ACTIVIDAD INTEGRADORA FINAL

DIPLOMATURA EN CIENCIA DE DATOS, INTELIGENCIA ARTIFICIAL Y SUS APLICACIONES EN ECONOMÍA Y NEGOCIOS

Alumnos: Caffaratti Donalisio, Vania; Martinatto, Iván; Maldonado Arana Paula Constanza

Introducción

Con el surgimiento de las nuevas tecnologías, los procesos de compra y venta se han ido modernizando cada vez más y adaptándose a este nuevo paradigma. En los últimos años, con la pandemia de coronavirus, esta modernización se aceleró a un ritmo casi exponencial. El e-commerce, las billeteras digitales, los pagos online y otros conceptos son algo ya común en nuestra vida diaria. Sin embargo, todas estas nuevas soluciones y ventajas, traen aparejadas consigo nuevos problemas o el aumento de otros ya existentes. Este último es el caso de los fraudes con tarjetas de crédito.

Según el diario Clarín (2022), los delitos de ciberseguridad se quintuplicaron en la Argentina desde los primeros meses de la pandemia. En este sentido, ante semejante crecimiento de casos, es necesario brindar soluciones que sean eficaces y rápidas. Las compañías crediticias deben poder detectar fácilmente transacciones fraudulentas de tarjetas de crédito para no cargar a sus clientes con gastos que no les corresponden. De esta manera, al eliminar las externalidades negativas a las que se enfrentan los usuarios al sufrir, por ejemplo, una clonación de sus tarjetas, las empresas mejoran la experiencia de los servicios financieros que otorgan.

Es así que entra en juego la Ciencia de Datos, y en especial los algoritmos de Machine Learning para enfrentar el problema de detección de fraude crediticio. La aplicación de modelos predictivos adecuados ayudará a disminuir las tasas de fraude, ya sea encontrando desvíos de patrones o comportamientos llamativos. De esta manera, impactará de manera positiva en la confianza de los clientes en el negocio financiero.

Por lo tanto, el objetivo de esta investigación es detallar la metodología de aprendizaje automático utilizada para el análisis de fraude. Las preguntas que se intentan responder son las siguientes: ¿Qué actividades tienen más incidencia en la ejecución de una maniobra fraudulenta vía tarjeta de crédito?¿Es posible identificar transacciones fraudulentas mediante modelos de clasificación supervisada? Para su abordaje, se propone primero un análisis exploratorio de los datos y luego se estiman dos modelos: árbol de decisión y análisis de regresión logística[1].

Datos

Descripción

Para el desarrollo de este informe se utilizó la base ***card_transdata*** provista por la Diplomatura en Ciencia de Datos e Inteligencia Artificial de la Universidad Nacional de Córdoba. El set de datos cuenta con muchos clientes, existen un millón de transacciones de tarjeta de crédito (número de filas) e incluye siete variables que serán utilizadas como variables explicativas para comprender el comportamiento de la variable dependiente o target (***'fraud'***). La información que se tiene es la siguiente:

[1] En el siguiente repositorio de Github se puede acceder a la base de datos y al código de trabajo

Variable	Descripción
distance_from_home	Distancia a la que la transacción ocurrió desde la dirección de facturación
distance_from_last_transaction	Distancia de la última transacción
ratio_to_median_purchase_price	Ratio del monto de transacción sobre la mediana
repeat_retailer	Si la transacción se dió sobre un retailer repetido
used_chip	Si la transacción es por chip
used_pin_number	Si la transacción es por PIN
online_order	Si la transacción es online
fraud	Si la transacción fue o no fraudulenta

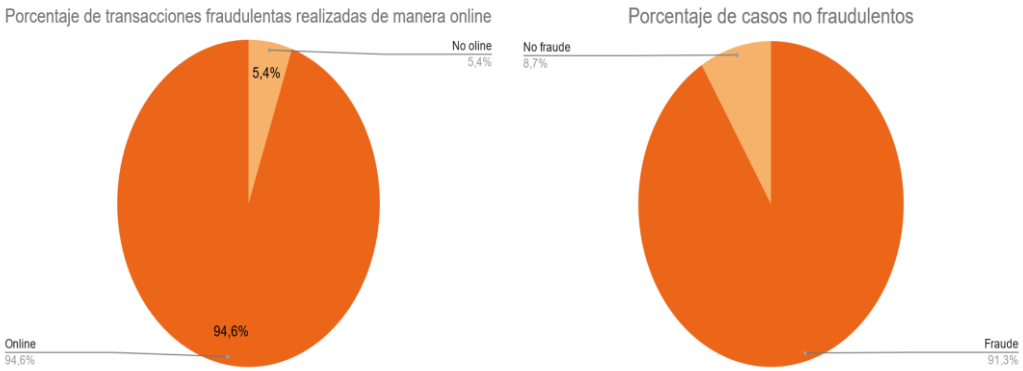
Dividiendo las variables entre numéricas y del tipo categóricas, se observa la siguiente distribución:

Variable	Mínimo	Máximo	Media
distance_from_home	0.004	10632.723	26.628
distance_from_last_transaction	0.000	11851.104	5.036
ratio_to_median_purchase_price	0.004	267.802	1.824

Variable categórica	1	0
repeat_retailer	881.536	118.464
used_chip	350.399	649.601
used_pin_number	100.608	899.392
online_order	650.552	349.448

En cuanto a la variable ***fraud***, la misma es una variable binaria, siendo el valor 1 una transacción fraudulenta y 0 indica que no se han detectado comportamientos irregulares. El número de muestras con fraudes es muy pequeño comparado con el número total de transacciones. Los fraudes representan solo el 8,7% del total.
Este desbalance de clases en la variable objetivo es un elemento muy importante a ser considerado al momento de crear nuestro modelo, ya que tendrá influencia en la precisión para identificar situaciones fraudulentas.

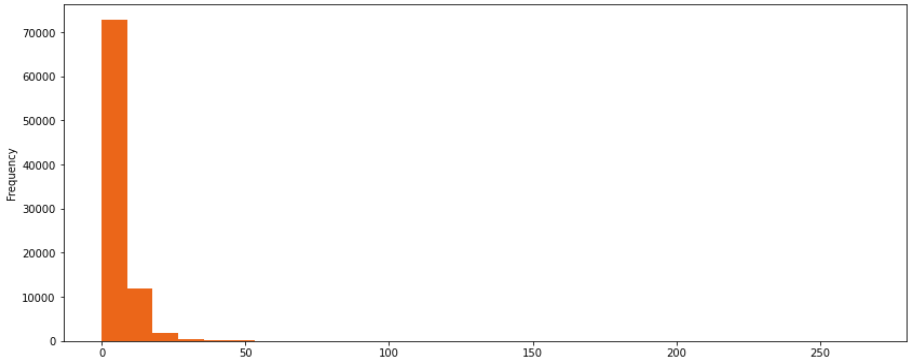
También es importante destacar que de los 87.403 casos de fraude que hay en la base, 82.711 son transacciones realizadas de manera online. Es decir, casi el 95% del total de situaciones fraudulentas presentes en los datos.
Ambas situaciones se pueden apreciar en los siguientes gráficos circulares:



Esto se debe a que la compra online es más riesgosa ya que en una transacción física se brinda mayor seguridad al solicitarse la firma u otra prueba identificadora como el DNI.
Por otro lado, teniendo en cuenta la variable ***used_pin_number***, en el 99,65% de los casos fraudulentos no se utilizó un número de PIN para llevar adelante la compra. A la hora de realizar una transacción mediante tarjeta de crédito, es muy importante que se solicite el número de identificación personal para acceder al sistema, ya que sin éste código numérico el sistema se vuelve más endeble.
Por último, a continuación se muestra la distribución del ratio del monto sobre la mediana en transacciones fraudulentas:

Cantidad	Media	DS	Mínimo	25%	50%	75%	Máximo
87.403	6,01	5,56	0,01	3,50	5,07	7,33	266,69

Como se puede observar en el histograma, los datos se encuentran concentrados a la izquierda, es decir, la mayoría de los valores son pequeños. La distribución claramente no es normal sino que presenta una asimetría positiva.



Tratamiento de datos

En base a los datos descritos previamente, y teniendo en cuenta el análisis exploratorio, procedimos con el tratamiento de los mismos. Ello, para poder limpiar lo más posible la base, en caso de que haya anomalías en la misma, y mejorar los resultados de los modelos a aplicar.

A partir del análisis se pueden sacar algunos puntos de interés: antes que nada, todas las variables parecen ser relevantes en el análisis a primera vista, tal cual se describe en el apartado anterior. Por otro lado, los datos no presentan valores faltantes, por lo que no se requirió recurrir a alternativas para mitigar este posible problema.
Con respecto a los outliers, estos si se encuentran presentes en la base de datos trabajada, en la variable "**distance_from_home**". Usualmente, los outliers se tratan como ruido, es decir, se busca solucionar este sesgo en los datos, pero dada la naturaleza del problema presentado, estos representan información valiosa para detectar casos de fraude crediticio, porque justamente reflejan un comportamiento atípico, lo cual puede llegar a ser un indicador de que la compra no está siendo realizada de manera lícita. Por lo tanto, no se realizó ninguna técnica para eliminar o sustituir estos valores anormales.
Una buena práctica dentro del aprendizaje supervisado es construir modelos que realicen una buena performance al ser ejecutados con nuevos datos. Muchas veces sucede que al momento de construirlo, no se cuenta con esta nueva data (como es nuestro caso), y una posible solución es "simular" mediante procedimientos como el train test split. Básicamente lo que hace esta técnica, es dividir a los datos en dos partes: el training set y el testing set. Como sus nombres lo indican, estos sirven para entrenar al modelo y luego testearlo, respectivamente. Previo a esto, se debe indicar qué variables son "features" y cual o cuales serán "target". Para nuestro caso, utilizamos como target a la variable "**fraud**" y el resto como features, y se dividió a los datos en 80% training set y el 20% restante como testing set.
Luego se estandarizaron los datos, y para ello se entrenó el StandardScaler únicamente con los datos del training set, para finalmente transformar las muestras de test y train con estandarizador entrenado. La razón de esto es que las variables medidas en diferentes escalas no contribuyen de igual manera al ajuste del modelo y al aprendizaje del mismo. lo que puede terminar creando un sesgo. Conocer la distribución de todos los datos podría influir en cómo el modelo detecta y procesa los valores atípicos. Por lo tanto, de esta manera se previene este "data leakage" de la distribución entre el training y el test.
Otro punto a tener en cuenta, y como ya previamente se mencionó, es el desbalance de clases en la variable objetivo. Dado que el modelo está muy balanceado en cuanto a casos de "fraude" por "no fraude", los modelos aplicados podrían caer en un sesgo y asumir que la mayoría de los casos son no fraudulentos. Para solucionar esto, se probaron distintas técnicas de balanceo, principalmente random undersampling y random oversampling, para igualar las clases y que el modelo no tienda hacia una de ellas.
Estos dos enfoques se basan en volver a muestrear aleatoriamente los datos eliminando parte de la clase mayoritaria (undersampling), que en nuestro caso son los "no fraudes", o duplicando la clase minoritaria (oversampling). Para el caso del undersampling, al estar reduciendo la cantidad de información, puede llegar a suceder que la pérdida de datos sea importante y afecte al modelo. Por ello, muchas veces el oversampling ajusta mejor, ya que se evita este problema. Dado esto último, decidimos adoptar la técnica de oversampling.

Modelo

Selección de Modelo

Para poder detectar las transacciones fraudulentas se probaron dos familias de algoritmos de clasificación supervisada: Regresión Logística y algoritmos basados en árboles[2],

[2]Otras opciones podrían haber incluido otros algoritmos basados en árboles como Random Forest y LightGBM o Máquinas de Vectores de Soporte (Support Vector Machines) que son también muy usados para este tipo de problemas.

específicamente un XGBoost (Chen, T., & Guestrin, C. , 2016). Este último dio los mejores resultados de los dos, por lo que fue elegido como el modelo a desarrollar.
Un XGBoost es una implementación de árboles de decisión con gradient boosting diseñados para óptima performance y rapidez. Este tipo de modelo de machine learning es altamente usado y efectivo, y suele llevarse los primeros puestos en las competencias de Kaggle para datos estructurados como los nuestros. Es además un modelo robusto frente a datos atípicos como presenta esta base de datos. El algoritmo se probó para la base de datos original, una versión con subsampling y otra con oversampling para balancear la muestra. A continuación mostraremos las decisiones tomadas y resultados del modelo entrenado para la muestra con oversampling, que como ya mencionamos fue la técnica escogida para este problema.

Tuneo de hiperparámetros

A diferencia de los parámetros, los hiperparámetros deben definirse y optimizarse antes de entrenar un modelo de machine learning. Por lo tanto, en primer lugar se realizó una búsqueda randomizada de parámetros con Randomised Search CV[3], con 5 validaciones cruzadas. Random Search es un método donde se seleccionan combinaciones aleatorias de hiperparámetros para entrenar al modelo. Permite elegir la mejor combinación de hiperparametros que maximice la métrica de evaluación elegida (en nuestro caso el ROC-AUC[4]) con validación cruzada. Los hiperparámetros y sus rangos fueron elegidos en función de su importancia en este tipo de modelos y de la performance que se obtuvo con los mismos. A continuación se muestran los rangos para los cuales se tunearon los hiperparámetros elegidos. Los mejores parámetros se detallan en la tercera columna de la siguiente tabla:

Hiperparámetro	Descripción	Efecto	Rango de búsqueda	Valor elegido
max depth	Máxima profundidad de un árbol	Aumentarlo complejiza el modelo, y hay mayor probabilidad de sobreajuste	[3, 4, 5, 6, 8, 10, 12, 15]	12
min child weight	Suma mínima de peso para hacer una partición adicional en un nodo del árbol	Más grande es, el algoritmo será más conservador	[1, 3, 5, 7]	3
gamma	Reducción mínima de pérdida requerida para hacer una partición adicional en un nodo de un árbol	A mayor gamma, el algoritmo será más conservador	[0.0, 0.1, 0.2, 0.3, 0.4]	0.4
colsample bytree	Es el ratio de subsampleo de	Más grande es, el algoritmo será más complejo	[0.3, 0.4, 0.5 , 0.7]	0.5

[3] No utilizamos un Grid Search a pesar de que es un algoritmo más exhaustivo de búsqueda de hiperparámetros porque es más taxativo computacionalmente.

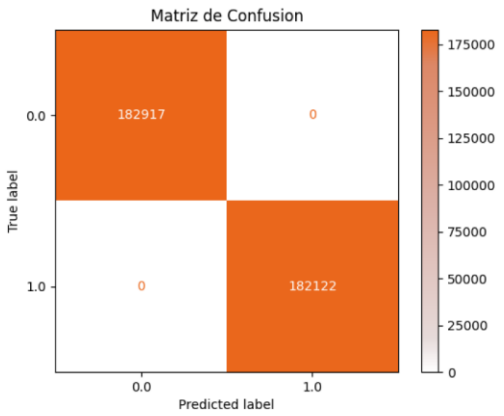
[4] La curva ROC permite ver el trade off entre la proporción de falsos positivos y falsos negativos para diferentes umbrales de clasificación en sistemas de clasificación binarios. El área debajo de esta curva o AUC puede presentar valores entre 0 y 1 y proporciona una medida agregada del rendimiento en todos los umbrales de clasificación posibles. Se eligió como métrica de evaluación principal porque es apropiada para muestras desbalanceadas, y a diferencia de la precisión, recall, y por lo tanto F1 no solo tiene en cuenta una foto con un solo punto de corte posible sino que todas las combinaciones posibles.

	columnas cuando se construye cada árbol			
n estimators	Número de árboles	Más grande es, el algoritmo será más complejo	[50, 100, 200, 300, 400, 500]	400
learning rate	Reduce la contribución de cada árbol multiplicando su influencia original por este valor.	Más chico es, el algoritmo será más complejo	[0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4]	0.3

Resultados

Una vez seleccionados los hiperparámetros, se entrenó el modelo utilizando los datos de la muestra de train para poder predecir la muestra de test.

La matriz de confusión muestra en el eje de las X los valores predichos de las dos clases y en el eje de la Y las clases verdaderas. Todos los valores se agrupan en la diagonal principal, lo que implica que se clasificaron bien todas las muestras del test, por lo que no hay falsos negativos ni falsos positivos. Se muestran además las principales métricas[5].

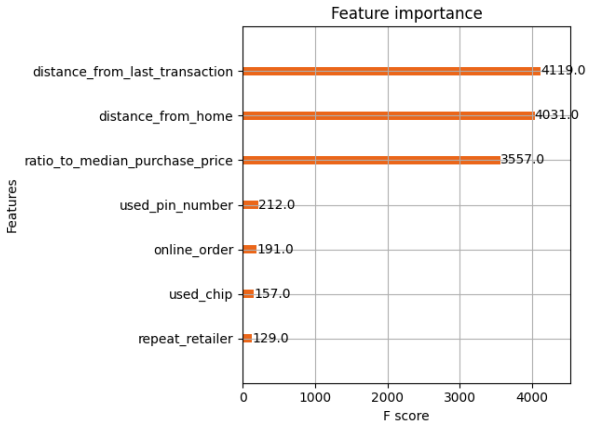


Accuracy	Precisión
1.00	1.00
Recall	**F1**
1.00	1.00
ROC-AUC	
1.00	

El modelo es muy exitoso para predecir qué actividades son fraudulentas, dado que todas las métricas muestran los mejores valores posibles (que van de 0 a 1, siendo 0 el peor valor posible). Un alto accuracy indica que se predijeron la mayoría de las transacciones correctamente. En el caso de muestras desbalanceadas, no es la métrica más acertada. Se puede tener un alto accuracy dado que estos modelos pueden llegar a clasificar correctamente la clase negativa a expensas de la clase positiva minoritaria. Sin embargo, sería un modelo que no es capaz de detectar efectivamente la clase en la que se tiene más interés. Las métricas de precisión y recall tienen un trade off entre ellas, donde la primera puede presentar valores muy altos si se predicen muy pocas transacciones de la clase positiva pero todas correctamente, a pesar de que en ese caso el recall sería muy malo, dado que aumentan los falsos negativos (aumentando el denominador y por lo tanto disminuye el score). La métrica F1, busca tener en cuenta el trade off realizando una media armónica entre los dos.

[5] Además del ROC-AUC se incluyeron otras métricas clásicas para entender la capacidad predictiva del modelo. El accuracy (Número total de predicciones correctas/Número total de predicciones), la precisión (True Positives / True Positives + False Positives), recall (True Positives/False Negatives+True Positives) y el F1 (2*(precision + recall)/(precision*recall)).

Podemos ver en el gráfico de feature importance que las variables más importantes para clasificar a las transacciones fraudulentas fueron “**distance from home**”, “**distance from last transaction**” y “**ratio to median purchase price**”.



Conclusión

Los fraudes de tarjetas de crédito representan un delicado problema de negocio, y pueden llevar a grandes pérdidas, tanto a los clientes como a las empresas. A la hora de realizar una transacción, existen ciertas variables que pueden indicar que la misma no es lícita, tales como la distancia entre transacción y domicilio del usuario, la distancia geográfica entre transacciones consiguientes, el monto de compra, entre otras.

Contar con un historial de datos de esas variables puede ayudar significativamente a reducir la ciberdelincuencia en estos casos. ¿De qué manera? Con una rápida detección de los mismos. Dado nuestro análisis, es posible predecir correcta y eficazmente el fraude crediticio utilizando la Ciencia de Datos y el Machine Learning, y más específicamente, algoritmos basados en árboles.

El mayor desafío en este tipo de problemas es la correcta manipulación de grandes bases de datos, y el criterio de selección de variables relevantes ya que distintas decisiones pueden alterar los resultados modelados, como por ejemplo, optar por eliminar o no outliers, balancear o no una muestra, entre otras.

Si bien los dos modelos entrenados en este trabajo dieron excelentes resultados, el modelo basado en árboles pudo clasificar todas las transacciones fraudulentas de manera exitosa. Sin embargo, es una situación llamativa. Este comportamiento puede darse porque el modelo está tan bien entrenado que no tendría tan buen desempeño si se utilizaran datos de distinta distribución. Por lo tanto, se deberían tomar con cautela las métricas obtenidas.

Referencias

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd international conference on knowledge discovery and data mining* (pp. 785-794).

Clarín.com. (2022, Enero). Crecen los fraudes con tarjetas y los usuarios quedan a la intemperie. Clarín. https://www.clarin.com/economia/crecen-fraudes-tarjetas-usuarios-quedan-intemperie_0_GxvdJ9orc.html