MODELOS Y APRENDIZAJE
ENTREGABLE CLASE 2 12/02/2024

Autores: Jorge Enriquez, Michelle Guamba, Francisco Lopez, Mario A. Parra, Carlos Rosero.

Ejercicio: Clasificación de Malware usando el dataset personalizado

Descripción del ejercicio:

Contexto: El aumento de la sofisticación del malware obliga a desarrollar enfoques innovadores para su detección. Este ejercicio te permitirá profundizar en el aprendizaje automático aplicado a la ciberseguridad, construyendo un clasificador para identificar archivos ejecutables maliciosos.

Objetivo y Alcance

Clasificación de Malware usando el dataset personalizado; el aumento de la sofisticación del malware obliga a desarrollar enfoques innovadores para su detección. Este ejercicio te permitirá profundizar en el aprendizaje automático aplicado a la ciberseguridad, construyendo un clasificador para identificar archivos ejecutables maliciosos.

El dataset contiene información extraída de archivos ejecutables de Windows, tanto maliciosos como benignos, en la descripción de la tarea se encuentran detalles del dataset y es proporcionado para este ejercicio en la plataforma de EIG.

Desarrollo

1. **Preparación del dataset:**

1.1.Descripción

El dataset personalizado de información extraída de archivos ejecutables, una vez realizado su preprocesamiento cuenta con:

- 369 registros sin datos perdidos de los 373 registros iniciales.
- 531 características expresadas binariamente.

- 1 columna numérica con valores (1 si es malicioso o 0 sino es malicioso)

1.2.Preparación

Para la preparación del dataset, se realiza la evaluación de valores nulos existentes y la descripción de los valores en cada campo existente.

- No hay valores nulos.
- El campo Label solo puede tener un valor entre: “no-malicious”, “malicious”
- El resto de los campos etiquetados como F_1…F_531 solo tiene valores 0 ó 1

Para el proceso de aprendizaje se separa el dataset en train y test con los parámetros test_size = 0.2 y random_state = 42

2. Selección de técnicas de aprendizaje automático

2.1.Investigación de que técnica aprendizaje es recomendable.

Se realiza una revisión de cuáles son las mejores técnicas de aprendizaje automático para clasificación de malware, tomando en cuenta el tamaño del conjunto de datos (pequeño) y los recursos disponibles; las recomendaciones que se encontraron son:

- Si tienen un conjunto de datos grande y recursos computacionales disponibles, se recomiendo usar redes neuronales.
- Si necesitan un modelo interpretable, se recomienda usar árboles de decisión.
- Si tienen un conjunto de datos pequeño, se recomiendo usar KNN o SVM.
- Si necesitan un modelo robusto, se recomienda usar Random Forest.

2.2.Random Forest.

- Proporciona altos niveles de clasificación, incluso generaliza bien con nuevos datos, esto es muy útil porque cada vez surgen nuevos tipos de malware.
- Maneja grandes conjuntos de características eficientemente.

- Es efectivo en clasificación de conjunto de datos desbalanceados, esto funciona para nuestro ejemplo donde las muestras de malware son menos frecuentes.
- Es robusto al ruido y valores inusuales, esto puede ser útil donde las muestras de malware tengan comportamientos inusuales.
- Es resistente al sobreajuste.

2.3.Decision Tree

- Son fácilmente interpretables y comprensibles.
- Los nodos representan características para tomar decisiones, esto puede ser muy valioso para identificar aspectos del código o del comportamiento del malware.
- Permiten manejar conjuntos de datos desbalanceados sin requerir técnicas adicionales, esto funciona para nuestro ejemplo donde las muestras de malware son menos frecuentes.
- Son robustos para valores atípicos, esto puede ser útil donde las muestras de malware tengan comportamientos inusuales.

2.4.SVM(Support Vector Machines).

Usaremos SVM porque en clase vimos KNN, el algoritmo SVM es robusto y puede ser efectivo para clasificar malware incluso con conjuntos de datos pequeños. Se usa los parámetros por defecto:

```
modelo_svc = SVC()
```

Vemos como se comporta de forma rápida con el reporte de clasificación y la matriz de confusión en forma texto.

```
              precision    recall  f1-score   support

   malicious       0.98      0.98      0.98        58
non-malicious      0.94      0.94      0.94        17

    accuracy                           0.97        75
   macro avg       0.96      0.96      0.96        75
weighted avg       0.97      0.97      0.97        75

[[57  1]
 [ 1 16]]
```

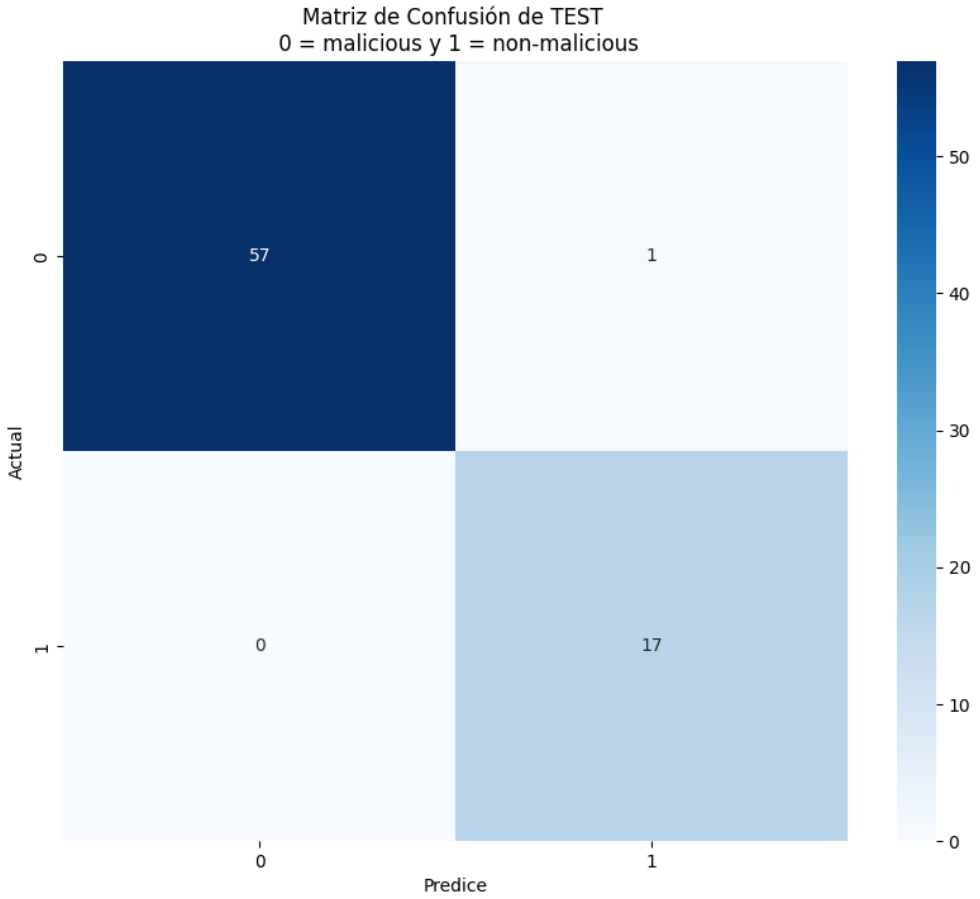
El resultado es idéntico al del Random_Forest, hay otros parámetros en este modelo que moveremos.

Parámetros de regularización propuestos:

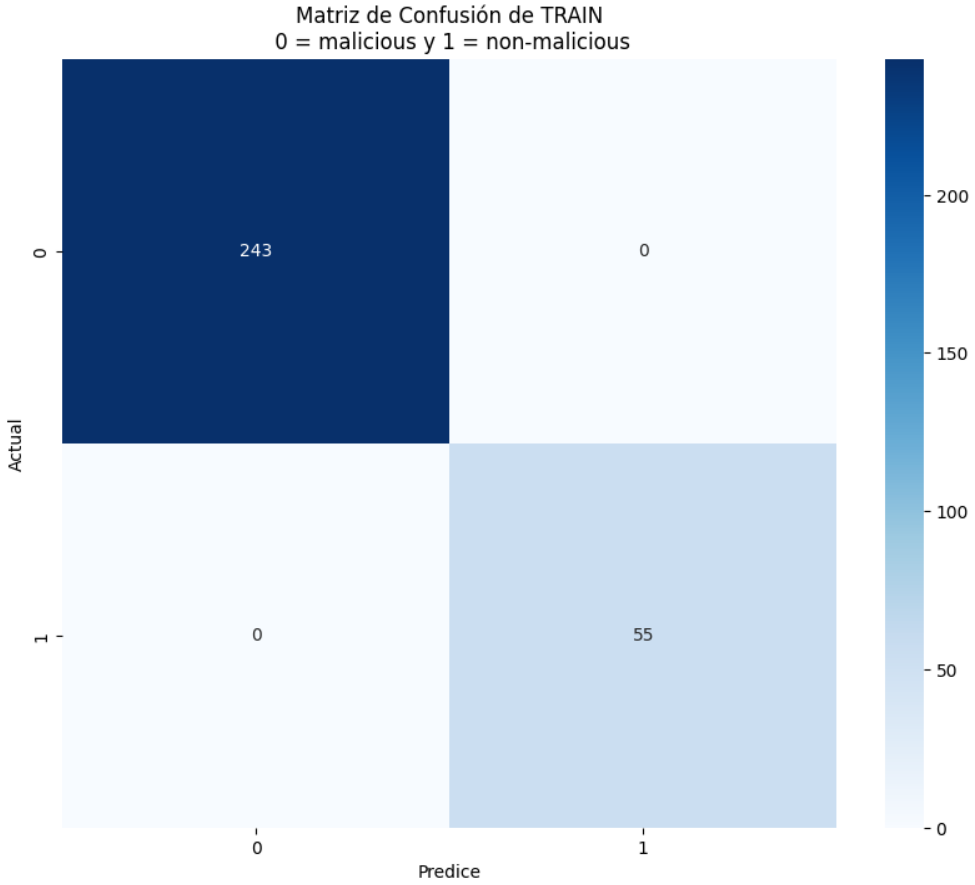
- **C:** Se recomienda un valor alto (entre 1 y 100) para aumentar la regularización y reducir el riesgo de sobreajuste, especialmente en conjuntos de datos pequeños o con características ruidosas.
- **gamma:** Se recomienda un valor bajo (entre 0.001 y 0.1) para que el modelo sea más sensible a los patrones locales en los datos.

Luego de realizar algunas pruebas vemos que el parámetro gamma = 0.1 causa un mejor efecto.

```
modelo_svc = SVC(gamma = 0.1)
```



En esta matriz de confusión hay un solo archivo que no lo reconoce de forma correcta.



En esta matriz de train hace las predicciones de forma correcta, esto implica que en el proceso de entrenamiento lo aprendió todo a detalle.

3. **Resultados del entrenamiento**

Implementar y entrenar el modelo utilizando el conjunto de entrenamiento (Random Forest)

	precision	recall	f1-score	support
0	1.00	0.98	0.99	42
1	0.93	1.00	0.96	13
accuracy			0.98	55
macro avg	0.96	0.99	0.98	55
weighted avg	0.98	0.98	0.98	55

La precisión para la clase 0 es del 100%, lo que indica que el modelo identifica correctamente todos los casos maliciosos (clase 0) en un 100% de las veces. Para la clase 1, la precisión es del 93%, lo que significa que el modelo identifica correctamente el 93% de los casos no maliciosos (clase 1) de entre todos los casos que clasifica como no maliciosos.

La sensibilidad o recall, la clase 0, el recall es del 98%, lo que significa que el modelo identifica correctamente el 98% de los casos maliciosos (clase 0) de todos los casos maliciosos reales. Para la clase 1, el recall es del 100%, lo que indica que el modelo identifica correctamente el 100% de los casos no maliciosos (clase 1) de todos los casos no maliciosos reales.

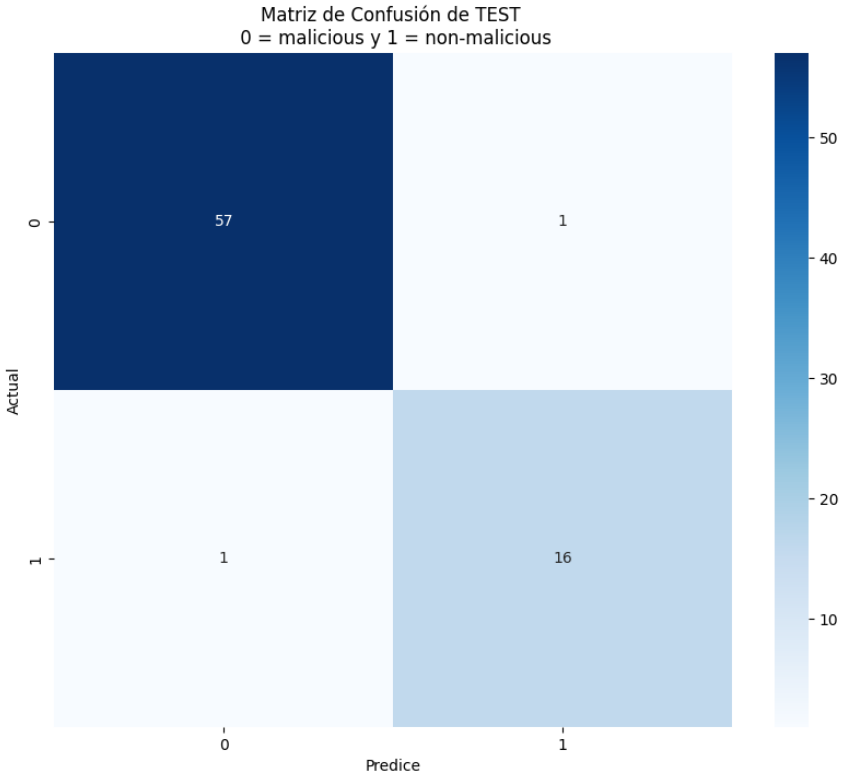
El F1-score para la clase 0 es del 99%, mientras que para la clase 1 es del 96%. El support es el número de ocurrencias reales de cada clase en los datos de prueba. En este caso, hay 42 muestras de la clase 0 y 13 muestras de la clase 1. La precisión global del modelo es del 98%.

4. **Evaluación del modelo**

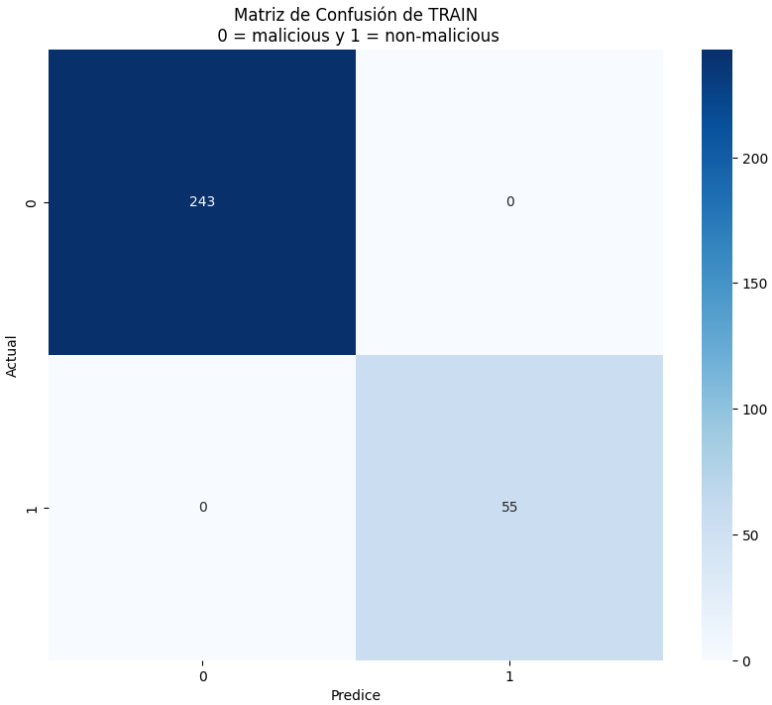
Se elige este modelo por ser robusto para que el modelo acierte de forma exacta la mayor cantidad de archivos maliciosos. Se configuran los siguientes parámetros en el modelo:

```
modelo_rf = RandomForestClassifier(n_estimators=50, random_state=42, max_depth = 5)
```

Se movieron los parámetros n_estimators entre 25, 50 y 100; el max_depth se puso valores bajos porque es un conjunto de datos pequeño y esto puede evitar el sobreajuste.



A nivel de predicción en los datos de TEST solo tiene un falso-positivo y un falso-negativo; a pesar de haber movido varios parámetros no fue posible mejorar aún más el modelo.



El modelo aprende todo en el proceso de entrenamiento, es por ello que la matriz de confusión de TRAIN no muestra falsos-positivos ni falsos-negativos.

Como resultado el modelo Random-Forest es bastante robusto y es muy adecuado su uso para este dataset de archivos de malware.

Optimiza los hiperparámetros

```
Best parameters: {'class_weight': 'balanced', 'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'best'}
Best score: 0.9845399698340875
Accuracy on val: 0.9454545454545454
ROC-AUC score val: 0.9634146341463415
Confusion matrix val:
 [[38  3]
 [ 0 14]]
Accuracy on test: 1.0
ROC-AUC score test: 1.0
Confusion matrix test:
 [[47  0]
 [ 0  9]]
```

El puntaje más alto obtenido durante la búsqueda de hiperparámetros (usando validación cruzada) fue de aproximadamente 0.9845, lo que indica que el modelo alcanzó una precisión del 98.45% en promedio en los datos de entrenamiento durante la validación cruzada.
Al evaluar este modelo en el conjunto de datos de validación, obtenemos una precisión del 94.55%. La matriz de confusión en el conjunto de datos de validación muestra que se clasificaron correctamente 38 muestras no maliciosas (clase 1) y 14 muestras maliciosas (clase 0), mientras que se clasificaron incorrectamente 3 muestras no maliciosas como maliciosas.
El modelo con los hiperparámetros optimizados, considerando el desbalance de clases, parece generalizar bien tanto en los datos de validación como en los de prueba, lo que sugiere que es un modelo robusto para la detección de malware.

5. Interpretación del modelo

En el caso de la detección de malware, donde lo que se intenta identificar es si una muestra es, en efecto, un programa malicioso, dado que la clasificación es binaria: la muestra es o no es un malware, lo tanto, aquel modelo que muestre con mayor precisión el ajuste a la base de datos es el que presenta mayor robustez en la presentación de resultados, la minimización de error y una mejor lectura del comportamiento de los datos analizados.

Para efectos del presente trabajo, los modelos con mayor eficiencia fueron Random Forest y el model de SVM, siendo que un Random Forest es un ensamble de árboles de decisión interconectados que utilizan variación de bagging,

Se determinó que el modelo de Random Forest presentó una mayor eficiencia en la predicción interacciones, a la vez que fue uno de los modelos más solidos y mas sencillos de entrenar. Por tanto, conforme a los resultados obtenidos después de la evaluación de diferentes modelos predictivos, los indicadores del modelo fueron los siguientes:

	precision	recall	f1-score	support
malicious	0.98	0.98	0.98	58
non-malicious	0.94	0.94	0.94	17
accuracy			0.97	75
macro avg	0.96	0.96	0.96	75
weighted avg	0.97	0.97	0.97	75

El modelo de Random Forest presentó un valor de accurrancy alto de 0.97, es decir el modelo predijo con exactitud al 97% el comportamiento de los datos. Generalmente se recomienda el modelo de Random Forest para el testeo de malware, según los análisis realizados en el presente trabajo investigativo el valor de acurrancy más elevado lo obtuvo el modelo de RF seguido por el modelo de SVM que también mostro indicadores robustos y un ajuste considerable a la predicción de datos.