UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Security Data Science

Proyecto 2

DIEGO JOSE FRANCO PACAY 20240

Parte 1

Investigación teórica

Para la investigación se seleccionaron los algoritmos: Random Forest, Máquinas de Vectores de Soporte (SVM).
Esto se debe a que ofrecen una buena combinación de robustez, precisión, eficiencia y flexibilidad.

Random Forest:

Capacidades:

- Robustez: Random Forest es un algoritmo robusto que no es susceptible al sobreajuste y puede manejar datasets con ruido.

- Interpretabilidad: Random Forest es relativamente fácil de interpretar y comprender, lo que facilita la identificación de las características más importantes para el modelo.

- Eficiencia: Random Forest puede ser entrenado de manera eficiente, incluso con datasets grandes.

- Escalabilidad: Random Forest puede ser escalado a datasets grandes utilizando técnicas de paralelismo.

Dificultades:

- Dificultad para manejar datasets con muchas características: Random Forest puede tener dificultades para manejar datasets con un gran número de características, ya que esto puede aumentar la complejidad del modelo y el riesgo de sobreajuste.

- Sensibilidad a la selección de hiperparámetros: El rendimiento de Random Forest puede verse afectado significativamente por la selección de los hiperparámetros, como el número de árboles y la profundidad de los árboles.

- Interpretabilidad limitada: Aunque Random Forest es más fácil de interpretar que otras técnicas de aprendizaje automático, todavía puede ser difícil comprender cómo funciona el modelo y qué características son las más importantes.

Máquinas de Vectores de Soporte (SVM):

Capacidades:

- Precisión: SVM es un algoritmo con alta precisión en problemas de clasificación binaria, especialmente cuando se trata de datasets con alta dimensionalidad.

- Eficiencia: SVM puede ser entrenado de manera eficiente utilizando técnicas de optimización.

- Flexibilidad: SVM puede ser adaptado a diferentes tipos de problemas de aprendizaje automático mediante la selección de diferentes kernels.

- Interpretabilidad: Aunque SVM no es tan fácil de interpretar como Random Forest, existen técnicas para visualizar y comprender los modelos SVM.

Dificultades:

- Sensibilidad a valores atípicos: SVM puede ser sensible a la presencia de valores atípicos en el dataset, ya que estos pueden afectar la frontera de decisión del modelo.

- Dificultad para manejar datasets con clases desbalanceadas: SVM puede tener dificultades para manejar datasets con clases desbalanceadas, ya que tiende a enfocarse en la clase mayoritaria.

- Interpretabilidad limitada: SVM es un modelo de caja negra, lo que significa que puede ser difícil comprender cómo funciona el modelo y qué características son las más importantes.

Metodología para Elegir Entre Reentrenamiento Total e Incremental

Análisis del Cambio en los Datos:
Si los nuevos datos son significativamente diferentes de los datos utilizados para entrenar el modelo original, entonces puede ser necesario un re-entrenamiento total.
Si los nuevos datos son similares a los datos utilizados para entrenar el modelo original, entonces el entrenamiento incremental puede ser una opción viable.

Monitoreo del Rendimiento del Modelo:
Monitorear el rendimiento del modelo en producción y evaluar si su precisión se ha degradado con el tiempo.
Si el rendimiento del modelo se ha degradado significativamente, entonces puede ser necesario un reentrenamiento total.

Consideraciones Computacionales:
El reentrenamiento total puede ser computacionalmente costoso, especialmente para modelos complejos.
El entrenamiento incremental es generalmente menos costoso que el reentrenamiento total, ya que solo se actualiza una parte del modelo.

Prioridades del Negocio:
Es importante considerar las prioridades del negocio al tomar la decisión de realizar un reentrenamiento total o incremental.
Si es necesario que el modelo tenga la mayor precisión posible, entonces un reentrenamiento total puede ser la mejor opción.
Si la velocidad y la eficiencia son más importantes, entonces el entrenamiento incremental puede ser una opción más viable.

Aunque realmente no existe una respuesta totalmente definida para cuándo es preferible un reentrenamiento total frente a uno incremental. Esta decisión debe tomarse en base a una evaluación de varios factores, como los mencionados anteriormente.

Implementación Práctica:

Se realizó la implementación de diferentes modelos en Python, siendo estos: ANN, Light GBM, XGBoost, Random Forest y SVM. Tanto el entrenamiento de estos modelos como las pruebas se realizaron con el dataset que quedó como resultado luego de haber creado las diferentes variables a partir de las originales. Además, se aplicaron una serie de pasos para hacer el balanceo del mismo dataset, utilizando la técnica SMOTE (Synthetic Minority Over-sampling Technique).

SMOTE se empleó para generar muestras sintéticas de la clase minoritaria, lo que permitió equilibrar las clases del dataset sin alterar la proporción original entre las clases. Esta técnica fue fundamental para mejorar la capacidad predictiva de los modelos al proporcionar un conjunto de datos más equilibrado para el entrenamiento.

Los detalles específicos de la implementación de SMOTE, así como de los modelos y las diferentes transformaciones, están documentados en los notebooks adjuntos. Estos documentos incluyen los pasos detallados para la aplicación de SMOTE y cómo se aseguraron las proporciones adecuadas en el balanceo de los datos.

Evaluación:

Se presenta una comparación del rendimiento de diferentes modelos de aprendizaje automático en términos de métricas de evaluación clave, incluyendo ROC-AUC, precisión, recall y F1-score. Se analizan los resultados antes y después del entrenamiento incremental para evaluar si hay una pérdida significativa en la capacidad de detección de transacciones fraudulentas.

Resultados:

Modelo	ROC-AUC antes	Precisión antes	Recall antes	F1-score antes	ROC-AUC después	Precisión después	Recall después	F1-score después
XGBoost	0.9999	0.9971	1.0	0.9985	0.9999	0.9979	1.0	0.9989
ANN	0.5	0.5012	1.0	0.6677	0.5	0.0	0.0	0.0
Random Forest	1.0	0.9995	1.0	0.9998	1.0	1.0	1.0	1.0
LightGBM	0.9997	0.9915	0.9993	0.9954	0.9997	0.9915	0.9994	0.9954
SVM	0.493	0.5012	1.0	0.6677	0.507	0.0	0.0	0.0

XGBoost, Random Forest y LightGBM: Estos modelos muestran un rendimiento sólido tanto antes como después del entrenamiento incremental, con métricas cercanas a la perfección (ROC-AUC de 1.0 y F1-score de 1.0). Esto sugiere que estos modelos son altamente efectivos en la detección de transacciones fraudulentas y no experimentaron una pérdida significativa de rendimiento después del reentrenamiento con nuevos datos.

ANN: El modelo de Red Neuronal Artificial (ANN) muestra un rendimiento deficiente tanto antes como después del entrenamiento incremental, con métricas de ROC-AUC, precisión, recall y F1-score de alrededor de 0.5. Esto indica que el modelo no fue efectivo en la detección de transacciones fraudulentas y no se observó mejora después del reentrenamiento.

SVM: El modelo de Máquinas de Vectores de Soporte (SVM) también muestra un rendimiento deficiente tanto antes como después del entrenamiento incremental, con métricas de ROC-AUC, precisión, recall y F1-score cercanas a 0.5. Esto sugiere que el modelo no fue efectivo en la detección de transacciones fraudulentas y no mejoró después del reentrenamiento.

Por lo tanto se puede decir que los modelos basados en árboles (XGBoost, Random Forest y LightGBM) demostraron ser altamente efectivos en la detección de transacciones fraudulentas y no experimentaron una pérdida significativa de rendimiento después del reentrenamiento con nuevos datos. Sin embargo, los modelos de Redes Neuronales y SVM mostraron un rendimiento deficiente y no mejoraron con el entrenamiento incremental.

<u>Matrices de confusión de cada modelo:</u>

XGBoost: La matriz de confusión muestra que el modelo predice correctamente la mayoría de las transacciones, con 367,067 verdaderos negativos y 368,982 verdaderos positivos. Sin embargo, hay 1,049 falsos positivos y 0 falsos negativos.

```
[[367067   1049]
 [     0 368982]]
```

ANN: Este modelo predice todas las transacciones como no fraudulentas, resultando en 368,116 verdaderos negativos y 368,982 falsos negativos, lo cual indica un rendimiento muy deficiente en la detección de fraudes.

```
[[368116      0]
 [368982      0]]
```

Random Forest: El modelo predice correctamente 368,058 verdaderos negativos y 368,982 verdaderos positivos, con solo 58 falsos positivos y 0 falsos negativos, lo que muestra una alta precisión en la detección de fraudes.

```
[[368058     58]
 [     0 368982]]
```

LightGBM: La matriz de confusión muestra 364,898 verdaderos negativos y 368,775 verdaderos positivos, con 3,218 falsos positivos y 207 falsos negativos, lo que indica un buen rendimiento, aunque con más errores en comparación con Random Forest.

```
[[364898   3218]
 [   207 368775]]
```

SVM lineal en TensorFlow: Este modelo tiene una deficiencia significativa, ya que predice todas las transacciones como fraudulentas, con 147,067 falsos positivos y 147,772 verdaderos positivos, sin ningún verdadero negativo o falso negativo, lo que sugiere una falla en la capacidad de generalización del modelo.

```
[[     0 147067]
 [     0 147772]]
```

Los resultados obtenidos a partir de las matrices de confusión son válidos y proporcionan una visión clara del rendimiento de cada modelo en la clasificación de transacciones fraudulentas.

Parte 2

Metodología desarrollada e implementada

Monitorización Continua del Rendimiento del Modelo:

- Se debe implementar un sistema de monitorización continua del rendimiento del modelo, evaluando regularmente métricas clave como ROC-AUC, precisión, recall y F1-score.
- Establecer umbrales aceptables para cada métrica con base en los requisitos del negocio y en la tolerancia al riesgo.

Análisis de la Variación en el Rendimiento:

- Realizar un análisis detallado de la variación en el rendimiento del modelo con el tiempo.
- Observar si hay una disminución significativa en las métricas de evaluación, lo que podría indicar la necesidad de un reentrenamiento total.

Evaluación del Tiempo desde el Último Entrenamiento Total:

- Registrar la fecha del último entrenamiento total del modelo.
- Comparar el tiempo transcurrido desde el último entrenamiento total con una ventana de tiempo definida.

Detección de Nuevas Tendencias en los Datos:

- Monitorizar los cambios en la distribución de los datos y la aparición de nuevas tendencias.
- Identificar si hay cambios significativos en la naturaleza de las transacciones fraudulentas o en el comportamiento de los usuarios.

Consideración de Factores Externos:

- Tener en cuenta factores externos como cambios en la regulación, actualizaciones en la infraestructura tecnológica o eventos significativos en el dominio del problema que puedan afectar la validez del modelo.

Decisiones Basadas en Umbrales Predefinidos:

- Establecer umbrales predefinidos para la variación en las métricas de rendimiento, el tiempo desde el último entrenamiento total y la detección de nuevas tendencias en los datos.
- Decidir sobre la base de estos umbrales si es necesario un reentrenamiento total o si un reentrenamiento incremental es suficiente.

Implementación de Políticas de Reentrenamiento:

- Desarrollar políticas de reentrenamiento que definan claramente los procedimientos y criterios para realizar reentrenamientos totales y/o incrementales.

Validación y Ajuste Continuo:

- Validar y ajustar continuamente la metodología en función de la experiencia operativa y los cambios en el entorno.

Conclusiones y recomendaciones

Conclusiones

Los modelos basados en árboles (XGBoost, Random Forest y LightGBM) demostraron ser altamente efectivos en la detección de transacciones fraudulentas y mantuvieron un rendimiento consistente incluso después del entrenamiento incremental.

Los modelos de Redes Neuronales y SVM mostraron un rendimiento deficiente y no mejoraron con el entrenamiento incremental, lo que sugiere que pueden no ser adecuados para este problema específico.

Las conclusiones anteriores se basan en las matrices de confusión presentadas anteriormente.

La implementación de un sistema de monitorización continua del rendimiento del modelo es crucial para detectar cualquier degradación en su eficacia con el tiempo.

Las decisiones sobre reentrenamientos totales o incrementales deben basarse en una evaluación integral de varios factores, como la variación en el rendimiento del modelo, el tiempo desde el último entrenamiento total y la detección de nuevas tendencias en los datos.

Recomendaciones

Realizar experimentos para optimizar los parámetros de los modelos existentes y explorar técnicas avanzadas de ajuste de hiperparámetros para mejorar aún más su rendimiento.

Explorar técnicas de interpretación de modelos para comprender mejor cómo funcionan los modelos en la detección de fraudes y para identificar características importantes en la toma de decisiones.

Explorar técnicas de reducción de dimensionalidad como PCA (Análisis de Componentes Principales) o t-SNE, que pueden ayudar a mejorar la eficiencia del modelo al reducir el número de variables.

Investigar y aplicar técnicas de ensamble como el Stacking o el Voting, que combinan las predicciones de múltiples modelos para mejorar la precisión general.

Referencias:

- Zhong, J., Liu, Z., Zeng, Y., Cui, L., & Ji, Z. (n.d.). *A Survey on Incremental Learning*. Retrieved from https://webofproceedings.org/proceedings_series/ECS/CAPE%202017/CAPE_1113034.pdf

- R.R, A., & P. R, D. (2013). Methods for Incremental Learning : A Survey. *International Journal of Data Mining & Knowledge Management Process*, *3*(4), 119–125. https://doi.org/10.5121/ijdkp.2013.3408

- Lange, S., & Grieser, G. (2002). On the power of incremental learning. *Theoretical Computer Science*, *288*(2), 277–307. https://doi.org/10.1016/s0304-3975(01)00404-2