MÁSTER EN DATA SCIENCE Y BUSINESS ANALYTICS

RESUMEN - USO DE MODELOS DE APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN DE TRANSACCIONES FRAUDULENTAS CON TARJETAS DE CRÉDITO

TFM elaborado por: Sara Estefanía Chamseddine Fajardo

Tutor de TFM: Abel Ángel Soriano Vázquez

- Madrid. Noviembre, 2023 –

Uso de modelos de aprendizaje automático para la detección de transacciones fraudulentas con tarjetas de crédito

Resumen

En la era actual, con el auge de las nuevas tecnologías, las tarjetas de crédito se han convertido en un componente crucial de la vida cotidiana. Esto se debe a su versatilidad para su uso en una amplia gama de transacciones, tanto en compras en línea como en establecimientos físicos. Sin embargo, este progreso tecnológico y financiero también ha dado lugar a un aumento en la creatividad de individuos maliciosos que buscan aprovecharse de estos avances para su propio beneficio. Se ha vuelto más evidente que nunca el robo y la falsificación de tarjetas de crédito.

Por lo tanto, es de suma importancia implementar nuevos procedimientos destinados a la identificación de transacciones potencialmente fraudulentas. Este enfoque resulta fundamental para combatir esta creciente amenaza. Al hacerlo, podemos salvaguardar los intereses tanto de los consumidores como de las entidades financieras, evitando pérdidas financieras significativas. Asimismo, contribuye a fortalecer y consolidar la confianza en el sistema financiero en su conjunto, fomenta el cumplimiento de la normativa vigente y permite una utilización responsable de las avanzadas tecnologías disponibles en la actualidad.

Construir modelos de Machine Learning para fraude financiero, representa un gran desafío pues normalmente se disponen de conjuntos de datos muy grandes que tienen pocas transacciones fraudulentas y esto genera un desequilibrio de clases. Cuando se enfrenta un desequilibrio de clases, los algoritmos de aprendizaje automático tienden a priorizar la maximización de la precisión general, lo que se traduce en la clasificación de la mayoría de las observaciones como

pertenecientes a la clase mayoritaria. En el contexto de la detección de fraudes financieros, esto resulta en una baja capacidad de predicción (recall) para la clase de interés, que es precisamente la clase minoritaria (transacciones fraudulentas).

El objetivo central de esta tesis es abordar el desafío de lograr un equilibrio en las clases dentro de un conjunto de datos, permitiendo así un entrenamiento eficiente de modelos de detección de fraudes en un entorno de procesamiento distribuido especialmente diseñado para manejar grandes volúmenes de datos, es decir, Big Data. En este contexto, se incluye la implementación y evaluación de varios modelos de aprendizaje automático con el fin de determinar cuál de ellos ofrece el mejor rendimiento en la detección de anomalías en transacciones con tarjetas de crédito. Además, se presta una atención específica a la detección de fraudes en entornos distribuidos. Todo este trabajo se realiza utilizando la plataforma Azure Databricks.

Es fundamental destacar que en este enfoque de detección de anomalías, priorizamos evitar omitir transacciones fraudulentas en lugar de etiquetar erróneamente transacciones legítimas como fraudulentas. Para escenarios de desequilibrio de clases, la métrica más relevante es la sensibilidad (recall o tasa de verdaderos positivos), ya que mide la capacidad del modelo para identificar con precisión las transacciones fraudulentas en comparación con las transacciones fraudulentas reales.

En este contexto, los verdaderos positivos son las transacciones fraudulentas detectadas correctamente, y los falsos negativos son las transacciones fraudulentas que el modelo no logra identificar como tales. Maximizar la sensibilidad es esencial para minimizar la omisión de transacciones fraudulentas, lo que resulta crítico cuando se prioriza la prevención de estos errores.

Para la elaboración del modelo se llevan a cabo una lista de actividades en distintos Notebooks de Databricks (desplegado en una instancia de Azure), utilizando diferentes clústeres dependiendo

de las necesidades del proceso a realizar. A continuación, se muestran las actividades generales realizadas para construir y escoger el modelo de detección de frade:

1. **Configuración del entorno de trabajo:**
    - Se crea una instancia de Databricks en Azure.
    - Para llevar a cabo las distintas etapas del proyecto se crean 5 distintos clústeres de distintos tamaños en Databricks. Esto con el objetivo de ahorrar dinero y tener el poder de cómputo adecuado para cada uno de los procesos que se llevan a cabo. Estos son: clúster para entrenar los modelos de ML, clúster para realizar feature engineering, clúster para realizar visualizaciones de datos, clúster para hacer reducción de dimensionalidad y clúster para hacer análisis exploratorio.
    - Descargar y configurar las credenciales de la cuenta de Kaggle que se utilizarán para la descarga del conjunto de datos a través del código y la API.
2. **Extraer y procesar un conjunto de datos con transacciones con tarjetas de crédito:** Para llevar a cabo los modelos, se utiliza principalmente un conjunto de datos (dataset) de Kaggle con transacciones realizadas con tarjetas de crédito. Este incluye datos acerca de la transacción, el usuario que la realizó y la tarjeta de crédito utilizada. Adicionalmente, se enriquece el conjunto de datos con información obtenida de una fuente externa que incluye el nombre completo del MCC (Merchant Category Code), en lugar de solo su código numérico. Estos conjuntos de datos son descargados y almacenados dentro de Databricks utilizando Python.
3. **Realizar el análisis exploratorio del conjunto de datos:** se utiliza PySpark en Databricks para llevar a cabo un análisis exploratorio de los datos con el propósito de comprender su naturaleza, identificar patrones y abordar posibles problemas de calidad de

datos. Este proceso es fundamental para preparar los datos y mejorar su idoneidad para la construcción de modelos de aprendizaje automático. Además, se realizan visualizaciones y se generan estadísticas del conjunto de datos, lo que permite confirmar completamente el desequilibrio de clases presente.

4. **Transformar el conjunto de datos y feature engineering:** se realizan transformaciones en los datos y se crean nuevas características (features) que resultan relevantes y beneficiosas para los modelos de aprendizaje automático que se desarrollan posteriormente.
5. **Modelado de aprendizaje automático:** se llevaron a cabo diversas tareas que involucraron la creación de características, la aplicación de transformaciones, codificaciones, reducción de dimensiones y selección de variables. El proceso para obtener un modelo adecuado implicó probar distintas combinaciones de parámetros, modelos, codificadores, transformaciones y variables en distintos experimentos. Esto quiere decir que la creación del modelo es un proceso iterativo, es decir, que se repite varias veces, y se refina continuamente el modelo hasta que se logra un mejor resultado, cada iteración implica aprender de los resultados anteriores y realizar mejoras en el modelo para maximizar su desempeño.
    - Se codifican las variables del conjunto de datos utilizando One-Hot-Encoding y String Indexer.
    - Se divide el conjunto de datos de manera balanceada para train y test. Como el conjunto de datos tiene un desequilibrio de clases, es necesario hacer una división del conjunto de datos en entrenamiento y pruebas teniendo esto en mente. Para ello, se lleva a cabo una división estratificada o división balanceada para las transacciones

fraudulentas y legítimas, manteniendo la proporción de clases en ambos conjuntos. Esto va a permitir que, tanto el conjunto de datos como el de prueba, contengan una representación adecuada de transacciones fraudulentas y legítimas.

- Se reduce la dimensionalidad del conjunto de datos utilizando PCA (Análisis de Componentes Principales). Esto con el fin de simplificar el análisis de los datos, eliminar multicolinealidad, preservar de la información importante, y mejorar el rendimiento del algoritmo.
- Se aplica oversampling al conjunto de datos. Este enfoque permite aumentar el número de ejemplos de la clase minoritaria, que en este caso son las transacciones fraudulentas. Existen varias formas de llevarlo a cabo, pero para este caso se duplican las instancias de dicha clase minoritaria.
- Se Entrenan tres (3) diferentes modelos de aprendizaje automático: dos con Python, SVM y Regresión Logística, ambos supervisados. Además, se implementa Isolation Forest utilizando Scala, este es no supervisado, pero funciona muy bien para detección de anomalías. Este último modelo se utiliza a pesar de tener la etiqueta de los datos.

6. **Se evalúan los diferentes modelos de aprendizaje automático implementados y se determina el mejor.** Luego de esto, se puede concluir lo siguiente.

- SVM tiene la sensibilidad más alta (0.8686), lo que significa que es el mejor modelo para identificar transacciones fraudulentas. Sin embargo, la precisión (0.0073) es baja, lo que indica que puede haber un número significativo de falsos positivos.

- La regresión logística tiene una sensibilidad (0.6035) menor en comparación con SVM, pero aun así es significativa. La precisión (0.0013) es aún más baja, lo que sugiere una mayor cantidad de falsos positivos.
- Para el caso de Isolation Forest, aunque este modelo es no supervisado, se poseen las etiquetas de los datos, por lo que solo para este caso es posible calcular las métricas que normalmente no serían posibles. Siendo así, el modelo tiene una sensibilidad muy baja (0.0975), lo que indica que es menos efectivo en la detección de transacciones fraudulentas. Sin embargo, la precisión (0.0112) es más alta en comparación con los otros modelos.

Dado que la prioridad es minimizar la cantidad de transacciones fraudulentas pasadas por alto, SVM es la mejor opción entre estos tres modelos debido a su alta sensibilidad. Sin embargo, también se debe considerar que la precisión de SVM es baja, lo que significa que puede generar más falsos positivos.

Además, es interesante notar que Databricks ofrece un entorno de trabajo robusto y versátil que es especialmente útil para abordar los desafíos relacionados con el procesamiento y análisis de conjuntos de datos voluminosos. Al permitir la creación de clústeres y la partición de datos, Databricks facilita la manipulación de grandes volúmenes de información, así como la capacitación y evaluación de modelos de aprendizaje automático en un entorno unificado. Así también, ofrece la capacidad de acceder a datos almacenados en data lakes y bases de datos desde el mismo entorno de trabajo de Databricks, simplificando la integración y la disponibilidad de datos necesarios para la detección de fraudes en transacciones con tarjetas de crédito.

Aun así, existen varias áreas de mejora para trabajos futuros, como lo pueden ser: utilizar librerías como TensorFlow y validar su rendimiento, probar con modelos de aprendizaje

automático no supervisados para ver si tienen un mejor rendimiento o no, o incluso, otros lenguajes de programación. También se recomienda utilizar un conjunto de datos con información real y más actualizada, pues el conjunto de datos utilizado fue generado de manera sintética y puede no ser el más idóneo.