estudió el efecto de la calibración de las probabilidades posteriores de los clasificadores individuales sobre los resultados de tres estrategias de combinación: *stacking, hard voting y soft voting*. Los resultados experimentales se presentan en dos secciones: primero, se muestran los resultados de la calibración de probabilidades y, segundo, el desempeño de los clasificadores en términos de las métricas de clasificación. Adicionalmente, las estrategias de combinación fueron construidas utilizando cuatro modelos base: LR, SVM, RF y XGBOOST para los cuales se utilizó optimización bayesiana para hallar los mejores valores de los hiperparámetros con base en el *F1-Score* y se utilizó *stratified k-fold cross validation* en el proceso de aprendizaje para garantizar la representatividad original de cada clase en los *folds*. Se obtuvieron resultados sobre la partición 2 destinada a prueba.

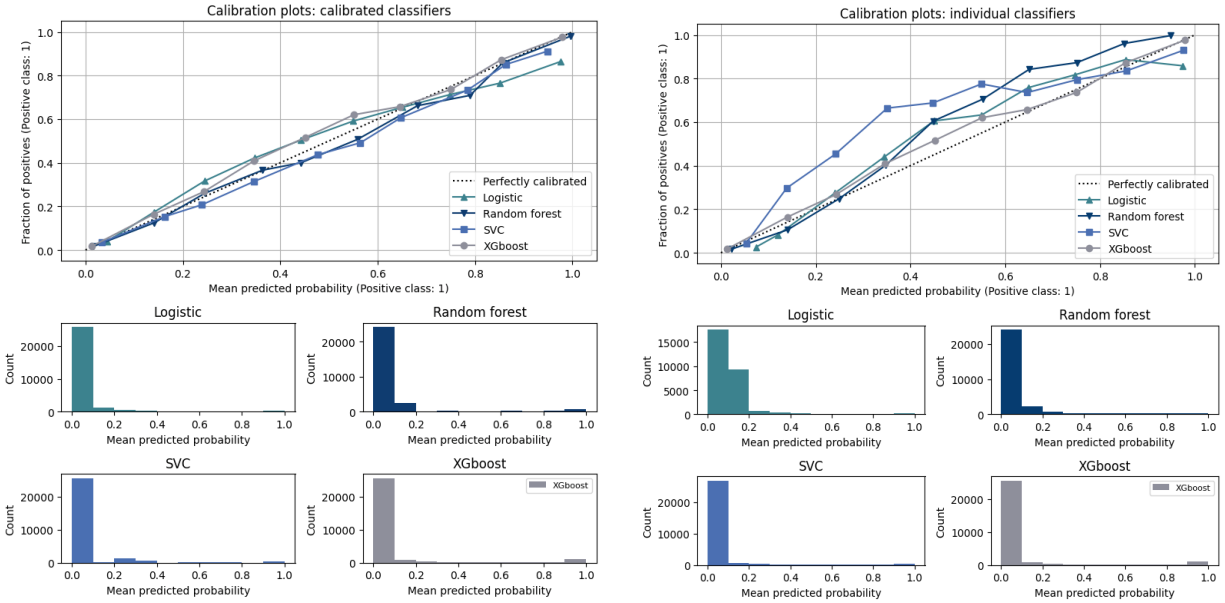
3.1. Calibración de modelos

La calibración se emplea con el método *CalibratedClassifierCV*[16] de ”sckit-learn”. Al evaluar la calidad de las predicciones de los clasificadores, se utilizaron las curvas de calibración o confiabilidad (Wilks, 1990), que muestran la relación entre la probabilidad estimada del evento condicional $P(Y = 1|predict - proba)$ y la frecuencia observada del mismo.

[16] Se define el parámetro cv=”prefit” dado que previamente los modelos son ajustados y optimizados sus hiperparámetros.

Figura 2

Diagramas de confiabilidad para modelos sin calibrar vs modelos calibrados



De acuerdo con estas curvas, se puede apreciar que los modelos que presentan una mayor concordancia entre las probabilidades estimadas y las reales previo a la calibración son los modelos LR y XGBOOST; esto se debe porque ambos modelos están generando directamente probabilidades. El LR está optimizado de forma predeterminada con *Log-loss* y el XGBOOST emplea el hiperparámetro *objective="binary:logistic",* es decir, la regresión logística para clasificación binaria, por lo que la salida se da en términos de probabilidad. Por el contrario, los demás modelos exhiben cierto sesgo en sus predicciones, que varía según el algoritmo utilizado, prevaleciendo en la mayoría una previsión insuficiente, con valores por encima de la diagonal.

Revisando la **Tabla 3**[17], se observa que el XGBOOST empeora su desempeño al calibrar sus predicciones, sin importar el método utilizado. Por esta razón, se opta por usar el resultado del modelo sin calibrar, ya que de igual forma produce valores probabilísticos. Los demás modelos mejoran su ajuste según el *Log-loss*, sin afectar negativamente su evaluación según el AUC- ROC.

[17] Estos resultados son obtenidos manteniendo el *threshold* por defecto de 0.5 para discretizar las probabilidades.

Tabla 3

Resultados Clasificadores

Métrica	LR	LR (Platt)	RF	RF (Isotonic)	SVC	SVC (Isotonic)	XGBOOST	XGBOOST (Isotonic)	XGBOOST (Platt)
Brier score	0.0281	0.0271	0.0353	0.0416	0.0232	0.0323	0.0494	0.0543	0.0543
Log-loss	0.2210	0.2076	0.1353	0.1339	0.2182	0.2004	0.1081	0.1145	0.1145
AUC-ROC	0.8200	0.8200	0.9300	0.9300	0.7700	0.7900	0.9500	0.9500	0.9500

3.2 Desempeño modelos de clasificación y esquemas de combinación

La **Tabla 4** muestra las métricas de clasificación tanto para los clasificadores individuales como los esquemas de combinación con y sin calibrar probabilidades.

Tabla 4

Métricas de clasificación

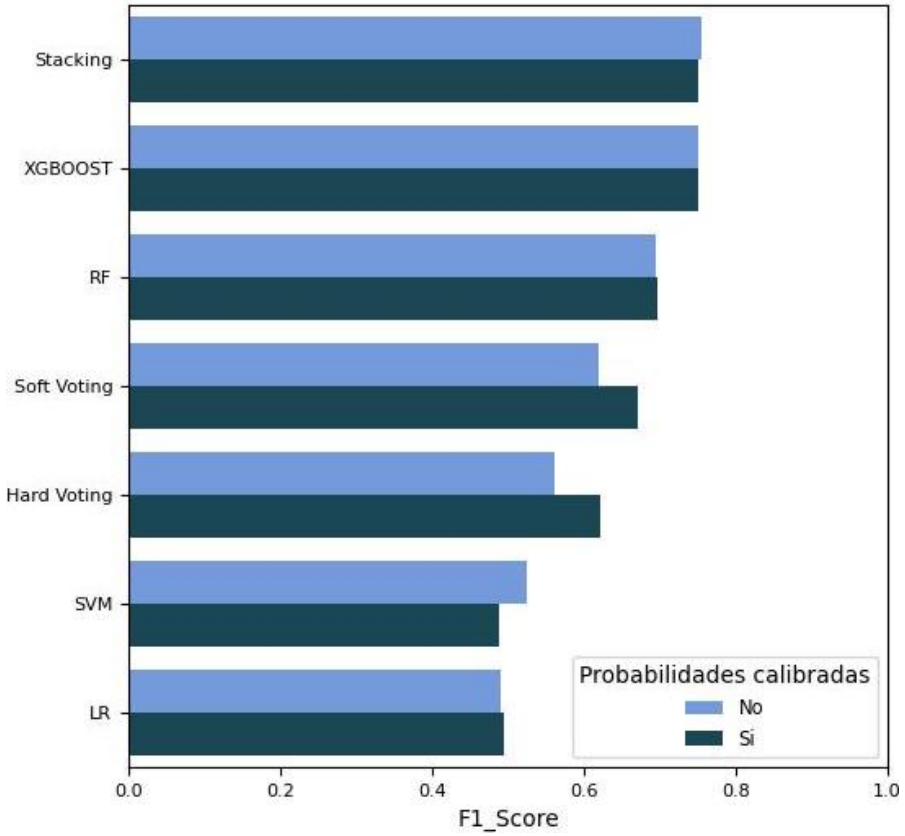
	Accuracy	Precision	Recall	F1_Score	F2_Score	ROC_ AUC	PR_AUC	FPR	FNR
Stacking	0.9654	0.8490	0.6802	0.7553	0.7084	0.9469	0.8157	0.0103	0.3198
Stacking Calibrado	0.9651	0.8554	0.6687	0.7506	0.6992	0.9487	0.8164	0.0096	0.3313
Extreme Gradient Boosting	0.9658	0.8798	0.6528	0.7495	0.6883	0.9523	0.8219	0.0076	0.3472
Random Forest Calibrado	0.9566	0.7707	0.6369	0.6975	0.6598	0.9292	0.7337	0.0161	0.3631
Random Forest	0.9583	0.8145	0.6069	0.6955	0.6395	0.9296	0.7526	0.0118	0.3931
Soft Voting Calibrado	0.9589	0.8992	0.5358	0.6715	0.5829	0.9393	0.7719	0.0051	0.4642
Hard Voting Calibrado	0.9546	0.8954	0.4766	0.6221	0.5258	0.9367	0.7568	0.0047	0.5234
Soft Voting	0.9550	0.9252	0.4647	0.6186	0.5160	0.9404	0.7833	0.0032	0.5353
Hard Voting	0.9506	0.9242	0.4037	0.5619	0.4550	0. 9374	0.7703	0.0028	0.5963
Support Vector Machine	0.9311	0.5718	0.4854	0.5251	0.5005	0.7750	0.5221	0.0310	0.5146
Logistic Regression Calibrado	0.9310	0.5817	0.4307	0.4949	0.4542	0.8181	0.4860	0.0264	0.5693
Logistic Regression	0.9268	0.5404	0.4492	0.4906	0.4649	0.8181	0.4860	0.0325	0.5508
Support Vector Machine Calibrado	0.9019	0.4134	0.5976	0.4887	0.5487	0.7910	0.4983	0.0722	0.4024

Según el F1-Score, los modelos con mejor desempeño en el conjunto de datos de prueba fueron obtenidos por modelos que hacen combinación de las predicciones de clasificadores base *stacking* (75.53%), *stacking* calibrado (75.06%), XGBOOST (74.95%), RF calibrado (69.75%), RF (69.55%), *soft voting* calibrado (67.15%), *hard voting* calibrado (62.21%), *soft voting* (61.86%), *hard voting* (56.19%), SVM (52.51%), LR calibrado (49.49%), LR (49.06%) y SVM calibrado (48.87%).

Para contrastar el efecto de la calibración sobre el desempeño de los clasificadores se puede observar en la **Figura 3** que, en general, los modelos alcanzaron un mayor *F1 Score* excepto para el *Stacking* y el SVM.

Figura 3

Comparación de F1 Score entre los modelos calibrados y no calibrados



Los esquemas de combinación *soft voting* y *hard voting* tuvieron un incremento considerable del *F1 Score* de 5,29% y 6,02%, respectivamente. Por su parte, la precisión y la sensibilidad tuvieron un comportamiento interesante. Con la calibración, los modelos *soft voting*, *hard voting*, RF y SVM presentaron un aumento en la sensibilidad y una disminución en la precisión y, los modelos *stacking* y LR tuvieron el comportamiento contrario (ver **Figura 4** y **Figura 5**).

Figura 4

Comparación de la precisión entre los modelos calibrados y no calibrados

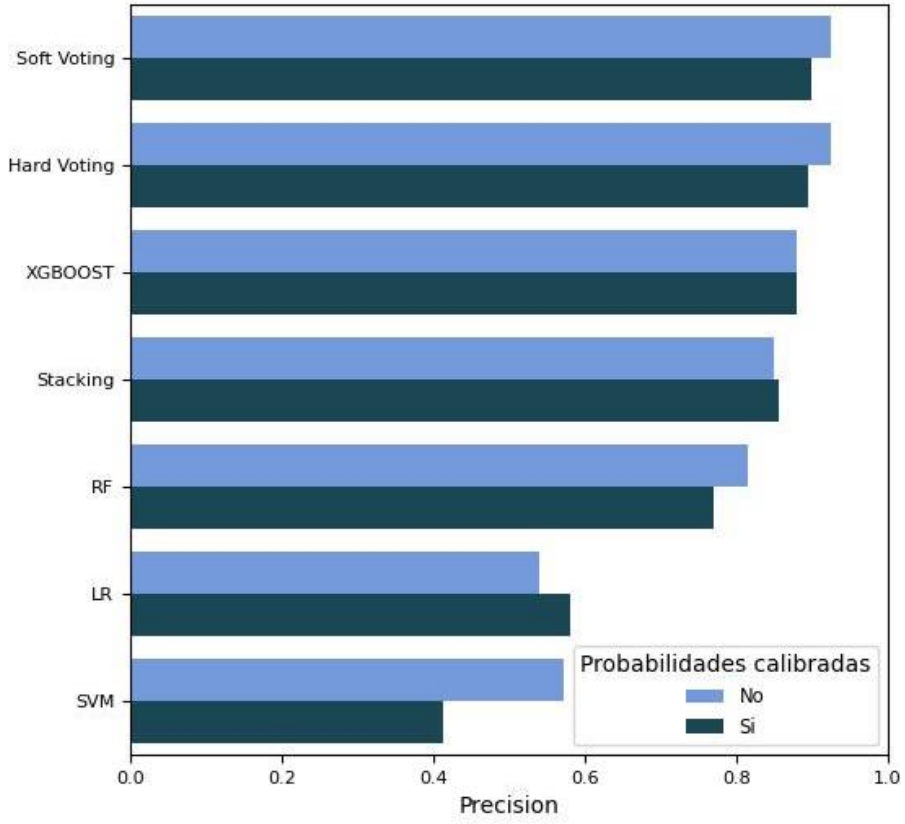
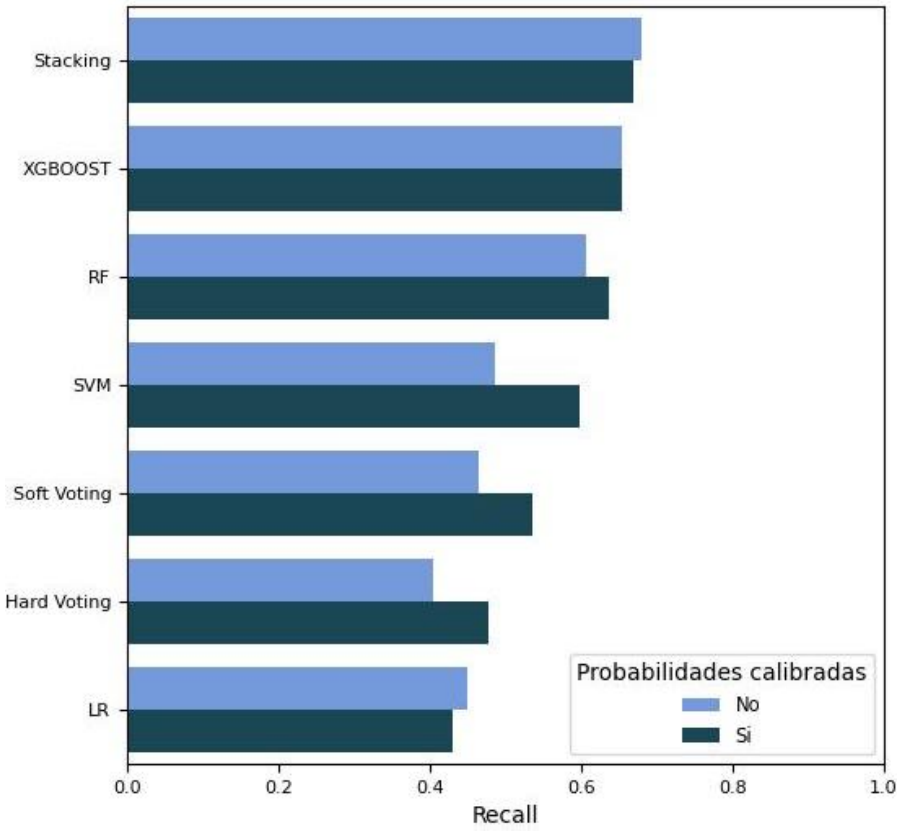


Figura 5

Comparación de la sensibilidad entre los modelos calibrados y no calibrados



Dado lo anterior, la calibración de las probabilidades representó una ganancia importante en la sensibilidad junto con una leve disminución en la precisión para las estrategias de combinación *soft voting* y *hard voting* que se materializó con un aumento del *F1-Score* en 5,29% y 6,02%, respectivamente. Por su parte, el *stacking* presentó un efecto compensatorio entre la precisión y la sensibilidad que mantuvo sin alteración el *F1 Score* alrededor del 75%.

Comparando el desempeño de las tres estrategias de combinación en la tarea de detección de fraude, se observa que en términos del *F1-Score* la estrategia de *stacking* tiene el mejor desempeño por encima de las otras dos estrategias (*soft voting* y *hard voting*) y, de los clasificadores individuales antes y después de calibrar las probabilidades.

4 Discusión

Este estudio propone un método para detectar el fraude por contracargo en transacciones con tarjetas de crédito en un comercio electrónico basado en la combinación de clasificadores con calibración. El fraude por contracargo ocurre cuando un cliente solicita la devolución del dinero de una compra realizada con su tarjeta de crédito, alegando que no la reconoce o que no recibió el producto o servicio (Liu & Lee, 2022). Este tipo de fraude representa un riesgo financiero y operativo, afectando la rentabilidad y la reputación de las empresas *e-commerce*, que deben invertir recursos en prevenirlo, mitigarlo y gestionarlo. El método presentado en este artículo utiliza una medida de criticidad para asignar pesos a los clasificadores individuales y combinarlos mediante tres estrategias: *stacking*, *hard voting* y *soft voting*. Además, se aplica una calibración de las probabilidades posteriores que busca evaluar posibles mejorías en el rendimiento de los clasificadores combinados, disminuir la incertidumbre y aumentar la eficiencia ante situaciones de riesgo con costos asimétricos.

El presente estudio demuestra que el método de agregación de los clasificadores influye en la precisión de las probabilidades generadas. El *stacking* produce probabilidades más exactas sin comprometer el rendimiento, mientras que el *soft voting* y *hard voting* llegan incluso a mejorar sus resultados al aplicar técnicas de corrección sobre el *score*, logrando disminuir el FNR en un 7,11% y 7,79% y, aumentar el *F1-Score* en un 5,29% y 6,02%, respectivamente después de la calibración. Esto sugiere que las metodologías de ensamble en conjunto con los procesos de calibración pueden adaptarse a fenómenos como el fraude, que se beneficiarían de tener un mayor nivel de confianza para tomar decisiones en situaciones de alta incertidumbre. Además, el uso de estas probabilidades ajustadas en casos de fraude aplicados permitiría establecer un criterio para la clasificación de costos mínimos esperados o la maximización de la utilidad (Silva Filho et al., 2023), cuando se conoce el costo diferencial de los errores y se busca asignar los recursos de forma óptima (Bahnsen et al., 2014; Brant & Hobæk Haff, 2023).

A nivel de desempeño, los resultados de los experimentos en el proceso de calibración no son del todo consistentes con lo encontrado en otros estudios a nivel empírico (Cohen & Goldszmidt, 2004), en donde se espera que se obtenga el mayor provecho de aplicar una corrección de la distorsión a nivel de los modelos individuales en los clasificadores que no pueden producir

[18]. Al aplicar la calibración, se observan comportamientos diferenciados entre las métricas enfocadas en la clase positiva y las enfocadas en la clase negativa con variaciones importantes, lo cual es un comportamiento que también se replica en los métodos combinatorios, con excepción de la técnica de *stacking*. Una posible explicación de estos resultados es la influencia del umbral de decisión utilizado para discretizar las clases. Cohen y Goldszmith (2004), por ejemplo, encuentran que el resultado de relacionar la calibración con la regla de decisión que minimiza el error de clasificación produce un procedimiento efectivo para encontrar umbrales óptimos en esta regla de decisión, por lo que, una posible extensión futura de este estudio podría ser analizar el efecto de la ganancia marginal de los métodos calibrados al combinarlo con el umbral de decisión.

La utilización de un método de optimización bayesiana para encontrar el mejor conjunto de hiperparámetros de los modelos base y los valores de los umbrales, permitió reducir los tiempos en la tarea de búsqueda[19]. Éste oscilo entre 28 segundos y 3 minutos aproximadamente. Además, la implementación utilizada con "Optuna" garantiza niveles de exactitud, eficiencia (en términos de tiempo de búsqueda) y robustez por encima de otras implementaciones como *Sckit-opt*, *Hyperopt* (Hanifi et al., 2024). Por su parte, la implementación para el cálculo del valor umbral, utilizado para discretizar las probabilidades posteriores con base en el *F1-Score*, representó un enfoque alternativo al comúnmente empleado de 0,5 y, en donde se tiene en cuenta el desbalance de las clases. Un resultado práctico de esto último es que los valores de los umbrales se situaron entre 0,1 y 0,4, lo que puede ser resultado del desbalance de las clases, donde la masa de la distribución de probabilidades posteriores estimadas se situó cercana a cero.

Los resultados obtenidos en este trabajo distan de ser comparables con las aplicaciones previas presentadas en el marco teórico. Esto, ya que los resultados de los trabajos previos se logran en conjuntos de datos diferentes. Sin embargo, se encuentran otros trabajos que han utilizado el conjunto de datos "IEEE-CIS Fraud Detection". Por un lado, Nguyen et al. (2022), proponen la implementación de dos modelos *CatBoost* y *Deep Neural Network* los cuales son entrenados y validados en poblaciones diferentes que se definen según el tipo de usuario entre conocidos y nuevos, respectivamente. Con esta implementación, los autores llegan a niveles de ROC-AUC de 97,4%, para usuarios conocidos y 84,2% para usuarios nuevos en los datos de prueba. Cabe

[18] El cual genera sólo una puntuación de confianza en el intervalo real $[-\infty; +\infty]$ que indican únicamente que tan seguro [inseguro] con respecto a la predicción (Bequé et. al. 2017)

[19] La ganancia más importante la obtuvo el método de SVM, con una reducción en tiempo mayor al 50% versus un método tradicional como la búsqueda en malla (*grid search*).

mencionar que los autores emplearon SMOTE para corregir el problema de desbalance de los datos previo a hacer la división del conjunto total de datos entre entrenamiento y prueba, por lo que los valores alcanzados de ROC-AUC pueden estar sobreestimados al obtenerse en un conjunto de datos del cuál ha sido modificado su distribución original. Por otro lado, está Chenoori y Kavuri (2022), quienes implementan una técnica de reducción de la dimensionalidad y un proceso de selección de características en conjunto con un modelo XGBOOST para clasificar las transacciones cómo fraudulentas o no fraudulentas. Con esta propuesta, alcanzan niveles de *F1 Score*, *Recall*, *Precision* y *ROC-AUC* de 92%, 88%, 97% y 93,67%. Estos resultados, son superiores a los obtenidos en el presente trabajo; lo cual se debe a que Chenoori y Kavuri (2022) tuvieron acceso al 100% de la base de datos y emplearon 590.540 registros para entrenar y 506.691 para pruebas, lo que implica que lograron capturar patrones de fraude adicionales presentes en los demás registros de la base de entrenamiento.

5 Conclusiones y recomendaciones

De acuerdo con la revisión de las fuentes bibliográficas, este es el primer trabajo que evalúa el efecto de implementar calibración de probabilidades en escenarios altamente desbalanceados sobre meta-clasificadores en un estudio cuantitativo aplicado a detección de fraude. Los resultados son consistentes con lo encontrado en otro estudio que lo utilizó de manera individual, dado que se mantienen las propiedades discriminatorias, en donde con los calibradores mejoran el ajuste entre el puntaje predicho y la probabilidad real, sin afectar la capacidad del clasificador para diferenciar las clases (Bequé et al., 2017). Este análisis, también permite examinar el rendimiento de tres esquemas combinatorios en la tarea de detección de fraude, obteniendo que el método de *stacking* genera los mejores resultados tanto con los modelos sin calibrar como con los modelos calibrados a nivel de la métrica global del *F1-Score*.

La metodología propuesta se puede replicar en otros campos de estudio donde los conjuntos de datos se caractericen por tener gran volumetría (*big data*), haya desbalance de las clases y haya alta dimensión del espacio de características. El proceso de calibración de probabilidades requiere tener particiones de datos independientes a la utilizada para el entrenamiento y la prueba, y una muestra de datos adicional para encontrar un valor óptimo del umbral para los distintos modelos. En este sentido, Niculescu-Mizil & Caruana (2005) recomiendan que cuando el tamaño del conjunto de calibración es pequeño (*i.e*., menos de 200 observaciones), se utilice el método *Platt Scaling*, dado que en los estudios realizados este supera los resultados del *Isotonic Regression*.

Además del umbral, podrían darse otras posibles contribuciones al artículo en el proceso de calibración y las funciones objetivas empleadas para la optimización y posterior evaluación de los métodos. Los métodos de calibración *post-hoc* existentes tienen una eficiencia de muestra limitada en los datos etiquetados del conjunto de desarrollo porque realizan una calibración a nivel de probabilidad y no pueden mejorar otras métricas ajenas a calibración, como la AUC (Pan et al., 2020). Por eso, los autores proponen dos aportaciones: una nueva métrica de evaluación llamada “error de calibración a nivel de campo”, que mide el sesgo en las predicciones respecto al campo de entrada sensible que interesa al tomador de decisiones; y un método de calibración *post-hoc* llamado “calibración neuronal”, que aprende a calibrar utilizando toda la información de campo disponible en el conjunto de validación. Otros estudios encontraron que con *Beta Calibration* se

puede obtener mejores resultados a nivel del *log-loss* y el *Brier score* al usarse en diferentes métodos de clasificación binaria como NB, *Adaboost*, LR, SVM, RF y MLP (Kull et al., 2017).

Se considera que otras direcciones prometedoras para el trabajo futuro sería aplicar la calibración con un método de *Cost Sensitive Learning*, que se ha mostrado eficaz para tratar datos desbalanceados en diversos ámbitos (Fernández et al., 2018; Mienye & Sun, 2021; Thai-Nghe et al., 2010). También se podría explorar algún método de reequilibrio de datos antes de la calibración, especialmente teniendo en cuenta los efectos del desbalance de clases reportados en la literatura (Wallace & Dahabreh, 2012), en los cuales las probabilidades de clase estimadas en contextos desbalanceados tienden a subestimar las de las clases minoritarias, aun cuando la calibración general es buena. Por último, se recomienda aplicar la metodología a otros conjuntos de datos, para soportar su robustez y capacidad de generalización en otros escenarios.

Referencias

Aganagic, M., & Okounkov, A. (2020). Elliptic stable envelopes. *Journal of the American Mathematical Society*, *34*(1), 79–133. https://doi.org/10.1090/jams/954

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631. https://doi.org/10.1145/3292500.3330701

Albashrawi, M. (2021). Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015. *Journal of Data Science*, *14*(3), 553–570. https://doi.org/10.6339/JDS.201607_14(3).0010

Amankwah-Amoah, J., Khan, Z., Wood, G., & Knight, G. (2021). COVID-19 and digitalization: The great acceleration. *Journal of Business Research*, *136*, 602–611. https://doi.org/10.1016/j.jbusres.2021.08.011

Asefi, S., Mitrovic, M., Ćetenović, D., Levi, V., Gryazina, E., & Terzija, V. (2023). Anomaly detection and classification in power system state estimation: Combining model-based and data-driven methods. *Sustainable Energy, Grids and Networks*, *35*, 101116. https://doi.org/10.1016/j.segan.2023.101116

Ashtiani, M. N., & Raahemi, B. (2022). Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review. *IEEE Access*, *10*, 72504–72525. https://doi.org/10.1109/ACCESS.2021.3096799

Bahnsen, A. C., Stojanovic, A., Aouada, D., & Ottersten, B. (2013). Cost Sensitive Credit Card Fraud Detection Using Bayes Minimum Risk. *2013 12th International Conference on Machine Learning and Applications*, *1*, 333–338. https://doi.org/10.1109/ICMLA.2013.68

Bahnsen, A. C., Stojanovic, A., Aouada, D., & Ottersten, B. (2014). Improving Credit Card Fraud Detection with Calibrated Probabilities. *Proceedings of the 2014 SIAM International Conference on Data Mining*, 677–685. https://doi.org/10.1137/1.9781611973440.78

Bajada, C., & Trayler, R. (2015). Technology-Driven Service Innovation in the Banking Industry. En R. Agarwal, W. Selen, G. Roos, & R. Green (Eds.), *The Handbook of Service Innovation* (pp. 319–343). Springer London. https://doi.org/10.1007/978-1-4471-6590-3_16

Baker, M. R., Mahmood, Z. N., & Shaker, E. H. (2022). Ensemble Learning with Supervised Machine Learning Models to Predict Credit Card Fraud Transactions. *Revue d'Intelligence Artificielle*, *36*(4), 509–518. https://doi.org/10.18280/ria.360401

Bequé, A., Coussement, K., Gayler, R., & Lessmann, S. (2017). Approaches for credit scorecard calibration: An empirical analysis. *Knowledge-Based Systems*, *134*, 213–227. https://doi.org/10.1016/j.knosys.2017.07.034

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. En J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 24). Curran Associates,

Inc. https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf

Bergstra, J., Yamins, D., & Cox, D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. En S. Dasgupta & D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning* (Vol. 28, Número 1, pp. 115–123). PMLR. https://proceedings.mlr.press/v28/bergstra13.html

Brant, S. B., & Hobæk Haff, I. (2023). The fraud loss for selecting the model complexity in fraud detection. *Journal of Applied Statistics*, *50*(10), 2209–2227. https://doi.org/10.1080/02664763.2022.2070137

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, *29*(2), 93–104. https://doi.org/10.1145/335191.335388

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Carta, S., Fenu, G., Reforgiato Recupero, D., & Saia, R. (2019). Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model. *Journal of Information Security and Applications*, *46*, 13–22. https://doi.org/10.1016/j.jisa.2019.02.007

Chakraborty, B., Chaterjee, A., Malakar, S., & Sarkar, R. (2022). An iterative approach to unsupervised outlier detection using ensemble method and distance-based data filtering. *Complex and Intelligent Systems*, *8*(4), 3215–3230. https://doi.org/10.1007/s40747-022-00674-0

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chenoori, Rathan Kumar, and Radhika Kavuri. "Online Transaction Fraud Detection Using Efficient Dimensionality Reduction and Machine Learning Techniques." Revue d'Intelligence Artificielle, vol. 36, no. 4, 31 Aug. 2022, pp. 621–628, https://doi.org/10.18280/ria.360415. Accessed 4 Oct. 2022.

Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., & Imine, A. (2023). Credit card fraud detection in the era of disruptive technologies: A systematic review. En *Journal of King Saud University—Computer and Information Sciences* (Vol. 35, Número 1, pp. 145–174). King Saud bin Abdulaziz University. https://doi.org/10.1016/j.jksuci.2022.11.008

Cohen, I., & Goldszmidt, M. (2004). Properties and Benefits of Calibrated Classifiers. En J.-F. Boulicaut, F. Esposito, F. Giannotti, & D. Pedreschi (Eds.), *Knowledge Discovery in Databases: PKDD 2004* (Vol. 3202, pp. 125–136). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30116-5_14

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cox, D. R. (1958). The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, *20*(2), 215–242. JSTOR.

Cybersource, Merchant Risk Council, & Verifi. (2023). *2023 Global eCommerce Payments and Fraud Report*. https://www.cybersource.com/en/solutions/fraud-and-risk-management/fraud-report.html

Dahouda, M. K., & Joe, I. (2021). A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access*, *9*, 114381–114391. https://doi.org/10.1109/ACCESS.2021.3104357

De', R., Pandey, N., & Pal, A. (2020). Impact of digital surge during Covid-19 pandemic: A viewpoint on research and practice. *International Journal of Information Management*, *55*, 102171. https://doi.org/10.1016/j.ijinfomgt.2020.102171

Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A Neural Network Ensemble with Feature Engineering for Improved Credit Card Fraud Detection. *IEEE Access*, *10*, 16400–16407. https://doi.org/10.1109/ACCESS.2022.3148298

Fajardo, V. A., Findlay, D., Jaiswal, C., Yin, X., Houmanfar, R., Xie, H., Liang, J., She, X., & Emerson, D. B. (2021). On oversampling imbalanced data with deep conditional generative models. *Expert Systems with Applications*, *169*, 114463. https://doi.org/10.1016/j.eswa.2020.114463

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Cost-Sensitive Learning. En A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, & F. Herrera, *Learning from Imbalanced Data Sets* (pp. 63–78). Springer International Publishing. https://doi.org/10.1007/978-3-319-98074-4_4

Gada, M., Haria, Z., Mankad, A., Damania, K., & Sankhe, S. (2021). Automated Feature Engineering and Hyperparameter optimization for Machine Learning. *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, *1*, 981–986. https://doi.org/10.1109/ICACCS51430.2021.9441668

Hanifi, S., Cammarono, A., & Zare-Behtash, H. (2024). Advanced hyperparameter optimization of deep learning models for wind power prediction. *Renewable Energy*, *221*, 119700. https://doi.org/10.1016/j.renene.2023.119700

Hawkins, D. M. (1980). *Identification of Outliers*. Springer Netherlands. https://doi.org/10.1007/978-94-015-3994-4

Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. En *Expert Systems with Applications* (Vol. 193). Elsevier Ltd. https://doi.org/10.1016/j.eswa.2021.116429

Howard, A., Bouchon-Meunier, B., Lei, J., Vesta, M., & Abbass, H. (2019). *IEEE-CIS Fraud Detection. Kaggle.* [dataset]. https://kaggle.com/competitions/ieee-fraud-detection

Huang, L., Zhao, J., Zhu, B., Chen, H., & Broucke, S. V. (2020). An Experimental Investigation of Calibration Techniques for Imbalanced Data. *IEEE Access*, *8*, 127343–127352. https://doi.org/10.1109/ACCESS.2020.3008150

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *Springer Texts in Statistics An Introduction to Statistical Learning*. http://www.springer.com/series/417

Kaggle. (2021). *Credit Card Fraud Detection: Anonymized credit card transactions labeled as fraudulent or genuine*. Machine learning Kaggle, Mienye y Sun. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Karthik, V. S. S., Mishra, A., & Reddy, U. S. (2022). Credit Card Fraud Detection by Modelling Behaviour Pattern using Hybrid Ensemble Model. *Arabian Journal for Science and Engineering*, *47*(2), 1987–1997. https://doi.org/10.1007/s13369-021-06147-9

Kendall, A., & Gal, Y. (2017). *What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?* https://doi.org/10.48550/ARXIV.1703.04977

Kim, M., & Hwang, K. B. (2022). An empirical evaluation of sampling methods for the classification of imbalanced data. *PLoS ONE*, *17*(7 July). https://doi.org/10.1371/journal.pone.0271260

King, G., & Zeng, L. (2001). Logistic Regression in Rare Events Data. *Political Analysis*, *9*(2), 137–163. https://doi.org/10.1093/oxfordjournals.pan.a004868

Kittler, J. (1998). Combining Classifiers: A Theoretical Framework. En *Pattern Analysis & Applic* (Vol. 1, pp. 18–27). Springer-Verlag London Limited.

Kull, M., Filho, T. M. S., & Flach, P. (2017). Beyond Sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, *11*(2), 5052–5080. https://doi.org/10.1214/17-EJS1338SI

Kumar, A. S., Sarawagi, S., & Jain, U. (2018). Trainable Calibration Measures for Neural Networks from Kernel Mean Embeddings. *Proceedings of the 35th International Conference on Machine Learning*, 2805–2814. https://proceedings.mlr.press/v80/kumar18a/kumar18a.pdf

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2016). *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. https://doi.org/10.48550/ARXIV.1612.01474

Lebichot, B., Verhelst, T., Le Borgne, Y.-A., He-Guelton, L., Oble, F., & Bontempi, G. (2021). Transfer Learning Strategies for Credit Card Fraud Detection. *IEEE Access*, *9*, 114754–114766. https://doi.org/10.1109/ACCESS.2021.3104472

Liu, D., & Lee, J.-H. (2022). CFLedger: Preventing chargeback fraud with blockchain. *ICT Express*, *8*(3), 352–356. https://doi.org/10.1016/j.icte.2021.06.001

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M.-S., & Zeineddine, H. (2019). An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection. *IEEE Access*, *7*, 93010–93022. https://doi.org/10.1109/ACCESS.2019.2927266

Mienye, I. D., & Sun, Y. (2021). Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Informatics in Medicine Unlocked*, *25*, 100690. https://doi.org/10.1016/j.imu.2021.100690

Mienye, I. D., & Sun, Y. (2023). A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection. *IEEE Access*, *11*, 30628–30638. https://doi.org/10.1109/ACCESS.2023.3262020

Muaz, A., Jayabalan, M., & Thiruchelvam, V. (2020). A Comparison of Data Sampling Techniques for Credit Card Fraud Detection. En *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 11, Número 6). www.ijacsa.thesai.org

Nguyen, Nghia, et al. "A Proposed Model for Card Fraud Detection Based on CatBoost and Deep Neural Network." IEEE Access, vol. 10, 2022, pp. 96852–96861, ieeexplore.ieee.org/abstract/document/9882043, https://doi.org/10.1109/ACCESS.2022.3205416.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting Good Probabilities With Supervised Learning. *In Proceedings of the 22nd international conference on Machine learning*. https://doi.org/10.1145/1102351.1102430

Nikolaou, N., Edakunni, N., Kull, M., Flach, P., & Brown, G. (2016). Cost-sensitive boosting algorithms: Do we really need them? *Machine Learning*, *104*(2–3), 359–384. https://doi.org/10.1007/s10994-016-5572-x

Nyberg, O., & Klami, A. (2021). Reliably Calibrated Isotonic Regression. En K. Karlapalem, H. Cheng, N. Ramakrishnan, R. K. Agrawal, P. K. Reddy, J. Srivastava, & T. Chakraborty (Eds.), *Advances in Knowledge Discovery and Data Mining* (Vol. 12712, pp. 578–589). Springer International Publishing. https://doi.org/10.1007/978-3-030-75762-5_46

Olowookere, T. A., & Adewale, O. S. (2020). A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach. *Scientific African*, *8*. https://doi.org/10.1016/j.sciaf.2020.e00464

Paldino, G. M., Lebichot, B., Borgne, Y.-A. L., Siblini, W., Oblé, F., Boracchi, G., & Bontempi, G. (2022). The role of diversity and ensemble learning in credit card fraud detection. *Advances in Data Analysis and Classification*. https://doi.org/10.1007/s11634-022-00515-5

Pan, F., Ao, X., Tang, P., Lu, M., Liu, D., Xiao, L., & He, Q. (2020). Field-aware Calibration: A Simple and Empirically Strong Method for Reliable Probabilistic Predictions. *Proceedings of The Web Conference 2020*, 729–739. https://doi.org/10.1145/3366423.3380154

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

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv Large Margin Classif 10(3):61–74. En *Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.). Advances in Large Margin Classifiers*.

MIT Press. https://home.cs.colorado.edu/~mozer/Teaching/syllabi/6622/papers/Platt1999.pdf

Pozzolo, A. D., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), 3784–3797. https://doi.org/10.1109/TNNLS.2017.2736643

Rakhshaninejad, M., Fathian, M., Amiri, B., & Yazdanjue, N. (2022). An Ensemble-Based Credit Card Fraud Detection Algorithm Using an Efficient Voting Strategy. *Computer Journal*, *65*(8), 1998–2015. https://doi.org/10.1093/comjnl/bxab038

Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit Card Fraud Detection Using AdaBoost and Majority Voting. *IEEE Access*, *6*, 14277–14284. https://doi.org/10.1109/ACCESS.2018.2806420

Saporta, G., & Maraney, S. (2022). *Practical fraud prevention: Fraud and AML analytics for Fintech and eCommerce, using SQL and Python*.

Sartor, D., Barbariol, T., & Susto, G. A. (2024). Bayesian active learning isolation forest (B-ALIF): A weakly supervised strategy for anomaly detection. *Engineering Applications of Artificial Intelligence*, *130*, 107671. https://doi.org/10.1016/j.engappai.2023.107671

scikit-learn. (2013). *Probability calibration*. ECML PKDD Workshop: Languages for Data Mining and Machine Learning. https://scikit-learn.org/stable/modules/calibration.html

Silva Filho, T., Song, H., Perello-Nieto, M., Santos-Rodriguez, R., Kull, M., & Flach, P. (2023). Classifier calibration: A survey on how to assess and improve predicted class probabilities. *Machine Learning*, *112*(9), 3211–3260. https://doi.org/10.1007/s10994-023-06336-7

Soleymanzadeh, R., Aljasim, M., Qadeer, M. W., & Kashef, R. (2022). Cyberattack and Fraud Detection Using Ensemble Stacking. *AI*, *3*(1), 22–36. https://doi.org/10.3390/ai3010002

Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. K. (2008). Credit Card Fraud Detection Using Hidden Markov Model. *IEEE Transactions on Dependable and Secure Computing*, *5*(1), 37–48. https://doi.org/10.1109/TDSC.2007.70228

Tavana, P., Akraminia, M., Koochari, A., & Bagherifard, A. (2023). An efficient ensemble method for detecting spinal curvature type using deep transfer learning and soft voting classifier. *Expert Systems with Applications*, *213*, 119290. https://doi.org/10.1016/j.eswa.2022.119290

Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2010.5596486

Tran, G.-L., Bonilla, E. V., Cunningham, J. P., Michiardi, P., & Filippone, M. (2018). *Calibrating Deep Convolutional Gaussian Processes*. https://doi.org/10.48550/ARXIV.1805.10522

Viaene, S., Derrig, R. A., & Dedene, G. (2004). A case study of applying boosting naive bayes to claim fraud diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, *16*(5), 612–620. https://doi.org/10.1109/TKDE.2004.1277822

Wallace, B. C., & Dahabreh, I. J. (2012). Class Probability Estimates are Unreliable for Imbalanced Data (and How to Fix Them). *2012 IEEE 12th International Conference on Data Mining*, 695–704. https://doi.org/10.1109/ICDM.2012.115

Wang, K., Xue, Q., & Lu, J. J. (2021). Risky Driver Recognition with Class Imbalance Data and Automated Machine Learning Framework. *International Journal of Environmental Research and Public Health*, *18*(14), 7534. https://doi.org/10.3390/ijerph18147534

Wilks, D. S. (1990). On the Combination of Forecast Probabilities for Consecutive Precipitation Periods. *Weather and Forecasting*, *5*(4), 640–650. https://doi.org/10.1175/1520-0434(1990)005<0640:OTCOFP>2.0.CO;2

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, *4*, 29–39. https://www.researchgate.net/publication/239585378_CRISP-DM_Towards_a_standard_process_model_for_data_mining

Zadrozny, B., & Elkan, C. (2001). *Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In ICML (pp. 609–616)*. 609–616.

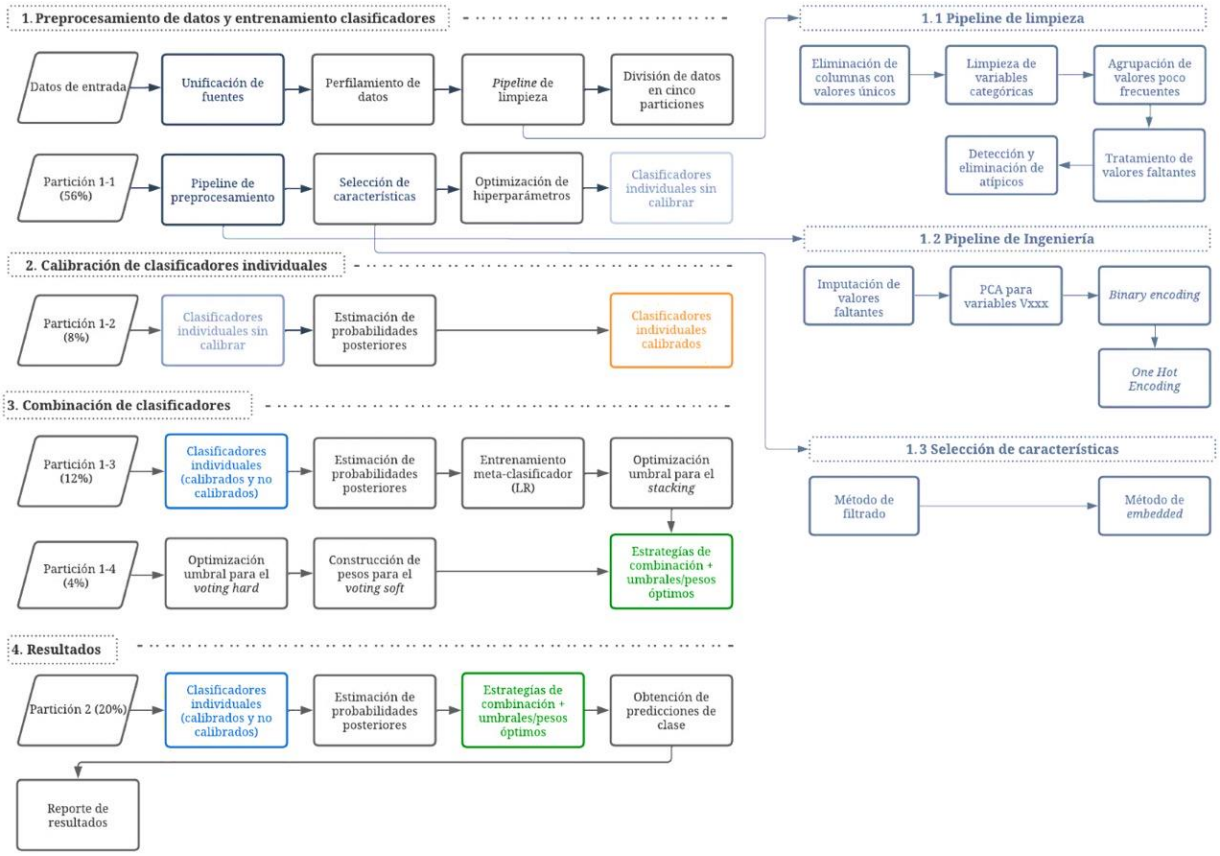
Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 694–699. https://doi.org/10.1145/775047.775151

Zhang, C., Cao, L., & Romagnoli, A. (2018). On the feature engineering of building energy data mining. *Sustainable Cities and Society*, *39*, 508–518. https://doi.org/10.1016/j.scs.2018.02.016

Zhang, G., Sun, S., Duvenaud, D., & Grosse, R. (2017). *Noisy Natural Gradient as Variational Inference*. https://doi.org/10.48550/ARXIV.1712.02390

Anexos

Anexo 1. Proceso metodológico en extenso



Anexo 2. Pipeline de limpieza

En el análisis exploratorio y la evaluación de la calidad de los datos, se utilizaron dos *frameworks* de código abierto de "Python": *dataprep* (*create_report: generate profile reports from a pandas DataFrame — DataPrep 0.4.0 documentation*), que permite generar informes de perfilamiento de un *DataFrame* de "pandas", y *deepchecks* (*Data Integrity Suite Quickstart — Deepchecks Documentation*) que ofrece herramientas para verificar la integridad de los datos.

Los reportes generados se pueden encontrar en los anexos externos "Anexo 2A_Data Profiling.html[20]" y "Anexo 2B_Data Integrity.html". Con los resultados obtenidos, se identificaron las siguientes alertas:

- El 26,9% de las filas tienen valores faltantes en alguna columna. Algunas variables, como dist1 y d11, tienen el 100% de sus valores faltantes.
- Las variables VXXX presentan distribuciones similares y una alta correlación entre ellas. Estas variables representan el 47% del total de las variables del conjunto de datos.
- La mayoría de las variables numéricas tienen una distribución asimétrica.
- Hay variables categóricas con una alta cardinalidad, como card1, que tiene más de 8000 valores únicos.
- Existen variables que son constantes o casi constantes, como algunas del grupo C1-C14 y otras del conjunto VXXX, con todos sus valores en cero y que no aportan información relevante.
- Hay problemas de multicolinealidad, especialmente entre las variables C1-C14 y VXXX.
- Se detectan categorías con variaciones debido a un etiquetado inconsistente, como macosx1012 y Mac OS X 10_12, que se refieren al mismo sistema operativo.
- Existen categorías con etiquetas de muy baja frecuencia en ocurrencia, menor al 5%.

[20] Se deja una muestra con las primeras 150 variables.

Teniendo en cuenta lo anterior en este trabajo se presenta un método para el preprocesamiento de datos tabulares que consta de las siguientes etapas:

1. Eliminación de columnas con valores únicos: se descartan aquellas variables que no aportan información al análisis por tener un solo valor posible.
2. Limpieza de columnas categóricas: se estandarizan los valores de estas variables convirtiéndolos a minúsculas, eliminando caracteres especiales y espacios en blanco.
3. Agrupación de valores poco frecuentes: se identifican los niveles de las variables categóricas que tienen una frecuencia menor a un umbral determinado y se reemplazan por una nueva categoría llamada "Otro".
4. Tratamiento de valores faltantes: se eliminan las filas que tienen todos los valores nulos o que superan un porcentaje máximo de nulos por columna. Para los valores nulos restantes en las variables categóricas, se crean una nueva categoría denominada "*missing*".
5. Detección y eliminación de valores atípicos: se aplican uno de los tres métodos disponibles para identificar y descartar los valores que se alejan significativamente de la distribución de los datos:
    - *Isolation Forest*: un algoritmo no supervisado basado en árboles de decisión que aísla las explícitamente las anomalías en función de sus propiedades cuantitativas, aprovechando el hecho de que las anomalías son minoritarias y tienen valores de atributos muy diferentes de los casos normales (Liu et al., 2008).
    - *Elliptic Envelope*: esta técnica asume que los datos siguen una distribución normal y ajusta una envolvente elíptica alrededor de ellos (Aganagic & Okounkov, 2020).
    - *Local Outlier Factor*: es un método que mide la desviación de la densidad local de una observación respecto a sus vecinos más cercanos, entendiendo el valor atípico no como una propiedad binaria, sino como el grado en que el objeto está aislado de su entorno circundante (Breunig et al., 2000).

Anexo 3. Validación del muestreo

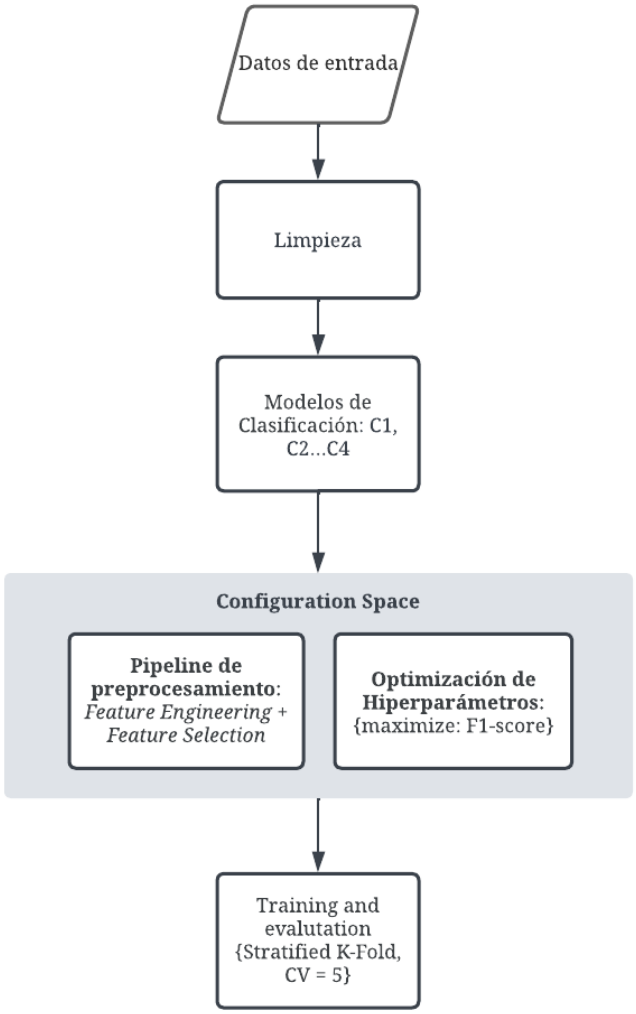
Se utiliza la *suite train-test validation de deepchecks* (*Train-Test Validation Suite Quickstart — Deepchecks Documentation*) para verificar que los subconjuntos de entrenamiento y prueba sean adecuados para el análisis. Con este paquete se evalúan las siguientes condiciones sobre los datos:

Tabla 4

Train-Test Validation Suite

Check	**Condición**	**Hallazgos**
Comparación de los tamaños de los conjuntos de datos.	La relación *test-train* es mayor a 0.1.	La relación de tamaño del tren de prueba es 0,36.
Categorías no observadas sobre test.	La proporción de muestras con una nueva categoría es menor o igual al 0%.	No se encontraron características relevantes para verificar.
Mezcla de muestras entre entrenamiento y prueba.	El porcentaje de muestras de datos de prueba que aparecen en los datos de entrenamiento es menor o igual al 5 %	Porcentaje de muestras de datos de prueba que aparecen en los datos de entrenamiento es: 0,98 %
Cambio de correlación en *feature label*	La diferencia del poder predictivo entre las características de prueba y entrenamiento es inferior a 0,2	Aprobado para 55 columnas relevantes.
Feature Drift	Puntaje de deriva sobre datos categóricos y numéricos < 0.2	La deriva más alta la tiene la columna numérica: pca18 = 0.0083
Label Drift	Puntaje de deriva sobre el label < 0.15	Puntuación de deriva del *label* usando Cramer's V es 0
Multivarie Drift	Puntaje de deriva < 0.25	Valor de deriva encontrado de: 0, correspondiente a un clasificador de dominio AUC de: 0.49
Nuevas etiquetas sobre test	El número de nuevos valores del *label* es menor o igual a 0.	Se encontraron 0 etiquetas nuevas en los datos de prueba.
Comparación de discrepancias en *strings*	No se permiten nuevas variantes en los datos de prueba.	No se encontraron columnas relevantes para verificar.

Anexo 4. Arquitectura de entrenamiento clasificadores



Anexo 5. Medidas de desempeño

Tabla 5

Métricas de evaluación para clasificadores y métodos combinados

Métrica	Fórmula	Enfoque
Accuracy (ACC)	$ACC = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$	Métrica general de la efectividad del clasificador.
Precisión (P)	$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$	Mide el porcentaje de resultados que son relevantes.
Recall (R / TPR)	$R = \frac{TP}{\mathrm{TP}+\mathrm{FN}}$	Porcentaje del total de resultados relevantes que fueron clasificados correctamente.
F_β Score (F_β)	$F_\beta = \frac{1+\beta^2}{\frac{\beta^2}{R}+\frac{1}{P}}$	Media armónica entre la precisión y la sensibilidad. Permite definir en qué medida es más importante la sensibilidad que la precisión.
AUC- ROC	$ROCAUC = \int TPR(FPR)\, d(FPR)$	Evalúa el performance general, al comparar la sensibilidad o tasa de verdaderos positivos, con la tasa de falsos positivos.
PR AUC	$PRAUC = \int P(R)\, d(R)$	Tiende a usarse para conjuntos de datos muy desequilibrados en los que se optimiza para la clase positiva.
False Positive Rate (FPR)	$FPR = \frac{FP}{\mathrm{TN}+\mathrm{FP}}$	Es la probabilidad de que ocurra un resultado positivo cuando el verdadero valor es negativo.
False Negative Rate (FNR)	$FNR = \frac{FN}{\mathrm{TN}+\mathrm{FP}}$	Es la probabilidad de que ocurra un resultado negativo cuando el verdadero valor es positivo.
Brier Score	$BS = \frac{1}{N}\sum_{i=1}^{N}(y_i - P_i)^2$	Es el error cuadrático medio entre el resultado real y las probabilidades estimadas.
Logarithmic loss (Log Loss)	$LL = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(P_i) + (1-y_i) log(1-P_i))$	Mide que tan cercana esta la probabilidad de predicción del valor real. En comparación con BS, asigna mayor peso a los errores más grandes.

Anexo 6. Otras métricas de clasificación

Figura 6

Matriz de confusión – Probabilidades no calibradas

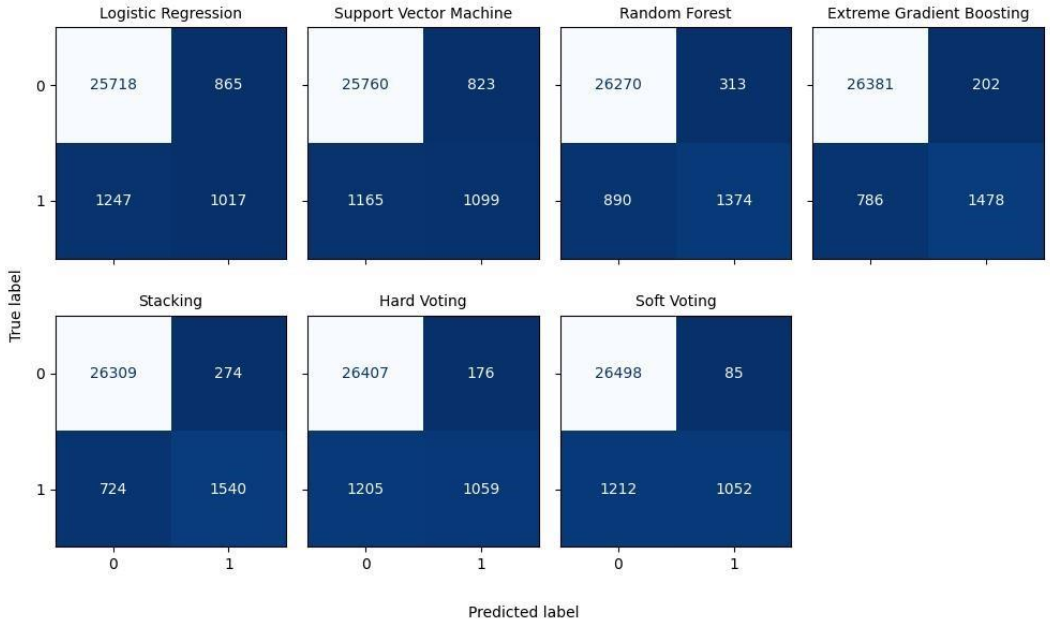


Figura 7

Matriz de confusión – Probabilidades calibradas

