Lectura 3

March 6, 2024

1 Aplicaciones de Minería de Datos I

1.1 Lectura 3: Exploración y Visualización de un Conjunto de Datos - Continuación

[1]:
```
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
import statistics
import datetime
warnings.filterwarnings('ignore')
```

1.1.1 Introducción a las herramientas

Existe una amplia gama de instrumentos destinados a la visualización de datos. En el presente documento se hará uso de matplotlib, herramienta de renombre por su sencillez y su habilidad para facilitar la visualización exhaustiva de gráficos, además de ofrecer la capacidad de almacenar los resultados obtenidos.

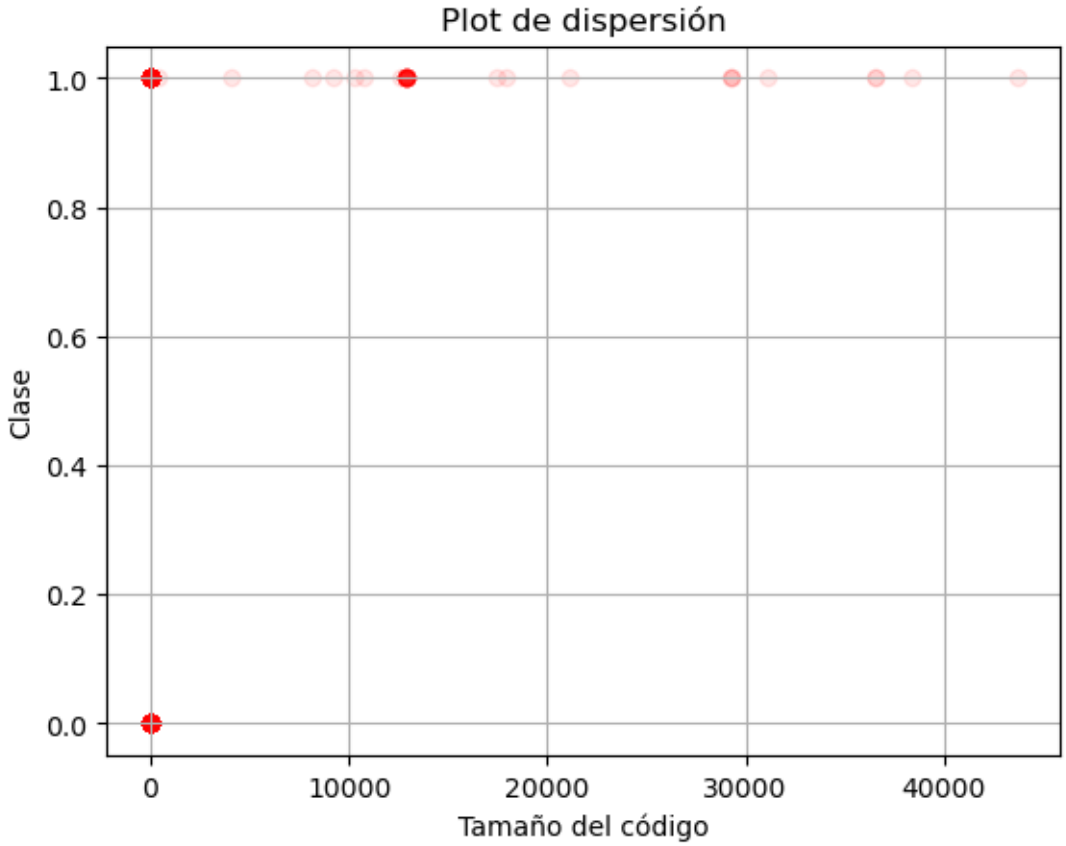
La gráfica subsiguiente presenta una comparación entre SizeOfCode, un campo opcional del Encabezado Portátil Ejecutable (PE), que detalla la magnitud del código fuente (representada por la suma total de todas las secciones de código), y el tipo de malware

Este tipo de gráfico es denominado scatter y sirve para visualizar la relación entre dos pares de valores en el conjunto de datos.

[2]:
```
dataset = pd.read_csv('https://alhernandezsua.gitlab.io/amd-misti/datasets/
↪dataset_malwares.csv',sep=',')
dataset = dataset.drop(columns=['Name'])
y = dataset['Malware']
X = dataset.drop(columns=['Malware'])

plt.scatter(X['e_cparhdr'],y,color='red',marker='o',linestyle='solid',alpha=0.
↪09)
plt.title("Plot de dispersión")
plt.xlabel('Tamaño del código')
plt.ylabel('Clase')
```

```
plt.grid()
plt.savefig("scatter.png",dpi=300)
plt.show()
```



1.1.2 Valores estadísticos

Por otro lado Seaborn, es otra biblioteca basada en matplotlib, con interfaces y desarrollo de gráficos más atractivos y estadísticamente informativos.

Si Python3 es el interprete por defecto:

pip install seaborn

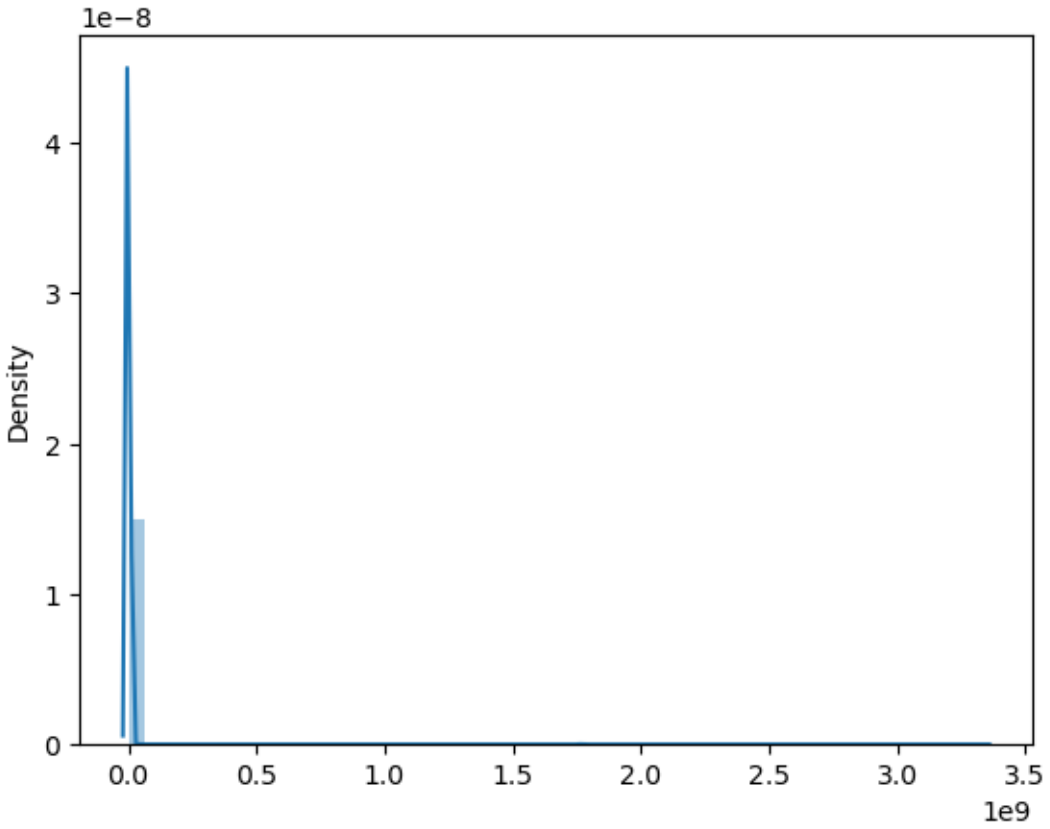
Si Python3 no es el interprete por defecto:

pip3 install seaborn

Se observa un valor mínimo de cero en el tamaño del código. **¿Será eso posible?, ¿Será algo malicioso?** Este artículo lo aborda: Benign and malware file size normalization

Si se pude reducir el número de valores o segmentarlos se podría observar que gran parte de las muestras oscilan entre los valores 0 y 7680.

[3]:
```
X['SizeOfCode'].values.sort()
sns.distplot(X['SizeOfCode'].values);
```



1.1.3 Oblicuidad (Asimetría) y Kurtosis

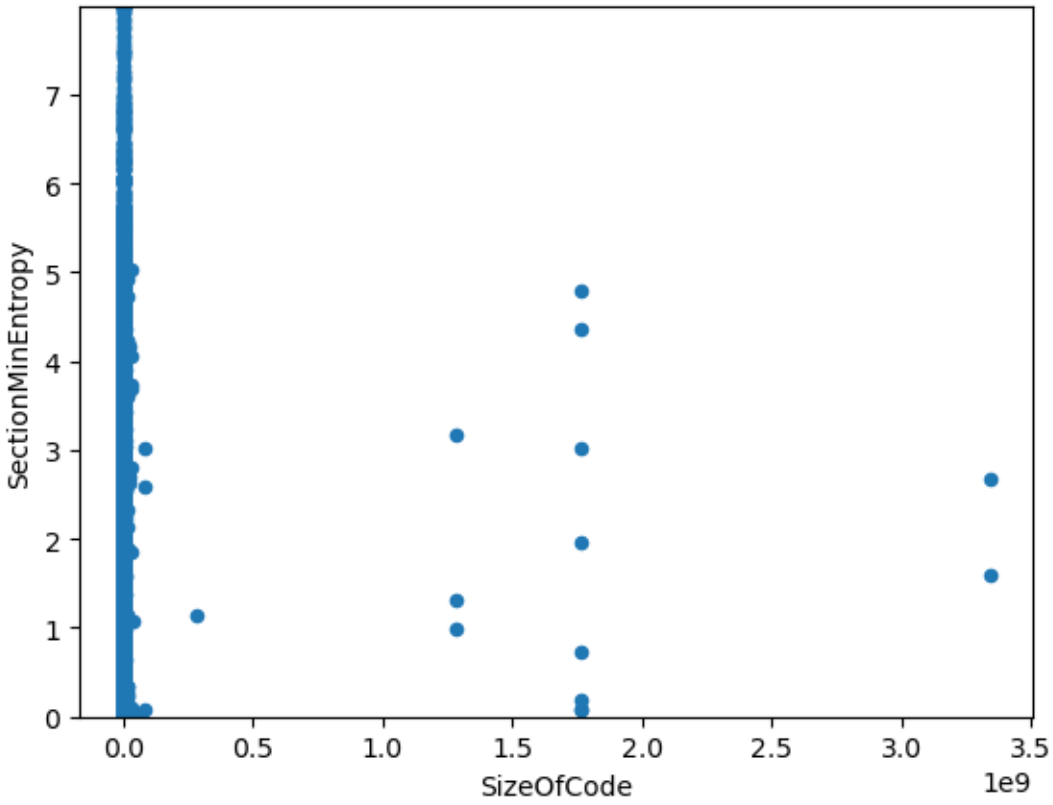
La oblicuidad, también conocida como asimetría, se emplea para estimar las probabilidades aproximadas dentro de las distribuciones, proporcionando una medida de su simetría. Este valor indica el punto de partida en el eje horizontal desde el cual se evalúa la simetría de la distribución

La kurtosis, también denominada apuntamiento, constituye una métrica de la forma que evalúa la prominencia o la planicie de una curva o distribución

[4]:
```
print("Oblicuidad: %f" % X['SizeOfCode'].skew())
print("Kurtosis: %f" % X['SizeOfCode'].kurt())
```

```
Oblicuidad: 46.199654
Kurtosis: 2394.381146
```

[5]:
```
var = 'SectionMinEntropy'
data = pd.concat([X[var],X['SizeOfCode']], axis=1)
data.plot.scatter(x='SizeOfCode', y=var, ylim=(0,max(X[var])));
```



1.1.4 Matriz de correlación(Sopa de plasma)

En estadística, la correlación constituye una métrica que describe el nivel de relación lineal entre dos variables, indicando hasta qué punto estas pueden variar de manera conjunta dentro de un rango constante

Una matriz de correlación es un cuadro que organiza las distintas características de un conjunto de datos en filas y columnas. Esta revela el grado de correlación entre cada par de características, exhibiendo en su diagonal principal la correlación perfecta que ocurre cuando una característica es comparada consigo misma

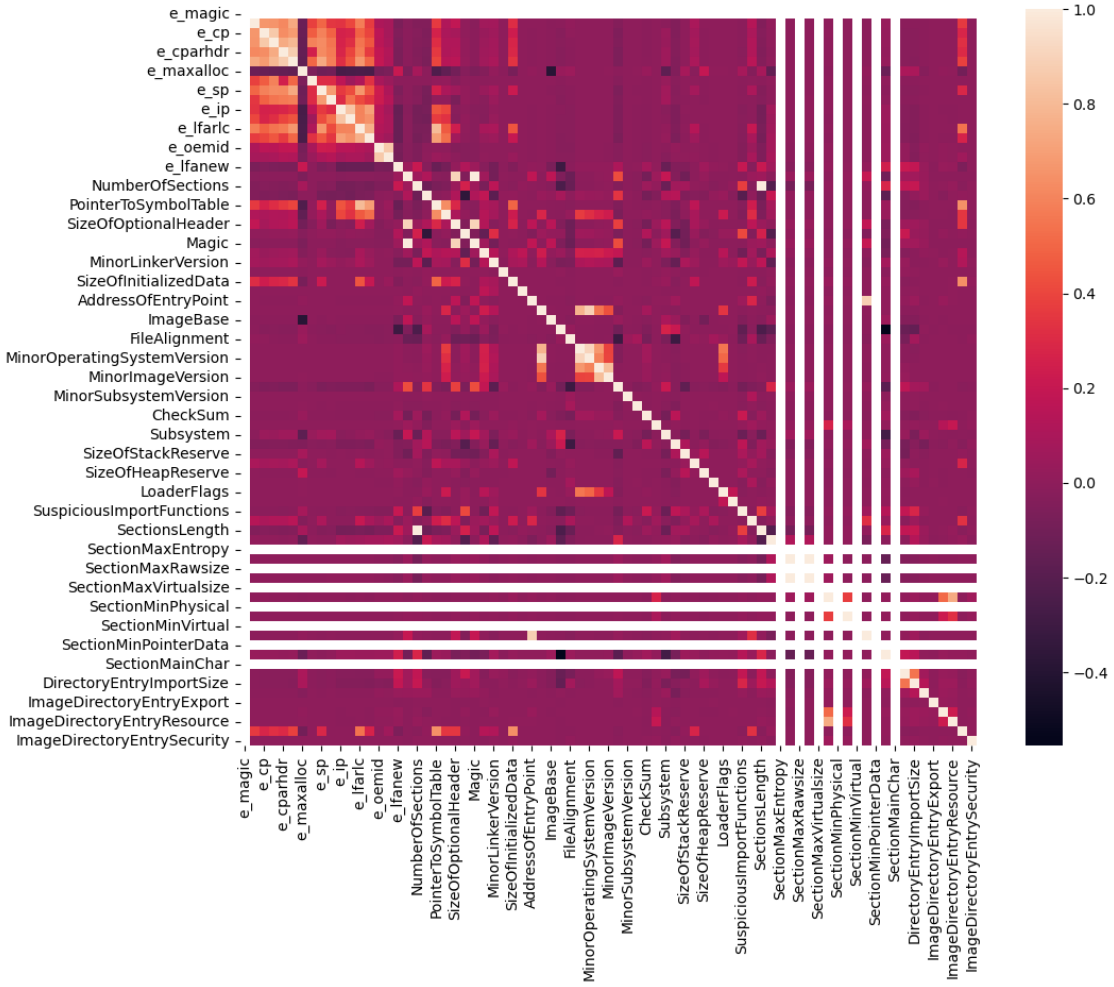
Un valor de -1 indica que hay una correlación lineal perfecta de manera negativa entre dos variables

Un valor de 0 indica que no hay una correlación lineal entre variables

Un valor de 1 indica una correlación positiva perfecta entre dos variables

[6]:
```
corrmat = X.corr()
f, ax = plt.subplots(figsize=(12, 9))
```

```
sns.heatmap(corrmat, vmax=1, square=True);
plt.savefig('corr.png',dpi=300)
```

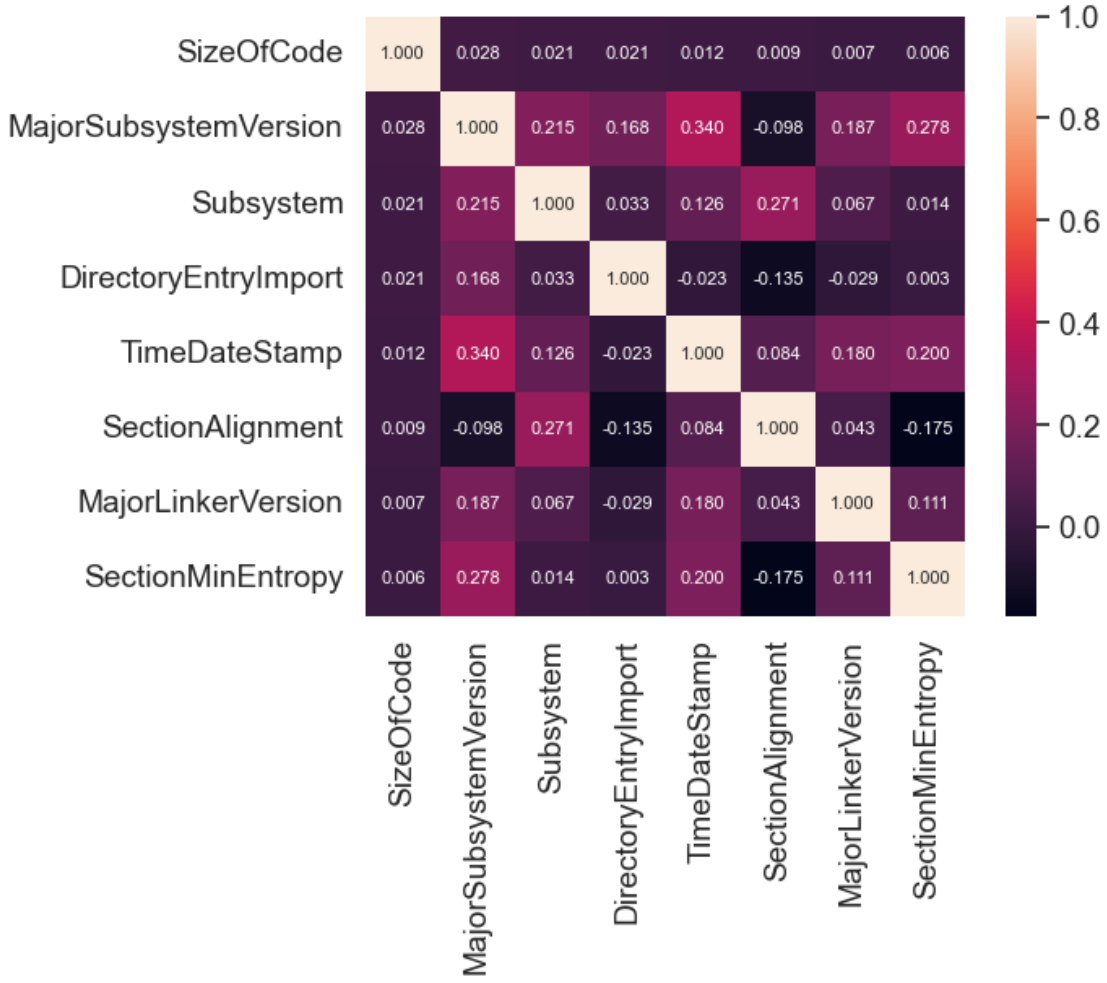


La matriz anterior se representa mediante un mapa de calor, los colores que tienden patrones más oscuros reflejan correlaciones negativas y los claros correlaciones positivas. Es importante observar que existen bastantes características con correlación positiva con respecto a otras. Por ejemplo, los valores de entropía (SectionMinEntropy) tienen correlación positiva con la mayoría de las características, una ligera variación y los valores podrían cambiar radicalmente.

1.1.5 La bola de cristal

[7]:
```
k = 8
cols = corrmat.nlargest(k, 'SizeOfCode')['SizeOfCode'].index
cm = np.corrcoef(X[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.3f',␣
 ↪annot_kws={'size':8}, yticklabels=cols.values, xticklabels=cols.values)
```

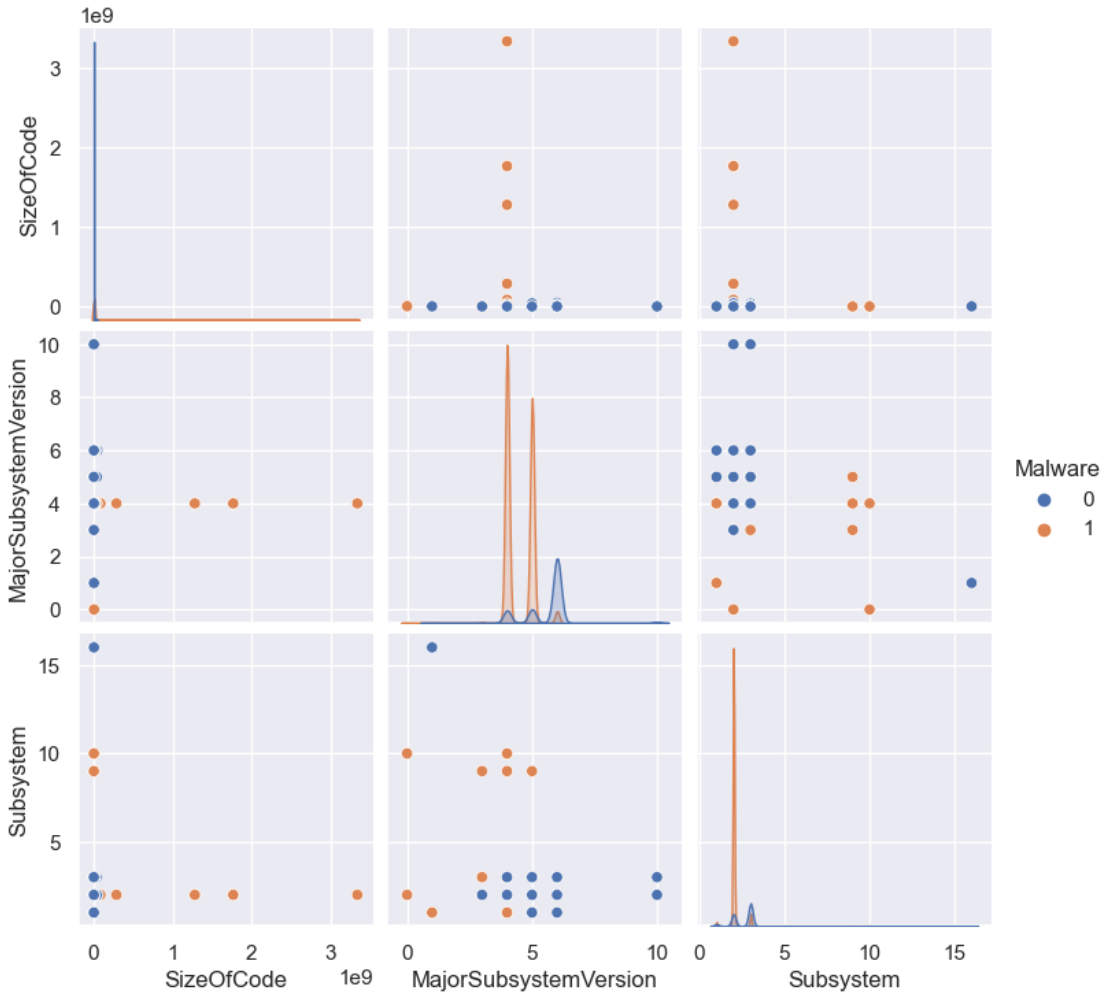
```
plt.savefig('bolacristal.png',dpi=300)
plt.show()
```



"Al representar gráficamente, por ejemplo, únicamente diez atributos asociados a la característica SizeOfCode, se puede apreciar que esta presenta una correlación insignificante en comparación con las restantes.

Podría afirmarse que es una característica óptima para el trabajo, dado que las variaciones en sus valores no influirán en el comportamiento de las demás.

[8]:
```
sns.set()
cols = ['SizeOfCode', 'MajorSubsystemVersion','Subsystem','Malware']
sns.pairplot(dataset[cols], size = 2.5,hue='Malware')
plt.show();
```



El cálculo del sesgo-varianza puede útil para mostrar las tendencias del conjunto de datos.

¿Pero qué es el sesgo? Representa una de las terminologías fundamentales en la Ciencia de Datos, describiendo el fenómeno mediante el cual un algoritmo genera resultados dañinos que afectan de forma sistemática el proceso de aprendizaje.

La varianza y el desempeño de un modelo están íntimamente conectadas.

Idealmente, se prefiere un modelo que combine un bajo sesgo con una baja varianza, aunque esto constituye un desafío en la práctica. De hecho, uno de los propósitos fundamentales de un modelo de aprendizaje consiste en equilibrar estos dos aspectos: reducir el sesgo a costa de incrementar ligeramente la varianza, o viceversa, aumentar la varianza para disminuir el sesgo.

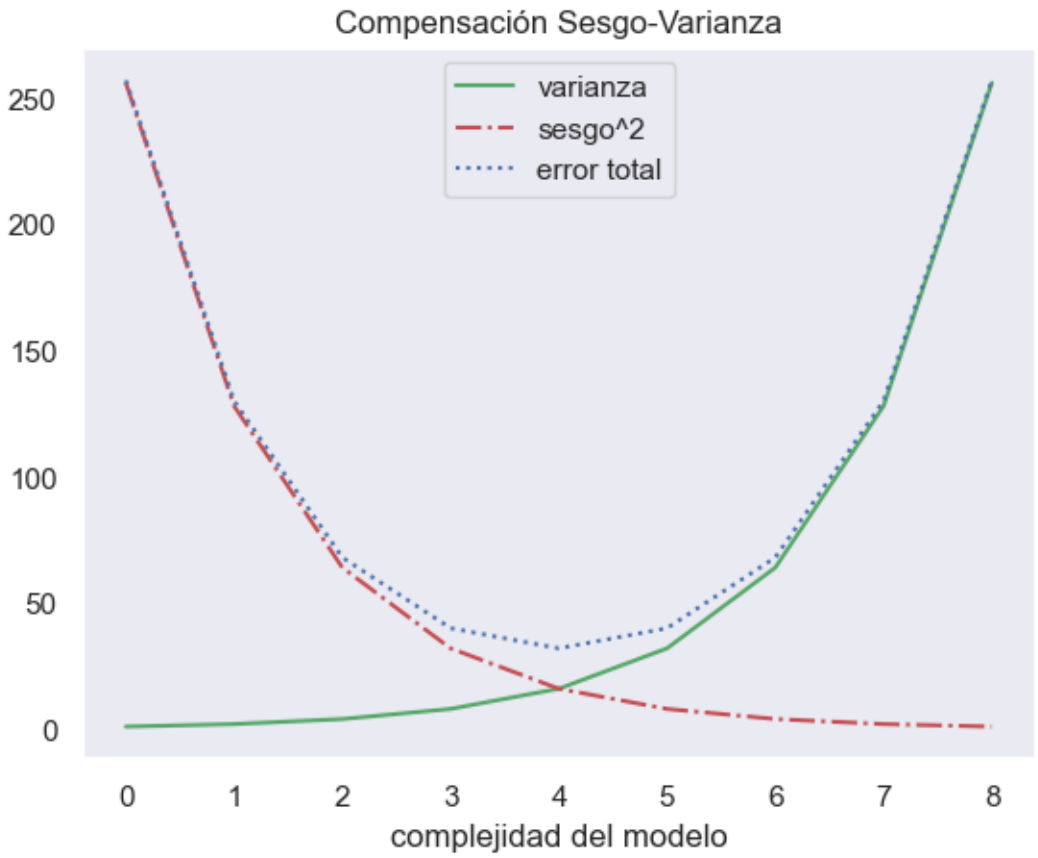
Lo anterior se describe como la balanza o compensación del sesgo-varianza.

El error total que se puede minimizar la balanza se define como:

$$error_{total} = Sesgo^2 + Varianza + Error_{irreducible} \tag{1}$$

[9]:
```
varianza = [1,2,4,8,16,32,64,128,256]
sesgo_al_cuadrado = [256,128,64,32,16,8,4,2,1]
error_total = [x + y for x,y in zip(varianza,sesgo_al_cuadrado)]
xs = [i for i,_ in enumerate(varianza)]

plt.plot(xs,varianza,'g-',label='varianza')
plt.plot(xs,sesgo_al_cuadrado,'r-.',label='sesgo^2')
plt.plot(xs,error_total,'b:',label='error total')
plt.grid()
plt.xlabel("complejidad del modelo")
plt.title("Compensación Sesgo-Varianza")
plt.legend()
plt.show()
```



1.1.6 Conjunto de datos de CVE

Common Vulnerabilities and Exposures son una lista de brechas de seguridad públicamente demostradas y categorizadas cve:

Las características del conjunto de datos son:

Índice: el identificador propuesto por CVE para cada vulnerabilidad: CVE-AÑO-NUMERO_DE_LA_VULNERABILIDAD

mod_date: fecha de modificación de la vulnerabilidad

pub_date: fecha de publicación de la vulnerabilidad

cvss (Common Vulnerability Scoring System): es un sistema de puntaje para caracterizar la severidad de una vulneabilidad en el rango de $\{0, 10\}$, en orden ascendente de impacto de la vulnerabilidad CVSS

cwe_code (Common Weakness Enumeration): es el código asociado a una taxonomía de debilidades de seguridad ya identificadas por el MITRE CWE

cwe_name: es el nombre asociado a una taxonomía de debilidades de seguridad ya identificadas por el MITRE

summary: resumen de la vulnerabilidad

[12]:
```
dataset_cve = pd.read_csv('https://alhernandezsua.gitlab.io/amd-misti/datasets/
↪cve.csv')
dataset_cve[dataset_cve['cvss']==10.0]
```

[12]:

	Unnamed: 0	mod_date	pub_date	cvss	cwe_code	cwe_name	summary
28	CVE-2011-2921	19/11/19 18:43	19/11/19 17:15	10.0	273	Improper Check for Dropped Privileges	
147	CVE-2019-15800	14/11/19 21:21	14/11/19 21:15	10.0	20	Improper Input Validation	
191	CVE-2013-3073	14/11/19 19:34	14/11/19 18:15	10.0	22	Improper Limitation of a Pathname to a Restri…	
280	CVE-2019-8248	14/11/19 16:53	14/11/19 16:15	10.0	119	Improper Restriction of Operations within the…	
281	CVE-2019-8247	14/11/19 16:53	14/11/19 16:15	10.0	119	Improper Restriction of Operations within the…	
…	…	…	…	…	…	…	
89544	CVE-2002-2365	05/09/08 20:33	31/12/02 5:00	10.0	20	Improper Input Validation	
89589	CVE-2002-2236	05/09/08 20:32	31/12/02 5:00	10.0	20	Improper Input Validation	
89597	CVE-2002-1874	05/09/08 20:31	31/12/02 5:00	10.0	20	Improper Input Validation	
89628	CVE-2007-1383	05/09/08 4:00	10/03/07 0:19	10.0	189	Numeric Errors	
89636	CVE-2005-1812	05/09/08 4:00	01/06/05 4:00	10.0	119	Improper Restriction of Operations within the…	

```
28      ktsuss versions 1.4 and prior has the uid set …
147     An issue was discovered on Zyxel GS1900 device…
191     A Symlink Traversal vulnerability exists in NE…
280     Adobe Illustrator CC versions 23.1 and earlier…
281     Adobe Illustrator CC versions 23.1 and earlier…
…                                                     …
89544   Simple WAIS (SWAIS) 1.11 allows remote attacke…
89589   Format string vulnerability in the awp_log fun…
89597   astrocam.cgi in AstroCam 0.9-1-1 through 1.4.0…
89628   Integer overflow in the 16 bit variable refere…
89636   Multiple stack-based buffer overflows in Futur…

[4496 rows x 7 columns]
```

[13]:
```
dataset_cve.describe()
```

[13]:
```
                cvss      cwe_code
count  89660.000000  89660.000000
mean       6.021429    199.690854
std        1.994757    176.177244
min        0.000000      1.000000
25%        4.300000     79.000000
50%        5.800000    119.000000
75%        7.500000    284.000000
max       10.000000   1188.000000
```

[14]:
```
dataset_cve.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 89660 entries, 0 to 89659
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  89660 non-null  object
 1   mod_date    89660 non-null  object
 2   pub_date    89660 non-null  object
 3   cvss        89660 non-null  float64
 4   cwe_code    89660 non-null  int64
 5   cwe_name    89660 non-null  object
 6   summary     89660 non-null  object
dtypes: float64(1), int64(1), object(5)
memory usage: 4.8+ MB
```

[15]:
```
# Seleccionar las columnas de interés del dataset
dataset_cve_cvss = dataset_cve[['Unnamed: 0','cvss','cwe_name','pub_date']].
  ↪head()
```

```
# Invocar el método style y el método highlight_max, para resaltar los valores más grandes
dataset_cve_cvss.style.highlight_max(color='red')
```

[15]:
```
<pandas.io.formats.style.Styler at 0x137ebeed0>
```

[16]:
```
# El atributo values de la biblioteca pandas, transforma una serie en una lista
#fechas_anio = dataset_cve_cvss["pub_date"].values
fechas_anio = [datetime.datetime.strptime(fecha,'%d/%m/%y %H:%M').year for fecha in dataset_cve["pub_date"].values]
```

[17]:
```
#fechas_anio
```

[18]:
```
# %d/%m/%y %H:M
#fecha_prueba = datetime.datetime.strptime('21/11/19 15:15','%d/%m/%y %H:%M')
#print(fecha_prueba.hour)
```

[19]:
```
#crear un objeto tipo DataFrame
fechas_df = pd.DataFrame(fechas_anio,columns=['anio'])
```

[20]:
```
fechas_df
```

[20]:

	anio
0	2019
1	2019
2	2019
3	2019
4	2019
…	…
89655	2007
89656	2007
89657	2007
89658	2007
89659	2007

[89660 rows x 1 columns]

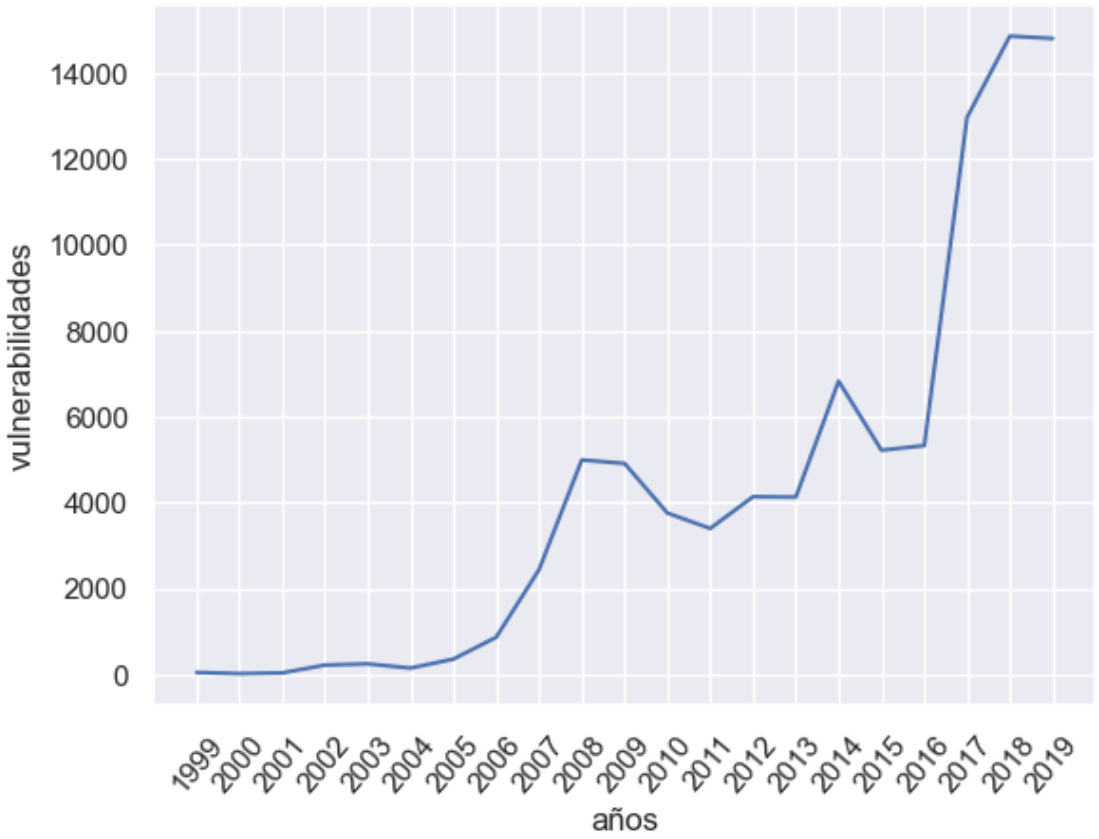
[21]:
```
# método groupby agrupa por el nombre de una columna a las muestras y el método count cuenta la
# frecuencia de cada fecha única
# keys() es el nombre del valor
# tolist() el vallor
# key: 1999 tolist(): 46
fechas_df.groupby('anio').anio.count()
```

[21]:
```
anio
1999       46
```

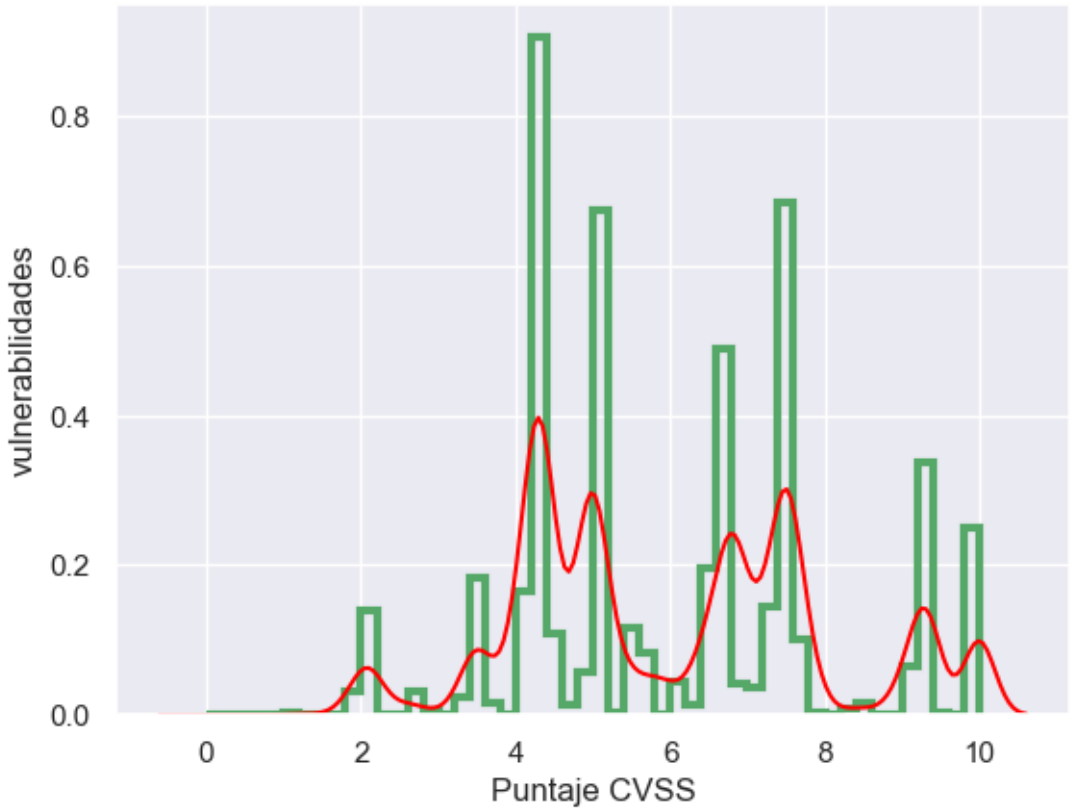
```
2000       18
2001       34
2002      217
2003      245
2004      147
2005      356
2006      864
2007     2435
2008     4991
2009     4909
2010     3755
2011     3396
2012     4135
2013     4125
2014     6825
2015     5217
2016     5325
2017    12965
2018    14855
2019    14800
Name: anio, dtype: int64
```

[34]:
```
anios = fechas_df.groupby('anio').anio.count().keys().tolist()
frecuencia = fechas_df.groupby('anio').anio.count().tolist()
```

[40]:
```
figura = sns.lineplot(x=anios,y=frecuencia);
figura.set(xlabel='años', ylabel='vulnerabilidades')
figura.set_xticklabels(anios,rotation=50)
plt.xticks(anios)
plt.show()
```



[41]:
```
#histtype: tipo de histograma
# linewidth: ancho de la línea
# alpha: opacidad
# color: color del histograma, donde g es "green"
# kws -> key:word arguments
figura = sns.distplot(dataset_cve['cvss'],color="red",hist_kws={"histtype":␣
 ↪"step", "linewidth": 3,"alpha": 1, "color": "g"});
figura.set(xlabel='Puntaje CVSS', ylabel='vulnerabilidades');
```



[ ]:
```
# La documentación de una clase, función o método
#help(sns.distplot)
```

[42]:
```
# Concat concatena n número de columnas en un solo DataFrame,
# siempre y cuando sean de la misma longitud
vulnerabilidades_df = pd.
 ↪concat([dataset_cve['cwe_name'],fechas_df['anio']],axis=1)
```

[43]:
```
vulnerabilidades_df.groupby(['cwe_name','anio']).anio.count()
```

[43]:
```
cwe_name                                          anio
 7PK - Code Quality                               2018     1
 7PK - Errors                                     2006     1
                                                  2016     9
                                                  2017    10
                                                  2018    22
                                                           ..
 XML Injection (aka Blind XPath Injection)        2013     1
                                                  2016     1
                                                  2017     7
                                                  2018     8
```

```
                                                  2019     14
Name: anio, Length: 916, dtype: int64
```

[44]:
```
pd.DataFrame(vulnerabilidades_df.groupby(['cwe_name','anio']).anio.count()).
↪to_csv('vulns.csv')
```

[ ]:

[ ]: