UIDE
Powered by
Arizona State University

MODELOS Y APRENDIZAJES

INTEGRANTES GRUPO 6:

- Renato Jacome
- Rubén Recalde
- Lilian Amán
- Santiago Borja

- **Link a repositorio público de Github**

  https://github.com/EvilWayz79/BigDataModelosYaprendizajesGrupo6/tree/main/SegundaSemana

- **Descripción del dataset personalizado**

  Se preprocesa el dataset para manejar valores nulos o inconsistentes, mostramos los valores nulos por columna.

  Preprocesar el dataset para manejar valores nulos o inconsistentes.
  nulos = malware.drop(columns=['Label']).isnull().sum()

  Mostrar los valores nulos por columna
  print("Valores nulos por columna:")
  print(nulos)
  No presenta valores nulos en el Dataset

  Imprimimos valores inconsistentes
  print(malware.describe())

  Esto nos permite conocer la media con valores de columnas menor a 0.09 y máximo 1, lo cual nos indica que existe una dispersión fuerte de los valores hacia el número cero.

  Características (X) y etiquetas (y)
  Excluir la columna de etiqueta
  X = malware.drop(columns=['Label'])
  y = malware['Label']

- **Técnicas de aprendizaje automático utilizadas y justificación de su elección.**

  Aplicamos la estrategia de muestreo en esta instancia se establece la técnica *oversampling* para el aprendizaje automático de nuestro modelo.

Esta técnica la utilizamos para aumentar la representación de las clases minoritarias al generar nuevas muestras sintéticas que son similares a las existentes en la clase minoritaria.

```
from imblearn.over_sampling import RandomOverSampler
oversampler = RandomOverSampler()

X_resampled, y_resampled = oversampler.fit_resample(X, y)
import numpy as np

Contar las clases en y
clases, conteo = np.unique(y_resampled, return_counts=True)

print("Clases después de oversampling:", dict(zip(clases, conteo)))
```

MÉTODO 1.

Este tipo de aprendizaje automático es supervisado para clasificación de *malware* porque utiliza un clasificador de bosque aleatorio (RandomForestClassifier).

Este clasificador es una técnica de aprendizaje automático que se basa en el ensamblaje de múltiples árboles de decisión entrenados en diferentes subconjuntos del conjunto de datos ya que se entrenó un modelo utilizando un conjunto de datos etiquetado con ejemplos de malware y software benigno.

El modelo luego utiliza esta información etiquetada para hacer predicciones sobre nuevos archivos ejecutables y clasificarlos como maliciosos o benignos.

MÉTODO 2.

Usando el método *LogisticRegression*, nos dio un resultado de evaluación del 100%. Por lo cual, para el ejercicio propuesto, el número de características se puede observar que el mejor método es el *LogisticRegression*, este resultado se puede dar porque la matriz es muy pequeña.

Por lo cual para probar los dos métodos seria de tener una matriz con los datos no menos desbalanceados y volver a evaluar los métodos.

- **Resultados del entrenamiento y la evaluación del modelo**

  Para el análisis de nuestro modelo hemos tomado el **método 1 RandomForestClassifier**

  Precisión: La clasificación para la clase maliciosa fue del 100% después de utilizar el *oversampling* y solo el 98% no maliciosa.

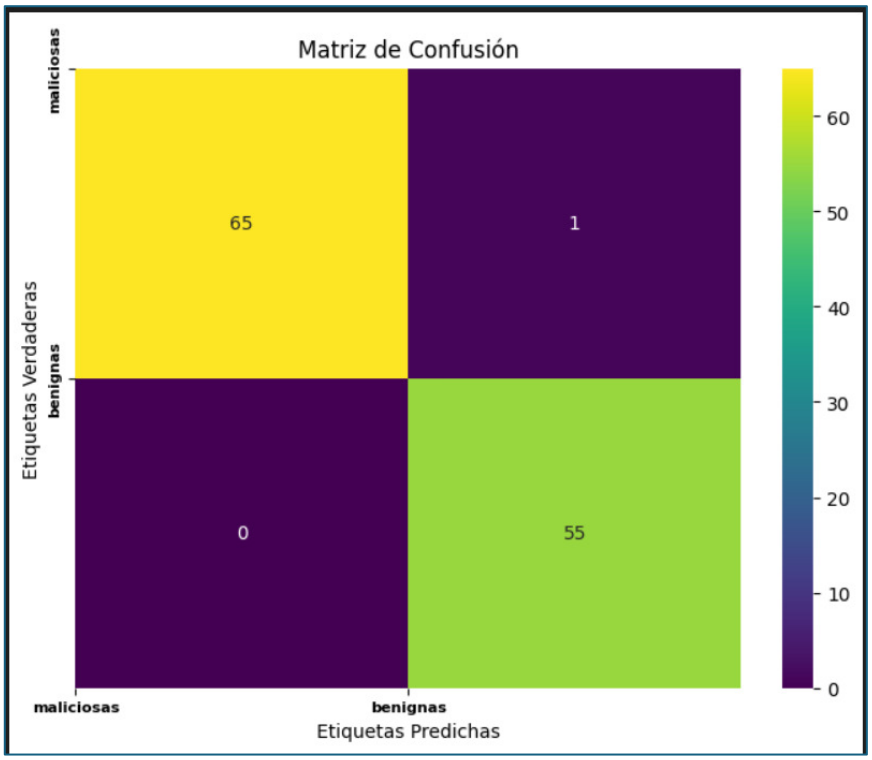
  Exhaustividad (recall): Para ambas clases, la exhaustividad es del 98% o superior, lo que indica que el modelo identificó correctamente la gran mayoría de las muestras positivas.

  f1-score: Adecuada proporcionalidad entre la precisión y la Exhaustividad (recall)

  El soporte indica el número de muestras de cada clase en el conjunto de prueba. Hay 66 muestras de la clase maliciosa y 55 muestras de la clase no maliciosa.

  Como resultado una exactitud del modelo del 99%

- **Análisis crítico de los resultados del modelo.**

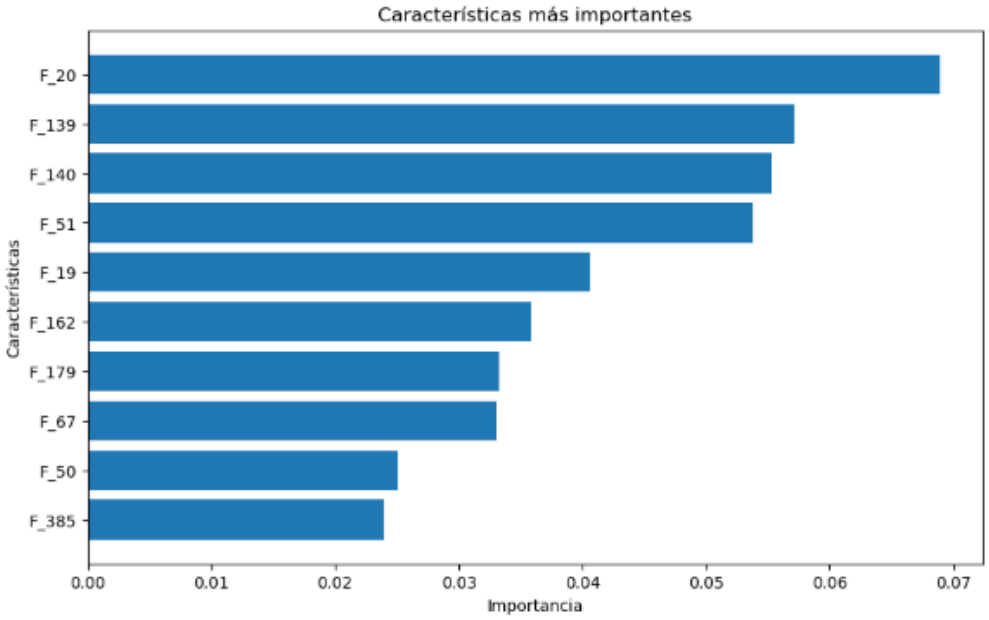


Según la matriz de confusión existen 65 programas maliciosas y 55 benignas, y 1 benigna que es clasificada como maliciosa.

- **Interpretación del modelo y características más relevantes**

Las características más importantes en un modelo de Bosque Aleatorio se determinan según su contribución a la reducción de la impureza en los nodos del árbol de decisión durante el proceso de entrenamiento.

Ayudan al modelo a hacer predicciones precisas se consideran las más importantes.


Características más importantes
F_20
F_139
F_140
F_51
F_19
F_162
F_179
F_67
F_50
F_385
Características
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Importancia