Universidad del Valle De Guatemala

Facultad de Ingeniería

Security Data Science

Jorge Yass

Proyecto 2: Entrenamiento Incremental en Modelos de Deep Learning y Machine Learning

Javier Mombiela
Carné: 20067
Sección: 10

Guatemala, 20 de mayo 2024

Parte 1: Entrenamiento Incremental

1.1 Investigación Teórica

Para fines de este proyecto, de todas las opciones listadas en las instrucciones, he decidido seleccionar XGBoost y LightGBM. La razón por la que escogí estos dos algoritmos, es debido a que ambos algoritmos son eficientes en problemas de regresión y clasificación. Adicionalmente, ambos algoritmos utilizan la metodología conocida como *gradient boosting*. *Gradient Boosting* se refiere a una metodología de aprendizaje automático en la que se utiliza un conjunto de alumnos débiles para mejorar el rendimiento del modelo en términos de eficiencia, precisión e interpretabilidad. Estos modelos suelen ser árboles de decisión y sus resultados se combinan para obtener mejores resultados generales (Saha, S. 2024).

Esto hace que ambos algoritmos sean muy útiles para este proyecto, el cual consiste en clasificar y predecir transacciones como fraudulentas o legítimas. Este tipo de algoritmos es muy útil para este tipo de problemas debido a su capacidad para capturar relaciones complejas, adaptarse a diferentes tipos de datos y reducir el riesgo de sobre ajuste, lo que conduce a una mejor capacidad de generalización y predicción precisa en nuevos datos (Allwright, S. 2023). A continuación, se explicarán las capacidades y limitaciones de ambos algoritmos en términos del entrenamiento incremental.

XGBoost

El algoritmo *Extreme Gradient Boostin*g o XGBoost es un algoritmo muy poderoso que construye un modelo en base a una colección de árboles de decisión y combina sus predicciones para poder predicciones precisas y robustas (Jana, M. 2023).

Cuando se trata de sus capacidades, XGBoost utiliza un algoritmo de optimización de gradientes que puede actualizarse incrementalmente con nuevos datos, lo que lo hace ideal para el entrenamiento incremental. Además, ofrece opciones de regularización que ayudan a prevenir el sobre ajuste durante este proceso, garantizando que el modelo se adapte bien a nuevos datos sin comprometer su capacidad de generalización. Asimismo, es conocido por su eficiencia en términos de tiempo de entrenamiento y recursos computacionales, lo que lo hace adecuado para aplicaciones que requieren actualizaciones frecuentes del modelo (Jana, M. 2023).

Por otro lado, a medida que se agregan más árboles al modelo XGBoost, su complejidad aumenta, lo que puede resultar en tiempos de entrenamiento más largos y mayores requisitos de memoria, siendo una limitación en escenarios de entrenamiento incremental con grandes conjuntos de datos. Además, el ajuste adecuado de los hiperparámetros puede ser crucial para el rendimiento del modelo durante el entrenamiento incremental, y la selección subóptima de hiperparámetros puede afectar negativamente la capacidad del modelo para adaptarse a nuevos datos de manera efectiva.

LightGBM

Similar a XGBoost, el algoritmo *Light Gradient Boosting Machine* o LightGBM es un marco de trabajo de refuerzo de gradientes que utiliza algoritmos de aprendizaje basados en árboles de decisión (Saha, S. 2024).

LightGBM se destaca por su eficiencia y velocidad de entrenamiento, gracias a su algoritmo basado en histogramas que realiza el agrupamiento de valores. Este enfoque no solo acelera el proceso de entrenamiento, sino que también reduce el uso de memoria, lo que lo hace adecuado para grandes conjuntos de datos. Además, LightGBM es compatible con aprendizaje paralelo y aprendizaje en GPU, lo que aprovecha al máximo los recursos de hardware disponibles para acelerar aún más el proceso de entrenamiento (GeeksForGeeks, 2024).

En cuanto al entrenamiento incremental, LightGBM puede manejarlo de manera efectiva. Cuando se actualiza a través del entrenamiento continuo (por ejemplo, a través de BoosterUpdateOneIter), LightGBM agregará más árboles. Si se utiliza refit, se estaran utilizando las estructuras de árboles existentes para actualizar la salida de las hojas en función de los nuevos datos. Es más rápido que volver a entrenar desde cero, ya que no se tiene que redescubrir las estructuras de árboles óptimas (StackExchange, 2023).

Sin embargo, hay que tener en cuenta que, casi con seguridad, tendrá un rendimiento peor (en los datos antiguos y nuevos combinados) que hacer un reentrenamiento completo desde cero en ellos. El rendimiento de LightGBM dependerá de los parámetros de entrenamiento que se utilicen y de cómo validemos las predicciones (StackExchange, 2023).

1.2 Descripción de la implementación práctica

Procesamiento de datos

Para el entrenamiento inicial de ambos modelos procesaron solamente los datos del año 2019. Inicialmente, se imputaron valores faltantes y se escalaron las características numéricas. Luego, los datos se dividieron en conjuntos de entrenamiento, desarrollo y prueba. Para abordar el desbalance de clases, se aplicó SMOTE al conjunto de entrenamiento, creando un conjunto de datos de entrenamiento resampleado con una proporción manejable de transacciones fraudulentas respecto a las legítimas (1:5), para mejorar la robustez de los modelos en la detección de fraudes.

XGBoost

La implementación del modelo XGBoost comienza con la conversión de los datos de entrenamiento resampleados y los datos de prueba al formato DMatrix de XGBoost. Luego, se definen los parámetros del modelo, configurando el objetivo como binary:logistic para

clasificación binaria y la métrica de evaluación como logloss para medir la pérdida logística. Con estos parámetros y los datos preparados, se entrena el modelo utilizando 100 rondas de actualización.

LightGBM

La implementación del modelo LightGBM comienza con la conversión de los datos de entrenamiento resampleados al formato Dataset de LightGBM. A continuación, se definen los parámetros del modelo, configurando el objetivo como binary para clasificación binaria y la métrica de evaluación como binary_logloss para medir la pérdida logística. Con estos parámetros establecidos, se entrena el modelo utilizando 100 rondas de actualización.

Reentrenamientos

Para habilitar el entrenamiento incremental y mantener la efectividad de los modelos en entornos cambiantes, se implementaron estrategias de reentrenamiento. Estas estrategias implicaron ajustes en el proceso de entrenamiento, donde los modelos anteriores fueron cargados y reentrenados con los nuevos batches, procesados, de datos del año 2020. Esto permitió evaluar la capacidad de los modelos para adaptarse a los cambios en los datos y mantener su rendimiento a lo largo del tiempo.

Se exploraron tres enfoques diferentes de reentrenamiento incremental:

- Reentrenamiento Semestral: Se actualiza el modelo cada seis meses utilizando los datos más recientes disponibles del año 2020.
- Reentrenamiento Trimestral: Se actualiza el modelo cada tres meses utilizando los datos más recientes disponibles del año 2020.
- Reentrenamiento Total: Se entrena un nuevo modelo desde cero con todos los datos disponibles hasta el momento (2019 y 2020).

Estos reentrenamientos se llevaron a cabo para evaluar la capacidad de los modelos para adaptarse a cambios en los datos y mantener su rendimiento a lo largo del tiempo.

1.3 Análisis de los resultados

En la evaluación de los modelos, se implementó un umbral basado en la curva ROC para la clasificación binaria. Este enfoque optimiza el recall, maximizando la tasa de verdaderos positivos a costa de la precisión, lo que puede resultar en una mayor detección de verdaderos negativos y una mayor probabilidad de falsos positivos. Sin embargo, es importante destacar que, en este contexto, la prioridad reside en la detección de transacciones fraudulentas.

Aunque un falso positivo puede resultar en una molestia para el usuario, en términos de seguridad financiera, es preferible errar en el lado de la precaución y clasificar una

transacción legítima como sospechosa, que pasar por alto una transacción fraudulenta. Por lo tanto, nuestro enfoque se centra en maximizar la detección de verdaderos negativos, es decir, la identificación precisa de las transacciones fraudulentas.

Es importante mencionar que las pruebas se realizaron utilizando el mismo conjunto de prueba para todos los modelos y sus diferentes enfoques de reentrenamiento. Esto permitió comparar de manera consistente el rendimiento de cada modelo y enfoque bajo las mismas condiciones, asegurando que los resultados fueran comparables y reflejaran con precisión la efectividad de cada método en la detección de fraudes.

LGBM

Método	Recall Promedio	TN Promedio
LGBM Completo	93.61%	259,431
LGBM Semestral	87.80%	245,154
LGBM Trimestral	81.92%	229,201

Tabla 1: Rendimiento promedio de los 3 reentrenamientos LGBM

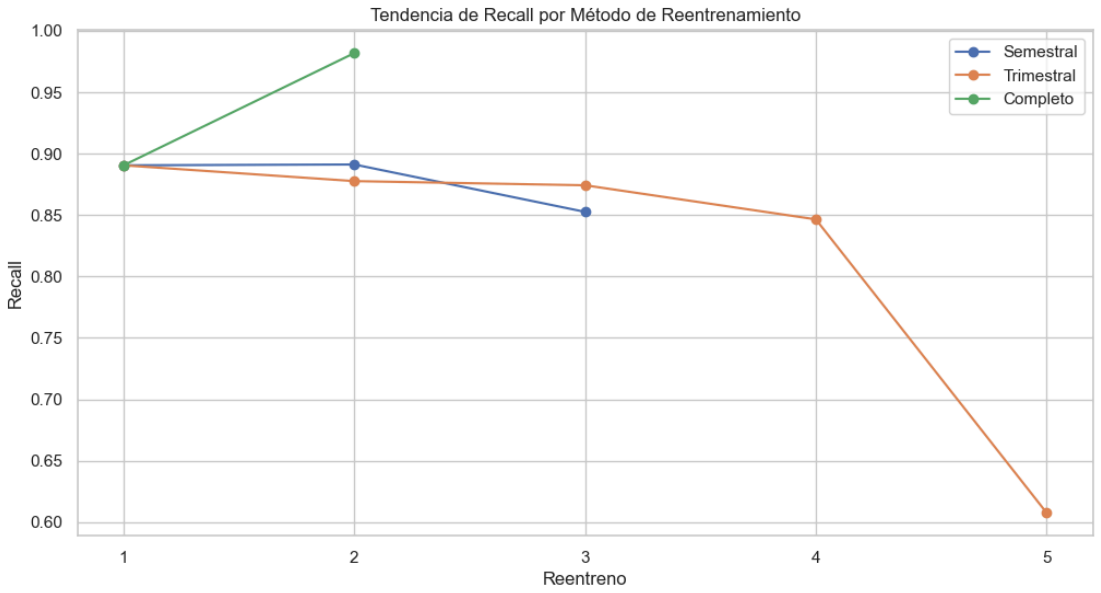

Gráfico 1: Tendencia del recall de cada reentrenamiento LGBM

Como se puede observar, los hallazgos revelan que el enfoque de reentrenamiento total se destaca como el más efectivo entre los tres, con un rendimiento generalmente superior y una tendencia positiva en la capacidad de detección de transacciones fraudulentas. Se puede observar que el recall del reentrenamiento total llego a 98%, lo que indica que este enfoque, podría predecir transacciones fraudulentas casi que a la perfección. Esto se debe a que este enfoque tiene todos los datos del dataset, lo que lo hace más efectivo.

En contraste, tanto el reentrenamiento semestral como el trimestral mostraron un rendimiento menos consistente, con tendencias negativas que sugieren una disminución en la eficacia del modelo con el tiempo. Se puede observar que el enfoque semestral, disminuye, pero se mantiene estable, siempre con recalls por encima del 85%. Esto significa que LGBM también es eficiente con este enfoque, pero también podría significar que con más semestres, la tendencia seguiría bajando, por lo que hay que tener cuidado con eso.

Por otro lado, el enfoque trimestral exhibió el peor rendimiento y una tendencia negativa más pronunciada, lo que indica una pérdida significativa de sensibilidad a los patrones de fraude más recientes. Podemos ver que en este enfoque el recall llega a ser tan bajo, casi llegando al 60% de efectividad. Esta disminución en el rendimiento puede atribuirse a la menor cantidad de datos disponibles en cada reentrenamiento, lo que limita la capacidad del modelo para capturar la complejidad de los patrones de fraude emergentes.

XGBoost

Método	Recall Promedio	TN Promedio
XGBoost Completo	84.71%	238,576
XGBoost Semestral	84.35%	237,505
XGBoost Trimestral	79.17%	240,865

Tabla 2: Rendimiento promedio de los 3 reentrenamientos XGBoost

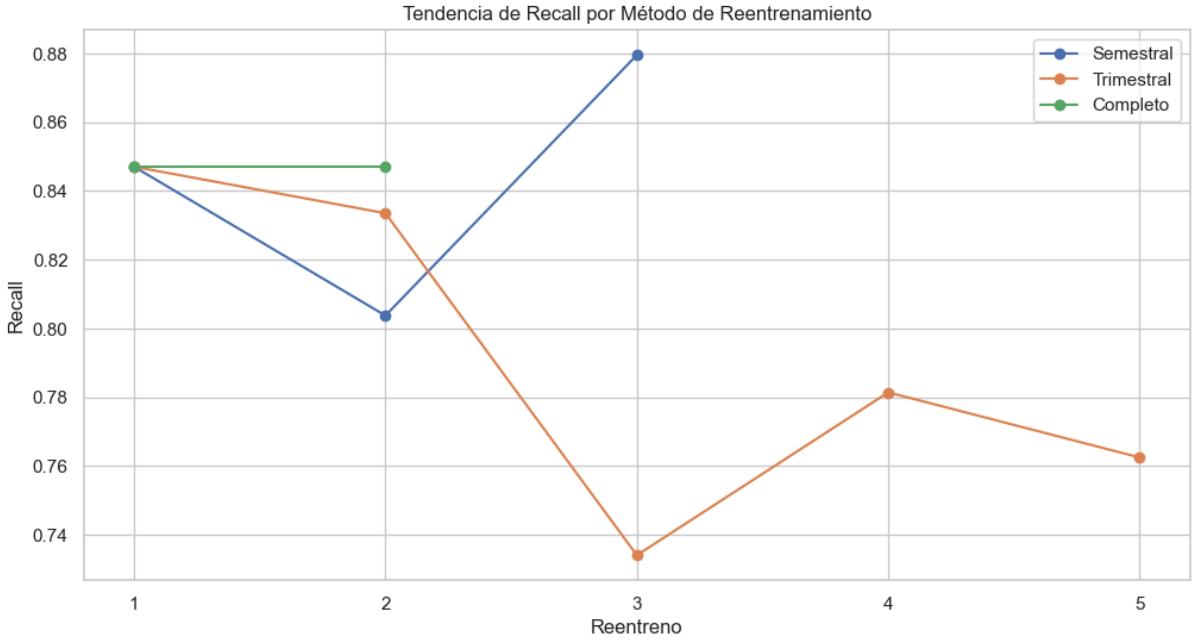

Gráfico 2: Tendencia del recall de cada reentrenamiento XGBoost

Como se puede observar, en el caso de XGBoost, ahora tenemos dos enfoques con tendencia positiva, el reentrenamiento total y el semestral. El enfoque total, aunque no cambia por mucho en el reentrenamiento, sí está en aumento, lo que significa que también es un enfoque aceptable para poder utilizar con XGBoost. El lado negativo es que como se encontró en la investigación, XGBoost puede sufrir cuando hay muchos datos, por lo que se tendría que tener cuidado con esto.

Con el enfoque semestral, se puede observar como el reentrenamiento final del enfoque logra el punto más alto de todos los reentrenamientos, sin embargo, el promedio entre el semestral y el completo es casi idéntico, ya que el completo no tiene ningún rendimiento tan bajo. Al igual que en LGBM, tendríamos que ver más semestres, para ver si el comportamiento se mantiene, o si disminuye.

Por otro lado, al igual que en LGBM, el enfoque trimestral, tuvo un mal rendimiento y con una tendencia negativa. Podemos notar que el promedio de este enfoque es el más bajo de todos los enfoques que se probaron en ambos modelos. Esto es aún, cuando con este modelo, el recall sí tuvo algún reentrenamiento en donde superaron al anterior, lo que es algo que no sucedió con el modelo LGBM.

En conclusión, se puede decir que ambos modelos tienen una buena efectividad con el reentrenamiento incremental y con el semestral, pero el modelo LGBM siempre fue superior a XGBoost, por lo que es una mejor opción, ya que nos permite predecir transacciones fraudulentas de una manera más precisa.

Parte 2: Criterios para Reentrenamiento

2.1 Metodología de reentrenamiento

En el proyecto se han explorado y comparado tres enfoques diferentes de reentrenamiento: reentrenamiento total, semestral y trimestral. Los resultados indican que, en el caso de LightGBM, el enfoque de reentrenamiento total supera consistentemente a los otros dos enfoques en términos de rendimiento, con una tendencia positiva en la capacidad de detección de transacciones fraudulentas a lo largo del tiempo. Esto respalda la recomendación de *StackExchange* de utilizar el reentrenamiento total como la metodología preferida para LightGBM.

Por otro lado, en el caso de XGBoost, si bien el enfoque de reentrenamiento total también muestra un rendimiento sólido, observamos que el enfoque semestral también puede proporcionar mejoras significativas en el rendimiento en comparación con el enfoque trimestral. Esto sugiere que el reentrenamiento semestral puede ser una opción eficaz si el reentrenamiento total no es factible debido a la cantidad excesiva de datos o recursos computacionales requeridos. Sin embargo, si se dispone de suficientes datos y recursos, se recomienda el reentrenamiento total para garantizar la máxima efectividad del modelo.

En general, la recomendación es utilizar un enfoque de reentrenamiento que tenga suficientes datos disponibles para mantener y mejorar la efectividad del modelo en la detección de transacciones fraudulentas. Esto significa que, en la mayoría de los casos, el reentrenamiento total será la mejor opción, ya que siempre tendrá datos suficientes para un buen rendimiento.

Sin embargo, si el reentrenamiento total resulta prohibitivamente costoso en términos de recursos, el reentrenamiento semestral puede ser una alternativa efectiva. Esto es debido a que, un semestre también tiene una gran cantidad de datos, lo cual podría resultar en un modelo eficiente. Adicionalmente, también es recomendable usar un reentrenamiento que cuente con más datos, como vendría siendo uno anual. Habiendo dicho esto, es importante mencionar que se tiene que evitar el enfoque trimestral, ya que los resultados indican que puede ser menos efectivo debido a la menor cantidad de datos disponibles en cada reentrenamiento.

Conclusiones

- El reentrenamiento total se muestra como la opción más efectiva, especialmente para modelos basados en LightGBM, al ofrecer una mejora continua en el rendimiento y una mayor efectividad en la detección de transacciones fraudulentas.
- El reentrenamiento semestral emerge como una alternativa viable en escenarios donde el reentrenamiento total no es factible, proporcionando mejoras significativas en el rendimiento en comparación con el enfoque trimestral, especialmente para modelos basados en XGBoost.
- El reentrenamiento trimestral no es opción en ninguno de los casos, ya que para ambos modelos, cada reentrenamiento resulto en una tendencia negativa. Esto es debido a la pequeña cantidad de datos que tiene cada reentrenamiento.
- En general, el modelo LightGBM demostró ser superior a XGBoost, ofreciendo una mayor precisión y efectividad en la predicción de transacciones fraudulentas.

Recomendaciones

- Utilizar umbral basado en la curva ROC para mejorar la detección de fraudes, ajustando las métricas de los modelos para maximizar el recall y optimizar la detección de verdaderos negativos.
- Aplicar SMOTE con una proporción de 5 a 1 para abordar el desbalance de clases, generando datos sintéticos de transacciones fraudulentas y conservando los datos originales de transacciones no fraudulentas. Esta estrategia puede mejorar significativamente la capacidad del modelo para detectar transacciones fraudulentas mientras mantiene un bajo índice de falsos positivos.
- Evitar el enfoque trimestral o cualquier enfoque con batches más pequeños, debido a la menor cantidad de datos disponibles en cada reentrenamiento, lo que puede afectar negativamente la efectividad del modelo en la detección de fraudes.

- Utilizar reentrenamientos con batches mayores, como semestrales o anuales, para asegurar un rendimiento más consistente y efectivo en la detección de transacciones fraudulentas.

Bibliografía

ArcGIS Pro. (s.f.). Cómo funciona el algoritmo LightGBM. Recuperado de: https://pro.arcgis.com/es/pro-app/latest/tool-reference/geoai/how-lightgbm-works.htm

GeeksForGeeks. (29 de abril de 2024). LightGBM (Light Gradient Boosting Machine). Recuperado de: https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/

Jana, M. (11 de septiembre de 2023). Exploring Machine Learning Models: A Comprehensive Comparison of Logistic Regression, Decision Trees, SVM, Random Forest, and XGBoost. Recuperado de: https://medium.com/@malli.learnings/exploring-machine-learning-models-a-comprehensive-comparison-of-logistic-regression-decision-38cc12287055

EthicalAds. (2022). Random Forests(™) in XGBoost. Recuperado de: https://xgboost.readthedocs.io/en/stable/tutorials/rf.html

Saha, S. (16 de abril de 2024). XGBoost vs LightGBM: How Are They Different. Recuperado de: https://neptune.ai/blog/xgboost-vs-lightgbm

StackExchange. (24 de septiembre de 2023). Recuperado de: https://stats.stackexchange.com/questions/453540/how-does-lightgbm-deals-with-incremental-learning-and-concept-drift