

# Malware en archivos PE

Integrantes  
Bryann Alfaro  
Diego Arredondo  
Raul Jimenez  
Donaldo García  
Oscar Saravia

# Motivación

El objetivo principal del proyecto de análisis de malware en archivos PE implementando data science es utilizar técnicas y herramientas de ciencia de datos para mejorar la detección, el análisis y la comprensión del comportamiento de los programas maliciosos en archivos Portable Executable (PE), utilizados en sistemas operativos Windows.

El proyecto incluye la aplicación de técnicas de análisis estadístico y de aprendizaje automático para identificar patrones y características comunes en el código malicioso, y la extracción de información importante del archivo PE. También se utilizan técnicas de visualización de datos para representar los resultados del análisis y facilitar la comprensión de los patrones y tendencias.

Además puede contribuir a la mejora de la seguridad informática en general al proporcionar una mejor comprensión de cómo prevenir y mitigar los ataques maliciosos en archivos PE.

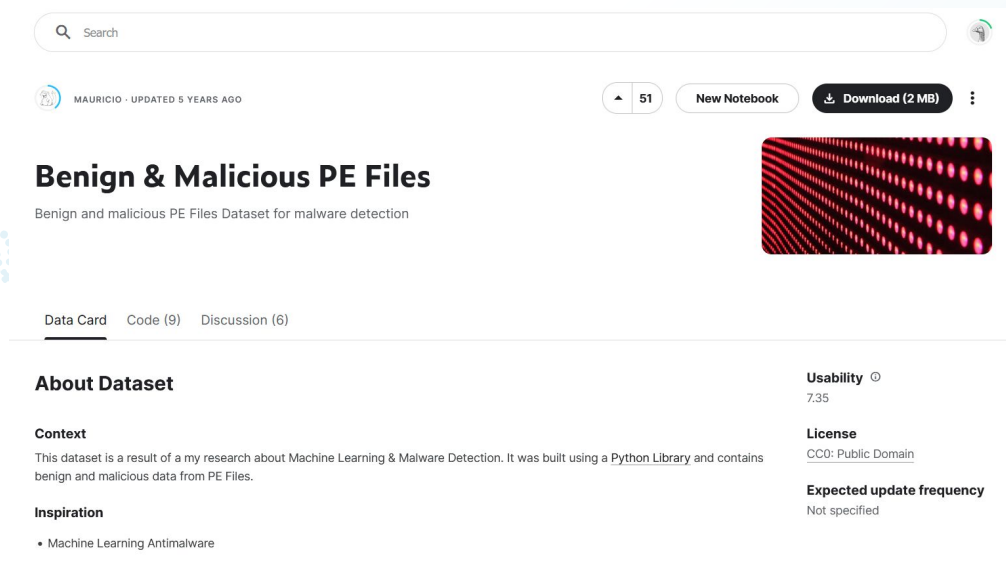
# Preguntas de investigación

- ¿Cuáles son las características más relevantes al momento de clasificar un archivo PE como malware?
- ¿Son los requerimientos mínimos solicitados por el PE útiles para determinar si es un malware?
- ¿El tamaño del código puede ser significativo al momento de detectar un archivo como malware?

# Recolección de datos

The background is a solid blue color. Overlaid on this background are several wavy, dotted lines in a slightly darker shade of blue. These lines flow from the bottom left towards the top right, creating a sense of movement and data flow. The dots are small and closely spaced, forming a continuous, undulating pattern.

1. Para la recolección de datos se utilizó la plataforma Kaggle. En esta plataforma se encontró un dataset de 19,612 archivos PE de entrenamiento y 18 de prueba
2. La base de datos cuenta con una variedad de tipos de malware y cuenta con una etiqueta que sirve para identificar si la observación corresponde a un archivo malicioso o no.
3. El dataset se encuentra bajo la licencia CC0: Public Domain lo que permite que se pueda utilizar de manera pública para cualquier propósito.



The screenshot shows the Kaggle dataset page for 'Benign & Malicious PE Files'. At the top, there is a search bar and a user profile for 'MAURICIO' updated 5 years ago. The dataset title 'Benign & Malicious PE Files' is prominently displayed, along with a description: 'Benign and malicious PE Files Dataset for malware detection'. A thumbnail image shows a grid of red and black dots. Below the title, there are tabs for 'Data Card', 'Code (9)', and 'Discussion (6)'. The 'About Dataset' section includes a 'Context' paragraph stating it's a result of research on Machine Learning & Malware Detection, built using a Python Library, and contains benign and malicious data from PE Files. The 'Inspiration' section lists 'Machine Learning Antimalware'. On the right side, there are metrics: 'Usability' (7.35), 'License' (CC0: Public Domain), and 'Expected update frequency' (Not specified).

Search

MAURICIO · UPDATED 5 YEARS AGO

51 New Notebook Download (2 MB)

## Benign & Malicious PE Files

Benign and malicious PE Files Dataset for malware detection

Data Card Code (9) Discussion (6)

### About Dataset

#### Context

This dataset is a result of a my research about Machine Learning & Malware Detection. It was built using a `Python Library` and contains benign and malicious data from PE Files.

#### Inspiration

- Machine Learning Antimalware

Usability ⓘ  
7.35

License  
[CC0: Public Domain](#)

Expected update frequency  
Not specified

# Limpieza de datos

The background features a solid blue color. Overlaid on this are several wavy, horizontal lines composed of small, dark blue dots. These lines create a sense of motion and depth, with some lines appearing more prominent than others, creating a layered effect.

## 1. Visualización de datos

train.head()

0.0s

Python

	Name	e_magic	e_cblp	e_cp	e_crc	e_cparhdr	e_minalloc	e_maxalloc	e_ss	e_sp	...	SectionMaxChar	SectionMainChar	DirectoryEntryImport	DirectoryEntryIn
0	VirusShare_a878ba26000edaac5c98eff4432723b3	23117	144	3	0	4	0	65535	0	184	...	3758096608	0	7	
1	VirusShare_ef9130570fddc174b312b2047f54cf0	23117	144	3	0	4	0	65535	0	184	...	3791650880	0	16	
2	VirusShare_ef84cdeba22be72a69b198213dada81a	23117	144	3	0	4	0	65535	0	184	...	3221225536	0	6	
3	VirusShare_6bf3608e60ebc16cbdf6ed5467d469e	23117	144	3	0	4	0	65535	0	184	...	3224371328	0	8	
4	VirusShare_2cc94d952b2efb13c7d6bbe0dd59d3fb	23117	144	3	0	4	0	65535	0	184	...	3227516992	0	2	

5 rows x 79 columns

+ Code + Markdown

## 2. Estandarización de nombres de columnas

```
1 # turn to lower case all the columns names
2 train.columns = map(str.lower, train.columns)
3 test.columns = map(str.lower, test.columns)
```

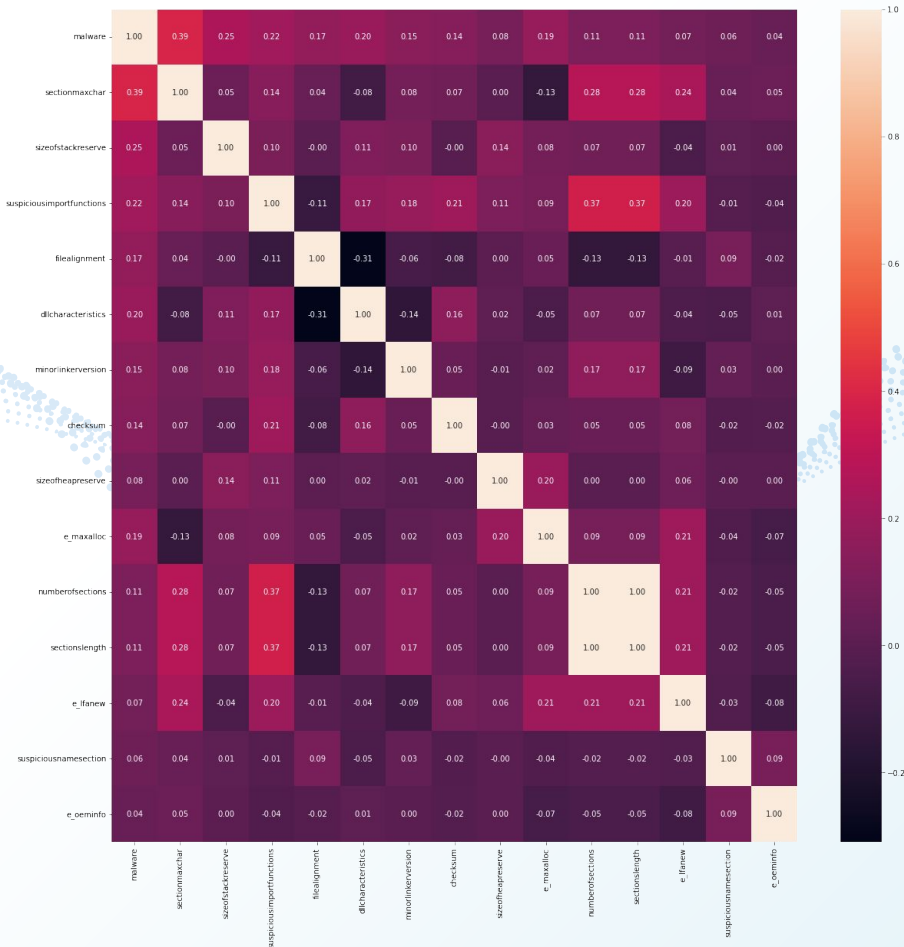
## 3. Eliminación de observaciones NA

```
1 train.dropna(inplace=True)
2 test.dropna(inplace=True)
3 train.shape
```

```
# show the correlation matrix of the train data
corr = train.corr()
corr['malware'].sort_values(ascending=False).head(15)
```

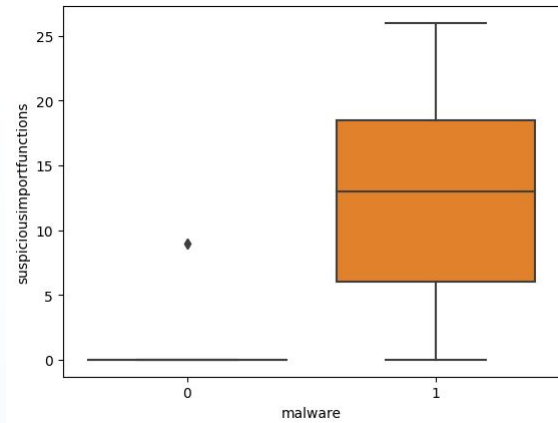
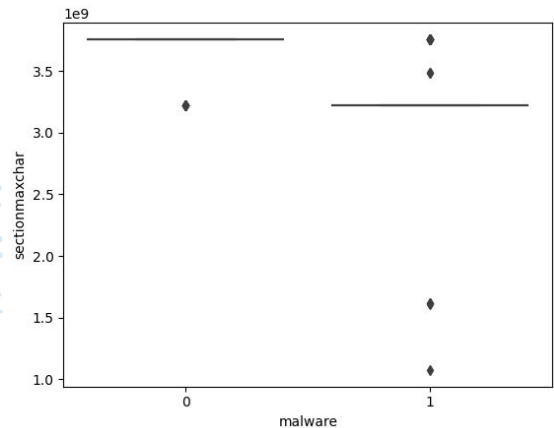
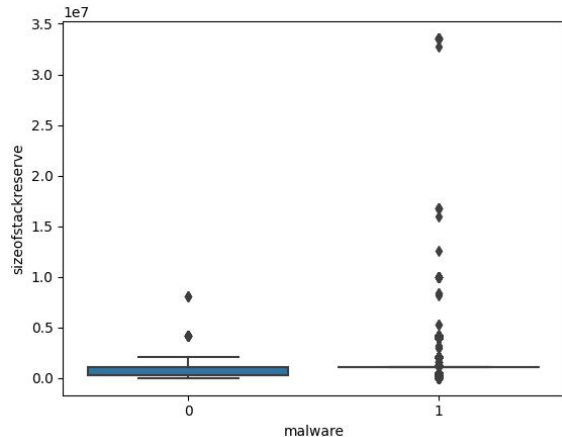
```
malware          1.000000
sectionmaxchar   0.393282
sizeofstackreserve 0.251791
suspiciousimportfunctions 0.216656
dllcharacteristics 0.197023
e_maxalloc       0.186079
filealignment    0.172926
minorlinkversion 0.145848
checksum         0.135325
numberofsections 0.109373
sectionlength    0.109309
sizeofheapreserve 0.084892
e_lfanew         0.074879
suspiciousnamesection 0.058088
sectionmaxpointerdata 0.045360
Name: malware, dtype: float64
```

4. Se analizó la correlación para poder elegir las variables que presentan la correlación más alta con el target “malware”





5. Por medio de gráficos de caja y bigotes se buscaron datos atípicos dentro de las observaciones para su tratamiento.



6. Se realizó un balanceo de datos ya que habían diferencias notorias entre los datos de malware y los benignos.

```
# Remove some of the rows that has value "1" in the malware column to balance the data
train = train[train['malware'] == 0].sample(5012).append(train[train['malware'] == 1].sample(5012))
train['malware'].value_counts()
```

```
C:\Users\raul\AppData\Local\Temp\ipykernel_20128\967369620.py:2: FutureWarning: The frame.append method is deprecated and will be removed in a future version. Use pandas.concat instead.
```

```
train = train[train['malware'] == 0].sample(5012).append(train[train['malware'] == 1].sample(5012))
```

```
0    5012
1    5012
Name: malware, dtype: int64
```

Gracias

