Universidad de Costa Rica

Escuela de Matemática

Modelo predictivo de Fraude Crediticio

Por:

Juan Carlos Aguilar Alfaro, A90081

Manfred D. Porras Rojas, B86119

Holmar Rivera, B86564

Dominick Rodríguez Trejos, B76600

Julio 2024

Índice

Índice de figuras

1. Introducción

En los últimos años, el fraude crediticio ha ido en aumento, afectando no solo a las personas, sino también a entidades financieras tanto a nivel nacional como internacional. En la era digital, con la mayoría de las entidades financieras ofreciendo servicios digitales y un incremento en las transacciones en línea, el fraude crediticio ha encontrado nuevas vías para expandirse. Hay varias formas en las que este tipo de fraude puede presentarse, como el robo de identidad, la manipulación de documentos y transacciones fraudulentas, entre otras.

Es fundamental evitar el fraude crediticio para proteger la estabilidad de los sistemas financieros y resguardar a los clientes de estas instituciones. Si no se aborda adecuadamente esta problemática, las repercusiones pueden ser devastadoras, tanto en el aspecto económico como en la confianza que los clientes depositan y en la reputación de las instituciones financieras.

Nuestro proyecto responde a este desafío proponiendo la incorporación de métodos y algoritmos sofisticados para la detección y prevención del fraude crediticio. Mediante el uso de técnicas de análisis de datos y machine learning, pretendemos detectar posibles actividades fraudulentas al identificar patrones y comportamientos sospechosos. Además de permitir una detección más rápida y precisa, este enfoque jugará un papel fundamental para disminuir considerablemente la cantidad de fraudes, lo cual brindará protección a los usuarios y reforzará el sistema financiero.

Nuestra propuesta tiene una gran importancia para la sociedad en el presente. Nuestro objetivo es ayudar a crear un entorno financiero más seguro y confiable mediante el uso de tecnologías avanzadas e innovadoras. Al implementar estas soluciones, nuestro objetivo es no solo reducir los riesgos relacionados con el fraude en créditos, sino también fortalecer la confianza en el uso de servicios financieros digitales.

Podemos definir fraude crediticio como lo hace Afriyie et al (2023) [1] en términos bastante sencillos como el uso de la tarjeta de crédito de un tercero sin su permiso previo, ya sea por hurto del plástico o por otros medios. Esta definición sirve como un punto de partida, el concepto en sí es simple y fácil de entender, pero puedo complicarse bastante en cuanto a la manera en la que se clasifican los delitos de este tipo y en como los criminales aprovechan las herramientas que tienen a disposición para cometer el delito.

A la definición anterior es sensato añadir un tipo diferente de fraude crediticio, el cuál no involucra el robo directo de la tarjeta de crédito de otro individuo, Baesens et all (2015) [3] brinda la definición de fraude en solicitudes, application fraud en ingles, él cuál consiste en el robo o manufacturación de credenciales e información personal para la solicitud de tarjetas de crédito, las cuales son usadas lo más rápido posible para evitar detección, en casos esta práctica tiene efectos sobre el crédito de los individuos cuyos datos han robado.

Con una definición más concreta de fraude crediticio podemos indagar un poco más en como se desarrolla esta problemática en la actualidad, ahora más que nunca el acceso a la información sensible de nuestras cuentas de crédito es sumamente importante, en un mundo de transacciones digitales y en linea, resguardar y mantener la información de nuestras cuentas crediticias en secreto es fundamental en la batalla contra el fraude, ya que con acceso a esta información es muy sencillo para los estafadores realizar transacciones sin consentimiento previo.

Para finalizar con esta sección, y continuar con una exploración de los diversos métodos y algoritmos investigados, se debe recalcar el impacto que ha tenido esta problemática en nuestra sociedad; como ejemplo tome lo sucedido entre los años 2021 y 2022 [10] época en la cuál los casos de fraude acumularon números preocupantes alcanzando su máximo durante el mes de marzo del 2022, donde el Organismo de investigación Judicial (OIJ) recibió un total de 722 denuncias por fraude, nótese que las 4886 denuncias de fraude reportadas en la publicación de Semanario Universidad provienen solamente de San José, lo cuál es alarmante.

Considerando todo lo anterior expuesto, y la dirección que nuestra sociedad esta tomando hacia un mundo de transacciones digitales y de métodos de pago digitales, nuestro estudio es fundamental para el desarrollo correcto de nuestro futuro financiero, por lo que se continua con la tarea de explorar los diversos métodos y recursos disponibles para solucionar esta problemática.

2. Metodología

2.1. Machine Learning

El aprendizaje automático es una subcategoría de la inteligencia artificial (IA). Su objetivo principal radica en comprender cómo se estructuran los datos y asociarlos a modelos que puedan ser comprendidos y utilizados por humanos. Aunque la inteligencia artificial y el aprendizaje automático suelen ser mencionados juntos, son conceptos distintos. La IA es un concepto más amplio que abarca máquinas de toma de decisiones, aprendizaje de nuevas habilidades y resolución de problemas, mientras que el aprendizaje automático es un conjunto dentro de la IA que capacita a los sistemas inteligentes para aprender de forma autónoma a partir de datos. [7]

Existen diferentes formas en las que una máquina aprende. En algunos casos, los modelos son entrenados con cierta idea en mente, mientras que en otros casos, aprenden por sí mismos. Existen tres tipos principales de aprendizaje automático:

- Aprendizaje supervisado: El aprendizaje supervisado es una técnica diseñada para aprender a partir de ejemplos proporcionados. En este enfoque, el proceso de aprendizaje se asemeja a tener un maestro supervisando el proceso de enseñanza. [7]
- Aprendizaje no supervisado: No se ocupan de datos etiquetados, lo que significa que existen datos de entrada pero no una variable de salida correspondiente. Esto es lo opuesto al aprendizaje automático supervisado. En el aprendizaje no supervisado, los usuarios no necesitan enseñar o supervisar el modelo. No hay una salida correcta ni un supervisor para enseñar. El algoritmo mismo aprende de los datos de entrada y descubre los patrones e información de los datos para aprender y agrupar los datos según similitudes.
- Aprendizaje de reforzamiento: El último tipo de aprendizaje automático. Se trata de un aprendizaje basado en retroalimentación. En el aprendizaje por refuerzo, la máquina aprende automáticamente utilizando retroalimentación sin datos etiquetados. Aquí, el modelo aprende por sí mismo a partir de su experiencia [7].

Para el desarrollo del proyecto, como se había propuesto desde el inicio, se aplicaron varios modelos de predicción a una base de datos para determinar si una observación es potencialmente fraude o no.

Se aplicaron un total de cinco modelos, cada uno con sus peculiaridades y además se aplicaron dos métodos de ensamblaje por medio de los cuales se esperaba observar mejores resultados a los obtenidos por solamente uno de los modelos.

A continuación podemos ver una breve descripción y explicación de cada uno de los modelos utilizados y al final de los dos métodos de ensamblaje.

2.2. Modelos de predicción

2.2.1. XGBoost

El modelo Extreme Gradient Boosting o XGBoost, es un algoritmo ideal para aprendizaje supervisado, este modelo se basa en Boosting, que se define por la manera en la que el modelo hace su aprendizaje, los modelos Boosting hacen su aprendizaje en secuencia, creando así algo similar a una serie de modelos donde cada interacción aprende de la anterior hasta llegar al que mejor predice nuestros datos, esto lo logra al hacer un análisis de cuales parámetros o variables de nuestros datos son más importantes para hacer la predicción, y con varias iteraciones logra alcanzar un modelo que toma en cuenta las variables más importantes para la predicción.

Cortez (2022) define el modelo XGBoost como una implementación de otro modelo llamado Gradient Boosting [5], él cual funciona muy similar a lo descrito anteriormente con el detalle de que en vez de ajustar el peso o importancia de las variables, lo que busca es que cada iteración corrija a su anterior, el nombre Extreme Gadient Boosting proviene del hecho de que el modelo permite hacer uso del potencial computacional del equipo, ya que es posible parametrizar su proceso.

2.2.2. Regresión Logística

Es bastante similar a los modelos de regresión lineal pero la gran diferencia es que la Regresión Logística es específicamente útil al momento de predecir variables llamadas dicotómicas [8], o variables que pueden ser o no pueden ser, como en nuestro caso que una observación sea fraude o no sea fraude.

A diferencia de una regresión lineal, la regresión logística permite observar la probabilidad de que la variable tome un valor en vez del otro, en nuestro caso la probabilidad de que una observación sea fraude, de esta manera es diferente pues no predice directamente, si no que toma en consideración la probabilidad de que sea fraude para su predicción, y así, el modelo de regresión logística toma en cuenta las variables de nuestros datos y a partir de ellas calcula la probabilidad de que una observación en concreto sea fraude o no y toma su decisión de predicción en base a esto.

2.2.3. Random forest

El modelo de Random Forest, también conocido como bosque aleatorio, es un algoritmo de aprendizaje supervisado que se utiliza tanto para problemas de clasificación como de regresión. Desarrollado por Leo Breiman y Adele Cutler, este modelo se basa en la creación de múltiples árboles de decisión, lo que permite mejorar la precisión de las predicciones y reducir el riesgo de sobreajuste en comparación con los árboles de decisión individuales.

Para construir un Random Forest, se inicia generando múltiples árboles de decisión a partir de variados subconjuntos del conjunto de datos de entrenamiento. Los conjuntos más pequeños se crean utilizando una técnica llamada bootstrap, que implica seleccionar muestras aleatorias con reemplazo de los datos originales. Significa que algunos ejemplos del conjunto de datos pueden aparecer en múltiples subconjuntos, mientras que otros podrían no ser seleccionados. La diversidad en los subconjuntos de datos permite generar diferentes árboles de decisión con características y errores propios.

En cada nodo de cada árbol, se considera un subconjunto aleatorio de las características disponibles para determinar la mejor división. Después de haber creado todos los árboles de decisión, el modelo procede a realizar una votación para hacer predicciones. En cuanto a la clasificación, cada árbol emite un voto por una clase y se elige como predicción final del modelo la clase que obtenga la mayoría de los votos. En cuanto a la regresión, se produce una media de las predicciones de todos los árboles para obtener una predicción final. Al utilizar este método de votación o de promedio, se logra suavizar las predicciones y disminuir la variabilidad, lo cual mejora la precisión del modelo.

2.2.4. LightGBM

LightGBM (Light Gradient Boosting Machine) es un marco de aprendizaje automático desarrollado por Microsoft para tareas de clasificación y regresión. Es una versión optimizada de los algoritmos de boosting, diseñada para ser más rápida y eficiente en términos de memoria en comparación con otros frameworks como XGBoost.

Boosting es el método utilizado por LightGBM para combinar varios modelos débiles, como los árboles de decisión, con el fin de crear un modelo más sólido. A diferencia de esto, utiliza una estrategia de crecimiento basada en hojas (leaf-wise) en lugar del nivel (level-wise), lo cual puede incrementar la exactitud, aunque también aumenta el riesgo de sobreajuste. Además, el uso de histogramas permite discretizar los datos continuos, lo que resulta en una reducción de la complejidad computacional y un aumento en la velocidad del entrenamiento.

Comienza entrenando un modelo simple en LightGBM y se centra en corregir los errores de predicción añadiendo nuevos árboles. Hasta que se alcance un número específico de iteraciones o se cumpla un criterio de parada, este procedimiento continuará repitiéndose.

2.2.5. Support Vector Classification

El modelo Support Vector Classification (SVC) es un algoritmo de aprendizaje supervisado utilizado para clasificar datos en dos clases distintas. Funciona encontrando un hiperplano óptimo en un espacio de características de alta dimensión que maximiza el margen entre las clases. Este margen es la distancia entre el hiperplano y los puntos de datos más cercanos de cada clase, llamados vectores de soporte. El SVC puede manejar datos no linealmente separables mediante el uso del truco del kernel, que mapea los datos a un espacio donde la separación lineal es posible. Es útil en aplicaciones como reconocimiento de patrones, detección de spam y diagnóstico médico, gracias a su capacidad para generalizar bien con datos nuevos y desconocidos.

2.3. Métodos de ensamblaje

En cuanto a los métodos de ensamblaje utilizados vamos a explorar dos opciones StackingClassifier y VotingClassifier. Cabe destacar que ambos tienen el mismo propósito, hacer uso de las predicciones y resultados obtenidos de varios modelos para alcanzar una mejor capacidad predictora, la idea es aprovechar que cada modelo captura y le da importancia a diferentes partes de la base de datos y al combinarlos se obtiene un mejor resultado.

2.3.1. Stacking Classifier

Este método funciona en dos capas [6], la primera es una capa donde se toman varios modelos de predicción (modelos base), se entrenan y ejecutan para obtener la predicción de cada uno, estos resultados se toman y se alimentan a la segunda capa, en esta se toma un modelo de predicción (modelo meta) el cuál tomará los resultados obtenidos en la primera capa y, tomando como base estos resultados, los datos de entrenamiento, y el algoritmo del modelo meta, produce una predicción final que, en teoría, será mejor a la que se obtiene cuando se aplica solamente el modelo por si solo.

2.3.2. Voting Classifier

Similarmente al anterior, este método se basa en los resultados y predicciones de varios modelos bases y así llegar a un resultado más acertado, la diferencia es la manera en como determina cuál es el resultado o predicción más acertada,

en este caso se tienen dos formas diferentes en las que esté método funciona:

- Hard Voting: es similar a una votación democrática, el método recibe la predicción de cada modelo y determina cuál es la predicción correcta dependiendo de cual tiene más votos, o cuál fue la más común entre los modelos bases. Por ejemplo, si de los diez modelos base 6 de ellos determinan que el resultado será 1 entonces esta será la predicción final.
- Soft Voting: esta forma es un poco más compleja a la anterior y se basa en probabilidades, en esta forma el método recibe las probabilidades de la predicción de cada modelo y determina cual será la más adecuada calculando el promedio de las probabilidades de las predicciones y la más alta es la elegida. En nuestro caso, los modelos base alimentan al método con las probabilidades de que una observación sea fraude o no, el método hace los calculos y determina si es fraude o no dependiendo de cual de los dos es más probable según los modelos base.

2.4. Validación Cruzada Estratificada

Por último, nos parece importante hablar un poco de lo que es Validación Cruzada Estratificada, lo cuál concierne directamente a como se maneja nuestro set de datos.

Se basa en tomar el conjunto de datos y dividirlo en k subconjuntos o *folds* en los cuales mantiene la proporción de las clases que se puede apreciar en el set entero, a partir de estos k subconjuntos toma *k-1* subconjuntos y con esos entrena al modelo, para al final hacer las predicciones sobre el *fold* restante, y así sucesivamente hasta usar todos los subconjuntos para la predicción, y al final promedia los resultados obtenidos, concluyendo en predicciones mucho más aceptables y que consideran en mayor medida la distribución de los datos.

2.5. Código

Todo el código utilizado para el proyecto puede encontrarse en el siguiente enlace que lleva al repositorio de Git correspondiente:

Repositorio de Git del Proyecto Grupal

3. Resultados

Es importante mencionar que estos resultados fueron obtenidos a partir de una muestra de 60,000 observaciones de la base de datos original. Esto se realizó debido al gran tamaño de la base de datos. La muestra se estratificó para asegurar que las proporciones entre las clases se mantuvieran consistentes.

3.1. Modelo XGBoost

:

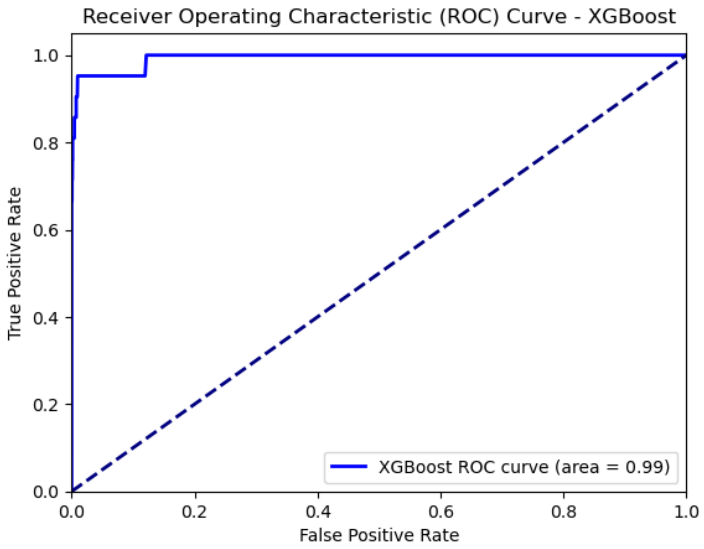

Figura 1: ROC AUC para el conjunto de prueba del modelo XGBoost

El modelo XGBoost mostró el mejor desempeño en términos de la métrica ROC AUC, con un valor de 0.9931 para el conjunto de prueba y 0.9965 para la validación cruzada estratificada.

3.2. Modelo Voting Classifier

:

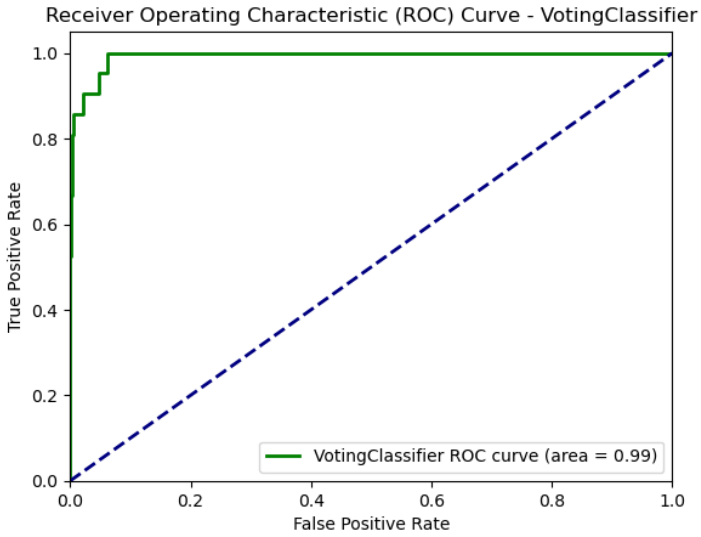

Figura 2: ROC AUC para el conjunto de prueba del modelo Voting Classifier

El modelo ensamblado Voting Classifier, entre los cinco modelos escogidos usando soft voting, quedo en segundo lugar de desempeño en términos de la métrica ROC AUC, con un valor de 0.9929 para el conjunto de prueba y 0.9881 para la validación cruzada estratificada.

3.3. Modelo Stacking Classifier

:

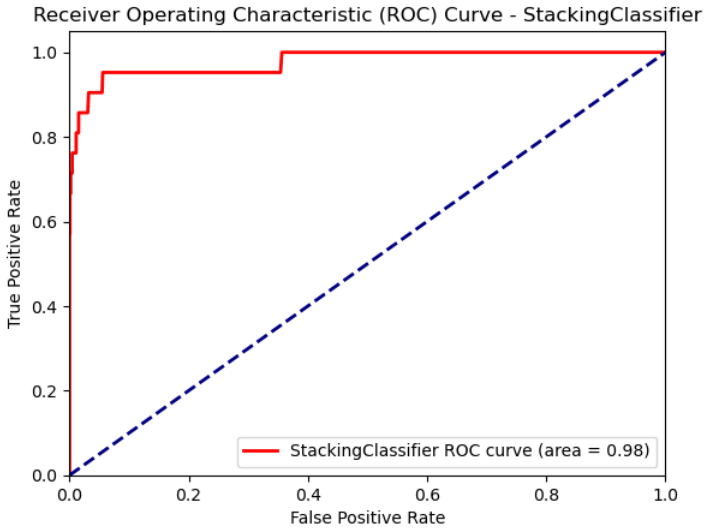

Figura 3: ROC AUC para el conjunto de prueba del modelo Stacking Classifier

El modelo ensamblado Stacking Classifier, que utilizó como modelo meta al XGBoost y otros modelos como estimadores base, mostró el peor desempeño en términos de la métrica ROC AUC. Obtuvo un valor de 0.9773 tanto para el conjunto de prueba como para la validación cruzada estratificada.

Es importante recalcar que, a pesar de los resultados obtenidos al comparar las diferentes métricas ROC AUC, el modelo Voting Classifier parece mostrar mejores resultados cuando el umbral es cercano a 0. A partir de un umbral de 0.4, todos los modelos presentan un desempeño prácticamente perfecto.

La importancia de cada variable para el modelo XGBoost se puede observar en el siguiente gráfico de barras:

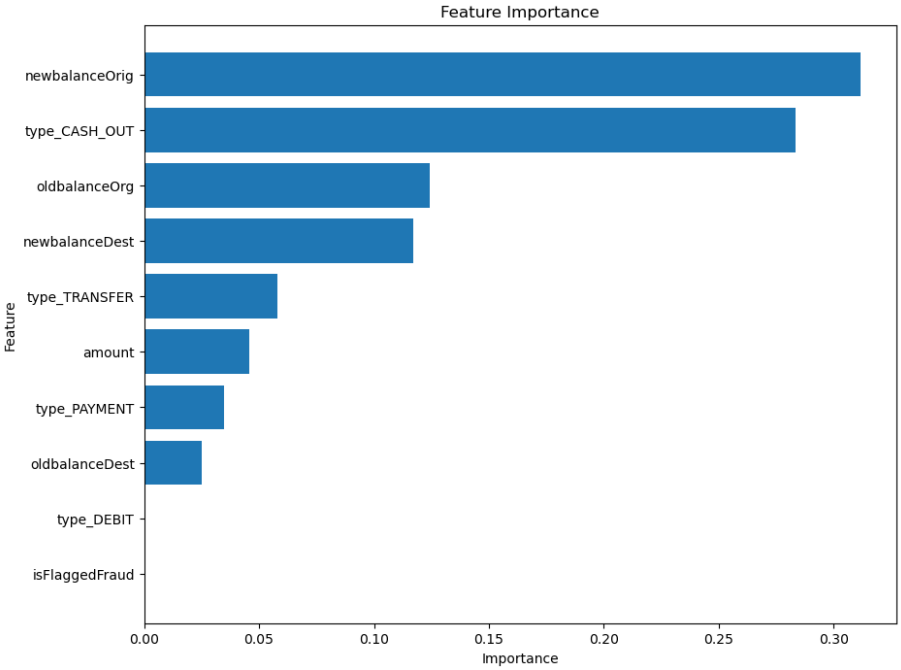

Figura 4: Importancia de cada variable para el modelo XGBoost

4. Conclusiones

- El modelo XGBoost fue el que mostró los mejores resultados a pesar de ser uno de los más sencillos y simples.
- A pesar de implementar métodos de ensamblaje como Stacking y Voting, los resultados obtenidos solamente con el modelo de XGBoost resultaron ser mejores.
- La variable que más brinda información es oldbalanceOrg.
- Los tres modelos aplicados (XGBoost, Stacking y Voting) dieron resultados excelentes en términos de predicción.

5. Recomendaciones

- Considerando el tema de interés, es primordial tener mucho cuidado al manejar esta información tan sensible y personal, y siempre se debe considerar mantener el anonimato de los usuarios ante todo.
- Si es posible, hacer la optimización de los hiperparametros de todos los modelos, pues puede resultar en un aumento significativo en el rendimiento del modelo.
- Al momento de embarcar en este tipo de proyectos, se debe tener en cuenta que algunos de estos modelos, en especial los más complejos, demanda mucho poder de calculo.
- Al momento de aplicar los métodos de ensamblaje se recomienda experimentar con varios modelos y tener una idea de lo que se desea obtener al mezclar estos modelos y cuales se complementan entre ellos.

Referencias

[1] Afriyie J., Tawiah K., Pels W., Addai-Hene S., Dwamena H., Odame E., Ayeh S. y Eshun J. *Decision Analytics Journal*, vol 6, *A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions*, Elsevier, 2023.

[2] Amat, J. *Máquinas de Vector Soporte (Support Vector Machines, SVMs)*, CienciadeDatos.net, 2017. Recuperado del sitio web: Máquinas de Vector Soporte (Support Vector Machines, SVMs)

[3] Baesens B., Van Vlasselaer V. y Verbeke W. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*, Wiley, 2015.

[4] Barrios, A. J. *Light GBM vs XGBoost ¿Cuál es mejor el algoritmo?*, Health Big Data, 2022. Recuperado del sitio web: Light GBM vs XGBoost ¿Cuál es mejor el algoritmo?

[5] Cortez, S.*Introducción a los Métodos de Ensamble y al Algoritmo de XGBoost: Caso Práctico.*, Medium, 2022. Recuperado del sitio web: Introducción a los Métodos de Ensamble y al Algoritmo de XGBoost: Caso Práctico.

[6] GeeksforGeeks. *Stacking in Machine Learning*, GeeksforGeeks: Sanchhaya Education Private Limited, 2021. Recuperado del sitio web: Stacking in Machine Learning

[7] Hiran, K. K., Jain, R. K., Lakhwani, K., and Doshi, R. *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples*, 1st ed., BPB Publications, 2021.

[8] IBM Corporation. *Regresión Logística*, IBM, 2023. Recuperado del sitio web: Regresión Logística

[9] IBM Corporation. *¿Qué es el random forest?*, IBM, 2023. Recuperado del sitio web: ¿Qué es el random forest?

[10] Zeledon N., *OIJ recibio 4886 denuncias de fraude a cuentas de bancos entre 2021 y 2022*, Semanario Universidad, 2022