Modelos de machine learning para aprendizaje no supervisado: Detección de anomalías para identificar transacciones fraudulentas con tarjetas de crédito

Karla Orozco
Jonathan Zapata
Juan Esteban Chavarria
Juan Fernando Gallego

Maestría en ciencias de los datos y analítica,
Aprendizaje Automático Avanzado, 2022-1,
Universidad Eafit, Medellín, Colombia

Resumen

Los atacantes cibernéticos desde siempre están al asecho buscado brechas de seguridad en los sistemas financieros o incluso en los sistemas de seguridad de las personas. Al hacer compras en línea con tarjetas de crédito si no tomamos las debidas precauciones nuestros datos o los de nuestras tarjetas pueden terminar fácilmente en las manos de una persona malintencionada que los use para dejar nuestras cuentas vacías. Aquí viene la necesidad de un sistema que pueda rastrear el patrón de todas las transacciones y si cualquier patrón es anormal entonces la transacción debe ser abortada. En este trabajo se planteará la solución al problema en cuestión mediante la aplicación de dos algoritmos de clasificación no supervisada los cuales serán comparados con el fin de decidir cuál de estos representa una solución mas adecuada el problema planteado y finalmente se determinará cuál es el mejor modelo que pueda ser usado en un ambiente real para detectar y abortar las transacciones sospechosas.

1. Introducción

Cuando hacemos cualquier transacción al comprar productos en línea una buena cantidad de personas prefieren las tarjetas de crédito. El límite de crédito de las tarjetas de crédito a veces nos ayuda a hacer compras incluso si no tenemos la cantidad en ese momento. Pero, por otro lado, estas características son mal utilizadas por los atacantes cibernéticos.

Cuando hablamos de seguridad en la vida digital, el principal reto es encontrar la actividad anormal.

Anomalía es un sinónimo de la palabra "outlier". La detección de anomalías (o detección de valores atípicos) es la identificación de elementos, eventos u observaciones poco comunes que levantan sospechas por diferir significativamente de la mayoría de los datos. Las actividades anómalas pueden estar relacionadas con algún tipo de problema o suceso raro, como fraudes bancarios, problemas médicos, defectos estructurales, equipos que funcionan mal, etc.

Para hacer frente a este problema necesitamos un sistema que pueda abortar la transacción si la encuentra sospechosa. Aquí viene la necesidad de un sistema que pueda rastrear el patrón de todas las transacciones y si cualquier patrón es anormal entonces la transacción debe ser abortada.

Hoy en día, tenemos muchos algoritmos de aprendizaje automático que pueden ayudarnos a clasificar las transacciones anormales. El único requisito son los datos pasados y el algoritmo adecuado que pueda ajustarse a nuestros

datos de la mejor manera.

Existen diversos métodos de aprendizaje tanto en el campo supervisado como no supervisado. Aquí, sin embargo, vamos a discutir cómo el aprendizaje no supervisado se utiliza para para encontrar outliers y también entender por qué la detección de anomalías utilizando el aprendizaje no supervisado es beneficioso en la mayoría de los casos.

En este trabajo se planteará la solución al problema en cuestión mediante la aplicación de dos algoritmos de clasificación no supervisada los cuales serán comparados con el fin de decidir cuál de estos representa una solución mas adecuada el problema planteado. En este caso se trabajará con los algoritmos Isolation Forest y Local Outlier Factor.

Finalmente, se obtendrá el mejor modelo que pueda clasificar la transacción en tipos normales y anormales.

2. Modelos usados

1). Isolation forest:

Es un algoritmo de detección de anomalías. Detecta las anomalías utilizando el concepto de aislamiento (la distancia de un punto de datos al resto de los datos), en lugar de modelar los puntos normales.

Este algoritmo utiliza las dos propiedades de las anomalías ("Pocas" y "Diferentes") para detectar su existencia. Como las anomalías son pocas y diferentes, son más susceptibles de ser aisladas. Este algoritmo aísla cada punto de los datos y los divide en valores atípicos o normales. Esta división depende del tiempo que se tarde en separar los puntos. Si intentamos separar un punto que obviamente no es un outlier, tendrá muchos puntos en su ronda, por lo que será realmente difícil de aislar. En cambio, si el punto es un outlier, estará solo y lo encontraremos muy fácilmente.

Isolation forests introduce un método diferente que aísla explícitamente las anomalías utilizando árboles binarios, demostrando una nueva manera de detectar anomalías más rápido que se dirige directamente a las anomalías sin perfilar todas las instancias normales. El algoritmo funciona bien con un gran volumen de datos.

Ventajas:

- Tiene una complejidad temporal lineal baja y un requisito de memoria pequeño
- Es capaz de tratar con datos de alta dimensión con atributos irrelevantes
- Puede ser entrenado con o sin anomalías en el conjunto de entrenamiento
- Puede proporcionar resultados de detección con diferentes niveles de granularidad sin necesidad de reentrenamiento

2). Local Outlier Factor (LOF):

Es un método de detección de anomalías no supervisado que calcula la desviación de la densidad local de un punto de datos determinado con respecto a sus vecinos. Se trata de un cálculo que mira a los vecinos de un determinado punto para averiguar su densidad y compararla con la densidad de los puntos vecinos más adelante. En resumen, podemos decir que la densidad alrededor de un objeto atípico es significativamente diferente de la densidad alrededor de sus vecinos. LOF considera como valores atípicos las muestras que tienen una densidad sustancialmente inferior a la de sus vecinos. Una de las grandes desventajas de este algoritmo es que sólo observa la vecindad local de un punto de datos y, por tanto, no puede hacer predicciones sobre puntos de datos fuera de la muestra. Por eso aquí trabajamos directamente con el conjunto de datos de X_{test}.

3. Conjunto de datos

El dataset contiene transacciones bancarias reales realizadas europeos en septiembre de 2013. Presenta transacciones ocurridas en dos días, donde tenemos 492 fraudes de 284.807 transacciones. La clase positiva (fraudes) representa el 0,172 % de todas las transacciones.

Por motivos de seguridad, no se compartieron las variables reales, sino que disponemos de versiones transformadas de estas (Componentes principales - PCA), una de las ventajas que tenemos con PCA es que los vectores de componentes son independientes entre sí. Como resultado, podemos encontrar 29 variables predictoras y 1 columna de clase final.

Las únicas características que no han sido transformadas con PCA son 'Tiempo' y 'Importe'. La característica "Tiempoçontiene los segundos transcurridos entre cada transacción y la primera transacción del conjunto de datos. La característica Importe.[es] el importe de la transacción; esta característica puede utilizarse para el aprendizaje sensible al costo en función de la muestra. La característica "Clase" es la variable de respuesta y toma el valor 1 en caso de fraude y 0 en caso contrario.

4. Ingeniería de características

Todos conocemos la importancia de unas buenas características para los modelos de aprendizaje automático. En la tarea de aprendizaje automático tenemos características que necesitamos procesar para hacerlas buenas y esto se hace mediante tareas de pre-procesamiento de datos. En la figura 1 se puede observar que el conjunto de datos está desbalanceado (Sólo el 17 % de las transacciones son fraudulentas). Teniendo en cuenta esta proporción de desequilibrio entre las clases, no se puede usar como criterio de exactitud los resultados que arroje una matriz de confusión ya que no es significativa para la clasificación desbalanceada. En vez de eso usamos como criterio de exactitud la medición del Área Bajo la Curva de Precisión-Recuperación (Area Under the Precision-Recall Curve - AUPRC).

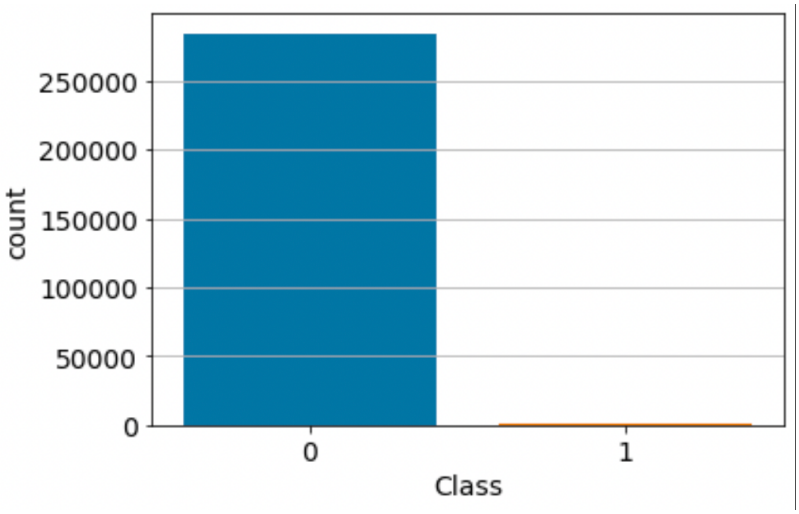


Figura 1: Comparación de las transacciones normales Vs. fraudulentas
Total de transacciones: 284807
Cantidad de transacciones normales: 284315
Numero de transacciones fraudulentas: 492
Porcentaje de transacciones fraudulentas: 17 %

Se verificó la no existencia de valores nulos en los datos.

La variable tiempo contiene los segundos transcurridos entre cada transacción y la primera transacción del dataset. Vamos a transformar esta característica en horas para tener una mejor comprensión:

En la figura 2 se evidencia claramente que la hora del día tiene repercusión en ella cantidad de casos de fraude, se ve como en las horas de la madrugada crece el número de hechos fraudulentos con respecto a los normales.

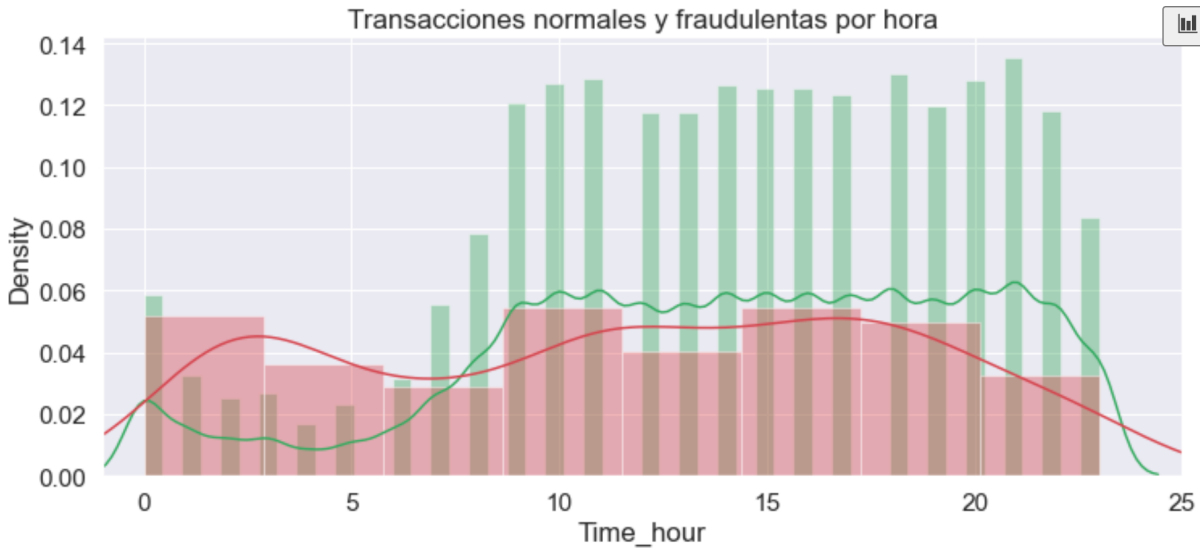


Figura 2: Gráfica de densidad de transacciones normales Vs. fraudulentas por hora del día

4.1. Transformación de características

Se observó que en los datos hay 28 características que son versiones transformadas de PCA, pero el Monto es el original. Y al comprobar el mínimo y el máximo se encontró que la diferencia es enorme, lo cual puede desviar el resultado:

```
(0.0, 25691.16)
```

En este caso es buena práctica escalar esta variable. Procedimos a utilizar un escalador estándar para normalizarla.

4.2. Selección de características

Vamos a realizar una prueba de hipótesis para encontrar/descartar columnas que no sean estadísticamente significativas, si se da el caso. Realizaremos la prueba Z con las transacciones válidas como población.

El caso es que tenemos que encontrar si los valores de las transacciones fraudulentas son significativamente diferentes de las transacciones normales o no para todas las características. El nivel de significancia es 0,01 y es una prueba de dos colas, por lo que tenemos un valor crítico de 2.58, el cual fue calculado de la siguiente manera:

Cálculo del valor crítico para una Prueba-Z de dos colas

$$V.C = \left|0{,}5 - \frac{\alpha}{2}\right| = \left|0{,}5 - 0{,}995\right| = 0{,}495$$

Al buscar este valor en la tabla de la prueba Z obtenemos el valor de $Z_{cr} = 2{,}58$

Escenario:

Transacciones válidas como nuestra población
Transacciones fraudulentas como muestra
Prueba Z de dos colas
Nivel de significancia de 0,01
El valor crítico correspondiente es 2,58

Hipótesis:
H0: No hay diferencia (no significativa)
H1: Existe una diferencia (significativa)

Fórmula para el Z-score:

$$Z_{score} = \frac{\overline{x} - \mu}{S.E.}$$

Resultados:

```
Time  es estadisticamente significativo
V1  es estadisticamente significativo
V2  es estadisticamente significativo
V3  es estadisticamente significativo
V4  es estadisticamente significativo
V5  es estadisticamente significativo
V6  es estadisticamente significativo
V7  es estadisticamente significativo
V9  es estadisticamente significativo
V10  es estadisticamente significativo
V11  es estadisticamente significativo
V12  es estadisticamente significativo
V14  es estadisticamente significativo
V16  es estadisticamente significativo
V17  es estadisticamente significativo
V18  es estadisticamente significativo
V19  es estadisticamente significativo
V20  es estadisticamente significativo
V21  es estadisticamente significativo
V24  es estadisticamente significativo
V27  es estadisticamente significativo
V28  es estadisticamente significativo
Amount  es estadisticamente significativo
```

Tenemos una variable más que es el tiempo, que puede ser un factor decisivo externo, pero en nuestro proceso de modelación, podemos dejarla de lado.

También comprobamos si hay transacciones duplicadas. Antes teníamos 284807 transacciones en nuestros datos. Después de eliminar las duplicadas nos quedaron 275663 lo que significa que habían 9144 datos duplicados.

Ahora tenemos los datos bien escalados, sin duplicados ni faltantes. Podemos proceder a separar los conjuntos para entrenamiento y pruebas para construir nuestro modelo.

5. Entrenamiento y aplicación de modelos

Dividimos el conjunto de datos en 70 % para entrenamiento y 30 % para pruebas.

Para el Isolation Forest se construyó un modelo ensamblado del mismo modelo con la técnica de bagging que cogía por cada uno de los modelos 10000 datos aleatorios de la muestra para entrenarse.

6. Evaluación de los modelos

Finalmente, se utilizaron un par de bibliotecas de calificación de modelos para medir y comparar los resultados de los dos modelos. Dado que estamos tratando con un conjunto de datos muy desequilibrado, la puntuación F1 se utiliza como un indicador del rendimiento del modelo así como el indicador Roc Auc.

Habíamos dicho que la matriz de confusión no era útil en estos casos porque puede llevar a pensar que el modelo está clasificando bien, sin embargo la podemos usar para tener una noción de qué porcentaje de aciertos estamos hablando, pero de nuevo, cabe aclarar que los resultados que arroje no son concluyentes debido a que sólo se está teniendo en cuenta la clase mayoritaria

6.1. Matriz de confusión para el modelo de Isolation Forests:

Figura 3

6.2. Matriz de confusión para el modelo de LOF:

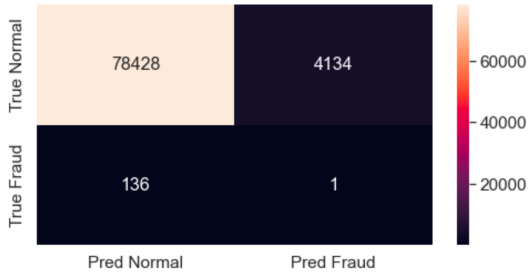
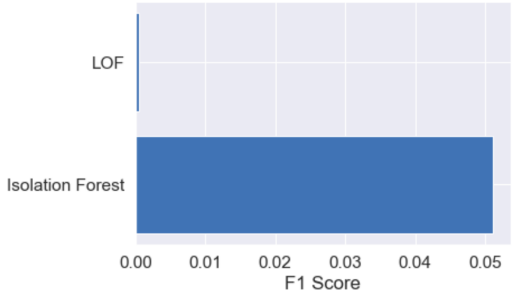


Figura 4

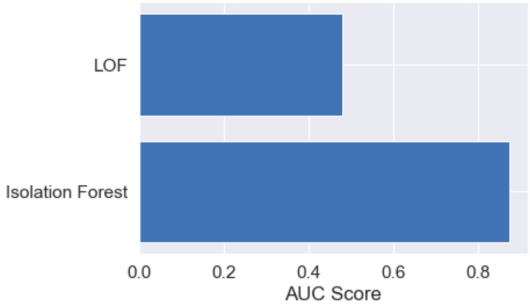
En la matriz de confusión de Isolation Forests vemos que el modelo detecta 44.870 transacciones normales correctamente y sólo 109 transacciones normales son etiquetadas como fraude, no es algo de qué preocuparse, sin embargo en la matriz de LOF si vemos una situación de mal performance en donde de 137 transacciones que debió haber clasificado como fraude, sólo acertó en una, sin embargo no es un motivo de alarma, ya que como se mencionó, no se puede basar en la matriz de confusión para tomar conclusiones.

Finalmente comparamos las métricas que si nos permiten tomar decisiones:

6.3. F1 score:



6.4. AUC score:



7. Conclusiones

En un escenario del mundo real, un modelo no supervisado se utiliza principalmente como semilla para crear datos etiquetados. A menos que se puedan formular reglas de riesgo basadas en el conocimiento del dominio para el problema, ahí si convendría más bien usar algoritmos tradicionales y métodos supervisados. En este caso se pudo analizar un set de datos sin tener mucho contexto del dominio, solo con el uso de herramientas de análisis de datos y aún así logramos una precisión mayor a la esperada (0.873 para el caso del Roc Auc y 0.051 para el F1 score, tomando el Isolation Forest como el modelo que mejor clasifica)

Tanto Isolation Forest como Local Outlier Factor obtuvieron los mismos resultados en la predicción de casos normales, pero Isolation Forest obtuvo mejores resultados en la detección de casos de fraude. En cuanto al rendimiento del modelo, comparando lasmétricas de rendimiento, el modelo de Isolation Forest es un claro ganador.

Las características de los datos son la versión transformada de PCA. Si las características actuales siguen un patrón similar, entonces se puede usar este mismo modelo para detectar transacciones fraudulentas en un ambiente real.

8. Anexo

El código fuente de este trabajo se encuentra publicado en github en la siguiente dirección: Repositorio de trabajos del materia de Aprendizaje automático

Referencias

[1] Machine Learning Group ULB (Owner). Anonymized credit card transactions labeled as fraudulent or genuine. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/metadata. Accessed: 2022-04-01.

[2] Alex Reinhart. Statistics done wrong. https://www.statisticsdonewrong.com/. Accessed: 2022-04-16.

[3] Srinath Perera. Introduction to anomaly detection: Concepts and techniques. https://iwringer.wordpress.com/2015/11/17/anomaly-detection-concepts-and-techniques/. Accessed: 2022-04-16.

[4] Clasificación con datos desbalanceados. https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/. Accessed: 2022-04-25.