ISSN 1870-4069

Detección de micro fraudes en transacciones de tarjetas de crédito utilizando aprendizaje automático: Generación de la base de datos y métodos de detección

Jaime Alberto Quiñones-Beltrán[1], Karina M. Figueroa-Mora [1], Juan Pablo Maldonado-López[2]

[1] Universidad Michoacana de San Nicolás de Hidalgo, México

[2] University of New York in Prague, Czechia

karina.figueroa@umich.mx, pmaldonado@unyp.cz

Resumen. Los micro fraudes en Tarjetas de Crédito (carding en inglés) han ido en aumento en los últimos años, en México se estima que el 12 % de las tarjetas han sido comprometidas. Sin embargo, a pesar de ser un tipo de problema muy interesante para su estudio, existen limitadas bases de datos con este tipo de información. Lo cual es justificado pues debe ser prioritaria la seguridad de la información de los clientes para las compañías bancarias. Por lo tanto, en este artículo se propone una herramienta que permite construir bases de datos sintéticas con información de transacciones en tarjetas bancarias, donde un porcentaje muy pequeño de ellas son consideradas fraudulentas. La construcción de estas bases de datos sintéticas refleja ciertas hipótesis sobre el comportamiento de posibles clientes, lo cual es reflejado en este artículo considerando diversas distribuciones de transacciones en clientes. Con estas bases de datos sintéticas se analizaron con técnicas de aprendizaje de máquina buscando determinar si fuera posible la detección de dichas transacciones fraudulentas.

Palabras clave: Detección de fraudes, aprendizaje de máquina, finanzas.

Detection of Micro Frauds in Credit Card Transactions Using Machine Learning: Database Generation and Detection Methods

Abstract. Micro frauds in Credit Cards (carding in English) have been increasing in recent years. In Mexico, it is estimated that 12% of cards have been compromised. However, despite being a very interesting problem for study, there are limited databases with this type of information, which is justified since the security of customer information must be a priority for banking companies. Therefore, this article proposes a tool that allows the building of synthetic databases with information on bank card transactions, a small fraction of which

are considered fraudulent. The construction of these synthetic databases reflects certain hypotheses about the behavior of potential clients, as evidenced by the consideration of various distributions of client transactions in this article. These synthetic databases were then subjected to a meticulous analysis using advanced machine learning techniques, ensuring the validity and reliability of the research findings, to determine if it was possible to detect these fraudulent transactions.

Keywords: Fraud detection, machine learning, finance.

1. Introducción

Las tarjetas de crédito (TC) se han convertido en uno de los principales métodos de pago usados en el mundo, lo que las ha llevado a ser un objetivo común por defraudadores. Se sabe de antemano que la detección de patrones en fraudes en este tipo de transacciones es deseable pero muy difícil por diversas causas. Una de estas es el hecho de tener un enorme volumen de transacciones diarias que hace que la detección de fraudes sea casi imposible para un humano. En [1] muestran que las pérdidas globales durante el año 2022 por fraudes a tarjetas de pago ascienden a los 34 mil millones de dólares.

Existen diversos tipos de fraudes, entre ellos: el robo de la tarjeta física, la clonación, el robo de los datos para su uso en tiendas virtuales, etc. [5]. Un tipo de fraude casi indetectable por el mismo dueño de las tarjetas son los micro fraudes (carding) pues se hacen pasar por transacciones normales de cantidad moderada y realizados en sitios usuales. Por ejemplo, el costo de la entrada a una función de cine con una descripción relacionada a tiendas comunes. Al menos el 12 % de los mexicanos han sido víctimas de un ataque de este estilo [2].

Otro gran problema es que las compañías bancarias no revelan bases de datos con este tipo de transacciones por razones de seguridad. Por lo tanto, estudiar el problema es prácticamente imposible. Básicamente solo se tiene acceso a una base de datos en el sitio web de Kaggle [3], sin embargo, ésta tiene distintos tipos de fraudes y los datos son información resultante de algún PCA, además la información solo es de una institución bancaria. En este artículo se presenta una herramienta de creación de datos sintéticos con información bancaria y con registros marcados como micro fraudes. Además se muestra el análisis con técnicas de aprendizaje de máquina empleadas sobre ellos.

2. Antecedentes

La única base de datos existente y que además es ampliamente usada se puede ver en [3]. Esta base de datos tiene 284,807 registros de los cuales solo el 0.172 % (492) son fraudes con transacciones en septiembre de 2013 hechas por clientes europeos. La evolución del Carding durante la pandemia del 2020 ha sido enorme. Las debilidades de esta base de datos son evidentes, en 2013 los patrones de fraudes han evolucionado fechas recientes y el número de transacciones diarias por cliente es completamente distinto a lo popularizado de hoy en día. Esta base de datos solo contiene información

Algorithm 1 Generación de la base de datos

```
function GENERAR_DATASET(c=n_clientes, w=n_webs, f=n_días, t=fecha_inicio)
    tabla_perfil_clientes ← GENERAR_TABLA_PERFIL_CLIENTES(n_clientes)
    tabla_perfiles_web ← GENERAR_TABLA_PERFILES_WEB(n_webs)
    transacciones_df ← TRANSACCIONES(cliente, tabla_perfil_clientes, tabla_perfiles,
f=n_días) return transacciones_df
end function
carding_victimas ← RANDOM.CHOICE(clientes, tamaño=12 %(clientes))
transacciones_df['CARDING'] ← 0
for clientes en carding_victimas do
    4_transacciones ← RANDOM.CHOICE(transacciones_del_cliente, tamaño=4)
    para cada transacción en 4_transacciones hacer
    transacciones_df[transacción, 'cantidad'] ← GENERAR_FRAUDE
    transacciones_df[transacción, 'CARDING'] ← 1
end for
```

Algorithm 2 Generación de las transacciones

```
function TRANSACCIONES(cliente, tabla_perfil_clientes, tabla_perfiles, f=n_días)).
    for cliente en tabla_perfil_clientes do
        GENERAR_TABLA_TRANSACCIONES(webs, n_días)
    end for
end function
```

numérica la cual es resultado de una transformación PCA. Los autores argumentan razones de confidencialidad para no poner disponibles los datos originales ni el contexto de las 28 características con las que cuenta. La únicas características no procesadas son el tiempo y la cantidad. Los resultados sobre esta base de datos presentado son casi del 100 % en precisión y recall.

Finalmente, dada la construcción de la base de datos, ésta no permite concentrarse en un tipo específico de fraude. En [4] el autor describe una revisión de la aplicación de modelos de Aprendizaje de Máquina en la detección de fraudes en tarjetas de crédito a la base de datos del sitio web de Kaggle [3]. El autor concluye que a pesar de haber comparado nueve técnicas no asegura tener un rendimiento óptimo.

Los autores en [5] presentan una revisión de la literatura sobre el uso de modelos de Aprendizaje de Máquina y Aprendizaje Profundo en ciberfraudes con tarjetas de crédito. En este trabajo se examinaron 181 artículos publicados entre 2019 y 2021, donde 108 utilizaban Aprendizaje de máquina, 34 Aprendizaje Profundo y 39 una combinación de ambos.

La mayoría de los algoritmos aplicados fueron supervisados, destacando Random Forest, SVM y Regresión logística. El artículo concluye destacando la importancia de que los bancos proporcionen conjuntos de datos públicos con diversos tipos de fraudes para investigaciones futuras.

En la mayoría de los trabajos aseguran no poder compartir los datos por cuestiones de seguridad. Finalmente, en [6] se presenta una metodología para detectar fraudes en tarjetas de crédito. Éste incluye la creación de datos sintéticos y la predicción de fraudes mediante técnicas de Aprendizaje de Máquina y profundo.

Algorithm 3 Generación de la tabla de perfil de clientes

```
function GENERAR_TABLA_PERFIL_CLIENTES(c=n_clientes, w=n_webs, f=n_días,
t=fecha_inicio)
    for cliente_id en rango(c) do
      monto_promedio ← DISTRIBUCIÓN_UNIFORME(5, 100)
      desv_est_monto ← monto_promedio / 2 num_transacciones_promedio_por_dia ←
DISTRIBUCIÓN_UNIFORME(1, 5)
      añadir [cliente_id,
            monto_promedio, desv_est_monto,
            num_transacciones_promedio_por_dia]
    end for
end function
```

Algorithm 4 Generación de la tabla de perfil web

```
function GENERAR_TABLA_PERFIL_WEB(c=n_clientes, w=n_webs, f=n_días, t=fecha_inicio)
    for web_id en rango(n_webs) do
      porcentajes← %Porcentajes de las categorías más compradas %
      categoría ← ELECCIÓN_ALEATORIA(porcentajes)
      categoría_id ← CODIFICAR(categoría)
      añadir [web_id, categoría, categoría_id]
    end for
end function
```

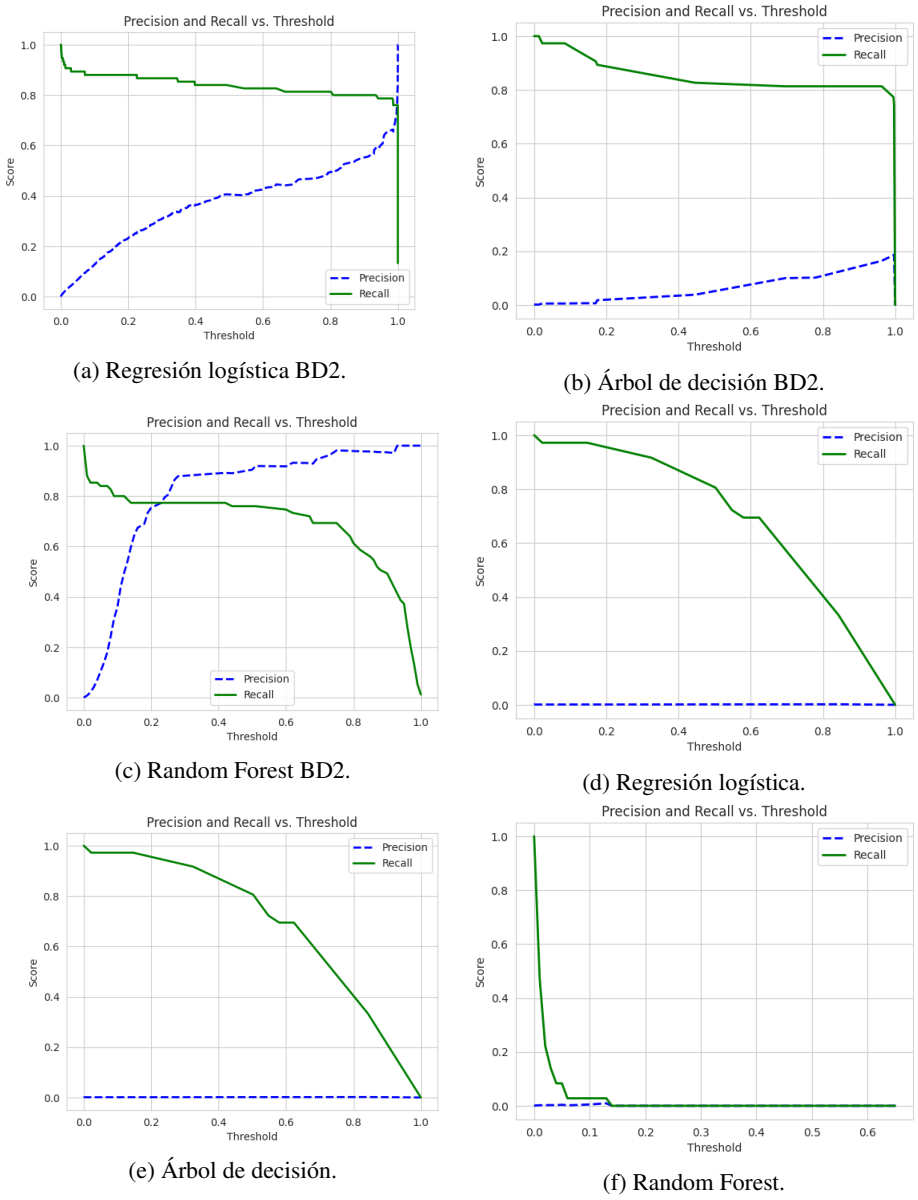
La Base de datos generada considera transacciones hechas en terminales físicas dentro de un radio establecido por usuario. Así mismo, genera tres tipos de fraudes, de los cuales en uno de ellos lo considera fraude siempre que sea mayor a 220. Esta es la principal literatura en este tipo de bases de datos sintéticas, y que además es coherente con las prácticas convencionales.

3. Propuesta

En este trabajo se presenta una herramienta de creación de datos sintéticos con transacciones bancarias, algunos de los cuales serán marcados como micro-fraude. El algoritmo propuesto considera tres premisas, que se describen a continuación:

- **Premisa 1**: Un consumidor escoge una distribución uniforme entre 1 y 5 el número de compras que usualmente se realizan en un día.
- **Premisa 2**: Para cada compra, el consumidor escoge una distribución uniforme entre 5 y 100 para decidir cuanto gasta en ese día.
- **Premisa 3**: el atacante escoge (estocásticamente) al 12 % de los consumidores, y los estafa, con montos dentro de una distribución normal con parámetros (μ, σ). Estos parámetros pueden ser calculados estadísticamente por una institución bancaria a partir de un fraude detectado.

En la siguiente sección se describe a detalle la creación de la base de datos.



(a) Regresión logística BD2.

(b) Árbol de decisión BD2.

(c) Random Forest BD2.

(d) Regresión logística.

(e) Árbol de decisión.

(f) Random Forest.

Fig. 1. Curva precision/recall en el conjunto de prueba para la base de datos de referencia (izquierda a,b,c) y en nuestra base de datos sintética (derecha d,e,f).

3.1. Generación de base de datos de micro fraudes

Para la creación de transacciones bancarias se consideró la creación de perfiles de clientes c y páginas web w de transacciones (el origen de la ésta). El primer paso fue crear una tabla de clientes que reflejara comportamientos de gastos únicos por

Tabla 1. Resultados de tres técnicas de aprendizaje de máquina en la base de datos en el conjunto de prueba.

Modelo	Precision	Recall	F1
LR	0.001	0.778	0.002
DT	0.001	0.806	0.002
RF	0	0	0

Tabla 2. Resultados en la base de datos de referencia sobre el conjunto de prueba.

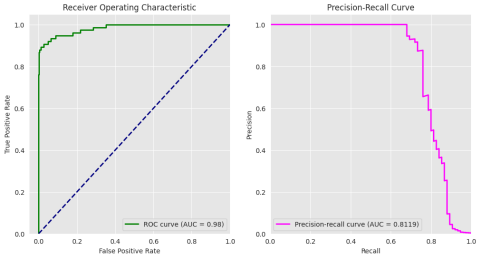
Modelo	Precision	Recall	F1
LR	0.403	0.827	0.542
DT	0.099	0.813	0.176
RF	0.891	0.76	0.82

cliente. Se generaron c=3,000 clientes con características específicas, como un monto promedio de transacción distribuido uniformemente en el intervalo m=[5, 100) y un número promedio de transacciones diarias también distribuido uniformemente en el rango f =[1, 5) (*premisas 1 y 2*). Estos números, en la práctica, pueden ser adaptados a los datos propios del banco.

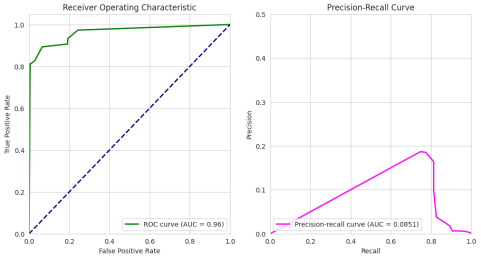
Posteriormente, se generó una segunda tabla que representaba w=100 páginas web donde se realizaron las transacciones. La generación de fraudes se llevó a cabo tomando el c_a =12 % de los clientes como usuarios afectados. En cada uno de estos casos, se seleccionaron estocásticamente f=4 transacciones para ser reemplazadas por transacciones fraudulentas que siguieron una distribución normal (*premisa 3*). Nuevamente, el porcentaje y cantidad de transacciones es adaptable a las necesidades.

Como resultado y de acuerdo a lo esperado, se obtuvo un conjunto de datos altamente des-balanceado, donde las transacciones etiquetadas como fraudes representaban solo el 0.06 % del total. Este conjunto de datos final fue de 2,605,617 registros y 9 características. Estas son: etiqueta, transacción_id, tiempo, cliente_id, web_id, categoría_web, cantidad, tiempo_segundos, tiempo_días (estos últimos dos entre transacciones). Todos los datos fueron procesados para convertir las características en números enteros o reales, lo que resultó en un conjunto de datos final listo para su uso en modelos de detección de fraude de la siguiente forma:

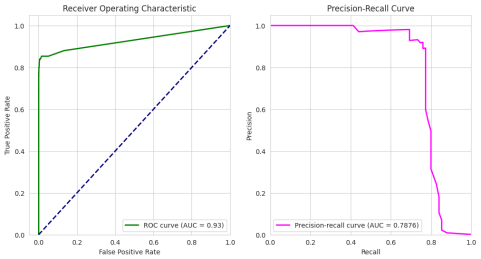
- **etiqueta**, un indicador consecutivo
- **web_id**, un id de un sitio web donde se efectuó la operación
- **cantidad**, monto de la transacción
- **durante_entresemana**, Tiene un 1 si la transacción fue realizada entre semana
- **durante_noche**, tiene un 1 si la transacción fue realizada de noche
- **número_transacciones_por_cliente_1día_ventana**, cuenta el número de transacciones realizadas anteriormente por cliente en 1 día



(a) Regresión logística.



(b) Árbol de decisión.

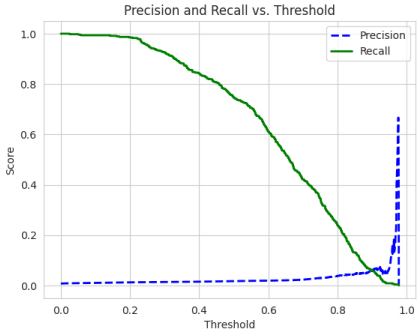


(c) Random Forest.

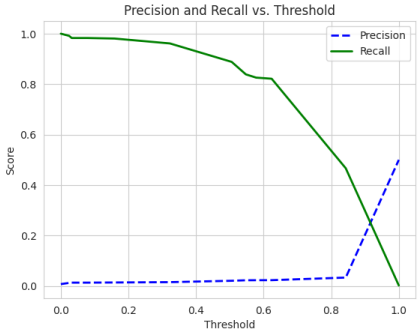
Fig. 2. Curvas ROC y precision-Recall en el conjunto de prueba para la base de datos de referencia.

– **promedio_cantidad_por_cliente_1día_ventana**, calcula el promedio de las transacciones realizadas anteriormente por cliente en 1 día
– **número_transacciones_por_cliente_7días_ventana**, cuenta el número de transacciones realizadas anteriormente por cliente en 7 días
– **promedio_cantidad_por_cliente_7días_ventana**, calcula el promedio de las transacciones realizadas anteriormente por cliente en 7 días
– **número_transacciones_por_cliente_30días_ventana**, cuenta el número de transacciones realizadas anteriormente por cliente en 30 días
– **promedio_cantidad_por_cliente_30días_ventana**, promedio de las transacciones realizadas anteriormente por cliente en 30 días

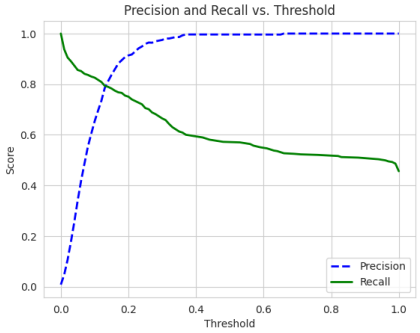
El pseudocódigo se presenta en los algoritmos 1, 2, 3, 4 mostrados a continuación.



(a) Regresión logística.



(b) Árbol de decisión.



(c) Random Forest.

Fig. 3. Curva precision/recall en el conjunto de prueba de la base de datos sintética con aumento de fraudes al 0.2 %.

3.2. Modelos estudiados

Una vez que es posible tener los datos sintéticos se analizaron con las siguientes propuestas: regresión logística (LR) [7] , Árboles de Decisión (DT) [8] con una profundidad de 4 utilizando la impureza Gini; y por último Random Forest (RF) [7, 9] usando 100 estimadores y la impureza de Gini sin profundidad máxima.

Métricas. Las métricas utilizadas para medir el desempeño de los modelos fueron precision, recall y F1 [7]. La matriz de confusión es una tabla que describe el rendimiento de un modelo de clasificación en un conjunto de datos, mostrando la cantidad de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN). La precisión (precision) mide la proporción de predicciones positivas que fueron correctas, y se calcula como:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

La sensibilidad (recall) mide la proporción de instancias positivas que fueron correctamente identificadas por el modelo, y se calcula como:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

El F1-score es la media armónica de la precisión y el recall, lo que proporciona un equilibrio entre ambas métricas. Se calcula como:

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{3}$$

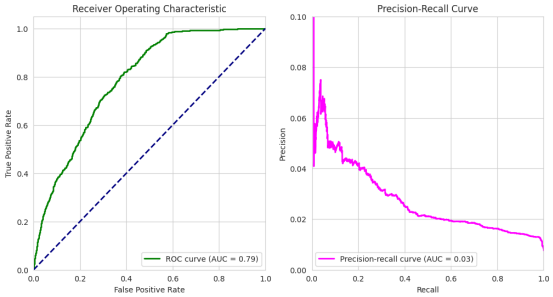
4. Experimentación

Para trabajar con esta base de datos evidentemente desbalanceada en las clases se hizo una validación presecuencial con folds=4 aplicados a tres modelos de Aprendizaje de Máquina: Regresión Logística, Árbol de Decisión y Bosque Aleatorio. Además de ello, se utiliza la técnica SMOTE (Synthetic Minority Oversamplic Technique) para lograr el balance de las clases. El conjunto de entrenamiento tuvo un 70 % y el de prueba 30 %. Los resultados se muestran en la tabla 1. Note que se priorizó un recall alto (threshold=0.5), con excepción del bosque aleatorio, el cual no logra destacar con sus predicciones.

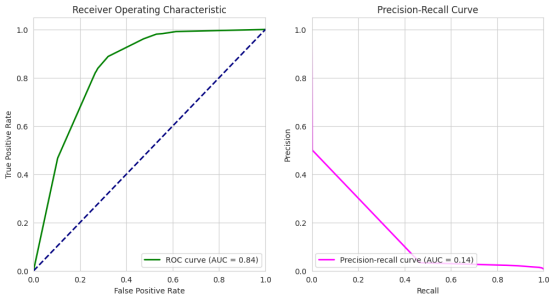
Considerando que solo existe una base de datos [3], la comparación se llevó a cabo uniformizando los criterios a ella, es decir, se usaron casi la misma cantidad de datos, es decir, 303,868. También se incrementó el número de fraudes en esa cantidad, esto es 178 fraudes (0.058 % del total). Los resultados de esta segunda base de datos (a la que llamaremos BD2) se muestran en la tabla 2. En ellos se destaca el desempeño de RF.

4.1. Variando el umbral

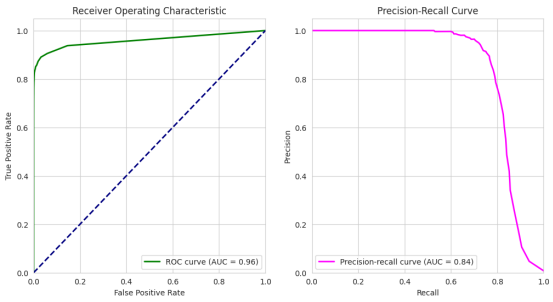
[t!] En esta sección se presenta el resultado de esta variar el umbral o threshold, es decir, cambiar la cantidad de fraudes detectados sobre la BD2. Evidentemente ampliar el umbral permite catalogar como fraudulentas las transacciones que tienen probabilidades más bajas como puede verse en la figura 1. En estas gráficas es claro observar el valor del umbral para cada algoritmo y mantener un equilibrio entre precisión y sensibilidad. Esto se puede ver reflejado también en la figura 2, donde los valores AUC resultan ser aceptables en general.



(a) Regresión logística.



(b) Árbol de decisión.



(c) Random Forest.

Fig. 4. Curvas ROC y precisión-recall en el conjunto de prueba de la base de datos sintética con aumento de fraudes al 0.2 %.

Si hacemos lo mismo con nuestra base de datos sintética, es posible conseguir valores más altos de recall al disminuir el umbral. Sin embargo, la precisión no logra despuntar debido a la muy poca cantidad de fraudes lo cual compromete considerablemente esta métrica. En la columna derecha de la figura 1 es posible el desempeño de los modelos sobre nuestra base de datos.

4.2. Variando el porcentaje de fraudes

La base de datos sintética tiene un mayor nivel de des-balance entre clases, lo que compromete su desempeño. Al aumentar prematuramente la cantidad de fraudes (clase 1) de tal forma que llegue del 0.058 % al 0.2 % de transacciones marcadas como fraude,

se consigue tener el mismo porcentaje de la base de datos de referencia. Los resultados obtenidos se muestran en la figura 3. Note que tanto la regresión logística como el árbol de decisión se vuelven ligeramente más precisos conservando un recall alto. Por otro lado, el Random Forest supera con creces a los dos modelos anteriores, a la par que con un umbral de $0{,}5$ se obtienen un F1 del $0{,}769$ que supera al resultado del conjunto de datos por referencia con este mismo umbral. De la misma manera, las curvas de la figura 4 presentan valores más altos comparados con el primer experimento.

5. Conclusiones y trabajo a futuro

Las transacciones a tarjetas de crédito han crecido enormemente en los últimos años, es de destacarse que el comportamiento de los clientes recientemente es muy distinto al de años previos a la pandemia. Además, es remarcable que también los fraudes han crecido considerablemente, en especial el conocido como micro fraude, el cual es de gran complejidad debido a lo desapercibidos que pueden ser incluso para el mismo cliente, básicamente se resume a transacciones con montos pequeños con nombres de establecimientos comunes (por ejemplo, un cargo en un centro de entretenimiento). Aunado a lo anterior, al no haber bases de datos públicas se vuelve aun más difícil encontrar un comportamiento de estos fraudes.

La propuesta presentada en este artículo consiste tanto en la generación de datos sintéticos de transacciones bancarias con micro fraudes controlada por diversos parámetros como el número de clientes afectados, en este caso al 12 %, y sustituyendo solamente 4 de todas sus transacciones como fraude, siguiendo una distribución normal. El código para la generación de las bases de datos sintéticas se encuentra disponible en[3]. En el presente trabajo se pudo observar que al incrementar la cantidad los fraudes se logró conseguir mejores resultados con las técnicas de aprendizaje de máquina empleadas.

En particular se resalta la técnica de Random Forest o bosques aleatorios, que superaron con su desempeño a la regresión logística y a los árboles de decisión. Es claro que al tener números tan bajos de transacciones fraudulentas, se vuelve una tarea compleja predecir tales movimientos con una precisión perfecta. Sin embargo, se insta a conseguir niveles de sensibilidad o recall más altos pues esto significa conseguir todos los fraudes, lo que a la vez se traduce a menor número de pérdidas monetarias para la ambas partes.

En conclusión, con esta herramienta mostramos que es viable para predecir micro fraudes en tarjetas de crédito de los que se tenga información sobre su distribución. Es decir, los modelos de aprendizaje automático si podrían distinguir entre datos generados de dos distribuciones, uniforme y normal. En la vida real, un banco podría hacer un análisis de sus propios datos y reemplazar los valores mencionados (1 y 5,5 y100, 12 %) por los propios, a los que tiene acceso en su data warehouse, así como estimar (μ, σ), que también tiene manera de estimar. Como trabajo a futuro se propone probar otros modelos que puedan ser más aptos para este problema, tal como PBC4cip para el balanceo de clases, entre otros.

[3] github.com/jqbeltran/Carding

Referencias

1. Radage, K.: Credit card fraud in 2023 (2023) https://www.clearlypayments.com/blog/credit-card-fraud-in-2023/
2. Rodríguez, D.: "Me hicieron un cargo al mes sin darme cuenta": Así son las estafas virtuales a las tarjetas de los mexicanos. EL PAÍS México (2022) https://elpais.com/mexico/2022-10-21/me-hicieron-un-cargo-al-mes-sin-darme-cuenta-asi-son-las-estafas-virtuales-a-las-tarjetas-de-los-mexicanos.html
3. Kaggle.: Worldline and the machine learning group of université libre de bruxelles: Credit card fraud detection (2023) https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?select=creditcard.csv
4. Faraji, Z.: A review of machine learning applications for credit card fraud detection with a case study. SEISENSE Journal of Management (2022) https://journal.seisense.com/jom/article/view/770.
5. Marazqah-Btoush, E. A. L., Zhou, X., Gururajan, R., Chan, K. C., Genrich, R., Sankaran, P.: A systematic review of literature on credit card cyber fraud detection using machine and deep learning. PeerJ Computer Science, vol. 9, pp. e1278 (2023) doi: 10.7717/peerj-cs.1278
6. Le Borgne, Y., Siblini, W., Lebichot, B., Bontempi, G.: Reproducible machine learning for credit card fraud detection - practical handbook. Université Libre de Bruxelles (2022) https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook.
7. Géron, A.: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. 2nd edition, O'Reilly Media, Inc., Sebastopol, CA (2019)
8. Shalev-Shwartz, S. Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge University Press (2014)
9. Kelleher, J., Namee, B., D'Arcy, A.: Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies. 2nd edition MIT Press (2015)