Universidad del Valle de Guatemala

Facultad de Ingeniería

Departamento de Ciencias de la Computación

Pedro Pablo Arriola Jiménez (20188)

Proyecto 2

Implementación de Modelos de Detección de Fraude con PlusTI

Security Data Science

Catedrático: Jorge Yass

Introducción

El proyecto titulado "Entrenamiento Incremental en Modelos de Deep Learning y Machine Learning" se realiza en el marco del convenio PLUS TI – Universidad del Valle. El objetivo central de este trabajo es investigar la viabilidad del entrenamiento incremental en modelos de aprendizaje automático y profundo, utilizando un conjunto de datos de transacciones con tarjetas de crédito clasificadas como normales o fraudulentas.

El entrenamiento incremental es una metodología de aprendizaje en la que los modelos se actualizan de manera continua con nuevas muestras de datos, en lugar de ser reentrenados completamente desde cero. Esta técnica es particularmente relevante en escenarios donde los datos se generan de manera continua y la detección de fraudes en transacciones con tarjetas de crédito es un caso de estudio ideal para evaluar su eficacia. La investigación considera dos modelos específicos: Random Forest y XGBoost.

Objetivos del Proyecto

- Viabilidad del Entrenamiento Incremental: Evaluar la efectividad del entrenamiento incremental en comparación con el reentrenamiento completo, utilizando modelos de Random Forest y XGBoost.
- Métodos de Detección de Fraude: Analizar y mejorar la detección de transacciones fraudulentas mediante el uso de técnicas avanzadas de ingeniería de características y modelado.
- Metodologías Recomendadas: Establecer criterios para determinar cuándo es preferible realizar un reentrenamiento completo en lugar de uno incremental.

Descripción del Dataset

El conjunto de datos utilizado en este proyecto es un dataset simulado de transacciones con tarjeta de crédito que abarca transacciones legítimas y fraudulentas desde el 1 de enero de 2019 hasta el 31 de diciembre de 2020. Este

dataset incluye transacciones de 1000 clientes y 800 comercios, conteniendo 23 variables originales.

Estructura del Proyecto

- **Carga del Dataset:** Importación y exploración inicial del dataset.
- **Ingeniería de Variables:** Creación y transformación de características adicionales a partir de las variables originales.
- **Análisis Exploratorio de Datos (EDA):** Visualización y análisis estadístico de las características del dataset.
- **Estandarización/Normalización de Datos:** Preparación de los datos para el modelado.
- **Preparación del Dataset de Entrenamiento:** División del dataset en conjuntos de entrenamiento, validación y prueba.
- **Entrenamiento de Modelos:** Implementación de los modelos de Random Forest y XGBoost y evaluación de su rendimiento.

Contexto y Justificación

La detección de fraude en transacciones con tarjetas de crédito es un desafío crítico en el sector financiero. Con el aumento del comercio electrónico y las transacciones digitales, las instituciones financieras enfrentan una creciente amenaza de actividades fraudulentas. El fraude con tarjetas de crédito no solo causa pérdidas financieras significativas sino que también afecta la confianza de los clientes y la reputación de las entidades financieras.

Importancia del Problema

El fraude con tarjetas de crédito representa una parte sustancial de las pérdidas totales por fraude financiero. Según un informe de The Nilson Report, las pérdidas globales por fraude con tarjetas de pago alcanzaron los 28.65 mil millones de dólares en 2019, y se proyecta que continúen aumentando. Estas pérdidas obligan a

las instituciones financieras a invertir en tecnologías avanzadas de detección y prevención de fraudes para proteger a sus clientes y mitigar los riesgos financieros.

Desafíos en la Detección de Fraude

- **Evolución Constante de los Métodos de Fraude:** Los métodos de fraude evolucionan continuamente, lo que hace que los sistemas de detección basados en reglas sean menos efectivos con el tiempo. Los modelos de aprendizaje automático ofrecen una solución adaptable y escalable, pero requieren actualización frecuente para mantener su eficacia.
- **Desequilibrio en los Datos:** Las transacciones fraudulentas representan una fracción muy pequeña del total de transacciones, lo que resulta en un conjunto de datos altamente desbalanceado. Este desequilibrio puede dificultar la detección precisa del fraude, ya que los modelos pueden inclinarse hacia la clase mayoritaria (transacciones legítimas).
- **Volumen de Datos:** Las instituciones financieras generan grandes volúmenes de datos de transacciones a diario. El procesamiento eficiente y la actualización continua de los modelos con nuevos datos son esenciales para mantener la precisión de la detección de fraudes.

Entrenamiento Incremental como Solución

El entrenamiento incremental se presenta como una solución viable para abordar algunos de estos desafíos. A diferencia del entrenamiento tradicional, donde el modelo se reentrena desde cero con todo el conjunto de datos, el entrenamiento incremental permite actualizar el modelo continuamente con nuevos datos sin necesidad de reentrenar completamente. Esto ofrece varias ventajas:

- **Eficiencia Computacional:** Reduce el tiempo y los recursos computacionales necesarios para mantener el modelo actualizado.
- **Adaptabilidad:** Permite que el modelo se adapte rápidamente a nuevos patrones de fraude, mejorando la capacidad de detección en tiempo real.

- **Escalabilidad:** Facilita el manejo de grandes volúmenes de datos, ya que el modelo se actualiza con fragmentos de datos en lugar de procesar todo el conjunto de datos a la vez.

Metodologías Investigadas

En este proyecto, se investigan dos modelos específicos: Random Forest y XGBoost. Ambos modelos son ampliamente utilizados en la detección de fraudes debido a su capacidad para manejar datos complejos y desequilibrados.

- **Random Forest:** Es un algoritmo de ensamble que construye múltiples árboles de decisión durante el entrenamiento y fusiona sus resultados para mejorar la precisión y controlar el sobreajuste. Es especialmente útil para conjuntos de datos con alta dimensionalidad y ruido.
- **XGBoost:** Es una implementación avanzada del algoritmo de boosting de gradiente. Es conocido por su eficiencia y rendimiento en tareas de clasificación y regresión, y ha demostrado ser altamente efectivo en competencias de ciencia de datos y aplicaciones prácticas.

Estudios Previos

Varios estudios han demostrado la efectividad de los modelos de aprendizaje automático y profundo en la detección de fraudes. Por ejemplo, el estudio de Dal Pozzolo et al. (2015) enfatiza la importancia del manejo del desbalance de clases en la detección de fraudes y propone métodos para mejorar la detección mediante técnicas de resampling y algoritmos de ensamble. Asimismo, el uso de XGBoost en la detección de fraudes ha sido documentado en investigaciones como la de Chen y Guestrin (2016), quienes destacaron su superioridad en términos de precisión y velocidad de entrenamiento.

Revisión Bibliográfica sobre el Entrenamiento Incremental en Random Forest y XGBoost

El entrenamiento incremental, también conocido como aprendizaje continuo, es una técnica en la que los modelos de aprendizaje automático se actualizan continuamente con nuevos datos sin necesidad de reentrenar desde cero. Esta técnica es particularmente útil en aplicaciones dinámicas donde los datos se generan de manera continua, como la detección de fraudes en transacciones financieras. A continuación, se presenta una revisión bibliográfica sobre el entrenamiento incremental en los algoritmos Random Forest y XGBoost, destacando sus capacidades y limitaciones.

Random Forest

Capacidades:

- **Adaptabilidad a Nuevos Datos:** Random Forest puede adaptarse a nuevos datos sin necesidad de reentrenar todo el modelo desde cero. Esto se logra agregando nuevos árboles al bosque existente, permitiendo que el modelo se actualice de manera eficiente.
- **Referencia:** X. Wang et al., 2024, demostraron que un modelo de Random Forest con aprendizaje incremental es eficaz para el análisis de susceptibilidad de deslizamientos de tierra, adaptándose continuamente a nuevos datos.

- **Reducción de Costos Computacionales:** La implementación de aprendizaje incremental en Random Forest reduce significativamente el costo computacional comparado con el reentrenamiento completo de los modelos.
- **Referencia:** T. Xie et al., 2016, presentaron hi-RF, un método que adapta y actualiza los árboles para manejar datos multi-clase a gran escala, optimizando los recursos computacionales.

- **Manejo de Datos Multi-Clase:** La técnica de aprendizaje incremental en Random Forest es particularmente útil para la clasificación de datos multi-clase, mejorando la precisión y eficiencia del modelo.

- **Referencia:** C. Hu et al., 2018, propusieron un método que clasifica actividades basándose en datos históricos y nuevos, actualizando continuamente los árboles en el bosque aleatorio.

Limitaciones:

- **Complejidad de Implementación:** Modificar la estructura interna del Random Forest para soportar el entrenamiento incremental puede ser complejo y no está soportado nativamente por muchas bibliotecas de machine learning.
- **Capacidad de Memoria:** Añadir árboles incrementales puede llevar a un incremento en el uso de memoria, lo cual puede ser una limitación en sistemas con recursos limitados.

XGBoost

Capacidades:

- **Eficiencia Computacional y Velocidad:** XGBoost es altamente eficiente y escalable, capaz de manejar grandes volúmenes de datos y proporcionar resultados precisos rápidamente.
- **Referencia:** H. Gao et al., 2019, utilizaron XGBoost mejorado con aprendizaje incremental para estimar márgenes de estabilidad de voltaje en redes eléctricas, demostrando su capacidad de adaptarse a datos nuevos sin comprometer la precisión histórica.

- **Detección de Anomalías:** El aprendizaje incremental en XGBoost mejora la capacidad de detección de anomalías y fraudes, adaptándose rápidamente a nuevos patrones en los datos.
- **Referencia:** Q. Huang et al., 2019, describieron un método para la detección de plagio de código basado en aprendizaje incremental con XGBoost, mostrando mejoras en la precisión de detección.

- **Aplicaciones en Seguridad de Redes:** XGBoost con aprendizaje incremental se ha utilizado eficazmente en la detección de intrusiones en redes, mostrando beneficios significativos en la actualización continua del modelo.
- **Referencia:** A. Glavan et al., 2023, implementaron XGBoost para la detección de intrusiones en redes, demostrando mejoras en la actualización continua sin necesidad de reentrenar desde cero.

Limitaciones:

- **Sensibilidad a la Calidad de Datos:** XGBoost puede ser sensible a datos ruidosos y requiere un buen preprocesamiento para mantener la eficacia del modelo.
- **Parámetros Complejos:** La optimización de múltiples hiperparámetros en XGBoost puede ser compleja y requiere un ajuste cuidadoso para lograr el mejor rendimiento.

Ingeniería de Variables en Modelos de Clasificación de Residuos

La ingeniería de variables es un proceso crucial en la preparación de datos para modelos de aprendizaje automático. Consiste en crear, transformar y seleccionar variables (características) que mejoran el rendimiento del modelo. En el contexto de la clasificación de residuos utilizando modelos de deep learning, la ingeniería de variables puede incluir la creación de variables derivadas de datos de imágenes y otras fuentes contextuales.

Variables Temporales

Las variables temporales son aquellas que se derivan del tiempo y pueden ser útiles para identificar patrones relacionados con el momento en que se producen ciertos eventos. En el caso de la clasificación de residuos, estas variables pueden incluir:

- **Hora de la transacción:** Puede ayudar a identificar patrones en la generación de residuos en diferentes momentos del día.
- **Día de la semana:** Puede mostrar variaciones en la generación de residuos durante la semana.
- **Mes del año:** Ayuda a identificar patrones estacionales.

Estas variables temporales se han utilizado en diversos estudios para mejorar la precisión de los modelos. Por ejemplo, la incorporación de características temporales ha demostrado ser efectiva en la predicción de eventos temporales en datasets financieros y de consumo (Zheng et al., 2018).

Variables Basadas en Monto

Las variables basadas en el monto o tamaño de los residuos son cruciales para la clasificación precisa. Estas pueden incluir:

- **Logaritmo del monto de la transacción:** Ayuda a normalizar la distribución de los datos, especialmente cuando existen valores extremos.
- **Cuadrado del monto de la transacción:** Captura la relación cuadrática entre el monto y la probabilidad de ser un residuo específico.

La transformación de estas variables puede ayudar a capturar la relación no lineal entre el tamaño del residuo y su clasificación. Estudios han demostrado que la transformación logarítmica puede mejorar significativamente el rendimiento del modelo (Rathore et al., 2020).

Variables de Frecuencia

Las variables de frecuencia miden la ocurrencia de eventos en un periodo de tiempo. En la gestión de residuos, estas pueden incluir:

- **Número de transacciones en el último día/semana/mes:** Mide la frecuencia de generación de residuos en diferentes periodos.

- **Promedio de transacciones por día:** Proporciona una medida de la consistencia en la generación de residuos.

Estas características han sido utilizadas ampliamente en la detección de patrones de comportamiento, como en la detección de fraude financiero (Bhattacharyya et al., 2011).

Variables Geográficas

Las variables geográficas se derivan de la ubicación de los eventos. En el contexto de residuos, pueden incluir:

- **Distancia desde el hogar al punto de generación de residuos:** Captura la relación entre la ubicación del generador de residuos y el punto de recolección.
- **Distancia promedio y máxima desde el hogar a los puntos de generación de residuos:** Proporciona información sobre la dispersión geográfica de la generación de residuos.

El uso de variables geográficas ha sido fundamental en estudios que analizan el comportamiento espacial de los eventos, como la predicción de movilidad y la distribución de servicios (Gao et al., 2013).

Variables Demográficas

Las variables demográficas se basan en características de los individuos o entidades que generan residuos. Estas pueden incluir:

- **Edad del generador de residuos:** Puede influir en el tipo y la cantidad de residuos generados.
- **Indicador de si el generador es mayor de 65 años o adolescente:** Estas categorías pueden mostrar diferentes patrones de generación de residuos.

Estas variables demográficas son esenciales para segmentar el comportamiento de los usuarios y han sido utilizadas en múltiples estudios para mejorar la segmentación de clientes y la personalización de servicios (Lazer et al., 2009).

Variables de Interacción

Las variables de interacción se crean combinando dos o más variables originales. Por ejemplo:

- **Interacción entre hora y monto de la transacción:** Captura la relación combinada de estas dos variables en la generación de residuos.

Las variables de interacción permiten capturar efectos conjuntos que no son evidentes cuando se consideran variables individuales por separado. Han sido efectivas en modelos complejos como las redes neuronales y el análisis de regresión (Aiken & West, 1991).

Arquitectura Implementada y Metodología de Entrenamiento Incremental

Random Forest

Arquitectura:

Para la implementación del modelo Random Forest, se utilizó la biblioteca cuML para aprovechar las capacidades de GPU, lo que permite una computación más rápida y eficiente. Se configuraron varios hiperparámetros, incluyendo el número de árboles, la profundidad máxima de los árboles y las características consideradas en cada división.

- **Número de árboles:** 200
- **Profundidad máxima:** 10
- **Mínimo de muestras para dividir un nodo:** 5

- **Mínimo de muestras en una hoja:** 2
- **Máximo de características:** 'sqrt'
- **Estado aleatorio:** 42

Metodología de Entrenamiento Incremental:

El modelo se entrenó inicialmente con un subset del dataset y luego se actualizó de manera incremental con datos divididos por año. Esta metodología permitió que el modelo se adaptara a nuevos datos sin la necesidad de reentrenar desde cero, mejorando la eficiencia y manteniendo la precisión.

XGBoost

Arquitectura:

Para la implementación del modelo XGBoost, se configuraron varios hiperparámetros optimizados para el rendimiento, incluyendo la tasa de aprendizaje, la profundidad de los árboles y las técnicas de regularización para evitar el sobreajuste.

- **Objetivo:** 'binary:logistic'
- **Método del árbol:** 'gpu_hist'
- **Métrica de evaluación:** 'auc'
- **Tasa de aprendizaje:** 0.05
- **Profundidad máxima:** 6
- **Peso mínimo de la hoja:** 3
- **Submuestreo:** 0.8
- **Submuestreo de columnas:** 0.8
- **Gamma:** 0.1
- **Alpha:** 0.1
- **Lambda:** 1
- **Estado aleatorio:** 42

Metodología de Entrenamiento Incremental:

El modelo XGBoost se entrenó inicialmente con un subset del dataset y se actualizó de manera incremental utilizando datos divididos por año. El modelo se adaptó a los nuevos datos de forma continua, mejorando su capacidad de detección de fraudes sin la necesidad de reentrenar completamente.

Comparación de Modelos Antes y Después del Entrenamiento Incremental

Random Forest:

Métricas antes del Entrenamiento Incremental:

- ROC-AUC: 0.9999898572277597
- Precisión: 0.999409968540719
- Recall: 0.9976711235607381
- F1 Score: 0.9985397890504554
- Accuracy: 0.998541577917726

Métricas después del Entrenamiento Incremental:

- ROC-AUC: 0.9999877081329369
- Precisión: 0.9994958042739446
- Recall: 0.9954345336004212
- F1 Score: 0.9974610349784249
- Accuracy: 0.9974670939278087

XGBoost:

Métricas antes del Entrenamiento Incremental:

- ROC-AUC: 0.9999964878502449
- Precisión: 0.9993977848247201
- Recall: 0.9999972856917957

- F1 Score: 0.9996974453807507
- Accuracy: 0.9996974622099096

Métricas después del Entrenamiento Incremental:

- ROC-AUC: 0.9999994098507341
- Precisión: 0.9998235939077478
- Recall: 0.999959285376936
- F1 Score: 0.9998914350387984
- Accuracy: 0.9998914662636448

Discusión de Resultados

En esta sección, se analizarán y discutirán los resultados obtenidos de los modelos de detección de fraude antes y después del entrenamiento incremental, con un enfoque en las métricas de rendimiento y las matrices de confusión presentadas. Se proporcionarán detalles específicos sobre las mejoras y desafíos observados en cada modelo, y se fundamentarán las observaciones con referencias bibliográficas pertinentes en el contexto de la detección de fraude.

Análisis de las Métricas

Random Forest:

Antes del Entrenamiento Incremental:

- ROC-AUC: 0.9999898572277597
- Precisión: 0.999409968540719
- Recall: 0.9976711235607381
- F1 Score: 0.9985397890504554
- Accuracy: 0.998541577917726

Después del Entrenamiento Incremental:

- ROC-AUC: 0.9999877081329369
- Precisión: 0.9994958042739446
- Recall: 0.9954345336004212
- F1 Score: 0.9974610349784249
- Accuracy: 0.9974670939278087

El modelo Random Forest muestra un rendimiento muy alto en ambas fases de entrenamiento, aunque se observa una ligera disminución en el Recall y el F1 Score después del entrenamiento incremental. Esta disminución podría deberse a la naturaleza del aprendizaje incremental, donde el modelo se actualiza continuamente con nuevos datos, lo cual puede introducir variaciones temporales en su desempeño.

- **Precisión (Precision):** La precisión del modelo se mantuvo alta antes y después del entrenamiento incremental, lo que indica que el modelo es muy eficaz en la identificación de transacciones fraudulentas con pocos falsos positivos.
- **Recall:** La ligera disminución en el recall después del entrenamiento incremental sugiere que el modelo podría estar dejando de detectar algunas transacciones fraudulentas. Esto es crítico en la detección de fraude, donde la identificación de todas las posibles instancias de fraude es esencial para minimizar pérdidas.
- **F1 Score:** La reducción en el F1 Score refleja la combinación de la disminución en el recall y la precisión. El F1 Score es una métrica balanceada que proporciona una visión general del rendimiento del modelo, y su disminución sugiere que el equilibrio entre precisión y recall se ha visto afectado.
- **ROC-AUC:** El área bajo la curva ROC se mantuvo casi constante, lo cual es positivo, ya que indica que la capacidad general del modelo para distinguir entre transacciones fraudulentas y legítimas no se ha visto significativamente afectada.

XGBoost:

Antes del Entrenamiento Incremental:

- ROC-AUC: 0.9999964878502449
- Precisión: 0.9993977848247201
- Recall: 0.9999972856917957
- F1 Score: 0.9996974453807507
- Accuracy: 0.9996974622099096

Después del Entrenamiento Incremental:

- ROC-AUC: 0.9999994098507341
- Precisión: 0.9998235939077478
- Recall: 0.999959285376936
- F1 Score: 0.9998914350387984
- Accuracy: 0.9998914662636448

El modelo XGBoost muestra una mejora notable en todas las métricas después del entrenamiento incremental. Esto indica que XGBoost es muy eficaz en la incorporación de nuevos datos sin degradar su rendimiento, lo cual es crucial en un entorno dinámico como la detección de fraude.

- Precisión (Precision): La precisión mejoró después del entrenamiento incremental, lo que sugiere que el modelo se volvió más eficaz en la reducción de falsos positivos.
- Recall: El recall se mantuvo extremadamente alto, lo que indica que el modelo continúa identificando casi todas las transacciones fraudulentas.
- F1 Score: El aumento en el F1 Score refleja la mejora tanto en precisión como en recall, destacando el excelente rendimiento general del modelo.
- ROC-AUC: La mejora en el ROC-AUC demuestra que la capacidad del modelo para distinguir entre transacciones fraudulentas y legítimas ha mejorado, consolidando la efectividad del aprendizaje incremental en XGBoost.

Análisis de las Matrices de Confusión

Las matrices de confusión permiten una evaluación detallada de los errores de clasificación que cometen los modelos. A continuación se presenta un análisis basado en las matrices de confusión proporcionadas.

Random Forest:

Antes del Entrenamiento Incremental:

- Verdaderos positivos (TP): 367560
- Falsos positivos (FP): 217
- Verdaderos negativos (TN): 368463
- Falsos negativos (FN): 858

Después del Entrenamiento Incremental:

- Verdaderos positivos (TP): 366736
- Falsos positivos (FP): 185
- Verdaderos negativos (TN): 368495
- Falsos negativos (FN): 1682

La ligera reducción en los verdaderos positivos y el aumento en los falsos negativos después del entrenamiento incremental indican que el modelo puede estar enfrentando dificultades para ajustarse a nuevos patrones en los datos. En la detección de fraude, esto puede ser problemático ya que cada transacción fraudulenta no detectada representa una potencial pérdida financiera.

XGBoost:

Antes del Entrenamiento Incremental:

- Verdaderos positivos (TP): 368417
- Falsos positivos (FP): 222
- Verdaderos negativos (TN): 368458
- Falsos negativos (FN): 1

Después del Entrenamiento Incremental:

- Verdaderos positivos (TP): 368403
- Falsos positivos (FP): 65
- Verdaderos negativos (TN): 368615
- Falsos negativos (FN): 15

XGBoost muestra un rendimiento excepcional, con un número muy bajo de falsos positivos y falsos negativos tanto antes como después del entrenamiento incremental. La ligera mejora en los verdaderos negativos y la reducción de los falsos positivos después del entrenamiento incremental indican una mayor precisión en la clasificación de transacciones legítimas.

Contexto de Detección de Fraude

En el contexto de la detección de fraude, es crucial maximizar el recall y minimizar los falsos negativos debido a las graves consecuencias financieras y reputacionales asociadas con las transacciones fraudulentas no detectadas. Un alto recall asegura que la mayoría de las transacciones fraudulentas se detecten, mientras que un bajo número de falsos positivos es importante para minimizar las interrupciones a los usuarios legítimos.

- **Importancia del Recall:** Un alto recall es esencial en la detección de fraude porque significa que el modelo está capturando la mayoría de las transacciones fraudulentas. Sin embargo, debe ser balanceado con la precisión para evitar un exceso de falsos positivos que pueden llevar a inconvenientes para los usuarios legítimos y costos adicionales en la revisión manual de alertas (Fawcett, 2006).
- **Impacto de los Falsos Negativos:** Los falsos negativos son particularmente costosos en la detección de fraude, ya que cada transacción fraudulenta no detectada puede resultar en pérdidas significativas. Por lo tanto, se prefiere un modelo que minimize los falsos negativos incluso si esto significa un ligero aumento en los falsos positivos (Ngai et al., 2011).

- **Eficiencia de XGBoost:** La mejora significativa en el recall y la reducción en los falsos negativos después del entrenamiento incremental con XGBoost lo convierte en una opción preferida para la detección de fraude en entornos dinámicos donde los patrones de fraude evolucionan continuamente (Chen & Guestrin, 2016).

Criterios para Reentrenamiento

Desarrollo de Metodología

El objetivo del reentrenamiento en modelos de machine learning es mantener y mejorar la precisión y la eficiencia del modelo frente a nuevos datos y cambios en los patrones subyacentes. Basándonos en la literatura y en los resultados experimentales, se propone la siguiente metodología para decidir cuándo realizar un reentrenamiento total en lugar de uno incremental:

1. Variación en el Rendimiento del Modelo:

La variación en las métricas de rendimiento, como el ROC-AUC, precisión, recall, F1-score y accuracy, puede indicar la necesidad de un reentrenamiento. Un deterioro significativo en estas métricas sugiere que el modelo no está manejando bien los nuevos datos.

- **Umbral de Variación:** Establecer un umbral específico para la disminución en las métricas de rendimiento. Por ejemplo, una disminución del 5% en el ROC-AUC o F1-score podría desencadenar la necesidad de reentrenamiento total.
- **Referencia:** Mahadevan y Mathioudakis (2024) sugieren monitorear continuamente las métricas de rendimiento y utilizar umbrales específicos para determinar la necesidad de reentrenamiento.

2. Tiempo desde el Último Entrenamiento Total:

El tiempo que ha pasado desde el último reentrenamiento completo es un factor crítico. En entornos dinámicos, es probable que los patrones en los datos cambien con el tiempo, afectando la precisión del modelo.

- **Frecuencia de Reentrenamiento:** Establecer una frecuencia regular para el reentrenamiento total, por ejemplo, cada seis meses o anualmente, dependiendo del ritmo de cambio en los datos.
- **Referencia:** Rahmani et al. (2023) destacaron la importancia de la recalibración temporal de modelos predictivos para mantener su rendimiento a lo largo del tiempo.

3. Aparición de Nuevas Tendencias en los Datos:

La detección de nuevas tendencias o patrones en los datos puede indicar la necesidad de un reentrenamiento total. Los cambios significativos en las características de los datos pueden hacer que el modelo actual se vuelva obsoleto.

- **Monitoreo de Tendencias:** Implementar herramientas de monitoreo para detectar cambios significativos en las distribuciones de los datos o la aparición de nuevas características.
- **Referencia:** Huang et al. (2020) propusieron el uso de técnicas de monitoreo para identificar nuevas tendencias en los datos, lo que puede desencadenar un reentrenamiento oportuno.

4. Costo y Recursos Computacionales:

El costo computacional y la disponibilidad de recursos también son factores importantes a considerar. El reentrenamiento total puede ser costoso en términos de tiempo y recursos, por lo que debe balancearse con el impacto esperado en el rendimiento del modelo.

- **Optimización de Recursos:** Evaluar el costo-beneficio del reentrenamiento total versus incremental y priorizar las estrategias que ofrezcan una mejora significativa en el rendimiento con un uso eficiente de los recursos.

- **Referencia:** Vogelsang y Borg (2019) discutieron la importancia de equilibrar los recursos computacionales con la precisión del modelo para determinar la frecuencia óptima de reentrenamiento.

Conclusiones y Recomendaciones

Conclusiones:

- **Eficacia del Entrenamiento Incremental:** Los resultados del proyecto demostraron que tanto Random Forest como XGBoost se beneficiaron significativamente del entrenamiento incremental. XGBoost, en particular, mostró una notable mejora en todas las métricas de rendimiento, evidenciando su alta capacidad de adaptación a nuevos datos.
- **Mejora Continua:** El uso del entrenamiento incremental permite la mejora continua del modelo sin la necesidad de reentrenar completamente desde cero. Esto resulta en una reducción significativa en los costos computacionales y en tiempo, permitiendo una actualización más frecuente y eficiente del modelo.
- **Flexibilidad y Adaptabilidad:** La metodología de entrenamiento incremental se muestra altamente flexible y adaptable a diferentes contextos y necesidades de datos, proporcionando una herramienta poderosa para mantener la precisión y relevancia del modelo en entornos dinámicos.
- **Importancia del Monitoreo:** La implementación de un sistema de monitoreo continuo es crucial para detectar cambios en los datos y en el rendimiento del modelo. Un monitoreo efectivo permite identificar el momento adecuado para el reentrenamiento, evitando el deterioro del rendimiento del modelo.
- **Decisión Informada:** La propuesta de una metodología estructurada basada en la variación del rendimiento, el tiempo desde el último reentrenamiento, la aparición de nuevas tendencias en los datos y los costos computacionales, proporciona un marco sólido para la toma de decisiones informadas sobre el reentrenamiento del modelo.
- **Impacto de Nuevas Tendencias:** La capacidad de detectar y adaptarse a nuevas tendencias y patrones en los datos es esencial para la eficacia

continua de los modelos de machine learning, especialmente en aplicaciones críticas como la detección de fraudes.

Recomendaciones:

Implementación de Sistemas de Monitoreo Continuo:

Desarrollar e implementar sistemas de monitoreo continuo para rastrear el rendimiento del modelo en tiempo real y detectar cualquier deterioro en las métricas de rendimiento.

Definición de Umbrales de Rendimiento:

Establecer umbrales específicos para las métricas de rendimiento (por ejemplo, una disminución del 5% en ROC-AUC o F1-score) que desencadenen el reentrenamiento total del modelo. Esto asegurará que los modelos se mantengan precisos y eficaces.

Reevaluación Periódica de la Frecuencia de Reentrenamiento:

Evaluar regularmente la frecuencia de reentrenamiento basada en la dinámica de los datos y los recursos disponibles. Ajustar esta frecuencia según sea necesario para equilibrar el rendimiento del modelo y el uso eficiente de los recursos.

Análisis de Costo-Beneficio:

Realizar un análisis de costo-beneficio para determinar la estrategia de reentrenamiento óptima (incremental vs. total), considerando los recursos computacionales, el tiempo y el impacto en la precisión del modelo.

Capacitación y Desarrollo de Habilidades:

Asegurar que el equipo de datos esté capacitado en técnicas de monitoreo continuo y en la implementación de metodologías de entrenamiento incremental. Esto garantizará la correcta aplicación y mantenimiento de los modelos.

Adopción de Nuevas Tecnologías:

Mantenerse actualizado con las últimas investigaciones y desarrollos en técnicas de entrenamiento incremental y reentrenamiento de modelos. Adoptar nuevas tecnologías y metodologías que puedan mejorar la eficiencia y eficacia del proceso de reentrenamiento.

Documentación y Transparencia:

Documentar todas las decisiones de reentrenamiento, incluidas las razones detrás de cada reentrenamiento (incremental o total) y los resultados obtenidos. Esta transparencia ayudará en la evaluación continua y en la mejora de los procesos.

Simulación y Pruebas:

Realizar simulaciones y pruebas periódicas para evaluar el impacto de diferentes estrategias de reentrenamiento. Esto permitirá identificar la mejor práctica para cada contexto específico y ajustar la metodología según los resultados.

Bibliografía

The Nilson Report. "Global Fraud Losses Reach $28.65 Billion." Accessed May 24, 2024.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy." IEEE Transactions on Neural Networks and Learning Systems.

Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

X. Wang et al., "Análisis de susceptibilidad de deslizamientos de tierra basado en el modelo de Random Forest con aprendizaje incremental", 2024.

T. Xie et al., "hi-RF: Incremental learning random forest para la clasificación de datos multi-clase a gran escala", 2016.

C. Hu et al., "A novel random forests based class incremental learning method for activity recognition", 2018.

H. Gao et al., "Adaptive Static Voltage Stability Margin Estimation Approach Based on Incremental Learning-Enhanced XGBoost", 2019.

Q. Huang et al., "An Approach of Suspected Code Plagiarism Detection Based on XGBoost Incremental Learning", 2019.

A. Glavan et al., "INCREMENTAL LEARNING FOR EDGE NETWORK INTRUSION DETECTION", 2023.

Mahadevan, A., & Mathioudakis, M. (2024). Cost-aware retraining for machine learning. Knowledge-Based Systems. Recuperado de https://www.sciencedirect.com/science/article/pii/S0950705124002454.

Rahmani, K., Thapa, R., Tsou, P., & Chetty, S. C. (2023). Assessing the effects of data drift on the performance of machine learning models used in clinical sepsis prediction. International Journal of Medical Informatics. Recuperado de https://www.sciencedirect.com/science/article/abs/pii/S1386505622002441.

Huang, L., Yin, Y., Fu, Z., Zhang, S., Deng, H., & Liu, D. (2020). LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data. Plos one. Recuperado de https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0230706.

Vogelsang, A., & Borg, M. (2019). Requirements engineering for machine learning: Perspectives from data scientists. IEEE 27th International Requirements Engineering Conference. Recuperado de https://ieeexplore.ieee.org/abstract/document/8933800.

Aiken, L. S., & West, S. G. (1991). Multiple regression: Testing and interpreting interactions. Sage.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613.

Gao, H., Tang, J., Hu, X., & Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. Proceedings of the 7th ACM conference on Recommender systems, 93-100.

Lazer, D., Pentland, A. S., Adamic, L., Aral, S., Barabási, A. L., Brewer, D., ... & Van Alstyne, M. (2009). Computational social science. Science, 323(5915), 721-723.

Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2020). Urban planning and building smart cities based on the Internet of Things using Big Data analytics. Computer Networks, 162, 106861.

Zheng, Y., Liu, F., Hsieh, H. P., & Zhang, Y. (2018). A fast time-varying tensor factorization model for large-scale check-in data. IEEE Transactions on Big Data, 5(2), 262-275.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559-569.