MAESTRIA EN CIBERSEGURIDAD Y GESTION DE LA INFORMACION

CICLO 2022-01

Data Science Aplicado a la Ciberseguridad

TRABAJO FINAL

PROFESOR DEL CURSO: ARNAU SANGRÀ

PRESENTADO POR LOS ALUMNOS:

Fuerte Rubio, Julio	E202211003
Custodio Chavarría, Gianpaul	E202211050
Valles Ríos, Lenin	E202210917

LIMA, NOVIEMBRE 2023

ÍNDICE

1. Data Science

Pregunta 1

De las siguientes preguntas, clasifica cada una como descriptiva, exploratoria, inferencia, predictiva o causal, y razona brevemente (una frase) el porqué:

1. Dado un registro de vehículos que circulan por una autopista, disponemos de su marca y modelo, país de matriculación, y tipo de vehículo (por número de ruedas). Con tal de ajustar precios de los peajes, ¿Cuántos vehículos tenemos por tipo? ¿Cuál es el tipo más frecuente? ¿De qué países tenemos más vehículos?

   Tipo: Descriptiva

   Sustento:

   Con los datos indicados como la marca, modelo, matricula y tipo de vehículo, se busca describir las características y los tipos de datos, sin hacer inferencias sobre ellos.

2. Dado un registro de visualizaciones de un servicio de video-on-demand, donde disponemos de los datos del usuario, de la película seleccionada, fecha de visualización y categoría de la película, queremos saber ¿Hay alguna preferencia en cuanto a género literario según los usuarios y su rango de edad?

   Tipo: Exploratoria

   Sustento:

   Se busca explorar los datos para descubrir patrones o tendencias que no sean evidentes a simple vista. Entender la posible relación entre la preferencia de género literario y el rango de edad de los usuarios a partir de datos existentes de visualizaciones de un servicio de video-on-demand. El enfoque es exploratorio porque no busca establecer una relación causal o predecir comportamientos futuros, sino entender patrones o preferencias potenciales basados en la información disponible.

3. Dado un registro de peticiones a un sitio web, vemos que las peticiones que provienen de una red de telefonía concreta acostumbran a ser incorrectas y provocarnos errores de servicio. ¿Podemos determinar si en el futuro, los próximos mensajes de esa red seguirán dando problemas? ¿Hemos notado el mismo efecto en otras redes de telefonía?

   Tipo: Predictiva e Inferencia

   Sustento:
   La primera pregunta se clasificaría como predictiva, ya que busca prever si los próximos mensajes de una red de telefonía específica seguirán dando problemas en el futuro basándose en el patrón observado en el registro de peticiones.

La segunda pregunta se clasificaría como inferencia, ya que indaga si el mismo efecto problemático se ha observado en otras redes de telefonía basándose en la información disponible en el registro de peticiones.

4. Dado los registros de usuarios de un servicio de compras por internet, los usuarios pueden agruparse por preferencias de productos comprados. Queremos saber si ¿Es posible que, dado un usuario al azar y según su historial, pueda ser directamente asignado a un o diversos grupos?

   Tipo: Predictiva

   Sustento:

   Se busca predecir el comportamiento de los datos en el futuro.

   En resumen, las preguntas descriptivas se centran en describir las características de los datos. Las preguntas exploratorias se centran en descubrir patrones o tendencias en los datos. Las preguntas inferenciales se centran en sacar conclusiones sobre los datos. Las preguntas predictivas se centran en predecir el comportamiento de los datos en el futuro. Las preguntas causales se centran en determinar la relación causa-efecto entre variables.

   En los casos presentados, las preguntas no tienen un componente causal, por lo que no se pueden clasificar como causales.

Pregunta 2

Considera el siguiente escenario:

Sabemos que un usuario de nuestra red empresarial ha estado usando esta para fines no relacionados con el trabajo, como por ejemplo tener un servicio web no autorizado abierto a la red (otros usuarios tienen servicios web activados y autorizados). No queremos tener que rastrear los puertos de cada PC, y sabemos que la actividad puede haber cesado. Pero podemos acceder a los registros de conexiones TCP de cada máquina de cada trabajador (hacia donde abre conexión un PC concreto). Sabemos que nuestros clientes se conectan desde lugares remotos de forma legítima, como parte de nuestro negocio, y que un trabajador puede haber habilitado temporalmente servicios de prueba. Nuestro objetivo es reducir lo posible la lista de posibles culpables, con tal de explicarles que por favor no expongan nuestros sistemas sin permiso de los operadores o la dirección.

Explica con detalle cómo se podría proceder al análisis y resolución del problema mediante Data Science, indicando de donde se obtendrían los datos, qué tratamiento deberían recibir, qué preguntas hacerse para resolver el problema, qué datos y gráficos se obtendrían, y cómo se comunicarían estos.

Respuesta:

A. Obtención de datos

- Registros de conexiones TCP: Estos registros indican las conexiones establecidas por cada máquina hacia destinos externos.
- Registro de actividades de usuarios autorizados: Para contrastar con las actividades legítimas.

B. Tratamiento de datos

- Los datos obtenidos deben ser tratados para eliminar los registros que no sean relevantes para el análisis. Por ejemplo, los registros de conexiones a servicios autorizados, los registros de conexiones a clientes remotos, y los registros de conexiones temporales a servicios de prueba.
- Identificar patrones: Analizar los registros TCP para detectar anomalías en comparación con el comportamiento normal de los usuarios. Buscar conexiones inusuales o puertos no autorizados.
- Agrupación de usuarios: Clasificar usuarios según su actividad TCP para identificar posibles grupos de comportamiento similar.
- Temporalidad: Analizar el período en que se detectaron estas actividades no autorizadas y compararlo con otros registros para contextualizar.

C. Preguntas a hacerse

Las preguntas que se deben hacerse para resolver el problema son las siguientes:

- ¿Qué usuarios o máquinas muestran conexiones inusuales o puertos no autorizados durante el período en cuestión?
- ¿Se observan patrones de comportamiento similares entre varios usuarios?
- ¿Hay momentos específicos en los que se producen estas conexiones no autorizadas?
- ¿Qué usuarios han habilitado servicios de prueba recientemente?
- ¿Cuál es la relación entre las actividades detectadas y los lugares remotos donde se conectan los clientes legítimos?

D. Datos y gráficos

Los datos que se pueden obtener para responder a estas preguntas son los siguientes:

- Grafos de conexiones: Representación visual de las conexiones TCP entre máquinas para identificar relaciones entre usuarios sospechosos y destinos no autorizados.
- Histogramas de actividad: Mostrar la actividad de conexión a lo largo del tiempo para identificar picos inusuales.
- Mapas geográficos de conexiones: Visualizar las ubicaciones de los destinos de las conexiones para comparar con los lugares remotos legítimos de los clientes.

Estos datos se pueden representar en gráficos para facilitar su interpretación. Por ejemplo, se puede crear un gráfico de barras para mostrar la lista de usuarios que realizaron conexiones a puertos no autorizados. Se puede crear un gráfico de líneas para mostrar la cantidad de conexiones a puertos no autorizados realizadas por cada usuario durante un período de tiempo determinado. Se puede crear un mapa de calor para mostrar la distribución de las conexiones a puertos no autorizados por destino.

E. Comunicación de los resultados

- Informe detallado: Detallar los hallazgos, identificar usuarios sospechosos y explicar las actividades inusuales.
- Gráficos explicativos: Utilizar visualizaciones para respaldar los hallazgos.
- Recomendaciones: Sugerir acciones preventivas y resaltar la importancia de no exponer sistemas sin autorización.

La comunicación debería ser clara, destacando que se han identificado ciertos usuarios o máquinas con actividades sospechosas y recomendando medidas para evitar este tipo de incidentes en el futuro. Es importante mantener la confidencialidad de los datos y enfocarse en la resolución del problema sin acusaciones directas hasta tener pruebas concluyentes.

F. Proceso general detallado

El proceso detallado para el análisis y resolución del problema mediante Data Science sería el siguiente:

1. Obtención de datos. Se recopilan los registros de conexiones TCP de cada máquina de cada trabajador.
2. Tratamiento de datos. Se eliminan los registros que no sean relevantes para el análisis.
3. Preguntas a hacerse. Se definen las preguntas que se deben responder para resolver el problema.
4. Obtención de datos y gráficos. Se obtienen los datos y gráficos necesarios para responder a las preguntas.
5. Comunicación de los resultados. Se comunica a los responsables de la empresa los resultados del análisis.

Este proceso puede ayudar a reducir lo posible la lista de posibles culpables, lo que facilitará la investigación del problema.

2. Introducción a R y Datos Elegantes

Pregunta 1

1. ¿Cuáles son las dimensiones del dataset cargado (número de filas y columnas)?

 # Pregunta 1: Dimensiones Data Frame
library(readr)
epa_http <- read_table("epa-http/epa-http.csv", col_names = FALSE)
#Otorgamos nombres de cabecera a las columnas
colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
View(epa_http)
dim(epa_http)

```
# Pregunta 1.1: dimensiones data frame
library(readr)
epa_http <- read_table("epa-http/epa-http.csv", col_names = FALSE)
#Otorgamos nombres de cabecera a las columnas
colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
View(epa_http)
dim(epa_http)
```

```
> dim(epa_http)
[1] 47748     7
```

2. Valor medio de la columna Bytes.

 # Pregunta 2: Valor medio de la columna Bytes
colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
#Convertimos a tipo de datos numéricos el Campo Bytes
epa_http$Bytes <- as.numeric(epa_http$Bytes)
#Calculamos el valor medio de la columna Bytes
mean(epa_http$Bytes, na.rm = TRUE)

```
# Pregunta 1.2: Valor medio de la columna Bytes
colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
#Convertimos a tipo de datos numéricos el Campo Bytes
epa_http$Bytes <- as.numeric(epa_http$Bytes)
#Calculamos el valor medio de la columna Bytes
mean(epa_http$Bytes, na.rm = TRUE)
```

```
> mean(epa_http$Bytes, na.rm = TRUE)
[1] 7352.335
```

Pregunta 2

De las diferentes IPs de origen accediendo al servidor, ¿Cuántas pertenecen a una IP claramente educativa (que contenga ".edu")?

```
colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
#De las peticiones recibidas utilizaremos nrow que es la cantidad de registros (Filas)
#El filter nos permite realizar el filtro según una condición dada, buscando las coincidencias de patrones con grepl.
FiltroGet <- nrow(filter(epa_http, grepl("edu", IP)==TRUE))
```

```
#Pregunta 2: Ips educativos(que contengan ".edu")
colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
#De las peticiones recibidas utilizaremos nrow que es la cantidad de registros (Filas)
#El filter nos permite realizar el filtfo según una condició, buscando las coincidencias de patrones con grepl
FiltroGet <- nrow(filter(epa_http, grepl("edu", IP)==TRUE))
```

```
Values
  FiltroGet    6539L
```

Pregunta 3

¿De todas las peticiones recibidas por el servidor cual es la hora en la que hay mayor volumen de peticiones HTTP de tipo "GET"?

```
epa_http.filtrado <- filter(epa_http,grepl("GET",Tipo,ignore.case = TRUE))
View(epa_http.filtrado)
#Extraemos la cadena de tiempo (Horas)
unique(str_sub(epa_http.filtrado$Tiempo,start = 5, end = 6))
# Separar la fecha 29 y 30
epa_http.filtradoGET <- data.frame(x=str_sub(epa_http.filtrado$Tiempo,start = 2, end = 3),y=str_sub(epa_http.filtrado$Tiempo,start = 5, end = 6))
View(epa_http.filtradoGET)
dia29 <- filter(epa_http.filtradoGET,grepl("29",x,ignore.case = TRUE))
dia30 <- filter(epa_http.filtradoGET,grepl("30",x,ignore.case = TRUE))
dia29$y <- as.numeric(dia29$y)
dia30$y <- as.numeric(dia30$y)
View(dia29)
View(dia30)
#Suma de GET en cada hora el día 29
CantGet29 <- dia29 %>% group_by(y) %>% summarize(total_dia29 = n())
View(CantGet29)
hora_pico29 <- CantGet29 %>% filter(total_dia29 == max(total_dia29))
View(hora_pico29)
#Suma de GET en cada hora el día 30
CantGet30 <- dia30 %>% group_by(y) %>% summarize(total_dia30 = n())
View(CantGet30)
hora_pico30 <- CantGet30 %>% filter(total_dia30 == max(total_dia30))
View(hora_pico30)
#Respuesta: La mayor cantidad de peticiones HTTP de tipo GET es a las 14H
```

```r
#Pregunta 3: Mayor volumen de peticiones HTTP de tipo "GET"
#De todas las peticiones recibidas por el servidor cual es
#la hora en la que hay mayor volumen de peticiones HTTP de tipo "GET"?
# Para realizar filtro usar el "GET"
epa_http.filtrado <- filter(epa_http,grepl("GET",Tipo,ignore.case = TRUE))
View(epa_http.filtrado)
#Extraemos la cadena de tiempo (Horas)
unique(str_sub(epa_http.filtrado$Tiempo,start = 5, end = 6))
# Separar la fecha 29 y 30
epa_http.filtradoGET <- data.frame(x=str_sub(epa_http.filtrado$Tiempo,start = 2, end = 3),y=str_sub(epa_http.filtrado$Tiempo
View(epa_http.filtradoGET)
dia29 <- filter(epa_http.filtradoGET,grepl("29",x,ignore.case = TRUE))
dia30 <- filter(epa_http.filtradoGET,grepl("30",x,ignore.case = TRUE))
dia29$y <- as.numeric(dia29$y)
dia30$y <- as.numeric(dia30$y)
View(dia29)
View(dia30)
#Suma de GET en cada hora el día 29
CantGet29 <- dia29 %>% group_by(y) %>% summarize(total_dia29 = n())
View(CantGet29)
hora_pico29 <- CantGet29 %>% filter(total_dia29 == max(total_dia29))
View(hora_pico29)
#Suma de GET en cada hora el día 30
CantGet30 <- dia30 %>% group_by(y) %>% summarize(total_dia30 = n())
View(CantGet30)
hora_pico30 <- CantGet30 %>% filter(total_dia30 == max(total_dia30))
View(hora_pico30)
#Respuesta: La mayor cantidad de peticiones HTTP de tipo GET es a las 14H
```

y	total_dia30
14	4546

Pregunta 4

De las peticiones hechas por instituciones educativas (.edu), ¿Cuantos bytes en total se han transmitido, en peticiones de descarga de ficheros de texto ".txt"?

#Filtramos todas las instituciones educativas .edu
epa_http.edu <- filter(epa_http,grepl(".edu",IP,ignore.case = TRUE))
View(epa_http.edu)
#Filtramos todas las peticiones txt
epa_http.edu.txt <- filter(epa_http.edu,grepl(".txt",URL,ignore.case = TRUE))
View(epa_http.edu.txt)
epa_http.edu.txt$Bytes <- as.numeric(epa_http.edu.txt$Bytes)
#Realizamos la suma de Bytes
suma.bytes <- sum(epa_http.edu.txt$Bytes,na.rm = TRUE)
View(suma.bytes)
#La suma es 3017871

```r
#Pregunta 4: De las peticiones hechas por instituciones educativas (.edu),
#¿Cuantos bytes en total se han transmitido, en peticiones de descarga
#de ficheros de texto ".txt"?

#Filtramos todas las instituciones educativas .edu
epa_http.edu <- filter(epa_http,grepl(".edu",IP,ignore.case = TRUE))
View(epa_http.edu)

#Filtramos todas las peticiones txt
epa_http.edu.txt <- filter(epa_http.edu,grepl(".txt",URL,ignore.case = TRUE))
View(epa_http.edu.txt)

epa_http.edu.txt$Bytes <- as.numeric(epa_http.edu.txt$Bytes)

#Realizamos la suma de bytes
suma.bytes <- sum(epa_http.edu.txt$Bytes,na.rm = TRUE)
View(suma.bytes)

#La suma es 3017871
```

Name	Type	Value
suma.bytes	double [1]	3017871

Pregunta 5

Si separamos la petición en 3 partes (Tipo, URL, Protocolo), usando str_split y el separador " " (espacio), ¿Cuántas peticiones buscan directamente la URL = "/"?

IP	Tiempo	Tipo	URL	Protocolo	Codigo	Bytes
141.243.1.172	[29:23:53:25]	"GET	/Software.html	HTTP/1.0"	200	1497
query2.lycos.cs.cmu.edu	[29:23:53:36]	"GET	/Consumer.html	HTTP/1.0"	200	1325
tanuki.twics.com	[29:23:53:53]	"GET	/News.html	HTTP/1.0"	200	1014
wpbfl2-45.gate.net	[29:23:54:15]	"GET	/	HTTP/1.0"	200	4889
wpbfl2-45.gate.net	[29:23:54:16]	"GET	/icons/circle_logo_small.gif	HTTP/1.0"	200	2624
wpbfl2-45.gate.net	[29:23:54:18]	"GET	/logos/small_gopher.gif	HTTP/1.0"	200	935
140.112.68.165	[29:23:54:19]	"GET	/logos/us-flag.gif	HTTP/1.0"	200	2788
wpbfl2-45.gate.net	[29:23:54:19]	"GET	/logos/small_ftp.gif	HTTP/1.0"	200	124
wpbfl2-45.gate.net	[29:23:54:19]	"GET	/icons/book.gif	HTTP/1.0"	200	156
wpbfl2-45.gate.net	[29:23:54:19]	"GET	/logos/us-flag.gif	HTTP/1.0"	200	2788

#El conjunto de datos filtrado resultante se almacena en el objeto "separador"
#El código filtra el conjunto de datos epa_http para que solo incluya filas donde el valor de la columna URLsea igual a /
#y luego cuenta el número de filas en el conjunto de datos filtrado.
separador <- filter(epa_http, URL=="/")
#La función nrow() se utiliza para contar el número de filas en un data frame.
peticiones<- nrow(separador)

```
#Pregunta 5: Usando str_split y el separador " " (espacio), ¿cuantas peticiones buscan directamente la URL = "/"?

#La primera línea de código filtra el conjunto de datos epa_http para que solo incluya filas donde el valor de la columna
#de peticiones sea igual a /.
#El conjunto de datos filtrado resultante se almacena en el objeto "separador"
#El código filtra el conjunto de datos epa_http para que solo incluya filas donde el valor de la columna URLsea igual a /
#y luego cuenta el número de filas en el conjunto de datos filtrado.
separador <- filter(epa_http, URL=="/")
#La función nrow() se utiliza para contar el número de filas en un data frame.
peticiones<- nrow(separador)
```

```
peticiones                2382L
```

Pregunta 6

Aprovechando que hemos separado la petición en 3 partes (Tipo, URL, Protocolo) ¿Cuantas peticiones NO tienen como protocolo "HTTP/0.2"?

colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
View(epa_http)
#Filtramos todas las consultas HTTP/0.2
epa_http.protocolof <- filter(epa_http,substr(epa_http$Protocolo,1,8)!="HTTP/0.2")
Pregunta6<- NROW(epa_http.protocolof$Protocolo)
View(epa_http.protocolof)
View(Pregunta6)
#Respuesta 6: La cantidad de peticiones que no es HTTP/0.2 es 47747

```
#Pregunta 6: Cantidad de peticiones que NO tienen como protocolo HTTP/0.2
colnames(epa_http) <- c("IP", "Tiempo", "Tipo", "URL", "Protocolo", "Codigo", "Bytes")
View(epa_http)
#Filtramos todas las consultas HTTP/0.2
epa_http.protocolof <- filter(epa_http,substr(epa_http$Protocolo,1,8)!="HTTP/0.2")
Pregunta6<- NROW(epa_http.protocolof$Protocolo)
View(epa_http.protocolof)
View(Pregunta6)
#Respuesta 6: La cantidad de peticiones que no es HTTP/0.2 es 47747
```

```
Pregunta6      47747L
```