Analisis de Dataset en R

Aleksandra Dubovik, Brian Calatrava y Sergio Suárez

January 9, 2023

Abstract

En este informe presentamos nuestro tarbajo con el dataset "Data Breaches"[1]. En la introducción hablamos de interese en este dataset, porque lo elegimos. En sección 2 damos discripción del dataset y su visualización. El las secciónes sigiuentes ponemos las preguntas de ML que hemos desarollado individualemnte, que metodos de analisis de datos hemos elegido para tratar el problema y que resultados tenemos.

1 Introducción

Los algoritmos de aprendizaje automático se dividen en dos categorías principales: supervisados y no supervisados.

Los algoritmos supervisados requieren que se proporcione un conjunto de datos etiquetados, es decir, un conjunto de entradas y sus correspondientes salidas deseadas. A partir de estos datos, el algoritmo construye un modelo que puede hacer predicciones sobre nuevos ejemplos de entrada. Los ejemplos de algoritmos supervisados incluyen regresión lineal y árboles de decisión.

Los algoritmos no supervisados, por otro lado, no requieren que se proporcionen etiquetas de salida. En su lugar, el algoritmo trata de encontrar patrones o relaciones en los datos de entrada. Los ejemplos de algoritmos no supervisados incluyen k-means y el algoritmo de agrupamiento jerárquico.

El preprocesamiento de datos es el proceso de limpieza y transformación de los datos de entrada antes de entrenar un modelo de aprendizaje automático. El objetivo del preprocesamiento de datos es preparar los datos de manera que sean adecuados para su uso en el modelo de aprendizaje automático. Esto puede incluir tareas como la eliminación de datos faltantes o ruidosos, la normalización de variables numéricas y la codificación de variables categóricas.

La visualización de datos es el proceso de crear gráficos y otras representaciones visuales de los datos con el fin de comprender mejor su significado y su relación con otros datos. La visualización de datos es una herramienta valiosa para explorar y entender los datos antes de entrenar un modelo de aprendizaje automático, así como para evaluar y comparar los resultados del modelo. Algunos ejemplos comunes de visualizaciones de datos son gráficos de dispersión, gráficos de barras y gráficos de línea.

2 Exploracion de Datos

El dataset consiste de 352 filas y 7 columnas. Las features que nos interesan, son:

- *Entity* - parametro categorico en string format. El nombre de la organización víctima que sufrió un ciberataque.
- *Year* - parametro numerico, un integer. El año en que se produjo la intrusión.
- *Records* - parametro numerico, un integer. El número de registros que se vieron comprometidos en la intrusión. Algunas filas tienen una entrada "unknown" debido a que no se pudo definir un numero exacto de datos robados. Ademas algunas filas tienen una cantidad de records escrita en palabras, como "over 5,000,000" o "tens of thousands"

[1] https://www.kaggle.com/datasets/thedevastator/data-breaches-a-comprehensive-list

- *Organization type* - paramtro categórico en string format. El tipo de organización que fue infectada.
- *Method* - parametro categorico en string format. El método que se utilizó para vulnerar la organización.

Explorando la columna "Year" hemos encontrado que el dataset tiene datos desde 2004 hasta el 2022.

2.1 Preprocesamiento de datos

Como se puede comprobar en Kaggle, nuestro dataset está en formato .CSV y tiene volaración de 10. Esto significa que no hay ni valores perdidos, ni valores de tipos falsos como datos categoricos en columnas numericas. El dataset está listo para el uso desde el principio. Pero a comprobar esto en RStudio, hemos encontrado muchos errores, por ejemplo que todas las columnas eran de tipo "caracter" como en la imagen 1.

```
> str(data)
'data.frame':   352 obs. of  7 variables:
 $ X                : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Entity           : chr  "21st Century Oncology" "500px" "Accendo Insurance Co." "Adobe Systems Incorporated" ...
 $ Year             : chr  "2016" "2020" "2020" "2013" ...
 $ Records          : chr  "2200000" "14870304" "175350" "152000000" ...
 $ Organization.type: chr  "healthcare" "social networking" "healthcare" "tech" ...
 $ Method           : chr  "hacked" "hacked" "poor security" "hacked" ...
 $ Sources          : chr  "[5][6]" "[7]" "[8][9]" "[10]" ...
```

Figure 1: Los tipos de datos en el dataset antes del preprocesamiento

Hemos convertido los tipos de columnas "Year" y "Records" en integers. Además, tuvimos que preprocesar los datos de las columnas "Records", "Organization.type" y "Method". Debido a que hubo los valores categóricos, tuvimos que cambiarlas a numericas a mano en columna "Records". Por ejemplo en vez de "ten of thousands" hemos puesto "59,500" porque "ten of thousands" puede ser entre 10,000 y 99,000. Nuestro valor es el medio entre los dos. Para rellenar otras valores, tuvimos que investigar por ejemplo cuantas data records tiene una libreria. En casos en que la investigacion no era posible (como "G20 data") hemos rellenado las celdas con valor mediano.
Para columnas "Organization.type" y "Method" hemos listado los valores únicos. Despues hemos reorganizado las catagorías para que no se repitan. Por ejemplo "healthcare" y "health" son las mismas categorias. En columna method "hacking" y "hacked" tambien son lo mismo. Al final hemos hecho 9 categorias de 70 en la columna "Organization.type" y 8 categorias de "Method" en vez de 26 categorias. En la imagen 2 se puede ver la version final de dataset:

```
> str(data)
'data.frame':   355 obs. of  7 variables:
 $ X                : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Entity           : chr  "21st Century Oncology" "500px" "Accendo Insurance Co." "Adobe Systems Incorporated" ...
 $ Year             : num  2016 2020 2020 2013 2019 ...
 $ Records          : num  2.20e+06 1.49e+07 1.75e+05 1.52e+08 7.50e+06 ...
 $ Organization.type: chr  "healthcare" "social networking" "healthcare" "tech" ...
 $ Method           : chr  "hacked" "hacked" "poor security" "hacked" ...
 $ Sources          : chr  "[5][6]" "[7]" "[8][9]" "[10]" ...
```

Figure 2: Los tipos de datos en el dataset despues del preprocesamiento

Finalmente hemos explorado este dataset sin aplicar los algoritmos de ML:

- El mayor número de datos robados es 885.000.000
- Observamos los registros de vulneraciones de datos de los últimos 18 años, desde 2004 hasta 2022.
- La categoría más atacada es la financiera, con 84 ataques registrados.
- La categoría menos atacada es la militar, con 9 ataques registrados.
- El método más utilizado para robar datos es el hacking con 197 ataques registrados.

- El método menos utilizado para robar datos es la ingeniería social, con 3 ataques registrados.

Ya se puede ver que nuestro dataset tiene mucho **bias**. El tipo de ataque más usado es 'hacking' porque el dataset no tiene la información sobre todos los ataques de tipo 'social engineering'. Este dataset explora información que esta público pero mucha información sobre ciberataques está secreta. Por eso no podemos confiar mucho en este dataset, pero independiente de eso podemos sacar conocimientos interesantes.
Durante el análisis hemos visto que los años 2021 y 2022 no tienes las filas suficientes para incluir estos años en análisis, por eso analizamos el periodo 2004-2020 en siguientes secciones.

2.2 Visualización de datos

En este sección visualizamos los datos primeros:

- Visualizar el progreso de los ataques a lo largo de los años utilizando el conjunto de datos Data Breaches

```r
attacks_by_year <- dataBreaches %>%
  group_by(Year, Method) %>%
  summarize(count = n())

# Crea el gráfico de barras apiladas
ggplot(attacks_by_year, aes(x=Year, y=count, fill=Method)) +
  geom_bar(stat="identity", position="stack")
```

Figure 3: Codigo en R para realizar la visualizacion

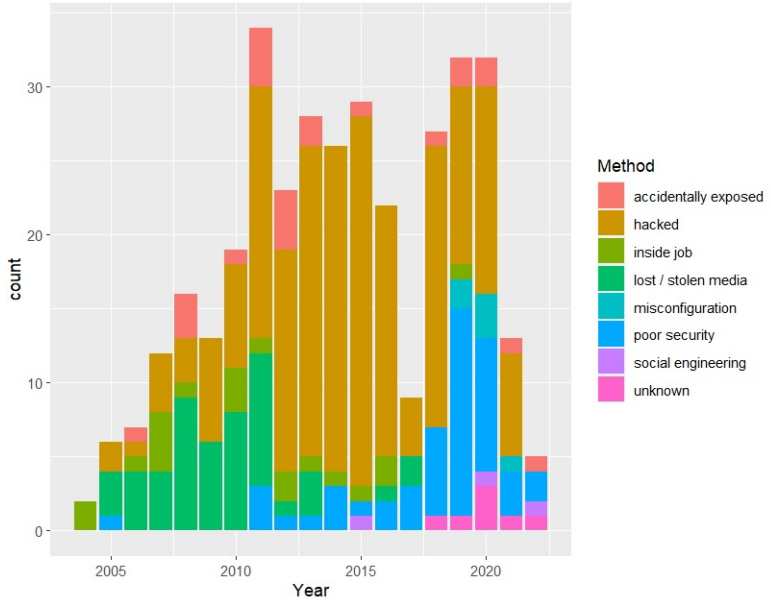

Figure 4: Crecimiento del número de ataques a través de los años

- ¿Hay alguna tendencia en la cantidad de datos que se han hackeado en brechas de datos a lo largo del tiempo?
- ¿Y todos estos datos hackeados, en cuantos ataques han sido robados?

```r
data_hacked <- dataBreaches %>%
  group_by(Records,Organization.type,Method) %>%
  filter(Method == 'hacked')

data_hacked$Sources <- NULL
data_hacked$X <- NULL
data_hacked$X.1 <- NULL

print(data_hacked)

ggplot(data_hacked, aes(x = Year, y=Records,fill=Method) ) +
  geom_bar(width = 0.9, stat="identity", position = position_dodge())
```

Figure 5: Codigo en R para realizar la visualizacion

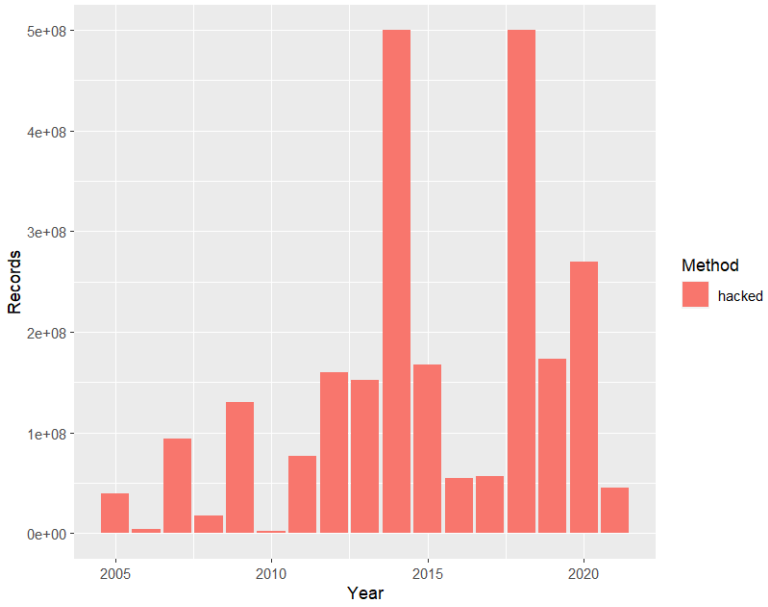

Figure 6: Tendencia de los hackeos a través de los años

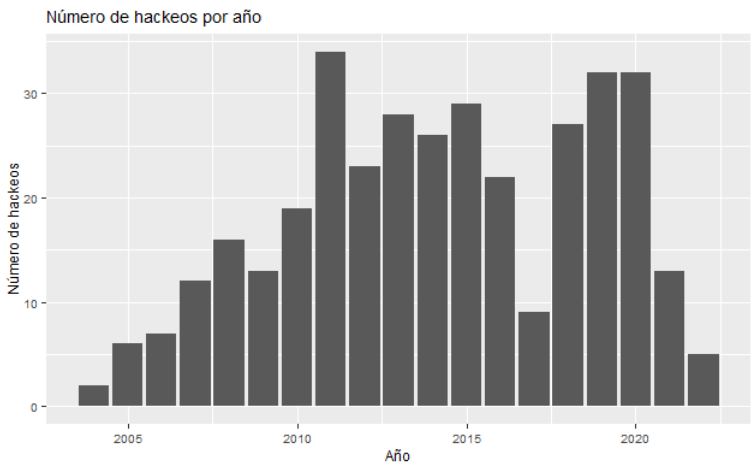

Figure 7: Numero de hackeos por año

3 Analisis Dataset: Supervised

3.1 Regression Predictions

Usamos los algoritmos de ML no supervisados para predecir valores con metodos de regressión. Hemos elegido la regressión lineal porque solo tenemos una variable (número de ataques por categorías difer-

entes), que cambia durante los años. En este subsección usamos regresión linean para responder las siguientes preguntas:

1. ¿Cuantos datos van a robar en 2023? en 2030? en 2040? (Numero de Records)
2. ¿Cuales serán las categorías de organisaciónes más atacadas en 2030?

Para empezar, creamos un plot de relacion entre variable "Year" y "Records". Sumimos los Records por año. Cómo se puede ver en el imagen 8 los ultimos años 2021 y 2022 no tiene información suficiente.

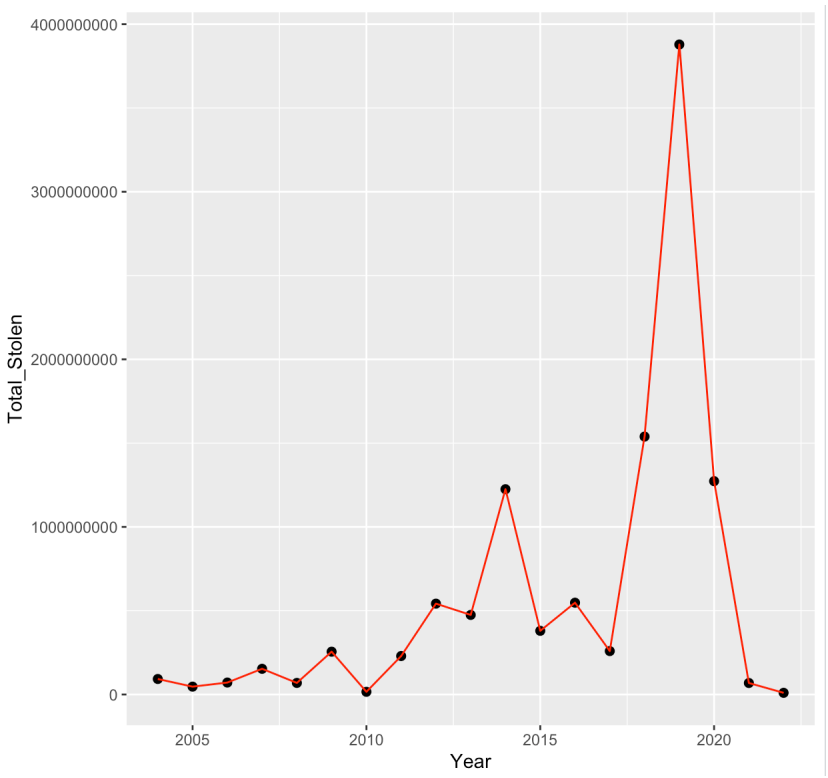

Figure 8: Relación entre año y número de ataques

Si comprobamos el numero de filas para los ultimos años, obtendremos la sugiente tabla:

	Number of rows with data breaches
2020	32
2021	13
2022	5

Figure 9: Número de filas para últimos tres años

El numero mediano de las filas per año durante 2004-2020 es 22 filas per año. Se nota que los años 2004, 2005 y 2006 también tienen un número de ataques muy pocos (2, 6 y 7 respectivamente). Pero en este caso vamos a dejarlas por dos razones. Por primero, en estos años, el campo de la ciberseguridad estaba en pleno desarrollo, por lo que muchos ataques no habrían podido ser reconocidos, registrados y analizados. Es lógico que en estos años el número de filas sea bajo. Además se nota un crecimineto pequeño en numero de ataques que refelja la realidad con crecimiento de ciberamenazas en todos los años de desarollo del internet.

3.1.1 Predecir número de datos robados

Entrenando un modelo de regresión lineal hemos obtenido la predicción de número de datos robados para los siguientes años:

	Predicted number of stolen data records
2023	2013187767
2030	2880539291
2040	4119612896

Figure 10: Predicción de Regresión Lineal para todos datos robads

Además podemos ver en el plot que la linea de las ataques crece muy rápido, aunqué hemos visto que el dataset no tiene los datos suficientes para el año 2020:

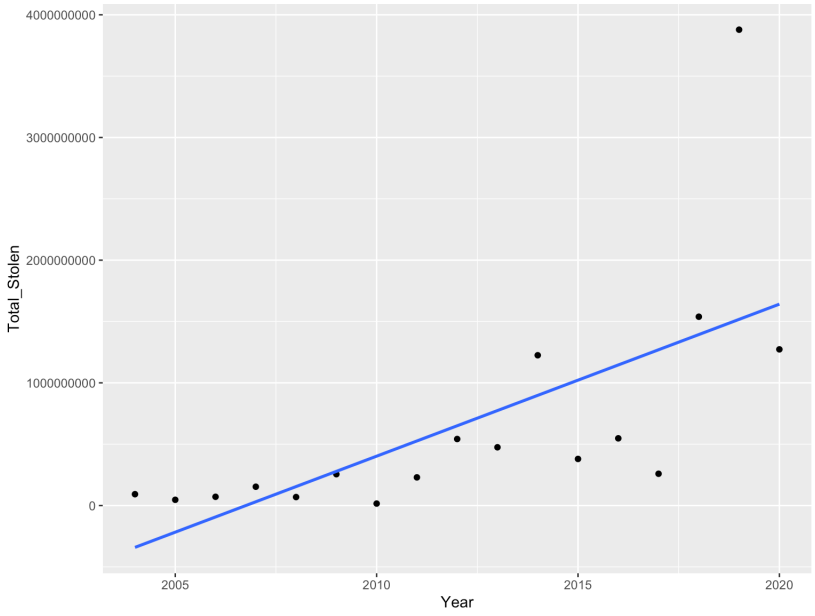

Figure 11: El Plot de la Regressión Lineal

Si visualizamos los datos de la imagen 10 obtenemos este gráfico: El crecimiento tan rápido aún con

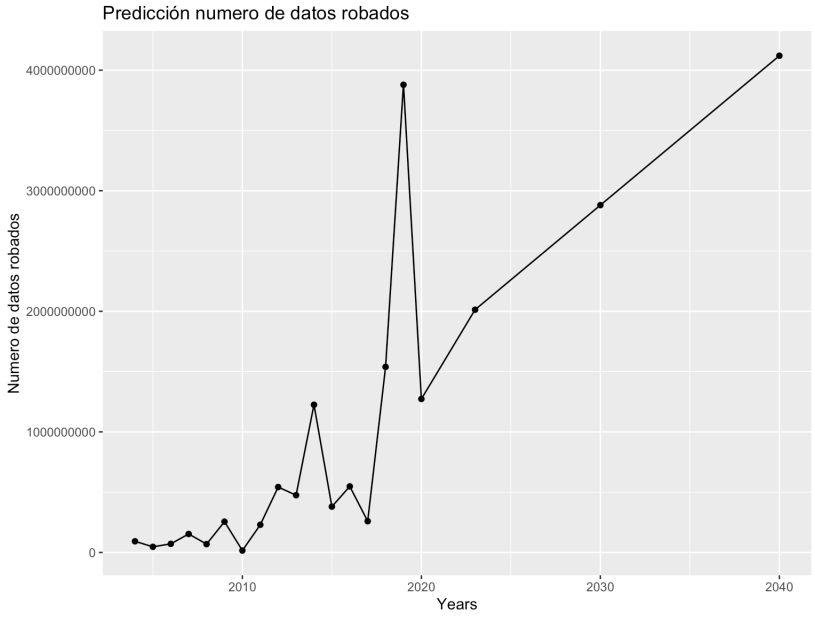

Figure 12: Predicción visualizada

datos ausentes de los años 2020, 2021 y 2022 nos demuestra que la regrsión linean ha reconocido cómo crecen los ataques con los datos que tuvimos.

3.1.2 Predecir las categorías más atacadas

Para cada categoria hay número de ataques durante el periodo. Podemos crear la regression linean para cada de categrías y predecir el crecimiento de ataques para año 2030. El dataset no refleja la realidad (mencionado en 2.1) Viendo las lineas podemos considerar que las categorías "financial",

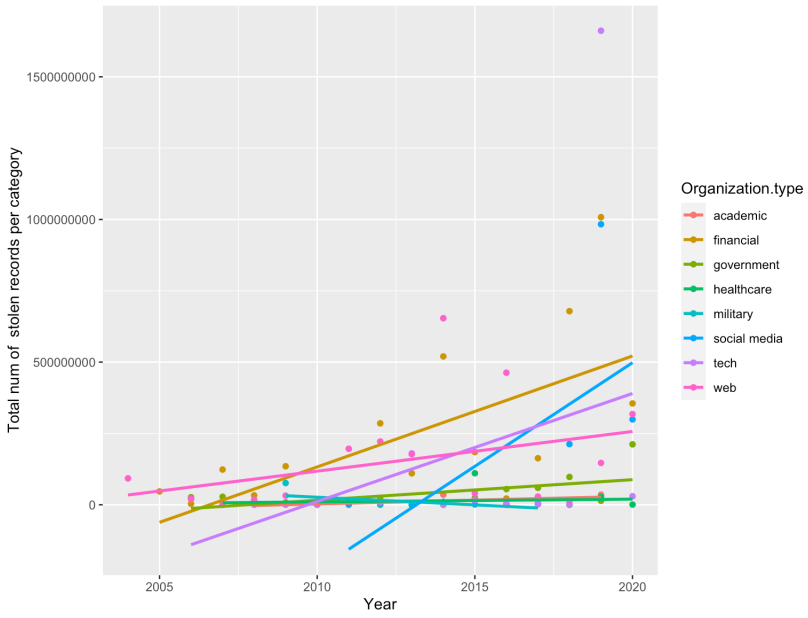

Figure 13: Regresión Lineal para cada categoría

"social media" y "tech" han perdido más datos que las otras en el año 2020. En comparasión con numero de ataques, las categorías más ataqadas en el año 2020 son "tech", "social media" y "web" (en orden ascendente). Para predicción omitimos las categorías "healthcare", "military" y "various" porque no tienen información suficiente.

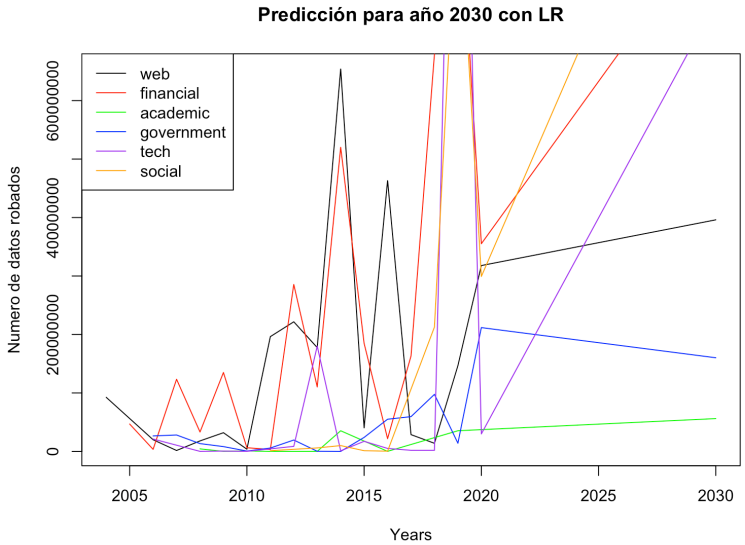

Figure 14: Predicción de número de datos robados en 2030 por categoría

Así podemos ver que las categorías "social media", "financial" y "tech" serán la más atacadas, pero la "social media" perderá mas datos que las otras. En 2020 la categoría "financial" está en primer lugar. Lo último que queremos comprobar es el numero de ataques. En nuestro dataset un ataque registrado es una fila del dataset. Con regresión lineal hemos visto que en el año 2030 las categorías más ataqadas serán "social media", "tech" y "web" (en orden ascendente).

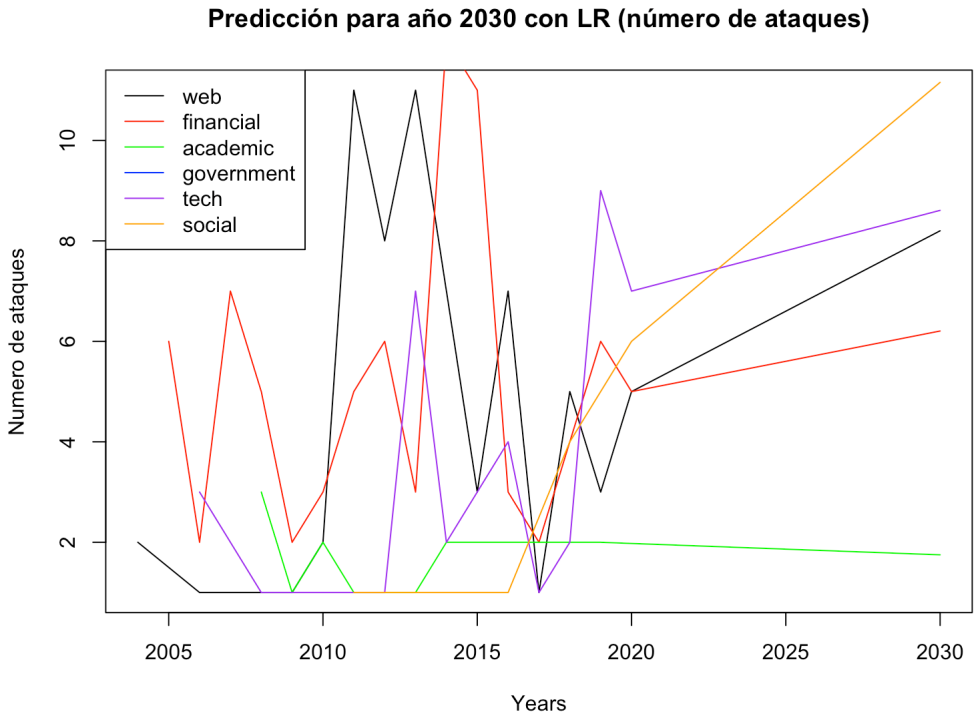

Figure 15: Predicción número de ataques por categoría

Aquí se nota la diferencia entre los datos robados y números de ataques. Por ejemplo la categoría "financial" perderá más datos, aunque será menos ataqada que las otras.

4 Analisis Unsupervised

4.1 Clustering

El Clustering es una tarea que se ejecuta con la finalidad de agrupar conjuntos de objetos no etiquetados para lograr construir subconjuntos de datos los cuales son los "Clusters".La dificultad de estos procesos es que dependiendo de las reglas y diseños que usemos para diseñar el cluster, este sera eficiente o no para el objetivo que queremos lograr.Podemos ver diferentes tipos de modelos de clustering:

- Algoritmo K-means.
  Con este metodo obtenemos k grupos de n observaciones, dependiendo del numero de grupos que usemos obtendremos un resultado más o menos eficiente.
- DBSCAN.
  Modelo de densidad discreto
- Mean Shift.
  Modelo de gradientes en densidad
- AGNES.
  Modelo jerárquico

4.2 KMeans

Antes de realizar el análisis podemos obtener una estimación de cuantos clusteres podemos obtener del dataset mediante el Indice de Hubert. Mediante la función NbClust en R obtenemos para este algoritmo k-means una estimación:

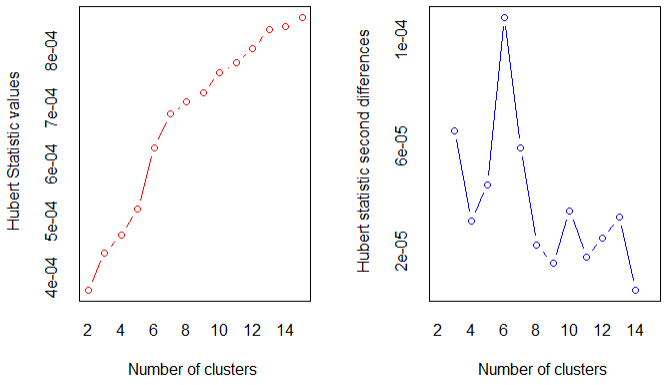

Figure 16: Plot del NbClust. Estimación de Clusteres

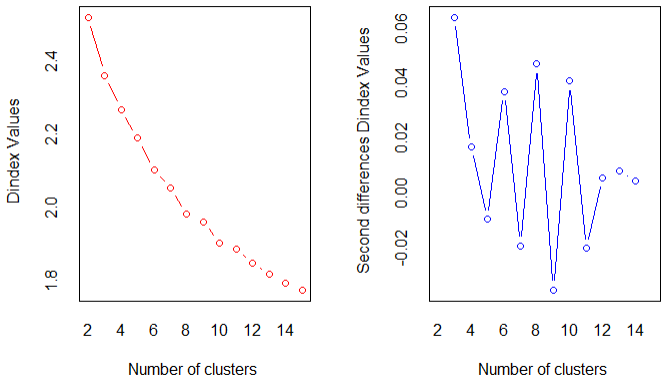

Figure 17: Plot del NbClust. Estimación de Clusteres

En la imagen 18 se puede ver que con 4 clusters el algoritmo reoconció bien las categorías más atacadas:

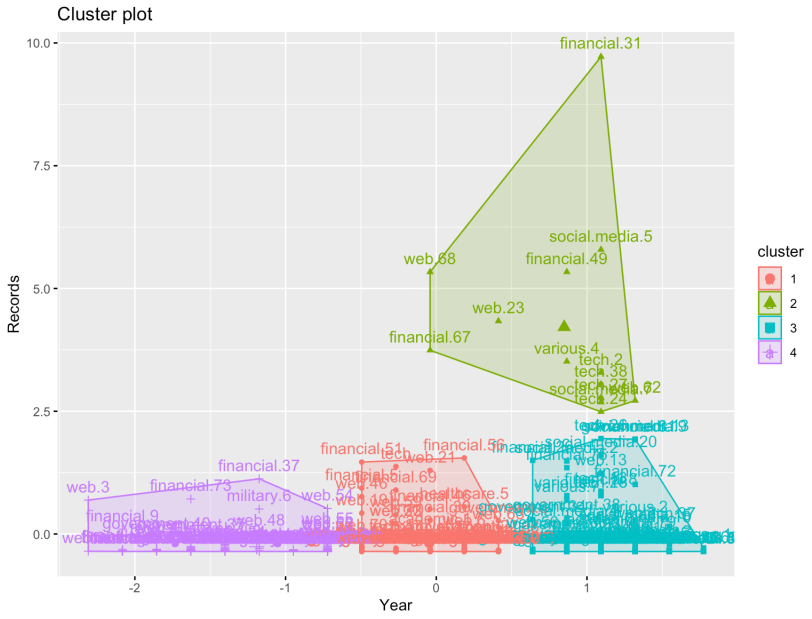

Figure 18: Clustering con k = 4

5 Anomaly Detection

[!ht] Los modelos de detección de anomalías se utilizan para identificar valores atípicos, o casos extraños, en los datos. A diferencia de otros métodos de modelado que almacenan reglas acerca de casos extraños, los modelos de detección de anomalías almacenan información sobre el patrón de comportamiento normal. Esto permite identificar valores atípicos, incluso si no se ajustan a ningún patrón conocido. Estos modelos tienen diferentes aplicaciones:

- Detección de fraudes
- Alertar a los técnicos de algún servicio en concreto sobre alguna anomalía.
- Filtrar anomalias para usar ciertos data sets en supervised learning.
- Evaluar la competencia de ciertos modelos.

La detección de anomalías puede examinar un gran número de campos para identificar clústeres o grupos de homólogos en los que hay registros similares, mientras que los métodos tradicionales de identificación de valores atípicos observan una o dos variables a la vez. Así, se puede comparar cada registro con el resto del grupo de homólogos para identificar posibles anomalías. Cuanto más alejado esté un caso del centro normal, mayor será la probabilidad de que sea extraño. Por ejemplo, el algoritmo podría agrupar registros en tres clústeres distintos y marcar aquellos que se sitúen lejos del centro de cualquier clúster.

References

IBM. (2021). Nodo Detección de anomalías. Recuperado de https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-anomaly-detection-node