Métodos de aprendizaje automático aplicados a la ciberseguridad: Clasificación binaria de ejecutables del Sistema Operativo Windows

Luis Alberto Glaría Silva

Universitat Rovira i Virgili (URV) y Universitat Oberta de Catalunya (UOC)
Máster Universitario en Ingeniería Computacional y Matemática
Área: Ciberseguridad

Tutor:
Prof. Ángel Elbaz Sanz

Defensa de la Tesis
7 de Junio de 2023

Índice

Introducción

Descripción del problema y motivación

La creciente digitalización de la sociedad ha llevado a un aumento exponencial de los incidentes relacionados con la seguridad informática, tanto en número como en impacto. La ciberseguridad se ocupa de prevenir y combatir estos incidentes, pero ante la complejidad y constante evolución de las distintas amenazas, las técnicas clásicas se tornan insuficientes

Descripción del problema y motivación

Se hace necesario explorar nuevas técnicas como el aprendizaje automático, capaces de adaptarse a nuevas amenazas y mejorar la detección.

Ciberseguridad

Definición

Definimos la ciberseguridad como el arte de proteger redes, dispositivos y datos de accesos no autorizados o usos delictivos y la práctica de garantizar la confidencialidad, integridad y disponibilidad de la información

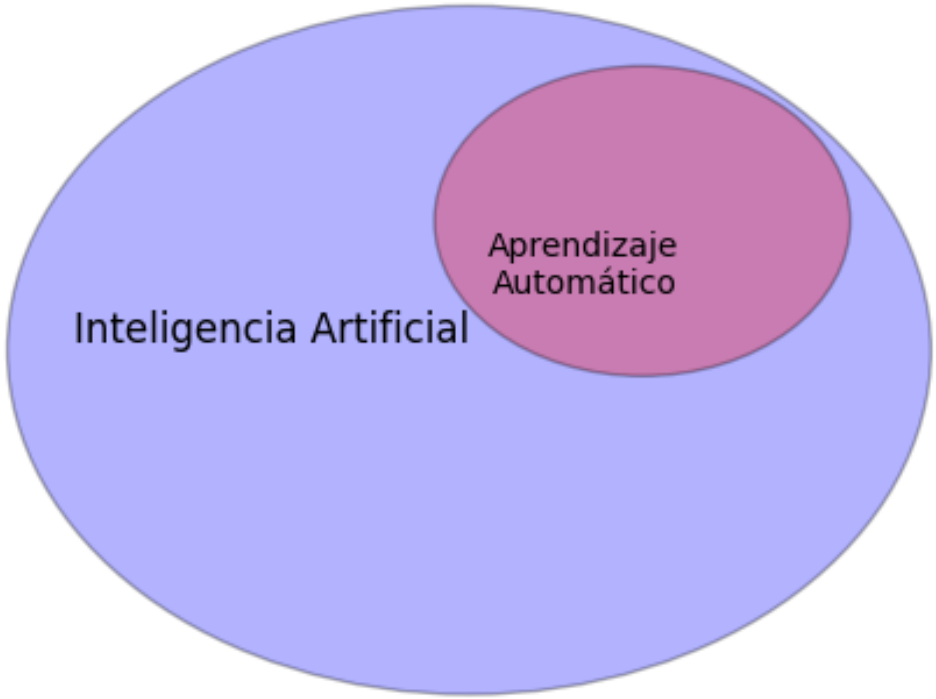
Principales amenazas a la ciberseguridad

Ingeniería Social Se basan en explotar vulnerabilidades humanas. Incluyen técnicas de suplantación de identidad como el *phishing* o el *smishing*

Ataques DoS y DDoS Tienen como objetivo interrumpir o degradar el funcionamiento de un servicio en línea.

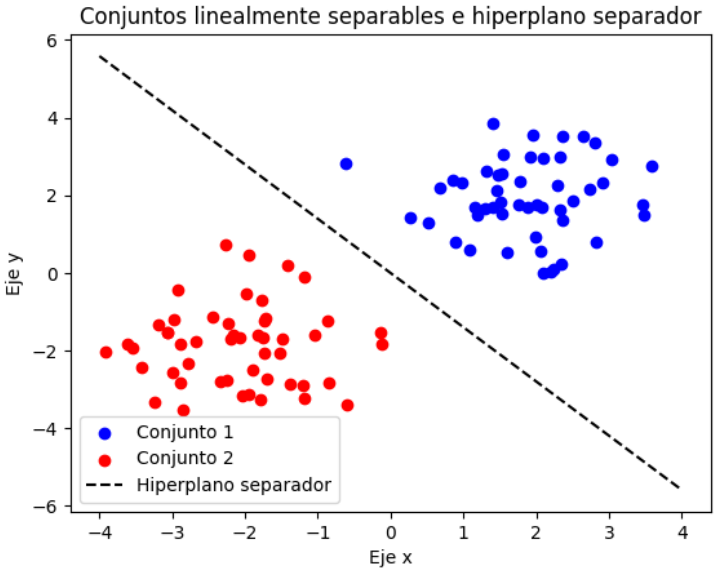
Malware Incluye una amplia variedad de software diseñado para infiltrarse, dañar o realizar acciones no autorizadas en sistemas informáticos, como virus, gusanos, troyanos, ransomware o adware.

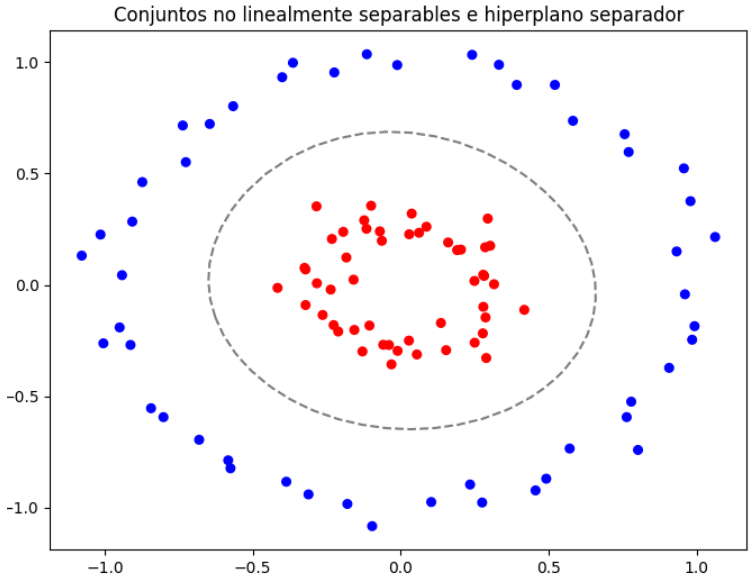
Aprendizaje automático



El aprendizaje automático es un campo dentro de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos estadísticos que aprenden de los datos, sin ser programados explícitamente para ello.

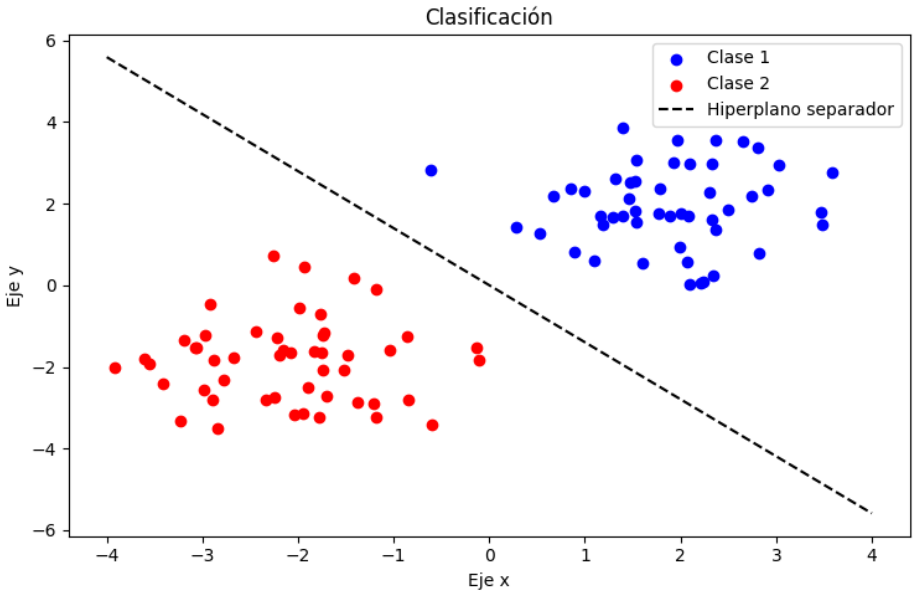
Aprendizaje automático: posibles clasificaciones

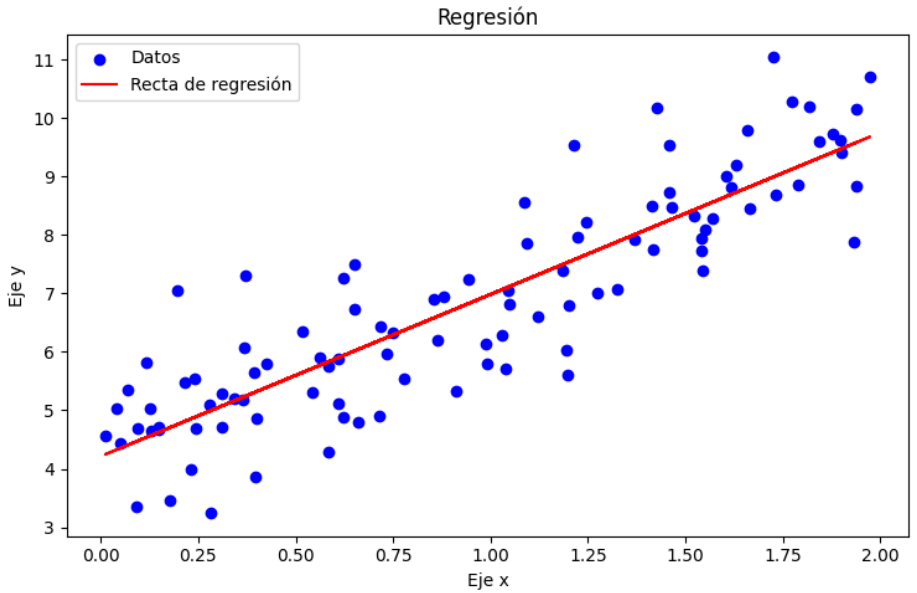




- Podemos clasificar los algoritmos según su capacidad para definir regiones de decisión no lineales en problemas de clasificación

Aprendizaje automático: posibles clasificaciones





- o según si se ocupan de predecir nuevos valores o clasificar elementos

Aprendizaje automático: algoritmos

Algoritmos lineales:

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Aprendizaje automático: algoritmos

Algoritmos lineales:

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

- Árboles de decisión
- Random forest
- Extreme gradient boosting tree (XGBoost)

Aprendizaje automático: algoritmos

Algoritmos lineales:

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

- Árboles de decisión
- Random forest
- Extreme gradient boosting tree (XGBoost)

- Máquina de vectores soporte (SVM)

Aprendizaje automático: algoritmos

Algoritmos lineales:

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

- Árboles de decisión
- Random forest
- Extreme gradient boosting tree (XGBoost)

- Máquina de vectores soporte (SVM)

- Redes neuronales

Aprendizaje automático: algoritmos

Algoritmos lineales:

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

- Árboles de decisión
- Random forest
- Extreme gradient boosting tree (XGBoost)

- Máquina de vectores soporte (SVM)

- Redes neuronales

- Ánalisis de componentes principales (PCA)

Aprendizaje automático aplicado a la detección de Malware

- **Detección de Malware en Android usando random forest** Alam, Mohammed S., and Son T. Vuong. "Random forest classification for detecting android malware." 2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing. IEEE, 2013.

- **Clasificación de Malware usando XGBoost** Kumar, Rajesh, and S. Geetha. "Malware classification using XGboost-Gradient boosted decision tree." Adv. Sci. Technol. Eng. Syst 5 (2020): 536-549.

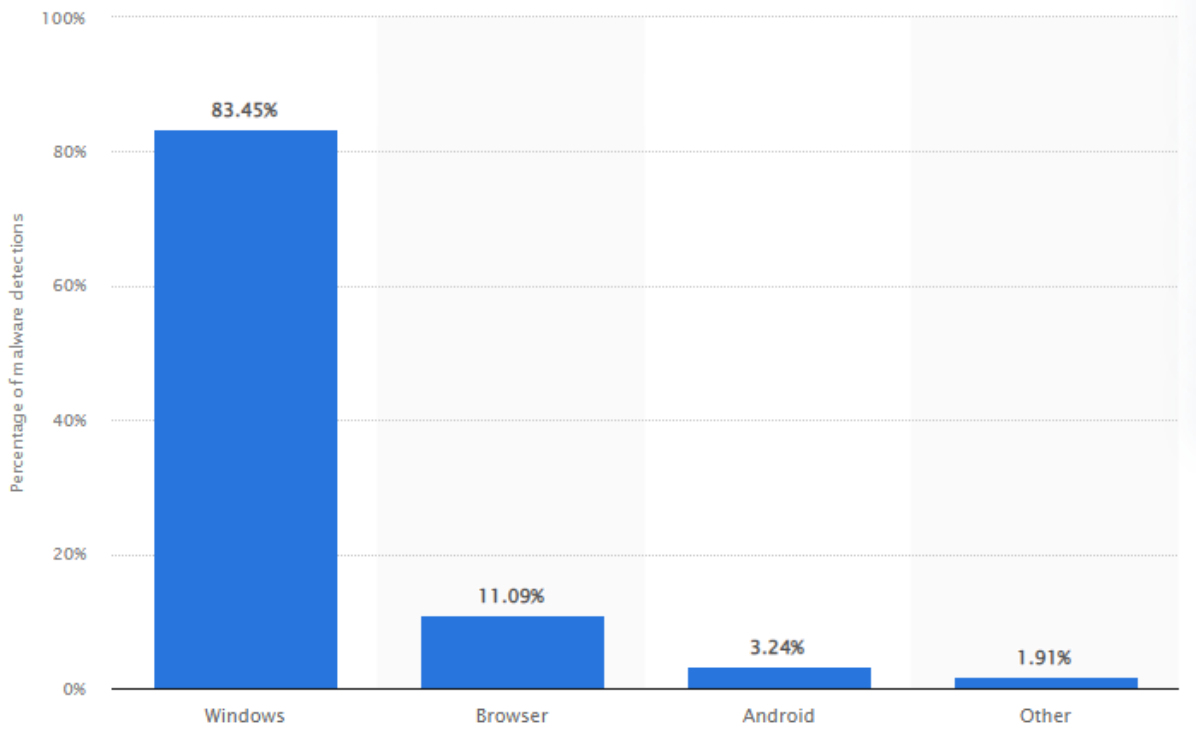
- **Clasificación de imágenes de Malware usando Deep Learning** Su, Jiawei, et al. "Lightweight classification of IoT malware based on image recognition." 2018 IEEE 42Nd annual computer software and applications conference (COMPSAC). Vol. 2. IEEE, 2018.

Soluciones comerciales y de código abierto

Soluciones comerciales			
Herramienta	**Uso de Machine Learning**	**Transparencia de Algoritmos**	**Tipo de código**
Kaspersky AntiVirus	Sí	Centros de transparencia	Propietario
Microsoft Security	Sí	Publicaciones de Microsoft	Propietario
Cylance	Sí	Publicaciones de Cylance	Propietario
Trend Micro	Sí	No se especifican algoritmos	Propietario

Soluciones de código abierto			
Herramienta	**Uso de Machine Learning**	**Transparencia de Algoritmos**	**Tipo de código**
Cuckoo Sandbox	No	Código Abierto	Código Abierto
Ember	Sí	Código Abierto	Código Abierto

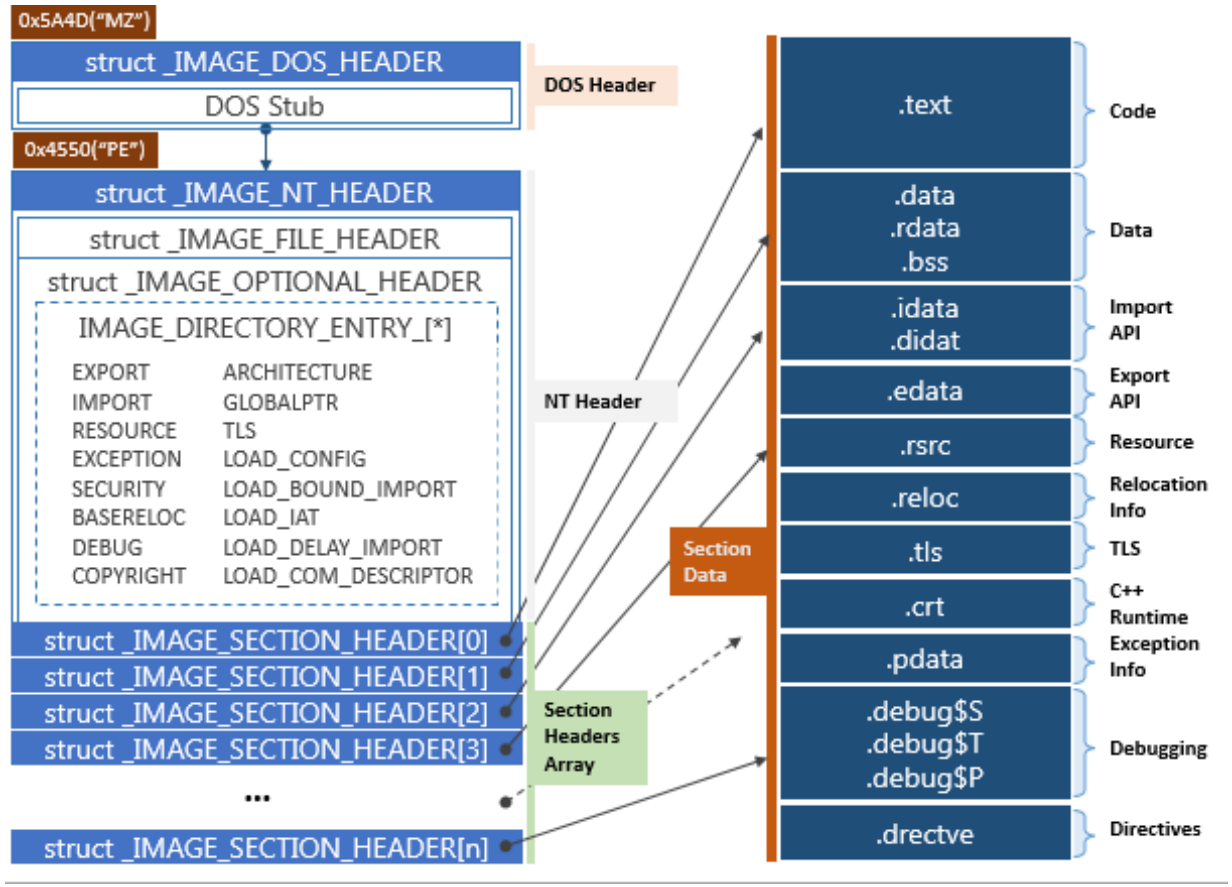
Implementación: temática



Más del 80% del Malware en 2020 tenía como objetivo el sistema operativo Windows

- Nos centraremos en la clasificación binaria de archivos ejecutables de Windows

Archivos ejecutables de Windows



- Estructura de los archivos ejecutables de Windows

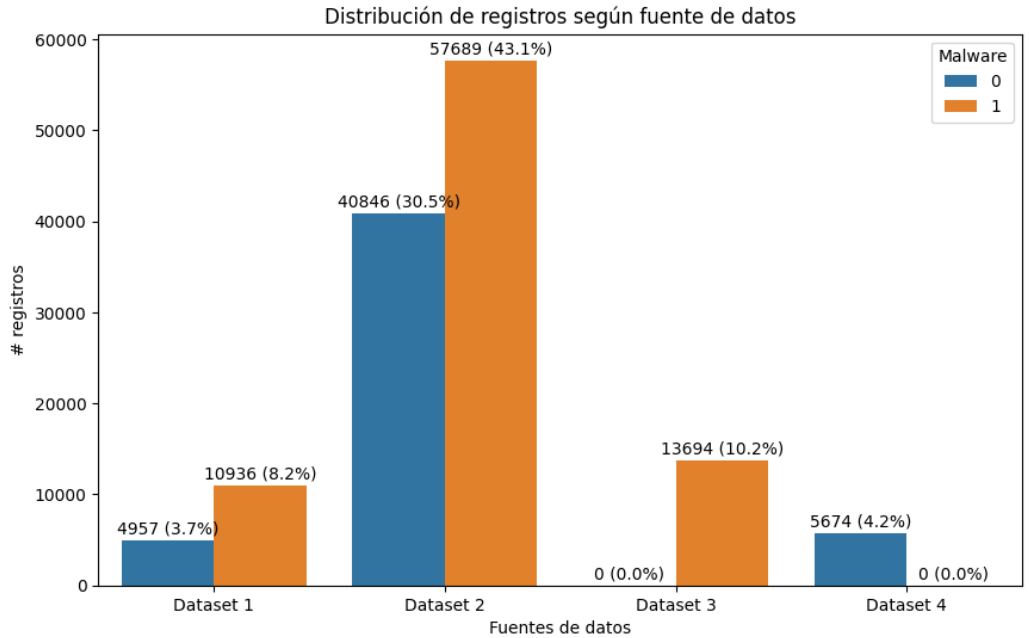
Implementación: lenguaje de programación y librerías

erocarrera/**pefile**

pefile is a Python module to read and work with PE (Portable Executable) files

Implementación: fuentes de datos y preparación

- Repositorios Github
- Conjunto de datos clasificados de Kaggle
- Fuente propia (software benigno)



A partir de estos datos se construye un dataset balanceado

Implementaciones: algoritmos

Algoritmos lineales (con y sin PCA):

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Implementaciones: algoritmos

Algoritmos lineales (con y sin PCA):

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

- Árboles de decisión
- Random forest
- Gradient Boosting trees (XGBoost)

Implementaciones: algoritmos

Algoritmos lineales (con y sin PCA):

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

- Árboles de decisión
- Random forest
- Gradient Boosting trees (XGBoost)

- Máquina de vectores soporte (SVM)

Implementaciones: algoritmos

Algoritmos lineales (con y sin PCA):

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

- Árboles de decisión
- Random forest
- Gradient Boosting trees (XGBoost)

- Máquina de vectores soporte (SVM)

- Método ensemble ad-hoc combinando Perceptrón, Árbol de decisión y SVM

Implementaciones: algoritmos

Algoritmos lineales (con y sin PCA):

- Análisis de Discriminante Lineal (LDA)
- Perceptrón

Métodos basados en árboles de decisión:

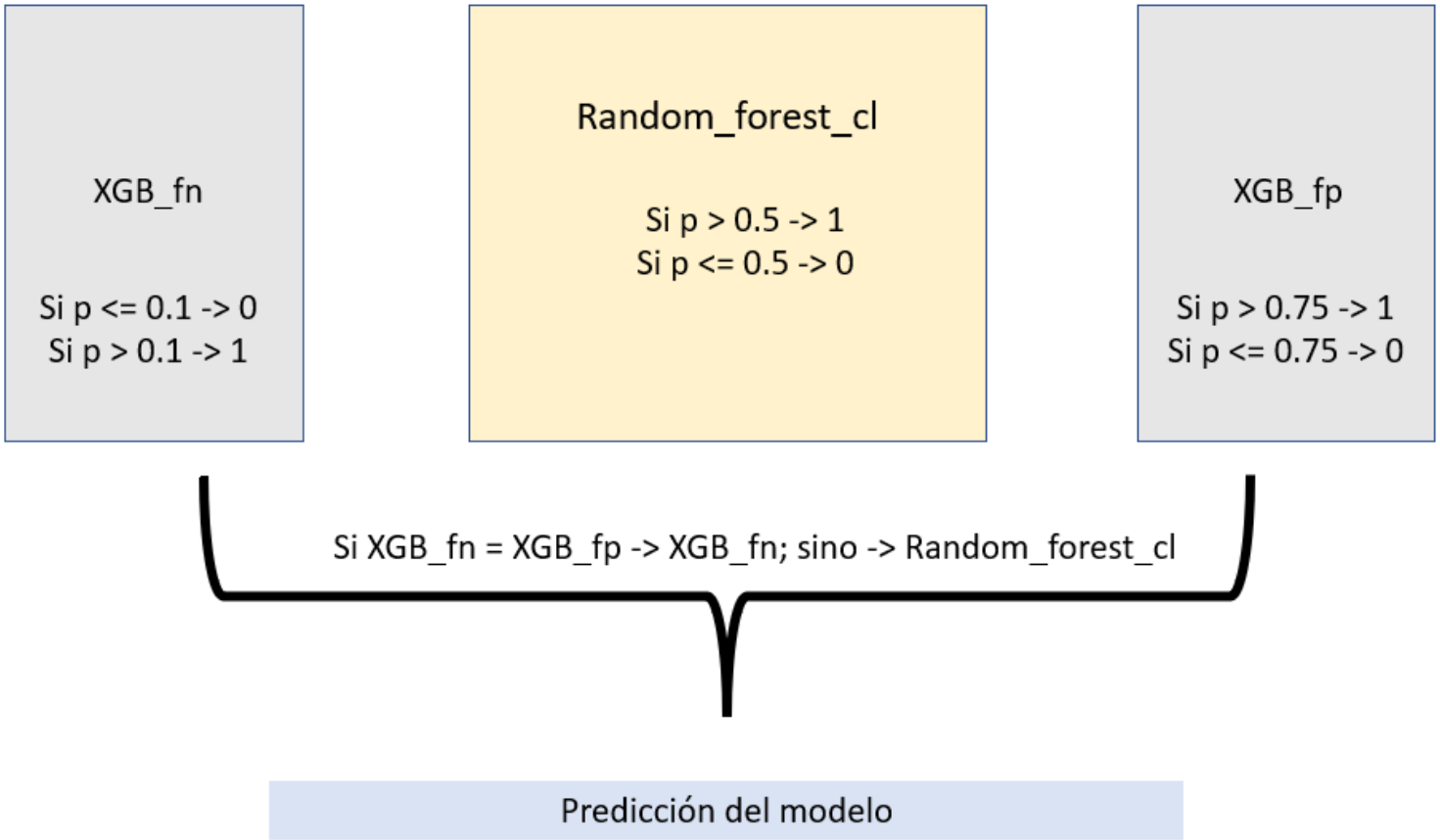
- Árboles de decisión
- Random forest
- Gradient Boosting trees (XGBoost)

- Máquina de vectores soporte (SVM)

- Método ensemble ad-hoc combinando Perceptrón, Árbol de decisión y SVM

- Red neuronal

Modelo final



- Con este modelo buscamos minimizar falsos positivos y falsos negativos

Métricas

Precision

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall (Sensibilidad)

$$\text{Recall} = \frac{TP}{TP+FN}$$

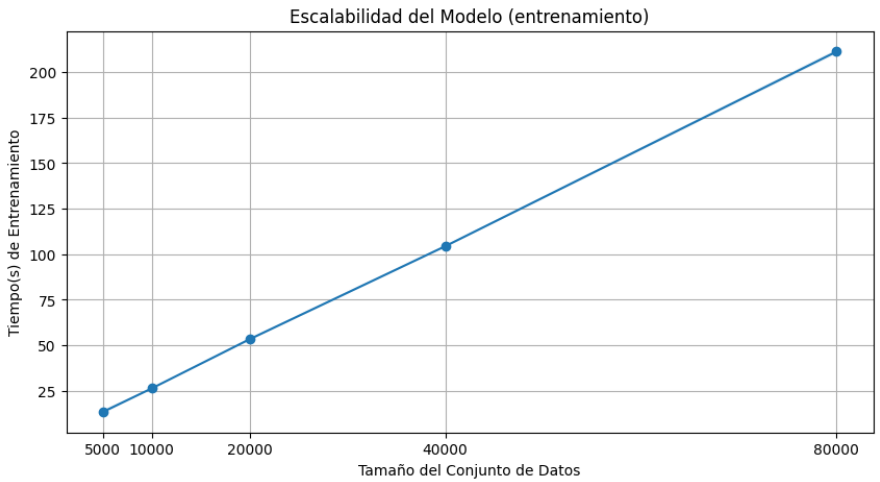
F1 Score

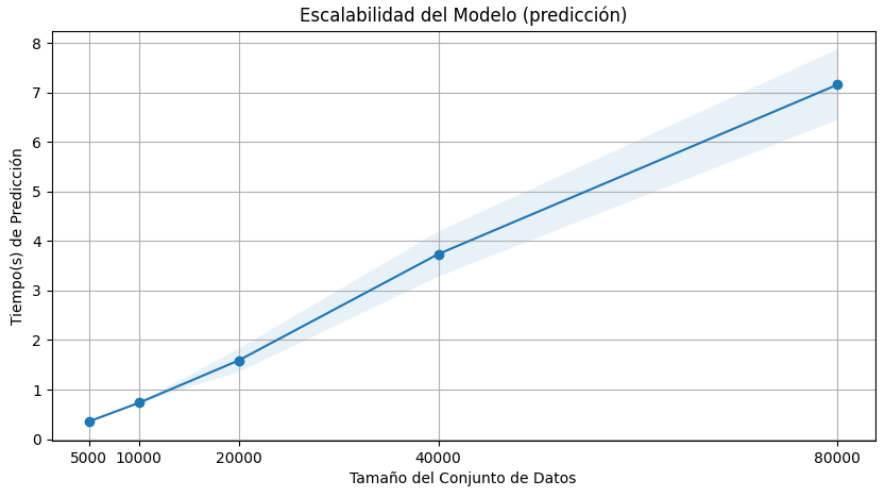
$$\text{F1 Score} = \frac{2\times(\text{precision}\times\text{recall})}{\text{precision}+\text{recall}}$$

Resultados

Modelo	F1 Score
Perceptron sin PCA	0.8305
LDA sin PCA	0.8449
Perceptron con PCA	0.8678
LDA con PCA	0.8472
Árbol de decisión	0.9660
Máquina de vectores soporte	0.8940
Método ensemble básico (votación)	0.9500
Random forest (parámetros por defecto)	0.9752
Random forest (mejores hiperparámetros)	0.9758
XGBoost (parámetros por defecto)	0.9587
XGBoost (mejores hiperparámetros)	0.9769
Red neuronal	0.9137
Modelo final (XGBoost + Random Forest)	0.9775

Escalabilidad



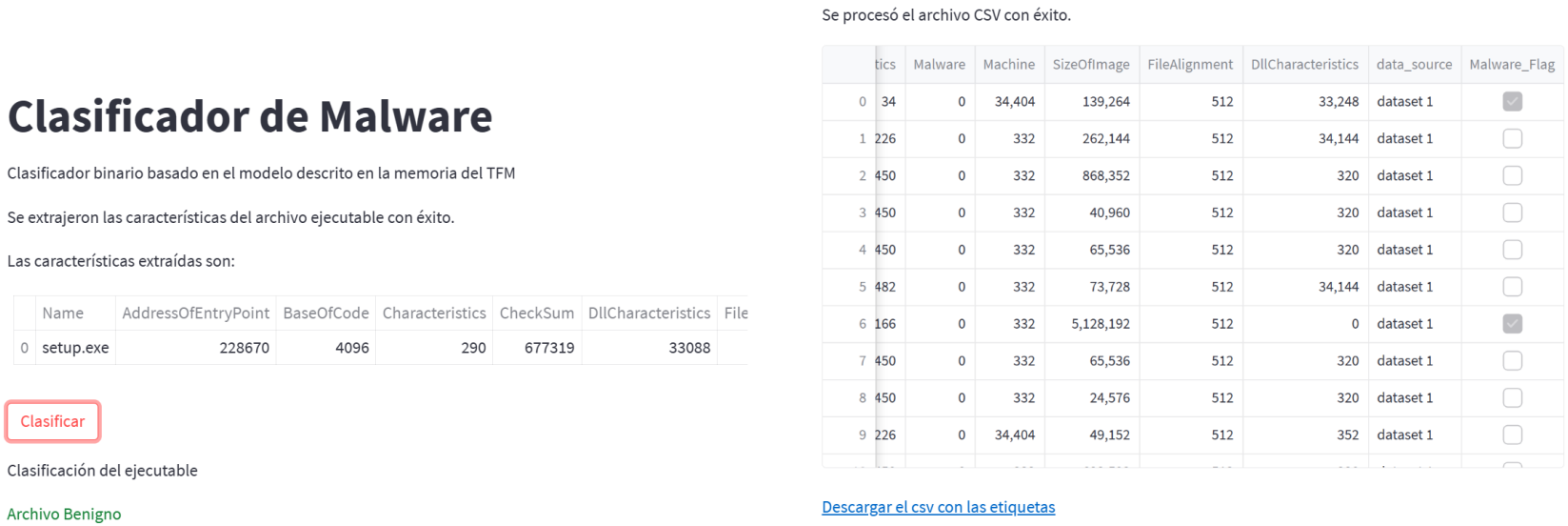


- El modelo escala linealmente tanto en el entrenamiento como realizando predicciones

Aplicación web

- La aplicación web, implementada y desplegada usando Streamlit, permite la carga tanto de archivos ejecutables como de CSV con características extraídas

Aplicación web



Clasificador de Malware

Clasificador binario basado en el modelo descrito en la memoria del TFM

Se extrajeron las características del archivo ejecutable con éxito.

Las características extraídas son:

	Name	AddressOfEntryPoint	BaseOfCode	Characteristics	CheckSum	DllCharacteristics	File
0	setup.exe	228670	4096	290	677319	33088	

Clasificar

Clasificación del ejecutable

Archivo Benigno

Se procesó el archivo CSV con éxito.

	tics	Malware	Machine	SizeOfImage	FileAlignment	DllCharacteristics	data_source	Malware_Flag
0	34	0	34,404	139,264	512	33,248	dataset 1	☑
1	226	0	332	262,144	512	34,144	dataset 1	☐
2	450	0	332	868,352	512	320	dataset 1	☐
3	450	0	332	40,960	512	320	dataset 1	☐
4	450	0	332	65,536	512	320	dataset 1	☐
5	482	0	332	73,728	512	34,144	dataset 1	☐
6	166	0	332	5,128,192	512	0	dataset 1	☑
7	450	0	332	65,536	512	320	dataset 1	☐
8	450	0	332	24,576	512	320	dataset 1	☐
9	226	0	34,404	49,152	512	352	dataset 1	☐

Descargar el csv con las etiquetas

- Dependiendo del tipo de archivo cargado se obtiene un resultado diferente al clasificar

Conclusiones

Hemos estudiado la importancia cada vez mayor de la ciberseguridad y la necesidad de recurrir al aprendizaje automático para combatir amenazas cada vez más sofisticadas.

Conclusiones

Hemos estudiado la importancia cada vez mayor de la ciberseguridad y la necesidad de recurrir al aprendizaje automático para combatir amenazas cada vez más sofisticadas.

Se ha creado un conjunto de datos balanceado, para el entrenamiento de algoritmos supervisados de aprendizaje automático en la clasificación binaria de archivos ejecutables de Windows.

Conclusiones

Hemos estudiado la importancia cada vez mayor de la ciberseguridad y la necesidad de recurrir al aprendizaje automático para combatir amenazas cada vez más sofisticadas.

Se ha creado un conjunto de datos balanceado, para el entrenamiento de algoritmos supervisados de aprendizaje automático en la clasificación binaria de archivos ejecutables de Windows.

Se han entrenado diferentes algoritmos y con los mejores se ha construido un modelo final.

Conclusiones

Hemos estudiado la importancia cada vez mayor de la ciberseguridad y la necesidad de recurrir al aprendizaje automático para combatir amenazas cada vez más sofisticadas.

Se ha creado un conjunto de datos balanceado, para el entrenamiento de algoritmos supervisados de aprendizaje automático en la clasificación binaria de archivos ejecutables de Windows.

Se han entrenado diferentes algoritmos y con los mejores se ha construido un modelo final.

Hemos creado y desplegado una aplicación web centrada en la clasificación tanto de archivos ejecutables como de sus características ya extraídas.

Muchas gracias