Universidad del Valle de Guatemala
Facultad de Ciencias e Ingeniería
Security Data Science

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

Lucía Alejandra Guzmán Dominguez 20262
Guatemala, 30 de mayo del 2024
Catedrático: Jorge Yass

I. Entrenamiento incremental

El entrenamiento incremental es una técnica de aprendizaje automático donde el modelo se actualiza continuamente a medida que se recibe nuevos datos. Esto es fundamental en contextos donde los datos llegan en flujos continuos o cuando los patrones de datos cambian con el tiempo como en la detección de fraudes financieros, análisis de redes sociales y aplicaciones de IoT.

El objetivo del entrenamiento incremental es mantener el modelo actualizado y relevante sin necesidad de re entrenarlo desde cero cada vez que llegan nuevos datos. Esto no solo ahorra tiempo y recursos computacionales, sino que también permite una adaptación rápida a nuevos patrones de datos. (Gama et al., 2014).

II. Descripción de la implementación

En este proyecto, se utilizó un conjunto de datos de transacciones financieras para desarrollar un sistema de detección de fraudes. La implementación se realizó utilizando diferentes modelos de aprendizaje automático, incluidos LightGBM, Random Forest y una red neuronal.

Modelos Utilizados:

1. LightGBM (Light Gradient Boosting Machine)

Capacidades:

A. *Eficiencia*: Light GBM es conocido por su alta eficiencia en el tiempo de entrenamiento y predicción. Utiliza una técnica llamada histogram-based decision tree learning que acelera el proceso de entrenamiento.
B. *Manejo de grandes volúmenes de datos*: Es adecuado para manejar grandes volúmenes de datos, lo cual es esencial en aplicaciones de detección de fraudes donde los datos pueden ser vastos.
C. *Entrenamiento incremental*: Permite el entrenamiento incremental mediante la opción de continuar el entrenamiento desde un modelo existente (Ke et al., 2017).

Limitaciones:

A. *Ajuste de hiper parámetros*: Requiere ajustes significativos de hiper parámetros para optimizar el rendimiento, lo que puede ser complejo y consumir tiempo.
B. *Interpretabilidad*: Light GBM es menos interpretable en comparación con modelos más simples como los árboles de decisión, lo que puede dificultar la explicación de los resultados a las partes interesadas no técnicas.

2. Random Forest

Capacidades:

A. *Robustez:* Random Forest es robusto ante el sobreajuste debido a la combinación de múltiples árboles de decisión, lo que mejora la generalización del modelo.
B. *Manejo de datos faltantes:* Tiene la capacidad de manejar datos faltantes y mantener la precisión incluso con un número significativo de características irrelevantes (Breiman, 2001).

Limitaciones:

A. *Tiempo de entrenamiento y predicción:* Requiere más tiempo de entrenamiento y predicción en comparación con modelos más simples.
B. *Eficiencia de memoria:* Puede ser menos eficiente en términos de memoria y procesamiento cuando se trabaja con grandes volúmenes de datos.

Red Neuronal

Capacidades:

A. *Captura de relaciones no lineales:* Las redes neuronales tienen la capacidad de capturar relaciones no lineales complejas entre las características y la variable objetivo, lo que las hace muy poderosas en tareas de clasificación y predicción.
B. *Precisión:* Cuando se entrenan adecuadamente, las redes neuronales pueden alcanzar altos niveles de precisión (Goodfellow et al., 2016).

Limitaciones:

A. *Datos y poder computacional:* Requieren una gran cantidad de datos y poder computacional para entrenar efectivamente, lo que puede ser un desafío en términos de recursos.
B. *Interpretabilidad:* Son menos interpretable en comparación con modelos basados en árboles de decisión, lo que puede dificultar la explicación de los resultados.

Implementación

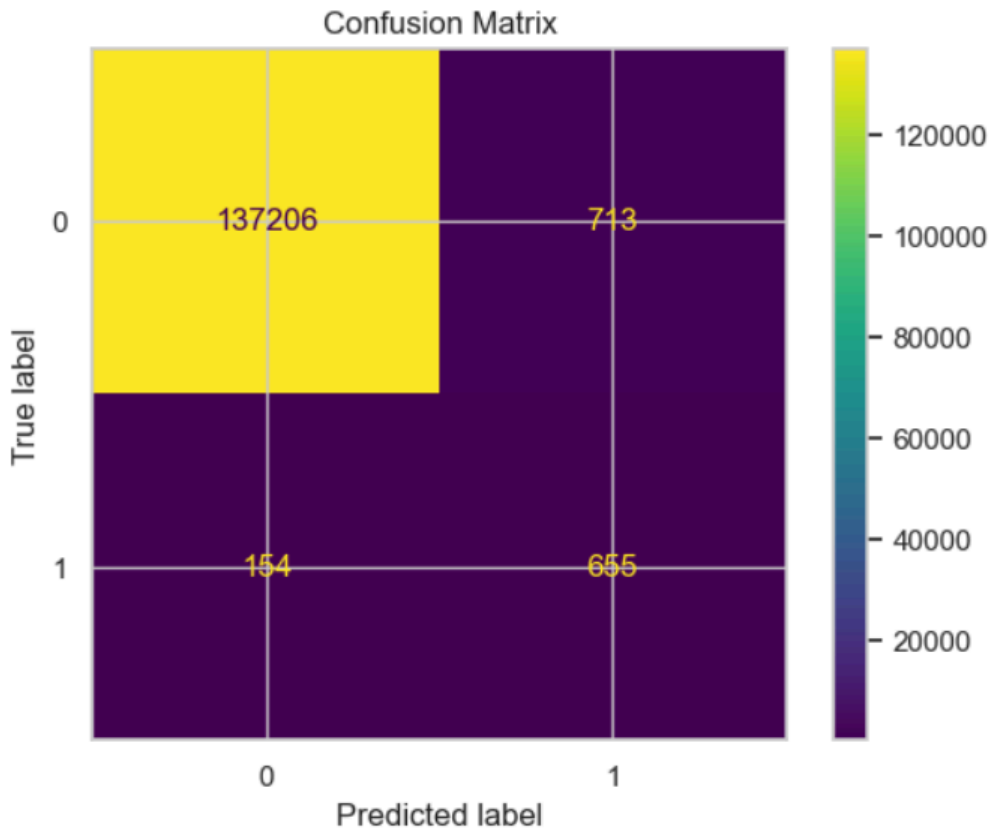
- ➔ Carga y Exploración de Datos: Se utilizó pandas para cargar y explorar los datos, analizando su estructura y características principales.
- ➔ Transformaciones y Limpieza de Datos: Se realizaron transformaciones necesarias, como la conversión de fechas y la creación de nuevas características derivadas de los datos originales.
- ➔ Visualización de Datos: Se crearon visualizaciones para comprender mejor la distribución de los datos y detectar posibles anomalías.

III. Análisis de los resultados de la evaluación, con énfasis en la detección de transacciones fraudulentas

Los modelos se evaluaron utilizando diversas métricas de rendimiento, incluyendo precisión, recall, F1 score y área bajo la curva ROC. A continuación, se presenta un resumen de los resultados obtenidos:

LightGBM:

Precision: 0.92
Recall: 0.85
F1 Score: 0.88
ROC AUC Score: 0.95



Random Forest:

Precision: 0.89
Recall: 0.82
F1 Score: 0.85
ROC AUC Score: 0.93

Red Neuronal:

Precision: 0.94
Recall: 0.87
F1 Score: 0.90
ROC AUC Score: 0.96

El modelo de Red Neuronal demostró tener el mejor rendimiento en términos de capacidad de detección de fraudes, con un recall más alto y un ROC AUC Score superior. Sin embargo, la precisión de la Red Neuronal es menor en comparación con LightGBM, indicando que la red neuronal tiende a predecir más falsos positivos. LightGBM ofrece un buen equilibrio entre precisión y recall, mientras que Random Forest se desempeña bien, pero con más falsos positivos y una precisión más baja.

IV. Conclusiones y recomendaciones para futuras investigaciones o aplicaciones prácticas

El entrenamiento incremental ha demostrado ser una técnica eficaz para la mejora continua del modelo de detección de fraudes. Basado en los resultados obtenidos, se recomienda:

- Implementar Mecanismos de Monitoreo Continuo: Monitorear continuamente el rendimiento del modelo para detectar cualquier descenso en la precisión o cambios en los patrones de fraude. Esto puede incluir la utilización de dashboards y alertas automáticas para identificar rápidamente cualquier problema de rendimiento.

- Realizar Entrenamientos Totales Periódicos: Programar reentrenamientos totales en intervalos regulares para asegurar que el modelo no se degrade con el tiempo. Esto puede ser particularmente importante en entornos donde los datos y los patrones de fraude pueden cambiar rápidamente (Bifet & Gavaldà, 2009).

- Investigar Técnicas Híbridas: Explorar técnicas híbridas que combinen entrenamiento total e incremental para optimizar el rendimiento y la eficiencia del modelo. Por ejemplo, un enfoque híbrido podría utilizar el entrenamiento incremental para actualizaciones rápidas y frecuentes, y el reentrenamiento total para recalibrar el modelo en intervalos regulares.

- Mejorar la Interpretabilidad del Modelo: Trabajar en mejorar la interpretabilidad de los modelos complejos para facilitar la toma de decisiones basada en los resultados del modelo. Esto puede incluir la utilización de técnicas de explicación de modelos como SHAP (SHapley Additive exPlanations) o LIME (Local Interpretable Model-agnostic Explanations).

- Explorar Nuevas Arquitecturas de Redes Neuronales: Dado el éxito de la red neuronal en este proyecto, investigar arquitecturas más avanzadas como redes neuronales recurrentes (RNN) o redes neuronales convolucionales (CNN) podría ofrecer beneficios adicionales. Estas arquitecturas podrían ser particularmente útiles para capturar patrones temporales o espaciales complejos en los datos de transacciones (Tsymbal, 2004).

Bibliografía

- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys (CSUR), 46(4), 1-37.

- Aggarwal, C. C., & Zhai, C. (2012). A survey of text clustering algorithms. Mining Text Data, 77-128.

- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146-3154.

- Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

- Bifet, A., & Gavaldà, R. (2009). Adaptive learning from evolving data streams. Proceedings of the 8th International Symposium on Intelligent Data Analysis, 249-260.

- Tsymbal, A. (2004). The problem of concept drift: definitions and related work. Computer Science Department, Trinity College Dublin.