

Gestión de alertas en seguridad en base a RRSS

08 de agosto de 2022



Universidad
Internacional
de Valencia

Titulación:

Máster Universitario en Big
Data y Ciencia de Datos

Curso académico

2022-2023

Alumno/a: Charlo Reyes,
Emilio

D.N.I: 77.468699-F

Director/a de TFM: Concepción
Cordón Fuentes

Convocatoria:

Primera

De:

 Planeta Formación y Universidades

Índice

Resumen	6
Abstract	7
Palabras Clave	8
1. Introducción	9
2. Contexto y motivación.....	9
3. Objetivos.....	10
2.1. Valoración del riesgo a la seguridad desde las redes sociales.	10
2.2. Valoración del riesgo a la seguridad desde Dataset especializados.....	10
2.3. Entrenamientos de modelos para predecir análisis futuros de riesgo a la seguridad.....	11
4. Estado del Arte y Marco teórico	11
4.1. Herramientas utilizadas.....	11
4.1.1. Anaconda y Jupyter.....	11
4.1.2. MongoDB	13
4.1.3. R y RStudio.	15
4.1.4. Octoparse.....	17
4.1.5. Tableau	18
4.1.6. PowerBI.....	19
4.2. Proceso KDD.	21
4.2.1. Definición.....	21
4.2.2. Etapas del proceso KDD.....	22
4.3. Machine Learning.....	24
1. Desarrollo del proyecto	26
1.1. Metodología.....	26
1.2. Desarrollo del proyecto.....	27
1.2.1. Extracción, procesamiento y visualización de datos en MongoDB.	27
1.2.2. Proceso KDD con Base de datos especializada en Terrorismo utilizando Anaconda Jupyter con Python3.....	37
1.2.3. Proceso KDD con Base de datos especializada en Terrorismo utilizando RStudio con R.....	47
1.2.4. Machine Learning.....	52
1.2.5. Visualización en PowerBI de resultados.....	60

1.3. Resultados	60
2. Conclusión y trabajos futuros.....	67
3. Referencias	68
Apéndice I. Análisis descriptivo	69
Anexo I. Código selector de propiedades para KDD	73
Anexo II. Tweets a MongoDB	73
Anexo III. Búsqueda de Tweets	73
Anexos IV. Proceso KDD Dataset Terrorismo.....	73

Índice de ilustraciones

Ilustración 1. Panel principal de Anaconda.	12
Ilustración 2. Cuaderno en Jupyter.	13
Ilustración 3. Comparativa MongoDB con SQL.....	14
Ilustración 4. Formato de documento en JSON.....	14
Ilustración 5. Arquitectura de Cluster in Mongodb.	15
Ilustración 6. Panel de control de RStudio.	16
Ilustración 7. Panel de control de Octoparse con una Web scrapped ilustrativa.....	18
Ilustración 8. Panel de Tableau, ejemplo	19
Ilustración 9. Panel de Control tipo en PowerBI	20
Ilustración 10. Proceso KDD en Big Data.....	24
Ilustración 11. Proceso de aprendiza de un algoritmo en Machine Learning.....	26
Ilustración 12. Creación en cuenta Twitter developer.....	27
Ilustración 13. librerías necesarias para importar Tweets	28
Ilustración 14. Claves obtenidas en Twitter developer	28
Ilustración 15. Función que recupera 200 Tweets por llamada	28
Ilustración 16. Conexión a MongoDB.....	29
Ilustración 17. Resultado de cuentas en la colección de MongoDB	29
Ilustración 18. Número total de Tweets extraídos.	30
Ilustración 19. librerías necesarias para extraer tweets.....	31
Ilustración 20. Creación conexión MongoDB	31
Ilustración 21. búsqueda de palabras clave en Twitter.....	32
Ilustración 22. Extracción de tweets.....	32
Ilustración 23. Número de Tweets y número de usuarios.....	32
Ilustración 24. Diez primeros tweets con palabras clave de búsqueda.....	33
Ilustración 25. Número total de Tweets en Tweepsters.....	33
Ilustración 26. Hashtags más usados.	34

Ilustración 27. Hashtags relacionados con terrorismo en Europa cogiendo de base los 10 primeros anteriores.	34
Ilustración 28. número de tweets por cuenta objetivo.....	35
Ilustración 29. hashtags más seguidos para las cuentas objetivo.	36
Ilustración 30. Hashtags utilizados relacionados con zonas de conflicto	36
Ilustración 31. Ejemplo de extracción de Dataset para este estudio.....	37
Ilustración 32. Carga de librerías	40
Ilustración 33. Carga y Dataframe del Dataset.....	41
Ilustración 34. Tipo de datos del Dataset y número de registros y columnas	41
Ilustración 35. Registros numéricos y no numéricos.	42
Ilustración 36. Registros duplicados o vacíos	42
Ilustración 37. Datos repartidos por países.	43
Ilustración 38. Correlación entre las variables Nkill y Nwound.	43
Ilustración 39. Creamos índice por Year	44
Ilustración 40. Porcentaje de valores nulos o vacíos.....	44
Ilustración 41. Unión de los 3 Datasets.....	45
Ilustración 42. Valores vacíos	45
Ilustración 43. Porcentaje de valores nulos por variables.....	46
Ilustración 44. Correlación entre las variables al 50%.....	46
Ilustración 45. Ninguna variable con correlación alta.	49
Ilustración 46. Variables numéricas.	49
Ilustración 47. Covarianza de las variables numéricas.....	49
Ilustración 48. correlación entre el número de fallecidos y heridos en atentados con éxito y el año de su realización	50
Ilustración 49. Valores nulos.	50
Ilustración 50. número fallecidos basado en el éxito de ataque	51
Ilustración 51. Coeficientes de interceptación.	51
Ilustración 52. Número de heridos basado en el éxito de ataques.	52
Ilustración 53. Coeficientes de interceptación.	52
Ilustración 54. Histograma del Dataset	54
Ilustración 55. Diagrama de densidad.....	54
Ilustración 56. Gráfico de dispersión	55
Ilustración 57. Diagrama de correlación.....	56
Ilustración 58. Modelo predictivo para una sola variable	57
Ilustración 59. Visualización para un modelo predictivo en base a dos variables.	58
Ilustración 60. Modelo con las tres variables a analizar	59
Ilustración 61. División de datos	59
Ilustración 62. Visualización del Dataset especializado en terrorismo.....	60
Ilustración 63. Número total de Tweets.....	61
Ilustración 64. Total de Tweets de cada cuenta	61
Ilustración 65. Ranking de los cinco primeros idiomas en los que se ha escrito más tweets	61
Ilustración 66. Contar Tweets por tipo de media que lleva incrustado.....	62
Ilustración 67. Ordenar las cuentas de mayor influencia a menor (de acuerdo con el número de seguidores que posee cada una	62

Ilustración 68. Realizar una consulta en Mongo DB para listar los 20 hashtags más utilizados.....	63
Ilustración 69. Técnicas de ataque terroristas exitosas	64
Ilustración 70. Fallecidos y heridos en atentados	65
Ilustración 71. Atentados por Regiones	66

Índice de tablas

Tabla 1. Tipos de datos de las variables.....	39
Tabla 2. Resumen estadístico variable cuantitativas.....	47
Tabla 3. Resumen estadístico variables categóricas.....	48
Tabla 4. Matriz de correlaciones de las variables numéricas.	48

Resumen

Para el siguiente Trabajo Fin de Máster se ha realizado un estudio de las redes sociales, en base a Twitter, al objeto de vislumbrar un análisis de riesgos en seguridad con la información analizada de ellas. Para ello se ha realizado la extracción de datos siguiendo los procedimientos adquiridos en la asignatura de Fundamentos de Big Data.

En primer lugar, he procedido a crear una cuenta en Twitter developer. Con ella he procedido a extraer Tweets de las cuentas objetivo como fuente de datos utilizando el lenguaje de programación Python 3, escrito en la plataforma Anaconda Jupyter, y los he volcado en Mongo DB Atlas. Una vez allí, se procede a la transformación de estos comprendiendo la limpieza la integración e integración de datos (ETL) para dar con el Dataset definitivo, al objeto de analizar y consultar los mismos con Mongo DB Compass.

Posteriormente, se utiliza MongoDB Charts para realizar las visualizaciones pertinentes. A continuación de lo anterior, he procedido a realizar el proceso KDD (knowledge Discovery in databases) aprendido en la asignatura de minería de datos.

Del mismo modo, he realizado un Web Scraping haciendo uso del programa Octoparse para extraer información de Webs objetivos al objeto de hacerme con un Dataset complementario a los anteriores y darle consistencia.

Tras realizar el proceso KDD con Anaconda Jupyter de los mismos he utilizado los conocimientos obtenidos en la asignatura de Estadística avanzada para realizar un análisis en R del Dataset y obtener conclusiones de estos.

Como continuación, a fin de tener un panel de control donde visualizar los datos procesados, procedí a utilizar la herramienta de visualización Tableau y PowerBI trabajada en la asignatura de visualización de datos.

Finalmente, haciendo uso de los conocimientos adquiridos en la asignatura de ciencia de datos para la toma de decisiones estratégicas he procedido a crear un panel de control a fin de ser capaz de valorar los datos en relación con los riesgos para la seguridad y sacar conclusiones de estos.

RStudio es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

RStudio tiene la misión de proporcionar el entorno informático estadístico R. Permite un análisis y desarrollo para que cualquiera pueda analizar los datos con R.

Anaconda Jupyter es una plataforma gratuita y de código abierto escrita en Python, para ciencia de datos y procesamiento de datos masivos que soporta varios lenguajes

de programación entre ellos Python y R. Es ideal para la gestión y administración de paquetes de datos.

MongoDB es un sistema de gestión de base de datos NoSQL (no estructurados) orientada a documentos el cual maneja colecciones similares a tablas siendo ideal para fuentes de datos procedentes de Tweets. MongoDB almacena los datos en BSON (Notación estándar para el intercambio de datos Binarios) que facilita su tratamiento en JSON (Java Script Object Notation) mediante expresiones regulares.

Octoparse es una herramienta de Scraping visual que se utiliza para extraer datos de casi cualquier sitio Web. Proporciona los datos de salida en varios formatos, incluidos Excel, HTML, Txt y CSV.

Tableau es un software de análisis de datos con una capa de visualización y presentación, utilizada para el Business Intelligence y análisis de datos la cual simplifica los datos para presentarlos en un formato comprensible e intuitivo.

PowerBI es un software de análisis de negocio basado en la nube y visualización de datos. Esta herramienta utilizada para el Business Intelligence y análisis de datos simplifica los datos para presentarlos en un formato comprensible e intuitivo.

Abstract

For the next master's Thesis, a study of social networks has been carried out, based on Twitter, in order to glimpse an analysis of security risks with the information analyzed from them. To this end, data extraction has been carried out following the procedures acquired in the subject of Fundamentals of Big Data.

First of all, I proceeded to create an account in Twitter developer. With it I have proceeded to extract Tweets from the target accounts as a data source using the Python 3 programming language, written on the Anaconda Jupyter platform, and I've dumped them into Mongo DB Atlas. Once there, we proceed to the transformation of these comprising the cleaning, integration, and integration of data (ETL) to find the final Dataset, in order to analyze and consult them with Mongo DB Compass.

Subsequently, MongoDB Charts is used to make the relevant visualizations. Following the above, I have proceeded to perform the KDD process (knowledge Discovery in databases) learned in the subject of data mining.

Similarly, I have done a Web Scraping making use of the program Octoparse to extract information from objective Websites to get a complementary Dataset to the previous ones and give it consistency.

After carrying out the KDD process with Anaconda Jupyter of the same I have used the knowledge obtained in the subject of Advanced Statistics to perform an analysis in R of the Dataset and obtain conclusions from these.

Then, to have a control panel where I can visualize the processed data, I proceeded to use the Tableau visualization tool and PowerBI worked on in the subject of data visualization.

Subsequently, making use of the knowledge acquired in the subject of data science for strategic decision making, I proceeded to create a control panel to be able to assess the data in relation to the risks to security.

Finally, making use of the knowledge acquired in the subject of data science for strategic decision making, I have proceeded to create a control panel to be able to assess the data in relation to security risks and draw conclusions from them.

RStudio is an integrated development environment (IDE) for the R programming language, dedicated to statistical computing and graphics. It includes a console, syntax editor that supports code execution, as well as tools for plotting, debugging, and workspace management.

RStudio has the mission of providing the statistical computing environment R. It allows an analysis and development so that anyone can analyze the data with R.

Anaconda is a free and open-source platform written in Python, for data science and massive data processing that supports several programming languages including Python and R. It is ideal for managing and administering data packages.

MongoDB is a document-oriented NoSQL (unstructured) database management system which handles table-like collections and is ideal for data sources from Tweets. MongoDB stores the data in BSON (Standard Notation for the exchange of Binary Data) that facilitates its treatment in JSON (Java Script Object Notation) through regular expressions.

Octoparse is a visual Web scraping tool which can be used to extract data from almost any Websites. it provides the output data in various formats including Excel, HTML, Txt and CSV.

Tableau is a data analysis software with a visualization and presentation layer, used for Business Intelligence and data analysis which simplifies the data to present it in an understandable and intuitive format.

Power BI is a cloud-based business analysis and data visualization software. This tool used for Business Intelligence and data analysis simplifies the data to present it in an understandable and intuitive format.

Palabras Clave

Dataset, MongoDB, Twitter, Jupyter, R, Python 3, Terrorismo.

1. Introducción

Dado un conjunto Dataset, obtenidos mediante técnicas de Web Scraping, usando el software Octoparse y Python 3 con Jupyter se considerarán diversos factores de forma que se pueda evaluar si un territorio determinado se considera como seguro o por el contrario el riesgo de atentado terrorista es elevado. En el caso estudio para el presente trabajo, me he centrado en Europa.

Dentro de los factores externos se han analizado los históricos de incidentes, índice de criminalidad, tipo de organización terrorista, población, índices de pobreza y desempleo, etc que puedan interceder en los resultados y puedan ayudar a una mejor valoración de la amenaza terrorista.

Además, para realizar el análisis mencionado, se han utilizado diferentes técnicas y herramientas estudiadas en el Máster de Big Data y Data Science de la VIU. Algunas de las herramientas utilizadas han sido: Python, RStudio, Octoparse, Tableau, PowerBI y MongoDB.

Como resultado de los diferentes estudios los cuales se han apoyado en técnicas de visualización, estudiadas en el máster, se ha conseguido vislumbrar a través de un panel de control la situación en un determinado lugar en relación con la seguridad.

2. Contexto y motivación

En un mundo globalizado en el que el acceso a la información se ha maximizado, ha permitido avances en todos los campos, permitiendo el desarrollo de pueblos en los que otrora no tenían las mismas oportunidades por falta de conocimiento o acceso a la información. Pero del mismo modo que ha contribuido al avance y al desarrollo, lo que llamamos la democratización del conocimiento en beneficio de las sociedades y de los pueblos, las organizaciones criminales y terroristas se han hecho eco y han evolucionado para hacer uso de ellas, tanto en el plano de propaganda y reclutamiento a través de los medios de información globalizados como a la obtención de conocimientos para conseguir sus objetivos por medio de la violencia entre las diferentes motivaciones. Es por ello que cada vez más se refleja que procedimientos de atentados exitosos en cualquier lugar del mundo se reflejan al poco tiempo en otros y eso es sin duda alguna gracias al acceso que dan las tecnologías de la información.

La motivación que me lleva a hacer este estudio es precisamente en adquirir conocimientos y realizar predicciones basados en métodos estadísticos sobre la base del lenguaje de programación Python y R con la guía de la asignatura tanto de minería de datos como de estadística avanzada, que me proporcionen las herramientas necesarias para ser capaz de asesorar a las personas con capacidad de decisión, en el campo profesional al que me dedico, seguridad, siendo capaz de mediante el uso de tecnologías de Big Data y con la ayuda de la estadística, de realizar análisis que

ayuden a minimizar el impacto de estas capacidades adquiridas por las organizaciones antes citadas gracias a la globalización y a las tecnologías de la información.

3. Objetivos

Los principales objetivos marcados en el presente Trabajo de Fin de Máster de la Universidad Internacional de Valencia son los siguientes:

2.1. Valoración del riesgo a la seguridad desde las redes sociales.

En la actualidad existen multitud de Redes sociales de publicación e intercambio de información de diverso tipo. Para el objeto del presente estudio de Redes sociales he enfocado el trabajo en una primera parte en Twitter, la cual es un servicio de comunicación bidireccional en el que se comparte información en forma de micro blogs, mensajes cortos, o en otras palabras en forma de tweets. Es un formato idóneo de hasta 140 caracteres en el que se pueden compartir entre otros videos, imágenes y emoticonos.

Por lo que, en base a Tweets, se pretende realizar un análisis de riesgos derivados de los eventos compartidos por los usuarios.

2.2. Valoración del riesgo a la seguridad desde Dataset especializados.

Para el estudio de las áreas históricas de más riesgo para la seguridad, he procedido a descargar diversos Datasets de páginas como la página Web “Global Terrorism Database” (Maryland, 2022), la página Web de Rand Corporation (Corporation, 2022). A su vez, he procedido mediante Web Scraping (Mitchel., 2021) a extraer de diversas páginas de noticias, en particular BBCnews, información relacionada con la seguridad al objeto de completar el estudio para Europa. Del mismo modo, he procedido a descargar Datasets de la página de la OCDE (Organization for economic, cooperation and development, 2022) relacionados con potenciadores de riesgo para atentados terroristas como pueden ser la Tasa de desempleo, el número de habitantes, el porcentaje de iliteratos o la renta per cápita.

Con ello se pretende conseguir de forma precisa regiones en Europa que son susceptibles a tener riesgos para la seguridad, a veces provocado como consecuencia de variables aparentemente no deducibles.

2.3. Entrenamientos de modelos para predecir análisis futuros de riesgo a la seguridad.

En conjunción con los Dataset obtenidos en los pasos anteriores me propongo mediante los procedimientos de Machine Learning, utilizando Python y diversos algoritmos en base a un cuaderno en Jupyter, entrenar un modelo que me sirva para predecir análisis futuros de riesgo para la seguridad en Europa.

4. Estado del Arte y Marco teórico

4.1. Herramientas utilizadas.

El mundo de Ciencia de Datos es muy amplio y podemos encontrar un gran abanico de herramientas de las que podemos hacer uso de ellas. En concreto, en este proyecto nos hemos centrado en el uso de las herramientas de Anaconda para la programación en Python en sus Notebooks, de Octoparse para la extracción de los datos mediante técnicas de *Web Scraping* de las páginas Web objeto de estudio, de Jupyter anaconda con Python3 y de Rstudio (Grolemund., 2020) para R al objeto de analizar los Dataset y de PowerBI y Tableau para apoyar con visualizaciones extra nuestro estudio.

A continuación, ofrecemos un poco más de detalle acerca de estas herramientas, su uso, su creación y su estado actual, ya que han sido clave para la realización de este proyecto.

4.1.1. Anaconda y Jupyter.

Anaconda es una de las herramientas que se considera clave para la Ciencia de datos en el mundo actual. Nos ofrece posibilidades que permiten desarrollar aplicaciones y análisis de una forma más efectiva, rápida y fácil.

Es un conjunto de aplicaciones, librerías y conceptos en código abierto, diseñado para la Ciencia de Datos. Funciona como un gestor de entorno, gestor de paquetes y dispone de más de 20 paquetes de código abierto.

Para el presente TFM hemos usado la versión *Individual Edition* de Anaconda, la cuál es la más popular del mundo con más de 25 millones de usuarios.

Posee más de 7500 paquetes de ciencia de datos y aprendizaje automático de código abierto y, además, permite la administración de entornos como solución para crear, distribuir, instalar, actualizar y administrar *software*.

La interfaz es muy fácil de aprender al estar muy claramente definida y permite acceso a materiales de formación, documentación y otros recursos de Anaconda.

Uno de los factores clave en el uso de *Python* como lenguaje de código abierto, es la gran comunidad de código abierto que tiene en internet, donde se pueden encontrar muchas dudas resueltas y colaboración entre los diferentes consumidores.

Cabe mencionar que Anaconda también está disponible en otras versiones para empresa, que tienen un coste asociado.

Podemos ver el logo de Anaconda en la ilustración 1.

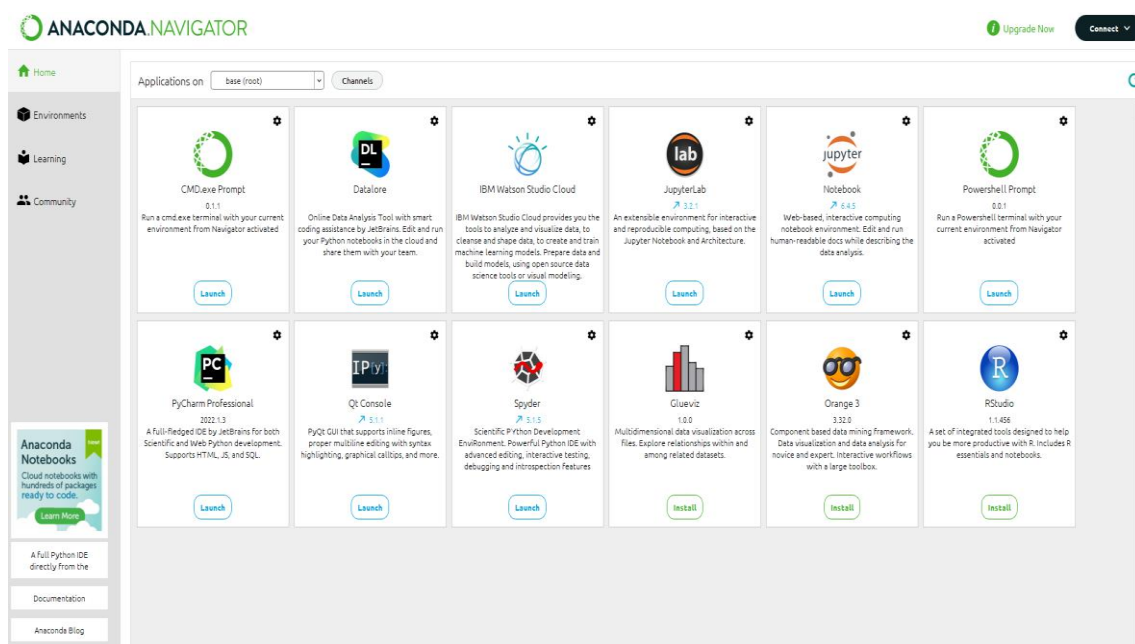


Ilustración 1. Panel principal de Anaconda.

En cuanto a Jupyter, un cuaderno incluido en Anaconda, decir que consiste en un IDE (Entorno de desarrollo integrado) de código abierto, el cual permite crear y compartir documentos que contienen código, cálculo interactivo basado en web, visualizaciones y texto. El cuaderno de Jupyter soporta diversos lenguajes de programación usados en Big Data que incluyen el ya mencionado Python, pero también otros como Julia, Scala, R y otros.

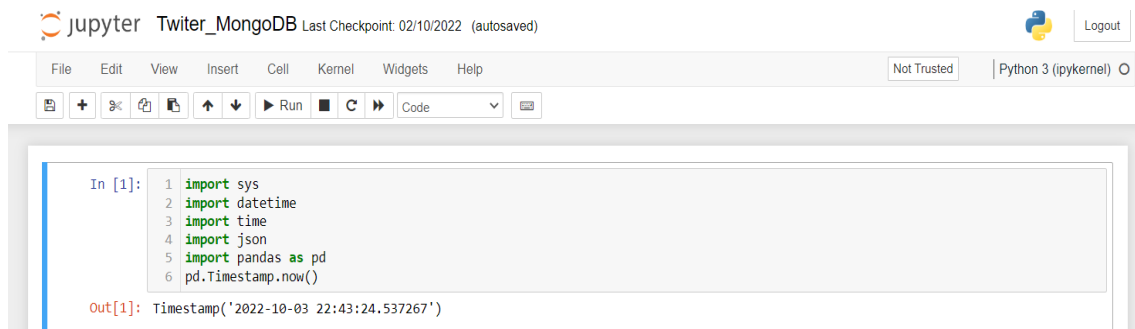


Ilustración 2. Cuaderno en Jupyter.

4.1.2. MongoDB

MongoDB (Shannon Bradshaw, 2019) es una base de datos NoSQL orientada a documentos. Aparte de crear, leer, actualizar, y borrar registros, es capaz de funcionar como un sistema de gestión de base de datos. Entre las capacidades específicas con las que cuenta tenemos las siguientes:

- ❖ Indexar. MongoDB soporta índices secundarios y es capaz además de proporcionar índices en estructuras jerarquizadas como documentos concatenados y arrays (estructura de datos en la que todos sus miembros son del mismo tipo).
- ❖ Agregaciones. MongoDB proporciona un marco basado en el concepto de canalizaciones de procesamiento de datos. Esto le permite crear procesos analíticos complejos mediante el procesamiento de datos divididos en etapas simples en el servidor, optimizando las bases de datos.
- ❖ Índices especiales. MongoDB soporta tipos especiales de índices como las colecciones que caducan en el tiempo (TTL), como colecciones activas según la actividad de la sesión o aquellos índices limitados a documentos que cumplen un determinado criterio de filtro.
- ❖ Almacenamiento de archivos. MongoDB tiene un protocolo fácil de usar para almacenar archivos muy pesados y metadatos.

Es ideal para el propósito de nuestro proyecto pues es en base a Tweets. Podemos ver la equivalencia entre una base de datos tradicional SQL, con una NoSQL, en este caso, con MongoDB.

MongoDB	SQL
Base de Datos	Base de Datos
Colecciones	Tablas
Documentos	Filas (registros)
Campos	Columnas

Ilustración 3. Comparativa MongoDB con SQL

Los tweets al almacenarlos en MongoDB lo harán en el formato JSON o BSON, en forma de documentos del tipo clave, valor, como podemos ver a continuación en la ilustración.

JSON (BSON)

```

{
  "nombre": "Juan",
  "edad": 25
  "dirección":
    {
      "ciudad": "Barcelona"
    },
  "aficiones":[
    {"nombre": "Fútbol" },
    {"nombre": "Esquí" }
  ]
}
```

Ilustración 4. Formato de documento en JSON

Una particularidad fundamental es que no existen esquemas predefinidos. El conjunto clave valor de los documentos no está fijo en cuanto a tamaño o tipo, lo cual facilita añadir o eliminar campos de este. Ello permite de una manera ágil crear modelos para los datos y elegir el que más nos interese para el proyecto en cuestión. El hecho de no tener filas o registros sino documentos permite representar relaciones jerárquicas complejas en un simple registro.

MongoDB Atlas proporciona una manera fácil de albergar y manejar Datasets en la nube lo cual permite tratar Datasets cuyo volumen no sería posible en una sola

plataforma en local. Para ello, este modelo de tratamiento de datos orientado a documentos trocea los datos entre varios servidores. MongoDB de manera automática balancea el reparto de los documentos, redistribuyendo los mismos y permitiendo en direccionamiento a los servidores apropiados para acceder a leer o escribir los documentos seleccionados como se puede ver en la ilustración 4.

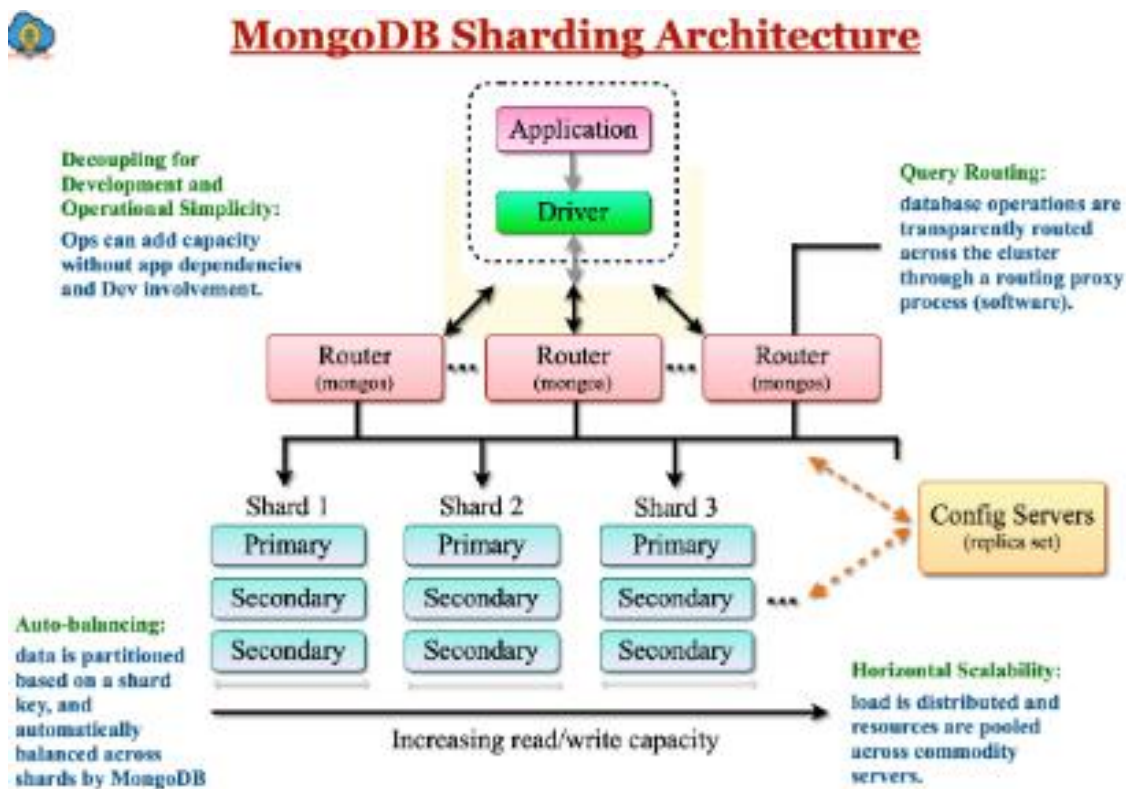


Ilustración 5. Arquitectura de Cluster in MongoDB.

MongoDB Compass nos permite analizar y entender el Dataset de una manera visual, importando y exportando bases de datos en formatos JSON y CSV. Permite explorar, analizar e interactuar con el contenido almacenado en un Dataset en MongoDB sin utilizar consultas. Permite, de manera análoga a MongoDB Shell, las siguientes utilidades:

- ❖ Visualizar y explorar los datos almacenados en su base de datos.
- ❖ Crear bases de datos e insertar, actualizar y eliminar datos de esta.
- ❖ Obtención de estadísticas del servidor en tiempo real.
- ❖ Entender los problemas de rendimiento de manera visual.
- ❖ Administración de índices.
- ❖ Validación de los datos con reglas de validación de esquema JSON

4.1.3. R y RStudio.

El lenguaje de programación R se basa en el uso de funciones matemáticas orientado a objetos y es multiplataforma ya que funciona en diversos sistemas.

Fue desarrollado en 1993 por miembros del Departamento de Estadísticas de la Universidad de Auckland y se inspira en su precursor S así como en Scheme, tal y como lo afirman sus creadores, Robert Gentleman y Ross Ihaka.

El lenguaje de programación R es de software libre y brinda un rango importante de herramientas estadísticas y gráficas. Se puede utilizar para cualquier tipo de cálculo numérico o análisis de información aplicada a una variedad de campos, tanto juegos sencillos como la gestión de los datos de los empleados de una multinacional.

Sigue experimentando un notable crecimiento en los últimos años, sobre todo por la exponencial expansión del Big Data y la minería de datos.

El lenguaje de programación R, es considerado por muchos expertos como el más potente y eficiente en el ámbito estadístico y de minería de datos.

El lenguaje R tiene varios sistemas para hacer gráficos y visualizaciones, pero ggplot2 es uno de los más versátiles. Con ggplot2 se implementa la gramática de gráficos, un sistema coherente para describir y construir gráficos.

En relación con RStudio, es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

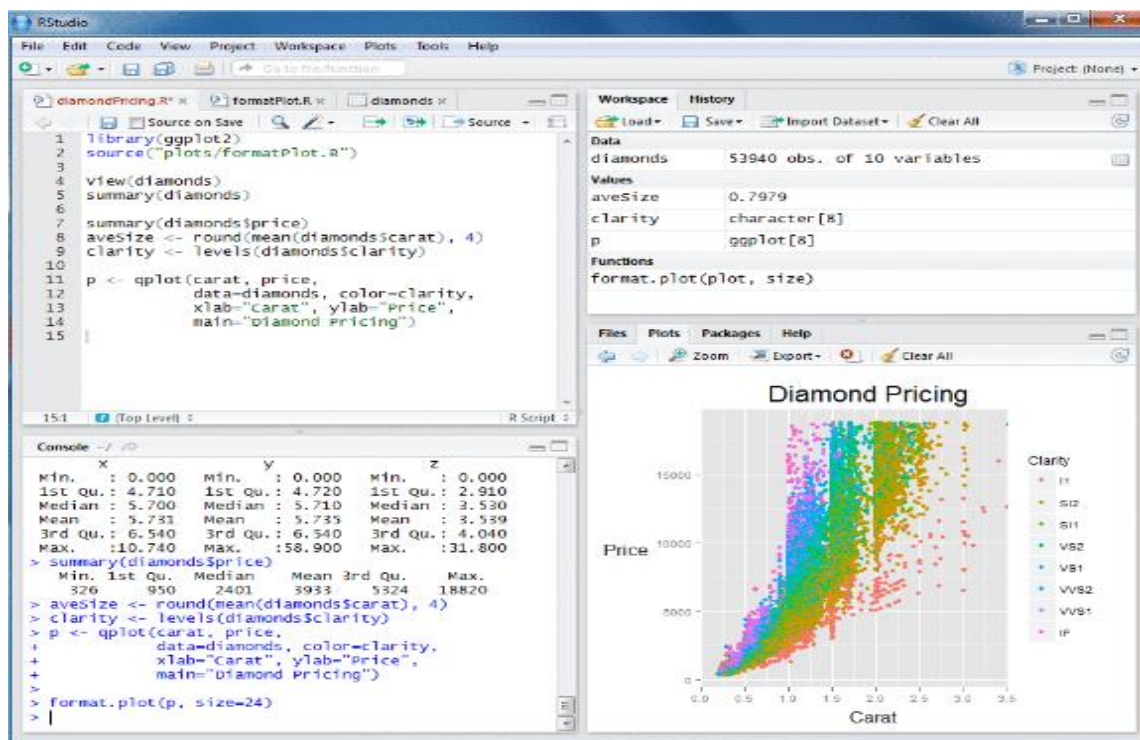


Ilustración 6. Panel de control de RStudio.

4.1.4. Octoparse

Octoparse es un software de extracción de datos Web visual. Está preparado para que tanto usuarios con experiencia como sin experiencia puedan usarlo para extraer información de sitios Web en grandes volúmenes (Web Scraping).

Tiene incorporada una serie de plantillas útiles que facilita a los usuarios la extracción de información de la Web. Es capaz de extraer todo tipo de datos, incluidos los datos de productos de los principales sitios de comercio electrónico como Amazon, eBay y más.

A su vez, la herramienta puede extraer información de redes sociales, como Facebook, Twitter, Instagram, YouTube, etc., para recuperar los mensajes, comentarios, imágenes y otros.

Entre las ventajas e inconvenientes de la aplicación tenemos como ventajas las siguientes:

- ❖ El software Octoparse es una solución rica en funciones y asequible para realizar Web Scraping de datos de todo tipo.
- ❖ El uso de servidores en la nube permite la extracción de datos 24/7.

Sin embargo, como inconvenientes tenemos la siguiente:

- ❖ A veces puede ser difícil adaptarlo a las necesidades específicas del usuario.

Es una aplicación que está preparada para Windows por lo que funciona bien para sitios estáticos y dinámicos. Además, nos permite exportar los datos extraídos en diferentes formatos como Excel, CSV, TXT, y otros. Además de muchas bases de datos como MySQL u Oracle.

Por otro lado, *Octoparse* tiene la versión en la nube, siendo esto una de las características más importantes ya que se puede realizar la ejecución lista en la nube para descargar, sin tener el ordenador encendido mientras se realiza el proceso de extracción.

Dentro de *Octoparse* se puede usar lenguaje REGEX y APIs.

A continuación, podemos visualizar el panel de control de Octoparse en un proceso de Scraping de una Web en la siguiente figura.

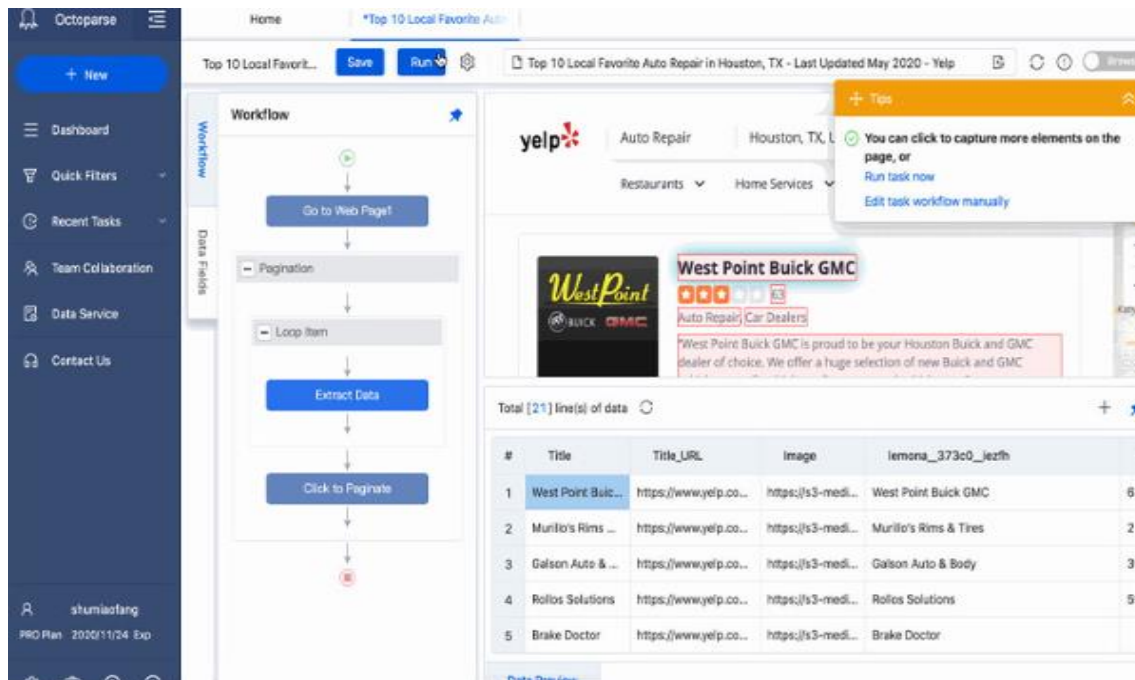


Ilustración 7. Panel de control de Octoparse con una Web scrapped ilustrativa

4.1.5. Tableau

Tableau es un software de análisis de datos con una capa de visualización y presentación, utilizada para el Business Intelligence y análisis de datos la cual simplifica los datos para presentarlos en un formato comprensible e intuitivo.

Se fundó en 2003 en la universidad de Stanford. Los cofundadores Chris Stolte, Pat Hanrahan y Christian Chabot desarrollaron y patentaron la tecnología central de Tableau: VizQL. Esta tecnología permite expresar datos visualmente al transformar acciones de arrastrar y soltar en consultas de datos, en una interfaz intuitiva. Entre las capacidades de la herramienta incluye lograr que el aprendizaje automático, las estadísticas, el lenguaje natural y la preparación de datos inteligente para el análisis de estos.

El análisis de datos se realiza muy rápidamente utilizando esta herramienta. Las capacidades principales de Tableau son el análisis en tiempo real, la visualización de datos y la colaboración.

Hay dos categorías de herramientas de Tableau: herramientas para desarrolladores y herramientas para compartir.

La primera categoría se utiliza para el desarrollo de paneles, gráficos, informes y visualizaciones. Son los denominados Tableau Desktop y Tableau Public. Permiten codificar y personalizar informes. Es posible crear gráficos e informes, combinarlos y formar un panel de control. También se puede conectar a un almacén de datos para el análisis de datos en tiempo real.

En relación con la segunda categoría, en cuanto a las herramientas para compartir, Tableau Server comparte libros de trabajo y visualizaciones. Una vez cargado en el servidor, se puede acceder en remoto a los paneles de visualizaciones creados en Tableau Desktop, permitiendo el uso en línea, de forma análoga a la herramienta Tableau Online con la única diferencia de ésta última en que los datos se almacenan en servidores alojados en la nube y son mantenidos por el editor.

Por último, tenemos la herramienta Tableau Reader la cual permite ver visualizaciones creadas con Tableau Desktop.

Tableau se conecta a diferentes fuentes para recuperar datos sean base de datos, archivos de Excel, PDF u otros.



Ilustración 8. Panel de Tableau, ejemplo

4.1.6. PowerBI

PowerBI (Lachev., 2022) es un conjunto de servicios de software, aplicaciones y conectores que nos permiten la conversión de diferentes fuentes de datos sin una relación entre sí, en información con sentido, interactiva y fácil de visualizar.

PowerBI se utiliza como una herramienta de inteligencia empresarial para convertir los datos en visualizaciones. Estos se actualizan automáticamente y permiten encontrar información y asesorar en la toma de mejores decisiones.

Por otra parte, es una herramienta de Business Intelligence (BI) denominada de autoservicio, lo que significa que está dirigida a usuarios no técnicos, acostumbrados a

generar páginas de informes a través de fórmulas y tablas dinámicas en Excel. PowerBI permite que los informes se programen una sola vez y luego sus actualizaciones sean automáticas.

Dispone de una gran flexibilidad sobre los ficheros de fuentes de datos que puede usar: se pueden usar CSV, Txt, Excel, conexión con base de datos (por ejemplo, SQL), conexión con SAP Hana, con SharePoint, etc., lo que hace que sea una herramienta muy versátil y enfocada a diferentes públicos.

Además, una de las novedades que ha incorporado es que se puede acceder a ella a través de su aplicación móvil, lo que hace que se puedan acceder a las visualizaciones en cualquier momento desde un *Smartphone*.

Actualmente y dadas las diferentes leyes de protección de datos (muchas de ellas estudiadas en la asignatura de Riesgo, Seguridad y Legislación en Sistemas de Información, impartida por los profesores Tamara Álvarez Robles y José Blanco Vega), PowerBI ha creado la diferencia de roles entre usuarios, con lo que podemos asociarles determinados permisos a unos usuarios o a otros dependiendo la región, ciudad o departamento en el que estén.

Podemos ver un panel de control en la ilustración adjunta.



Ilustración 9. Panel de Control tipo en PowerBI

4.2. Proceso KDD.

4.2.1. Definición.

En el mundo actual, los datos se generan a partir de numerosas fuentes de diferentes tipos y formatos, como, por ejemplo, transacciones económicas, biométricas, médicas, de servicios como medios de transporte, científicas, en sectores terciarios como el turismo, etc. Estas cantidades tan enormes de información que se intercambian en cada momento, hace necesario hacer uso de técnicas que sean capaces de extraer de las mismas, información estructurada y proporcionar datos confiables, de alta calidad y efectivos para su uso en diversos campos para la toma de decisiones. Aquí es donde KDD (descubrimiento de conocimiento en bases de datos) es tan útil.

El proceso KDD se define como un método para encontrar, transformar y refinar datos y patrones significativos de una base de datos sin procesar para ser utilizados en diferentes dominios o aplicaciones. El objetivo principal del proceso KDD es extraer información de bases de datos masivas empleando técnicas de minería de datos para determinar qué se considera conocimiento.

Este proceso de extracción de conocimiento es largo y complejo el cual implica muchos pasos e iteraciones ya que es clave para entender las bases de datos. Esto ayuda a decodificar patrones que pueden ayudar a completar tareas de manera más eficiente y rápida. Lo que eventualmente termina con es el descubrimiento de conocimiento que es refinado, confiable y altamente específico para su aplicación

KDD en minería de datos es un enfoque programado y analítico para modelar datos de una base de datos para extraer conocimiento que sirva para el propósito de la investigación. La minería de datos forma la columna vertebral de KDD y, por lo tanto, es fundamental para todo el método. Utiliza varios algoritmos para deducir patrones útiles a partir de los datos procesados. La minería de datos es la fase analítica del proceso de KDD. Incluye la introducción de algoritmos que examinan los datos, crean el modelo y descubren patrones previamente no descubiertos, considerándose el núcleo central del método KDD.

El proceso es una retroalimentación constante como un bucle cerrado donde se producen muchas iteraciones entre los diversos pasos según la demanda de los algoritmos y las interpretaciones de patrones.

Esta retroalimentación constante implica que es un proceso continuo en un proceso de análisis de datos, ya que se pueden ir añadiendo continuamente nuevos datos, por la vida y la estructura en sí de los mismos, por su avance o por su volatilidad, por ejemplo.

A su vez lleva a que sea un proceso recurrente y que una vez se llega a una conclusión, sea posible volver a realizar el proceso para mejorar los resultados y encontrar información adicional para que nuestros resultados o decisiones estén mejor soportados.

4.2.2. Etapas del proceso KDD.

El proceso se puede resumir en las siguientes etapas:

- ❖ Establecimiento de objetivos y comprensión de la aplicación. Implica elegir el Dataset base del que queremos partir nuestra investigación o análisis. Requiere una comprensión previa y conocimiento del propósito donde se aplicará. Se decide de qué manera se utilizarán los datos transformados y los patrones a los que llega la minería de datos para extraer conocimiento.
- ❖ Selección e integración de datos. Después de establecer las metas y objetivos, los datos recopilados deben seleccionarse y segregarse en conjuntos significativos basados en la disponibilidad, la importancia de la accesibilidad y la calidad. Estos parámetros son críticos para la minería de datos porque constituyen la base para ello y afectarán a los tipos de modelos de datos que se forman.
- ❖ Limpieza y preprocesamiento de datos. Se procede a la limpieza de dicho conjunto de datos (*outliers*, *missing*, etc.). Implica buscar datos faltantes ("missings") y eliminar datos ruidosos, redundantes y de baja calidad del conjunto de datos para mejorar la confiabilidad de los datos y su efectividad. Ciertos algoritmos se utilizan para buscar y eliminar datos no deseados basados en atributos específicos de la aplicación.
- ❖ Transformación de datos. Este paso prepara los datos que se alimentarán a los algoritmos de minería de datos. Por lo que los datos deben estar en formas consolidadas y agregadas. Se realiza la transformación de las variables o reducción del número de estas si estamos ante un conjunto de datos con un gran número de variables. Los datos se consolidan sobre la base de funciones, atributos, características, etc.
- ❖ Minería de datos. Este es el proceso raíz o columna vertebral de todo el KDD. Aquí es donde se utilizan algoritmos para extraer patrones significativos de los datos transformados, que ayudan en los modelos de predicción. Es una herramienta analítica que ayuda a descubrir tendencias a partir de un conjunto de datos utilizando técnicas como la inteligencia artificial, métodos numéricos y estadísticos avanzados y algoritmos especializados. Es en la minería de datos donde se crean nuevas variables a través de otras encontradas y se realiza la búsqueda y descubrimiento de

patrones no visibles a simple vista. En este proceso se pueden llevar a cabo técnicas de Clustering, asociaciones, patrones secuenciales, etc.

- ❖ Evaluación/interpretación de patrones. Una vez obtenidos los patrones mediante varios métodos e iteraciones de minería de datos, estos deben representarse en formas discretas como gráficos de barras, gráficos circulares, histogramas, etc. para estudiar el impacto de los datos recopilados y transformados durante los pasos anteriores. Esto también ayuda a evaluar la efectividad de un modelo de datos particular en vista del dominio. Con ello, se procede a sacar conclusiones del análisis previo y se lleva a cabo la evaluación de los modelos predictivos. En esta fase se evaluarán los resultados obtenidos para tomar decisiones que tendrán un impacto en nuestra investigación o estudio.
- ❖ Por último, se concluirá con la fase del descubrimiento y uso del conocimiento. Este es el paso final en el proceso de KDD y requiere que el "conocimiento" extraído del paso anterior se aplique a la aplicación o dominio específico en un formato visualizado, como tablas, informes, etc. Este paso impulsa el proceso de toma de decisiones para dicha aplicación dentro de la investigación objeto de estudio.

A continuación, en la siguiente ilustración podemos ver el proceso KDD descrito anteriormente de una manera más visual.

Este proceso es el que se ha realizado en el presente TFM, se ha realizado el proceso de KDD comprobando diferentes fuentes externas, añadiendo nuevas variables, viendo si mejoraban los resultados o no, incluyendo o no variables dependiendo de sus correlaciones, buscar de nuevo variables adicionales. A su vez se ha procedido a probar los modelos de Machine Learning (aprendizaje automático) como se verá en el siguiente apartado y ver cuáles son las variables más importantes y de nuevo proceder a la inclusión de nueva información que nos pudiera resultar útil para nuestro análisis y obtener conclusiones relevantes para el propósito de la investigación.

Data Mining: A KDD Process

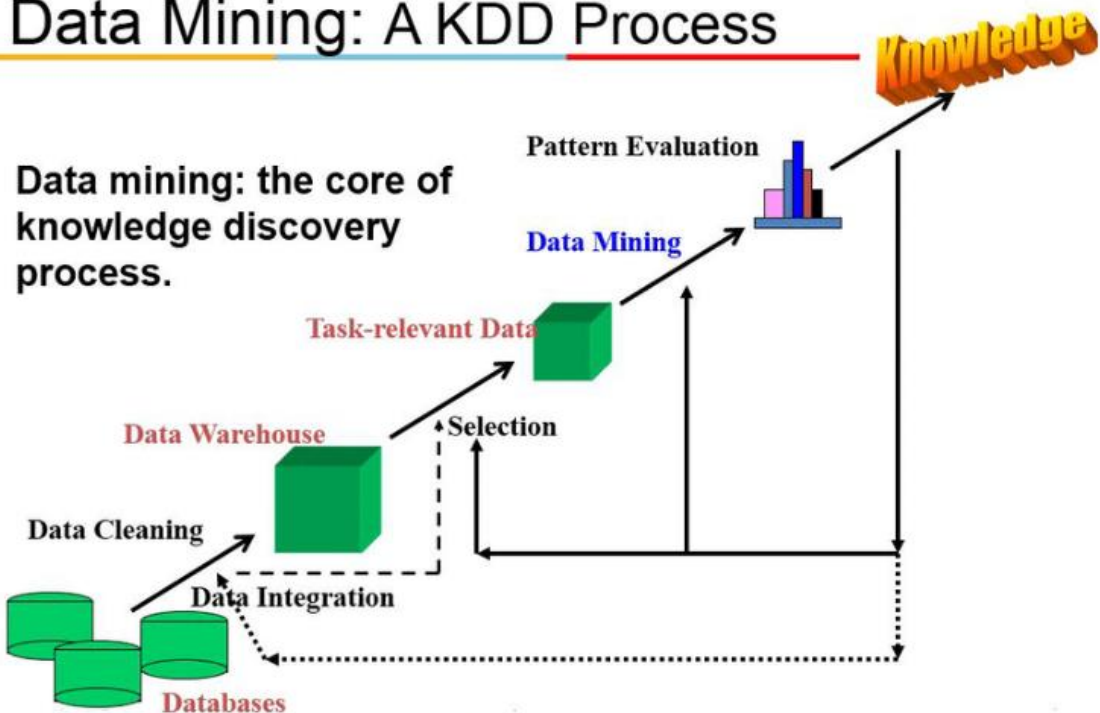


Ilustración 10. Proceso KDD en Big Data

4.3. Machine Learning.

A partir de 2006 el Machine Learning (Guido., 2018) apareció con más fuerza de mano de las empresas de IBM y Microsoft y empezó entonces a expandirse a nivel internacional. Desde entonces, compañías como Microsoft, Azure y Amazon han ido evolucionando en los diferentes productos y softwares que ofrecen para la utilización de técnicas de Machine Learning e Inteligencia Artificial en el mundo actual.

Machine Learning es un conjunto de disciplinas para extraer conocimiento a partir de datos. Ésta encuadrada en el campo de la investigación interviniendo otras disciplinas como estadística, inteligencia artificial, ciencias de computación.

También es conocida por sus capacidades de realizar análisis predictivos o aprendizaje estadístico. Hoy en día se encuentra extendido en todos los sectores de la sociedad, desde el industrial, al sanitario, pasando por las grandes corporaciones o las instituciones públicas.

A parte de las aplicaciones mencionadas, las disciplinas de Machine Learning han influenciado de manera central el modo en que los proyectos de investigación se realizan en la actualidad, como el estudio de planetas, o en el campo de la biología el estudio complejo del ADN, entre otros.

Existen dos grandes bloques de algoritmos para Machine Learning.

Aquellos que realizan el aprendizaje usando los pares de entrada/salida de datos son llamados algoritmos de aprendizaje supervisado. Estos algoritmos enseñan a los algoritmos de la deseada salida teniendo como patrón los ejemplos de los que realizaron el aprendizaje. Ejemplos de los usos de este tipo de algoritmos pueden ser en medicina el diagnóstico de cáncer basado en imágenes médicas, o en el campo bancario la identificación de transacciones fraudulentas, entre otros.

El otro gran grupo son los algoritmos no supervisados, en los que, en este caso, solo la entrada de datos es conocida, siendo en cambio la salida de datos desconocida a priori. Es por ello por lo que este tipo de algoritmos son más difíciles de entender y de evaluar a l no ser tan intuitivos como el anterior. Ejemplos de usos de este tipo de algoritmos los tenemos en plataformas como Netflix con recomendaciones sobre tipos de películas que nos gustan o en la plataforma de venta online, Amazon, en la que te hacen recomendaciones de artículos para venta en función de ventas anteriores. También en el campo de la seguridad informática, este tipo de algoritmos son usados para detectar accesos no autorizados o virus en las páginas Webs, entre otros.

La clave del éxito en ambos tipos de algoritmos consiste en representar los datos de entrada en un formato que el ordenador pueda entender. Para ello es necesario entender el conjunto de datos de como una tabla en la que cada fila de datos consiste en muestras de esos datos de entrada y las columnas consistirían en las propiedades de estas.

Es de suma importancia en el proceso de Machine Learning comprender el Dataset y cuál es el propósito del estudio de estos. A partir de ahí será necesario contestar una serie de preguntas a fin de identificar que algoritmos se ajustan mejor al propósito a conseguir. Entre las preguntas tenemos:

- ❖ Que pregunta estoy intentando resolver con el Dataset seleccionado.
- ❖Cuál es la mejor manera de enfocar el problema desde el punto de vista de Machine Learning.
- ❖ El Dataset seleccionado tiene suficientes registros o no.
- ❖ Las propiedades de los registros seleccionados son las adecuadas para realizar la predicción que buscamos o no.
- ❖ Como puedo medir el éxito de mi predicción.
- ❖ Como interactuará el producto final obtenido mediante Machine Learning con otros apartados del estudio.

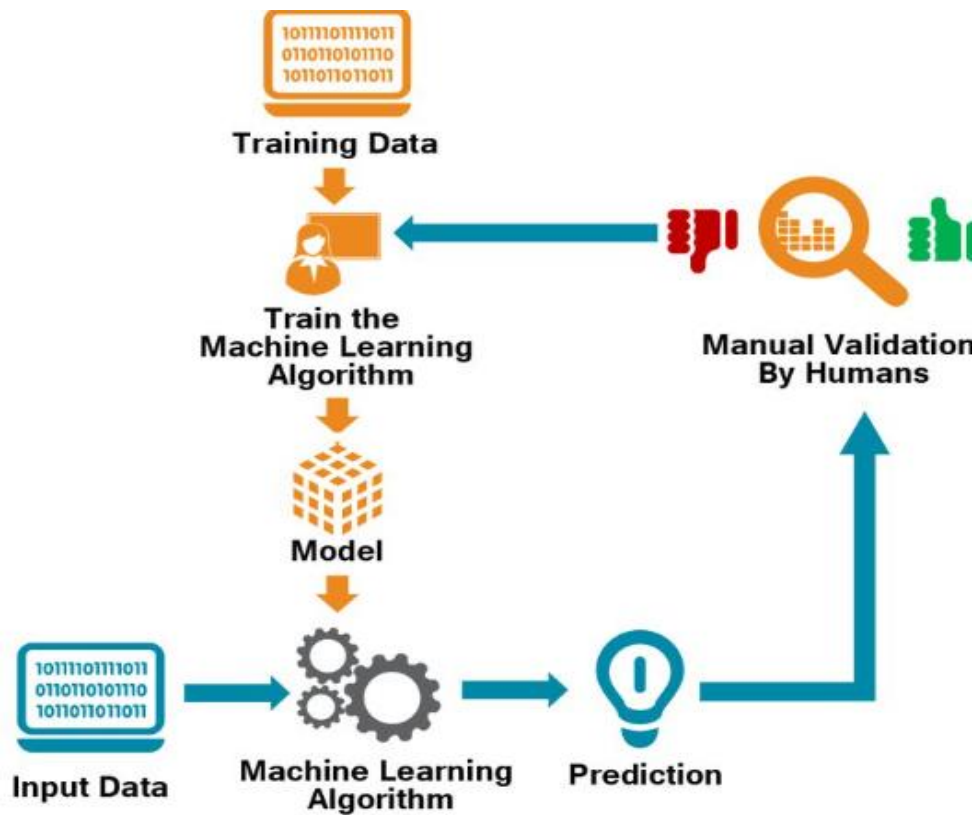


Ilustración 11. Proceso de aprendizaje de un algoritmo en Machine Learning

1. Desarrollo del proyecto

1.1. Metodología

Al comprender varios objetivos se han seguido varias metodologías para implementarlas.

En una primera instancia, se ha creado una cuenta en Twitter developer (Developer Portal, 2022) afín de extraer tweets con una determinadas palabras clave y también en unas páginas determinadas como expondré a lo largo del presente epígrafe. Estos se han cargados a MongoDB Atlas (Mongodb Atlas, 2022) para subirlos a la nube.

Del mismo modo, se han extraído información procedente de páginas Webs objetivo con la herramienta Octoparse y se ha cargado en MongoDB.

A continuación, se ha realizado un estudio mediante el proceso KDD de Dataset especializados en terrorismo. El conjunto de Dataset, se han analizado tanto en Jupyter como en RStudio a fin de sacar conclusiones a los mismos.

Seguidamente, se ha procedido a representar una vez los datos se han refundido, con la herramienta PowerBI y Tableau, al objeto de valorar los riesgos a la seguridad.

Por último, mediante métodos de Machine Learning, se ha procedido a probar a un algoritmo para predecir la predicción a futuro de la seguridad, basados en los datos del pasado.

1.2. Desarrollo del proyecto

1.2.1. Extracción, procesamiento y visualización de datos en MongoDB.

Una vez creada la cuenta en Twitter developer siguiendo las instrucciones adquiridas en la asignatura de Fundamentos de Big data, utilizamos las claves proporcionadas para fines académicos al objeto de introducirlas en el script de Python modificado para los propósitos del TFM a fin de extraer los tweets de las cuentas seleccionadas y almacenadas en Cuentas_Twitter_TFM.csv.

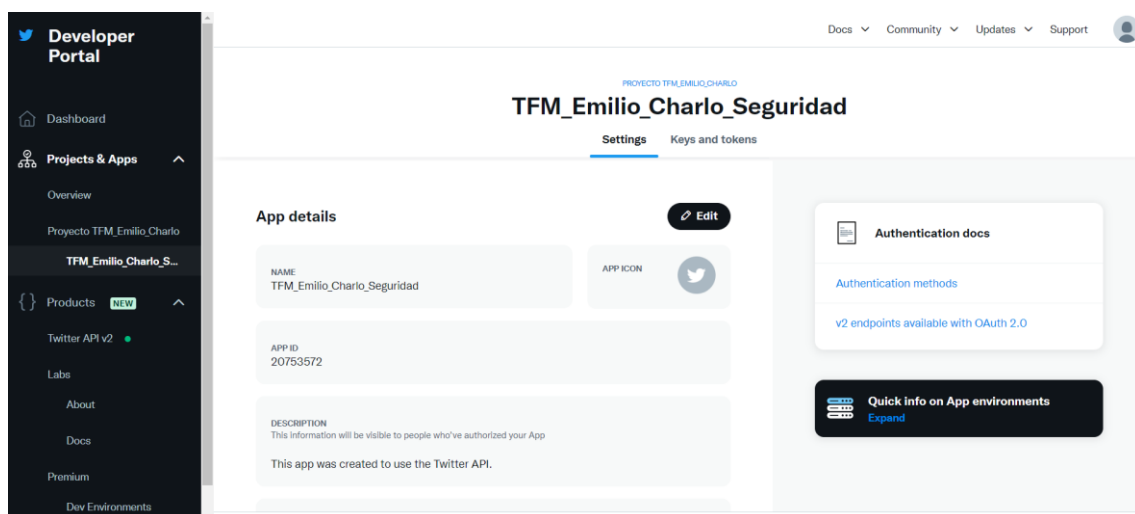


Ilustración 12. Creación en cuenta Twitter developer

Las cuentas seleccionadas para el propósito del siguiente estudio son relacionadas con noticias o del ámbito de seguridad, como se muestra a continuación.

- BBC Breaking News.
- The Guardian.
- Al Jazeera English.
- CNN Breaking News.
- Terrorism Watch.
- Israel Radar.

El proceso de extracción ha sido el siguiente:

En una primera instancia, sobre la base del IDE de Jupyter (Gutiérrez.), he importado las librerías necesarias como se muestra en la ilustración a

```
In [1]: 1 import sys
        2 import datetime
        3 import time
        4 import json
        5 import pandas as pd
        6 pd.Timestamp.now()

Out[1]: Timestamp('2022-10-03 22:43:24.537267')
```

```
In [2]: 1 !pip install pymongo

Requirement already satisfied: pymongo in c:\users\milom\anaconda3\lib\site-packages (4.2.0)
```

```
In [3]: 1 import pymongo
```

```
In [4]: 1 !pip install twython

Requirement already satisfied: twython in c:\users\milom\anaconda3\lib\site-packages (3.9.1)
Requirement already satisfied: requests-oauthlib>=0.4.0 in c:\users\milom\anaconda3\lib\site-packages (from twython) (1.3.1)
Requirement already satisfied: requests>=2.1.0 in c:\users\milom\anaconda3\lib\site-packages (from twython) (2.28.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\milom\anaconda3\lib\site-packages (from requests>=2.1.0->twython) (1.26.7)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\milom\anaconda3\lib\site-packages (from requests>=2.1.0->twython) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\milom\anaconda3\lib\site-packages (from requests>=2.1.0->twython) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in c:\users\milom\anaconda3\lib\site-packages (from requests>=2.1.0->twython) (3.2)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\milom\anaconda3\lib\site-packages (from requests-oauthlib>=0.4.0->twython) (3.2.1)
```

```
In [5]: 1 from twython import Twython
```

```
In [6]: 1 import timeit
```

continuación.

Ilustración 13. librerías necesarias para importar Tweets

A continuación, llamamos a las claves proporcionadas en Twitter developer.

```
In [7]: 1 APP_KEY = 'k1pDMFL3s9nFmvtLT3517VsGN' # API Key
        2 APP_SECRET = 'U4C3XqTcfaBXP58rf2iZG1OWFuJKUpMr1IondDFPUz3JHQC1BN' # API Secret Key
        3 OAUTH_TOKEN = '1389261920251883526-frxfbag986NqBPbJ2hXSAHFQC17Bx' # Access Token
        4 OAUTH_TOKEN_SECRET = 'cVbpQmznRVXSRZV8oj3TnSrXOL1L8j5XyyjHmbVMnQcAR' # Access Token Secret
```

```
In [8]: 1 twitter = Twython(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
```

Ilustración 14. Claves obtenidas en Twitter developer

Seguidamente, definimos una función que nos devuelva hasta un máximo de 200 Tweets por llamada.

```
In [9]: 1 def get_data_user_timeline_all_pages(kid, page):
        2     try:
        3         ...
        4         'count' especifica el número de tweets que se deben intentar recuperar, hasta un máximo de 200
        5         por solicitud distinta. El valor del conteo se considera mejor como un límite para
        6         el número de tweets que se devolverán porque se eliminó el contenido suspendido o eliminado
        7         después de que el recuento ha sido aplicado Incluimos retweets en el conteo, incluso si
        8         include_rts no se suministra. Se recomienda que siempre envíe include_rts = 1 cuando
        9         utilizando este método API.
        10        ...
        11        d = twitter.get_user_timeline(screen_name=kid, count="200", page=page, include_entities="true", include_rts="1")
        12        except Exception as e:
        13            print ("Error reading id %s, exception: %s" % (kid, e))
        14            return None
        15        #print (len(d)) #Número de entradas devueltas
        16        # Para visualizar Los campos (keys del diccionario) introducidos
        17        # if len(d)>0:
        18        #     print ("d.keys(): ", d[0].keys())
        19        return d
        20
        21 #Para visualizar Los campos (keys del diccionario) introducidos
        22 # if len(d)>0:
        23 #     print ("d.keys(): ", d[0].keys())
```

Ilustración 15. Función que recupera 200 Tweets por llamada

Una vez realizado lo anterior, es necesario configurar la base de datos de Mongo y las conexiones. Para ello estableceremos las conexiones a MongoDB Atlas. Es de señalar que se crean dos colecciones. Una primera colección llamada DBCollectionA = "Tweeter_acc" para almacenar las cuentas de Twitter objetivo que se encuentran en el archivo "Cuentas_Twitter_TFM.csv", y otra colección denominada DBCollectionT = "Tweeters" donde se van a almacenar los Tweets que extraiga de la anterior.

```
In [10]: 1 ##### PARTE 4: Configurar La base de datos Mongo y Las colecciones #####
2
3 #Establecimiento Conexión a MongoDB Atlas
4
5
6 # Datos de ejemplo, necesario modificar por vuestra instancia en MongoDB Atlas o local
7 dbStringConnection = "mongodb+srv://DB_TFM_ECR:AcR*1003@clustertfm.ss3qq9m.mongodb.net/?retryWrites=true&w=majority"
8 dbName = 'DB_TFM_ECR'
9 dbCollectionA = 'Tweeters_acc'
10 dbCollectionT = 'Tweeters'
```

Ilustración 16. Conexión a MongoDB

A continuación, intentamos leer las cuentas de Twitter y procedemos a añadirlas a MongoDB si es posible como se muestra en la ilustración adjunta.

```
In [12]: 1 ##### PARTE 5: Leer Las cuentas de Twitter (y añadir a MongoDB si es posible)
2
3 #Siempre trata de insertar las cuentas, si ya tenemos todas podemos cambiar a la otra versión que primero comprueba
4 #Substituir el fichero accountsEMBS.csv por el fichero csv propio con los datos de las cuentas de Twitter
5
6 df = pd.read_csv(r'C:\Users\milom\Desktop\TFM_Emilio_Charlo\Cuentas_Twitter_TFM.csv', encoding='latin-1')
7 repetidas = 0
8 cuentasTwitter = json.loads(df.T.to_json()).values()
9 print("Intentando insertar ", len(df), " cuentas de Twitter")
10
11 # Alternativa, solo leer y cargar a MongoDB si la colección de cuentas es vacía
12 if accounts.count_documents({}) < 1:
13     df = pd.read_csv(r'C:\Users\milom\Desktop\TFM_Emilio_Charlo\Cuentas_Twitter_TFM.csv', encoding='latin-1')
14     cuentasTwitter = json.loads(df.T.to_json()).values()
15     print ("No account data in MongoDB, attempting to insert", len(cuentasTwitter), "records")
16     try:
17         accounts.insert_many(cuentasTwitter)
18     except pymongo.errors.BulkWriteError:
19         print ('BulkWriteError\n');
20     pass
21 #else:
22 # print ("There are already", accounts.count_documents({}), "records in the *accounts* table")
23
24 # Listar documentos en la colección accounts
25 # list(accounts.find())[:1]
26
27 # Crea la lista de cuentas de twitter para descargar tweets
28 twitter_accounts = accounts.distinct('Twitter_URL')
29 #print len(twitter_accounts)
30 twitter_accounts[:5]
```

Intentando insertar 6 cuentas de Twitter

```
Out[12]: ['@AJEnglish', '@BBCBreaking', '@IsraelRadar_com', '@cnnbrk', '@guardian']
```

Ilustración 17. Resultado de cuentas en la colección de MongoDB

Por último, en base a la colección anterior objetivo, procedemos a extraer los Tweets almacenándolos en MongoDB en la colección DBCollectionT, dando como

```

88 #if twitter.get_application_rate_limit_status(jl_resources jl_search jl_search/tweets jl_remaining js).
89 if rate_limit < 5:
90     print ('Quedan menos de 5 llamadas API estimadas ... pausando por 5 minutos...')
91     time.sleep(300) #PAUSA POR 300 SEGUNDOS
92
93
94 elapsed = timeit.default_timer() - start_time
95 print ('# de minutos: ', elapsed/60)
96 print ("Número de nuevos tweets añadidos en esta ejecución: ", tweets.count_documents({}) - starting_count)
97 print ("Número de tweets actuales en la BD: ", tweets.count_documents({}), '\n', '\n')
98
99 ##### PARTE 7: Impresión del número de tweets en la base de datos por cuenta. #####
100
101 for org in db.tweets.aggregate([
102     {"$group": {"_id": "$screen_name", "sum": {"$sum": "1"}}}
103 ]):
104     print (org['_id'], org['sum'])

```

```

--- STARTING PAGE 4 ...llamadas a la API restantes estimadas: 856
.....llamadas a la API restantes estimadas: 855
--- PÁGINA FINALIZADA 4 PARA ORGANIZAR @terrorismwatch1 -- 200 TWEETS
--- STARTING PAGE 5 ...llamadas a la API restantes estimadas: 855
.....llamadas a la API restantes estimadas: 854
--- PÁGINA FINALIZADA 5 PARA ORGANIZAR @terrorismwatch1 -- 200 TWEETS
--- STARTING PAGE 6 ...llamadas a la API restantes estimadas: 854
.....llamadas a la API restantes estimadas: 853
--- PÁGINA FINALIZADA 6 PARA ORGANIZAR @terrorismwatch1 -- 200 TWEETS
--- STARTING PAGE 7 ...llamadas a la API restantes estimadas: 853
.....llamadas a la API restantes estimadas: 852
--- PÁGINA FINALIZADA 7 PARA ORGANIZAR @terrorismwatch1 -- 18 TWEETS
--- STARTING PAGE 8 ...llamadas a la API restantes estimadas: 852
No hubo tweets devueltos.....Desplazandose al siguiente ID
# de minutos: 8.134107825
Número de nuevos tweets añadidos en esta ejecución: 10816
Número de tweets actuales en la BD: 14395

```

Ilustración 18. Número total de Tweets extraídos.

resultado un total de 14395 tweets.

Por otro lado, al objeto de enriquecer más el Dataset objeto de estudio, he procedido a crear otro mediante una extracción de tweets en base a palabras clave e importarlas a MongoDB. Para ello he creado un código en Python modificando el anterior, para que, sobre un punto central en París, realicé una búsqueda de Tweets en un radio de 3000 kms. El idioma utilizado de búsqueda es el inglés y el francés

Las palabras clave utilizadas son las siguientes:

- Terrorist attack.
- Bomb attack.
- Attentat terroriste.
- Attentat à la bombe.

El proceso de extracción ha sido el siguiente:

En una primera instancia, sobre la base del IDE de Jupyter, he importado las librerías necesarias como se muestra en la ilustración a continuación. De ellas hay que destacar las librerías tweepy y geopy al objeto de extraer los tweets e intentar conseguir las coordenadas de los Tweets.

```
In [2]: 1 |pip install pymongo[srv]

Requirement already satisfied: pymongo[srv] in c:\users\milom\anaconda3\lib\site-packages (4.2.0)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in c:\users\milom\anaconda3\lib\site-packages (from pymongo[srv]) (2.2.1)

In [3]: 1 |pip install dnspython

Requirement already satisfied: dnspython in c:\users\milom\anaconda3\lib\site-packages (2.2.1)

In [4]: 1 |pip install tweepy

Requirement already satisfied: tweepy in c:\users\milom\anaconda3\lib\site-packages (4.10.1)
Requirement already satisfied: requests-oauthlib<2,>=1.2.0 in c:\users\milom\anaconda3\lib\site-packages (from tweepy) (1.3.1)
Requirement already satisfied: oauthlib<4,>=3.2.0 in c:\users\milom\anaconda3\lib\site-packages (from tweepy) (3.2.1)
Requirement already satisfied: requests<3,>=2.27.0 in c:\users\milom\anaconda3\lib\site-packages (from tweepy) (2.28.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\milom\anaconda3\lib\site-packages (from requests<3,>=2.27.0->tweepy) (2021.10.8)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\milom\anaconda3\lib\site-packages (from requests<3,>=2.27.0->tweepy) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\milom\anaconda3\lib\site-packages (from requests<3,>=2.27.0->tweepy) (3.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\milom\anaconda3\lib\site-packages (from requests<3,>=2.27.0->tweepy) (1.26.7)

In [5]: 1 |pip install twitter

Requirement already satisfied: twitter in c:\users\milom\anaconda3\lib\site-packages (1.19.6)
Requirement already satisfied: certifi in c:\users\milom\anaconda3\lib\site-packages (from twitter) (2021.10.8)

In [6]: 1 |pip install geopy

Requirement already satisfied: geopy in c:\users\milom\anaconda3\lib\site-packages (2.2.0)
Requirement already satisfied: geographiclib<2,>=1.49 in c:\users\milom\anaconda3\lib\site-packages (from geopy) (1.52)

In [7]: 1 import pymongo
2 from pymongo import MongoClient
3 import json
4 import tweepy
5 import twitter
6 from pprint import pprint
7 import configparser
8 import pandas as pd
9 from geopy.exc import GeocoderTimedOut
10 from geopy.geocoders import Nominatim
```

Ilustración 19. librerías necesarias para extraer tweets

Seguidamente, las claves proporcionadas en Twitter developer se extraerían del mismo modo y para la conexión a MongoDB, creamos una colección llamada dbCollectionA = “Búsqueda_tweets”, como se muestra en la ilustración adjunta.

```
In [11]: 1 dbStringConnection = "mongodb+srv://DB_TFM_ECR:Acr*1003@clustertfm.ss3qq9m.mongodb.net/?retryWrites=true&w=majority"
2 dbName = 'DB_TFM_ECR'
3 dbCollectionA = 'Busqueda_tweets'
4 client = pymongo.MongoClient(dbStringConnection)
5 # Definición de la base de datos MongoDB
6 db = client[dbName]

In [12]: 1 # Get MongoDB Cluster connection
2 client = MongoClient(dbStringConnection)
3 # Use or create a database named demo
4 db = client.dbName
5 # Create a collection tweet_collection
6 tweet_collection = db.tweet_collection
7 # Create an index with "id" and make sure the collected tweets are unique
8 tweet_collection.create_index([("id", pymongo.ASCENDING)],unique = True)

Out[12]: 'id_1'
```

Ilustración 20. Creación conexión MongoDB

A continuación, se ha procedido a seleccionar el enfoque de nuestra búsqueda. Para ello, haciendo uso de Google Earth, he obtenido las coordenadas de París. Sobre ellas, he creado en Jupyter un radio de 5000 kilómetros al objeto de obtener todos los Tweets de la temática seleccionada. Al objeto de seleccionar las palabras clave, he creado la búsqueda como sigue q = ("terrorist" and "attack") or ("Bomb" and "attack") or ("Attentat" and "terroriste") or ("Attentat" and "bombe"). En la siguiente ilustración se puede apreciar como queda el código en Jupiter.


```
In [36]: 1 # Number of tweets that needs to be returned from REST API. Default and Maximum value is 100.
2 count = 100
3 # Latitude value de Paris , ON
4 latitude = 48.856613
5 # Longitude value de Paris, ON
6 longitude = 2.352222
7 # Search Range of tweets in Kilometers
8 max_range = 5000
9 geocode = "48.856613, 2.352222, 5000km" # Location de Paris, ON , Retrieved with the help of Google Maps
10 geocode = "%f,%f,%dkm" % (latitude, longitude, max_range)
11 # Search Keyword - terrorist attack
12 q = ("terrorist") or ("Bomb") or ("Attentat" and "terroriste") or ("Attentat" and "bombe")
13 # Get the Search Results using the above values
14 search_results = rest_api.search.tweets( count = count, q = q, geocode = geocode)
15 # Get all the tweets from the JSON response
16 statuses = search_results["statuses"]
17
18 # Below Print required during debug
19 pprint(search_results["statuses"])
20 pprint(tweet_collection)
21 pprint(statuses[1])
```

Ilustración 21. búsqueda de palabras clave en Twitter.

A continuación, he procedido a almacenar en la colección que hemos creado con anterioridad, "tweet_collection", los tweets extraídos mediante palabras clave. Además, se ha añadido la fecha de la extracción de los mismos.

```
In [17]: 1 # Below ID to be used to extract further data and saving the tweets into MongoDB
2 since_id_new = statuses[-1]['id']
3 # Loop all the result
4 for statuse in statuses:
5     try:
6         #print(statuse) # Used during debug to check the values
7         tweet_collection.insert_one(statuse) # Inserting an individual value into MongoDB instance
8         pprint(statuse['created_at'])# print the date of the collected tweets , so that we know how many tweets have been in
9     except:
10         pass
```

```
'Mon Oct 31 19:53:08 +0000 2022'
'Mon Oct 31 19:53:05 +0000 2022'
'Mon Oct 31 19:52:12 +0000 2022'
'Mon Oct 31 19:52:10 +0000 2022'
'Mon Oct 31 19:51:33 +0000 2022'
'Mon Oct 31 19:51:19 +0000 2022'
'Mon Oct 31 19:50:41 +0000 2022'
'Mon Oct 31 19:50:26 +0000 2022'
'Mon Oct 31 19:50:04 +0000 2022'
'Mon Oct 31 19:49:24 +0000 2022'
'Mon Oct 31 19:49:12 +0000 2022'
'Mon Oct 31 19:47:54 +0000 2022'
```

Ilustración 22. Extracción de tweets.

Posteriormente, procedo a ver cuántos tweets me he descargado y cuantos usuarios. En total han sido 7213 tweets y 5768 usuarios de tweets.

```
In [40]: 1 # Display the number of tweets collected
2 print(tweet_collection.estimated_document_count())
3
4 # Display the unique twitter users for the saved tweets , just to get an idea about the data
5 user_cursor = tweet_collection.distinct("user.id")
6 print (len(user_cursor))
```

```
7213
5768
```

Ilustración 23. Número de Tweets y número de usuarios

Finalmente procedemos a imprimir una muestra de los diez primeros tweets para comprobar que efectivamente se ha realizado la búsqueda con los criterios marcados como se puede apreciar en la ilustración adjunta.


```
In [23]: 1 # Display only first 10 tweets using the cursor created above
2 for document in tweet_cursor[0:10]:
3     try:
4         print ('----')
5         # pprint (document) # use pprint to print the entire tweet document
6         print ('name:', document["user"]["name"]) # user name
7         print ('text:', document["text"]) # tweets
8     except:
9         print ("***error in encoding")
10        pass

----
name: GaudiumPress - English Edition
text: Religious Sister Killed in Terrorist Attack in Congo

Terrorist attack in Congo resulted in the burning of a Catho... https://t.co/XRtD165lyI
----
name: THE BIAFRAN RAVEN
text: Breaking News: Terrorists Strikes Near Abuja

8 confirms dead as terrorists attack Nigerian Military Barracks at th... https://t.co/vmeKe18Wf0
----
name: J.R.
text: @semper_vincit Washington or London are the only 2 possible culprits for this terrorist attack.
----
name: WinterGinL
text: @stclairashley Because it was a terrorist attack and needed to be documented. Thank God Because #RepublicansLieAboutEver
ything
----
```

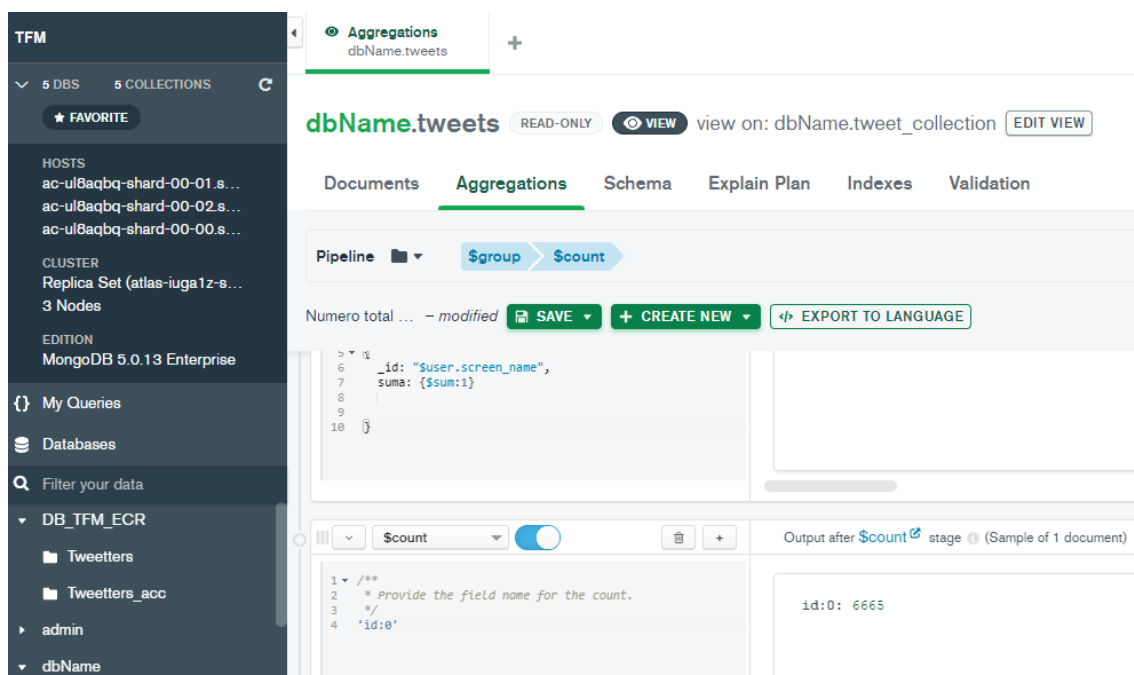
Ilustración 24. Diez primeros tweets con palabras clave de búsqueda.

Una vez importadas las bases de datos extraídas en MongoDB, he procedido a analizar las mismas.

En primer lugar, se han realizado filtros para descartar los tweets no relacionados con el objeto de búsqueda.

A continuación, desde MongoDB atlas, conecto con MongoDB Compass a fin de realizar análisis y sacar conclusiones de los datos extraídos.

En relación con el Dataset creado a partir de palabras clave relacionadas con terrorismo, en la colección "Tweets", estos son los resultados obtenidos analizando desde MongoDB Compass.



The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the database structure: TFM, 5 DBS, 5 COLLECTIONS. The main panel shows the 'dbName.tweets' collection. The 'Aggregations' tab is active, showing a pipeline with two stages: '\$group' and '\$count'. The '\$count' stage is selected, and the output is displayed as a single document: {'id': 0, 'count': 6665}.

Ilustración 25. Número total de Tweets en Tweetters

Para ver cuáles son los hashtags más utilizados relacionados con los parámetros de búsqueda, utilizamos la siguiente expresión:

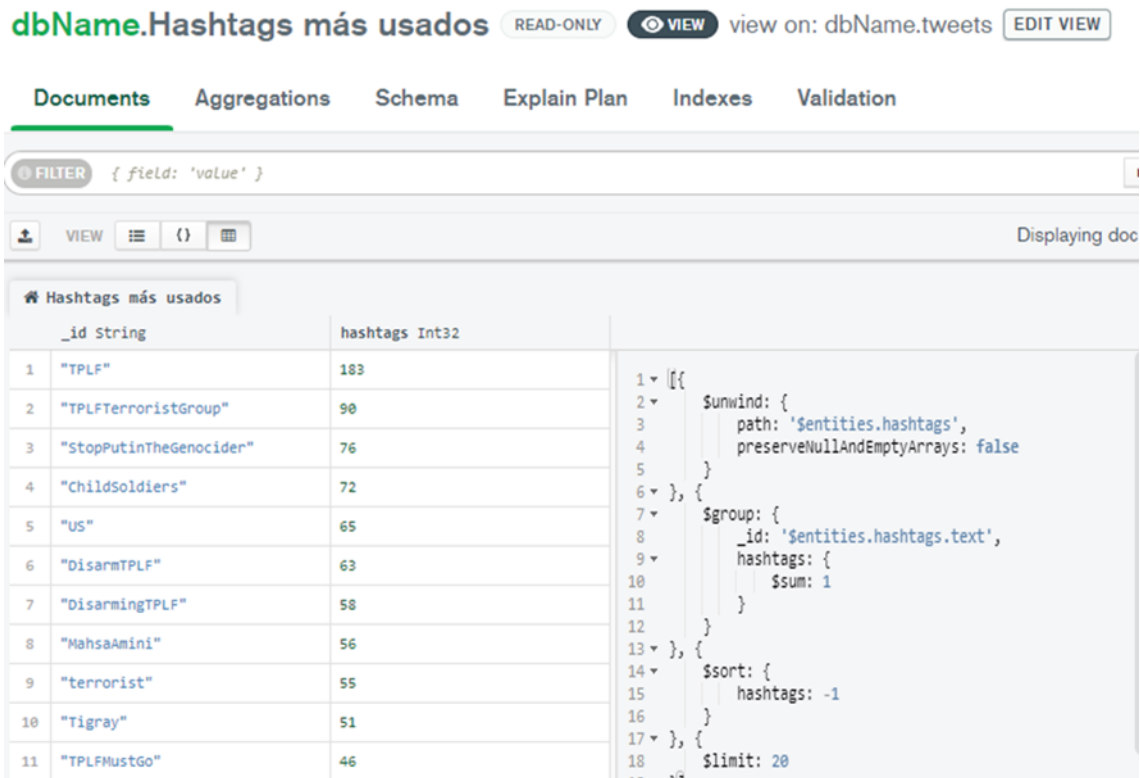


Ilustración 26. Hashtags más usados.

Entre ellos podemos observar que el hashtag “terrorist” se encuentra el noveno con 55 repeticiones en la muestra extraída. Selecciono el hashtag relacionado con terrorismo y visualizo desde donde se han producido.

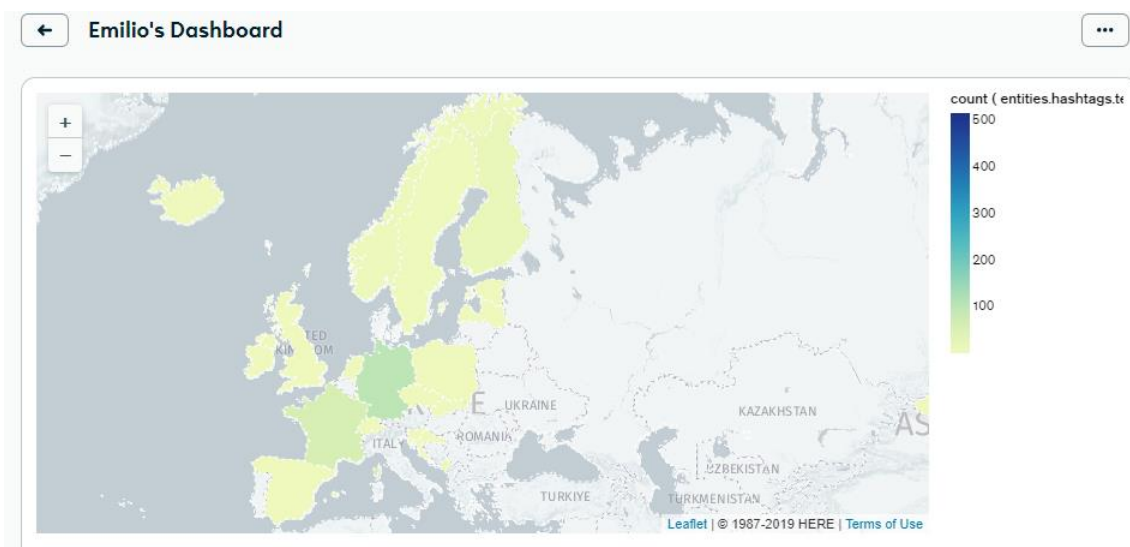


Ilustración 27. Hashtags relacionados con terrorismo en Europa cogiendo de base los 10 primeros anteriores.

En relación con el Dataset creado a partir de cuentas objetivos, en la colección “Tweeters”, estos son los resultados obtenidos analizando desde MongoDB Compass.

DB_TFM_ECR.Tweets por cuenta objetivo READ-ONLY VIEW view on: DB_TFM_ECR.Tweeters |

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } **OPTIONS**

VIEW [] [] [] [] Displaying documents 1 -

	_id String	suma Int32
1	"BBCBreaking"	828
2	"A3English"	4765
3	"IsraelRadar_com"	3313
4	"cnnbrk"	3336
5	"terrorismwatch1"	1217
6	"guardian"	9221
7	"Milo12169262"	4

```

1  [ [ {
2    $group: {
3      _id: '$user.screen_name',
4      suma: {
5        $sum: 1
6      }
7    }
8  } ] ]

```

Ilustración 28. número de tweets por cuenta objetivo

Vemos tras realizar un estudio de los hashtags más seguidos en las cuentas objetivo que no se encuentra específicamente ninguno entre los 30 primeros, relacionados específicamente con terrorismo.

En cambio, sí que encontramos multitud de ellos relacionados con zonas de conflictos como Israel, Irán, Ucrania y Rusia. Podemos observar en la ilustración adjunta el ranking de los hashtags.

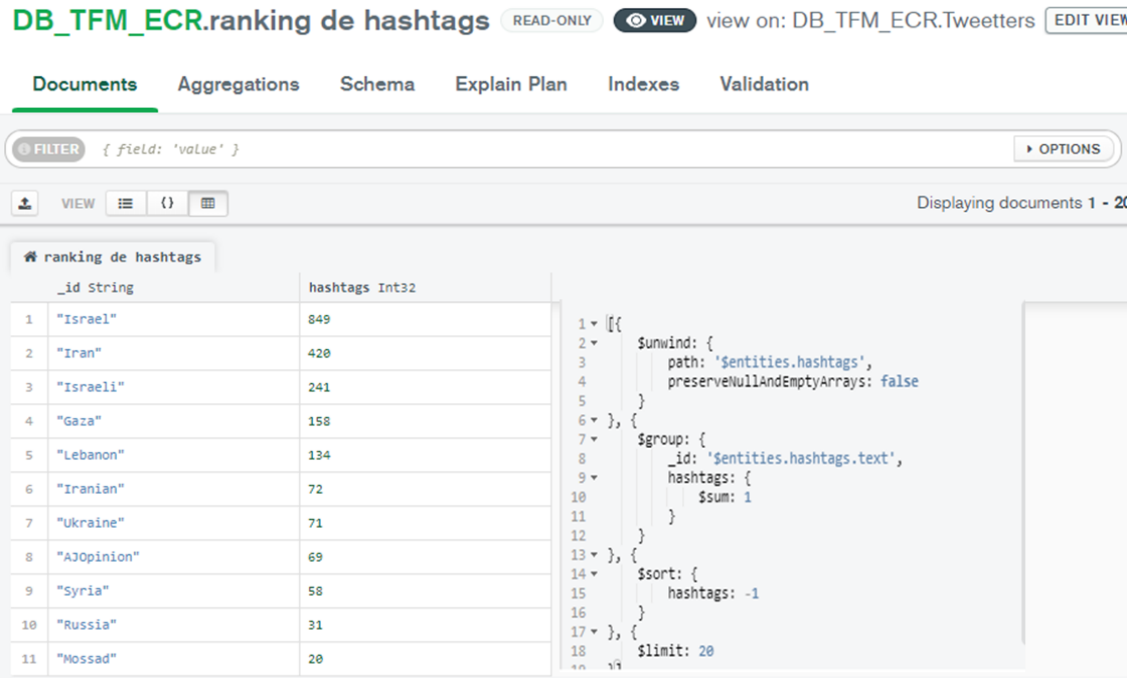


Ilustración 29. hashtags más seguidos para las cuentas objetivo.

Como resultado vemos que en las cuentas de Twitter seleccionadas para el estudio, el resultado es más genérico y se centra más en las zonas de conflicto

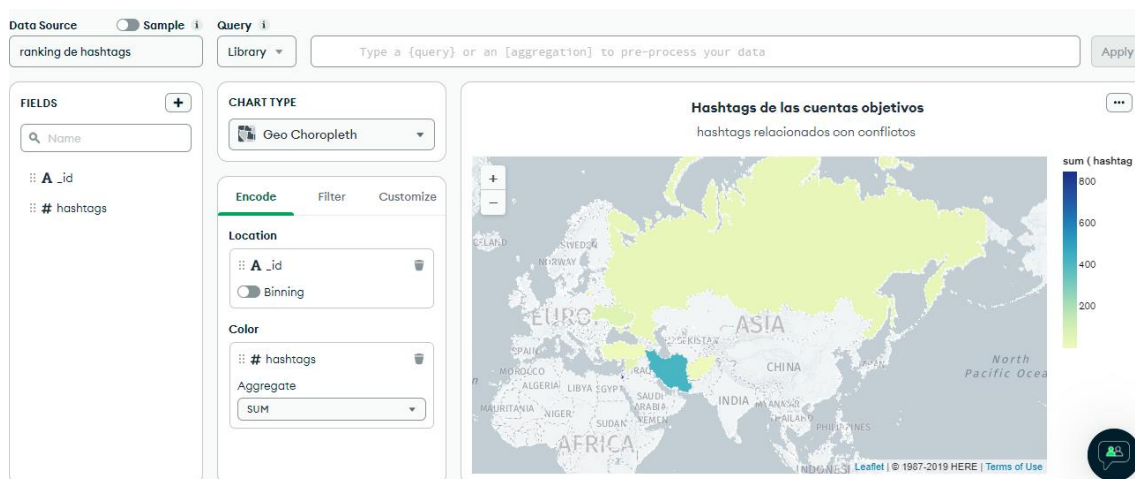


Ilustración 30. Hashtags utilizados relacionados con zonas de conflicto

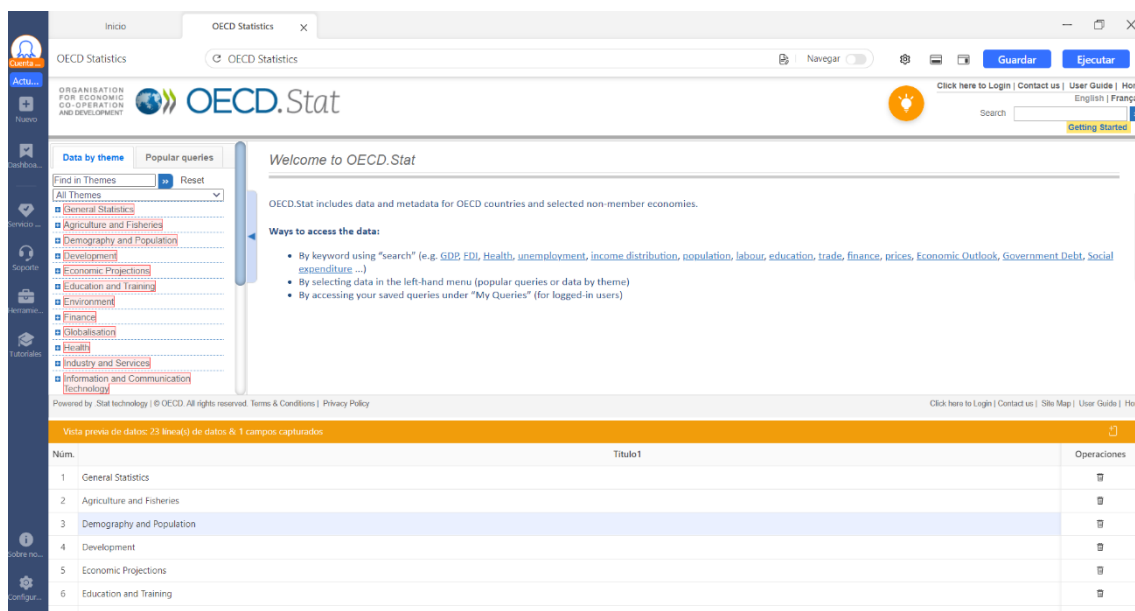
que en el terrorismo per se. Como conclusión de los dos métodos utilizados vemos que es más fiable realizar el estudio por palabras clave dirigidos a las temáticas a buscar que por cuentas objetivo. Aun así podemos llevar a la conclusión a su vez, que al no proporcionar de datos georreferenciados la mayor parte de los usuarios de Twitter, solo podemos aproximarnos en el mapa en función de la región origen del usuario, pero no se puede tener constancia

del lugar al que se refieren cuando usan la palabra terrorismo. Por lo tanto, esta metodología es útil para realizar estudios realizados encaminados a vislumbrar las preocupaciones o sentimientos de los usuarios de Twitter pero no es adecuada para identificar zonas de conflicto por lo anteriormente expuesto.

1.2.2. Proceso KDD con Base de datos especializada en Terrorismo utilizando Anaconda Jupyter con Python3.

Para el siguiente estudio, he seleccionado la temática de terrorismo. Para ello he acudido a la página Web <https://gtd.terrorismdata.com/> la cual recopila Dataset relacionados con terrorismo a lo largo del globo. Al objeto de la siguiente actividad, he elegido un Dataset que recopila los atentados terroristas desde el año 1970 hasta el 2019. Para el propósito del presente estudio, he acotado para incidentes desde el año 2000 hasta el 2019, centrándome en el área de Europa.

Por otro lado, he procedido a realizar un Web Scraping haciendo uso tanto de la herramienta Octoparse como de programación en Python3 al objeto de enriquecer aún más la búsqueda de atentados terroristas.



Núm.	Titulo1	Operaciones
1	General Statistics	
2	Agriculture and Fisheries	
3	Demography and Population	
4	Development	
5	Economic Projections	
6	Education and Training	

Ilustración 31. Ejemplo de extracción de Dataset para este estudio.

Para ello me he centrado en varias páginas Web de la organización especializada en conflictos Rand.org, la cual sigue los atentados más significativos además de otras de tipo demográfico como la página web de la OCDE, la cual dispone de multitud de estadísticas.

- <http://www.rand.org/nsrd/projects/terrorism-incidents.html>.
- <https://ourworldindata.org/terrorism#how-and-why-do-estimates-of-deaths-from-terrorism-vary>.
- <https://stats.oecd.org>.

Los Dataset extraídos se han procedido a realizar el proceso KDD de los mismos a fin de depurar los datos y extraer conocimiento de estos.

Me propongo en primer lugar central el estudio del Dataset para lo cual seguiré el proceso de selección, preproceso (Data Cleaning), transformación, minería de datos.

Intentaré vislumbrar si los patrones son consistentes a fin de predecir la tendencia más allá de los datos reflejados hasta el momento.

Todo ello mediante el estudio del Dataset el cual se componen de 201184 registros iniciales. Para ello definiré un subconjunto del Dataset focalizándome el propósito del estudio.

A continuación, realizamos el proceso KDD.

De todos los campos y datos del Dataset, he generado una selección para realizar mi estudio particularizando para los siguientes Campos:

Columna	Descripción	Tipo de dato
Eventid	Identificador unico	Numérico, Entero
Date	Fecha	String
región (id)		String
región_txt	región	
Country (Id)		String
Country_txt	Pais	
provstate	provincia	
city	ciudad	
Latitud	Coord geográficas	String
longitud	Coord geográficas	String
multiple	Ataque multiple	String valores 0 y 1
success	Éxito del ataque	String. Valores 0 y 1
attacktype1		String
attacktype1_txt	Tipo de ataque	
targettype1(id)		String
targettype1_txt	Tipo de objetivo	

gname	Grupo armado	
weaptype1		String
weaptype1_txt	Método de ataque	
weapsubtype1		String
weapsubtype1_txt	Sub método de ataque	
nkill	Numero fallecidos	
nwound	Número de heridos	

Tabla 1. Tipos de datos de las variables.

En el preproceso, he procedido a eliminar los siguientes campos que no eran objeto de estudio:

- Extended, Specificity, vicinity, location, crit1, crit2, crit3, doubtterr, targtype2 (id), targtype2_txt, targetsubtype1, targetsubtype1_txt, corp1, target1, natlty, natlty1_txt, suicide, guncertain, individual, nperps, nperpcap, claimed, nkillus, nkillter, nwoundus, nwoundte, property, propextent, propextent_txt, propvalue, propcomment, ishostkid, , ransom, , addnotes, scite1, scite2, scite3, dbsource, int_log, int_ideo, int_misc, int_any, relate, dbsource.

Los siguientes campos han sido suprimidos por encontrarse vacíos en más del 80% de los mismos:

- Approdate, resolution, summary, alternative, alternative_txt, attacktype2, attacktype2_txt, attacktype3, attacktype3_txt, targtype3, targtype3_txt, targtype2, targtype2_txt, targetsubtype2, targetsubtype2_txt, corp2, target2, natlty2, natlty2_txt, targtype3, targtype3_txt, natlty3, natlty3_txt, gsubname, gname2, gsubname2, gname3, gsumname3, motive, guncertain2, guncertain3, claimmode, claimmode_txt, claim2, claimmode2, claimmode2_txt, claim3, claimmode3, claimmode3_txt, compclaim, weaptype2, weaptype2_txt, weapsubtype2, weapsubtype2_txt, Weaptype3, weaptype3_txt, weapsubtype3, weapsubtype3_txt, Weaptype4, weaptype4_txt, weapsubtype4, weapsubtype4_txt, nhostkid, nhostkidus, nhours, ndays, diver, kidhijcountry, ransomamt, ransomamtus, ransompaid, ransompaidus, ransomnote, hostkidoutcome, hostkidoutcome_txt, nreleased.

He procedido a eliminar las filas con días en (0), un total de 872 registros y con meses en 0, un total de 20 registros.

En relación con el contenido de algunas columnas, he procedido a eliminar de las siguientes columnas, los siguientes atributos por no ser específicos o ser desconocidos:

- De la entidad attacktype he procedido a eliminar el atributo desconocido siendo un total de 9508 registros eliminados.

En cuanto al agrupamiento de campos, he procedido a agrupar los campos idday, idmonth, idyear en el nuevo creado Date con el siguiente formato DD/MM/AAAA.

Después de haber realizado la limpieza (“Cleaning”) y haber estructurado los datos de acuerdo con el propósito que me planteé me quedan 27878 registros en mi Dataset, ya que he acotado la búsqueda desde el año 2000 en adelante, para poder cruzarla con los datos demográficos y de desempleo. Al objeto de particularizar para Europa, he procedido a realizar un filtro por regiones, quedándome con 6040 registros a estudiar.

A continuación, con la ayuda de Google Colab, he procedido a continuar con el proceso KDD a fin de obtener conocimiento de los Datasets.

En primer lugar, antes de empezar cualquier análisis hay que asegurar que las librerías básicas y de uso general (numpy, pandas, etc.) están correctamente importadas.

```
[ ] # carga de datos
import pandas as pd
import io
from google.colab import files
# manipulación y visualización
import matplotlib.pyplot as plt
import numpy as np
import itertools
import seaborn as sns
import plotly.tools as tls
from plotly.offline import init_notebook_mode, iplot, plot
import plotly.graph_objs as go
```

Ilustración 32. Carga de librerías

A continuación, procedemos a la carga de los datos del Dataset, descargado mediante Octoparse anteriormente y creamos un DataFrame de los mismos.


```
[15] df = upload_files()

Choose Files | atentados_t...europa.xlsx
• atentados_terroristas_europa.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 434786 bytes, last modified: 10/31/2022 - 100% done
Saving atentados_terroristas_europa.xlsx to atentados_terroristas_europa (1).xlsx
User uploaded file "atentados_terroristas_europa.xlsx" with length 434786 bytes

[19] data_1="atentados_terroristas_europa"

[20] data_1 = pd.read_excel(data_1 + ".xlsx", header=0, parse_dates=False, index_col=0)
```

frames = [data_1]
Terrorismo = pd.concat(frames)
Terrorismo=Terrorismo.drop_duplicates()
Terrorismo

	Country	City	Latitude	Longitude	Multiple	Success	Attacktype	Target_type	Group	Weapon_type	Nkill	Nwound
Year												
2000	Kosovo	Peje	42.659809	20.307119	0	1	Bombing/Explosion	Private Citizens & Property	Unknown	Explosives	0	1
2000	Kosovo	Gorazhdec	42.640556	20.369722	0	0	Armed Assault	Private Citizens & Property	Unknown	Firearms	0	0
2000	Spain	Galdacano	43.230556	-2.845833	0	1	Armed Assault	Military	Unknown	Incendiary	0	1
2000	Spain	Guernica	43.317073	-2.678975	0	1	Armed Assault	Business	Unknown	Incendiary	0	0
2000	Germany	Erfurt	50.973734	11.022435	0	0	Armed Assault	Government (General)	Autonomous Decorators	Incendiary	0	0
...

Ilustración 33. Carga y Dataframe del Dataset

A continuación, procedo realizar una copia de seguridad en Excel y muestro las cinco primeras líneas para tener una idea global.

Seguidamente, procedo a comprobar el tipo de datos que contiene el Dataset. Vemos en la ilustración adjunta que tenemos un total de 5017 filas o registros y un total de 13 columnas.

```
[27] Terrorismo=Terrorismo.reset_index()
```

Terrorismo.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5017 entries, 0 to 5016
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Year            5017 non-null  int64
1   Country         5017 non-null  object
2   City            5017 non-null  object
3   Latitude        5017 non-null  float64
4   Longitude       5017 non-null  float64
5   Multiple        5017 non-null  int64
6   Success         5017 non-null  int64
7   Attacktype      5017 non-null  object
8   Target_type     5017 non-null  object
9   Group           5017 non-null  object
10  Weapon_type     5017 non-null  object
11  Nkill           5017 non-null  int64
12  Nwound          5017 non-null  int64
dtypes: float64(2), int64(5), object(6)
memory usage: 509.7+ KB
```

Ilustración 34. Tipo de datos del Dataset y número de registros y columnas

Con posterioridad, vemos el nombre de las columnas que tenemos y identificamos las numéricas y las no numéricas.

Veamos el nombre de las columnas en nuestro dataset

```
[31] Terrorismo.columns
Index(['Year', 'Country', 'City', 'Latitude', 'Longitude', 'Multiple',
      'Success', 'Attacktype', 'Target_type', 'Group', 'Weapon_type', 'Nkill',
      'Nwound'],
      dtype='object')
```

Veamos las variables que son numericas

```
[32] df_numeric = Terrorismo.select_dtypes(include=[np.number])
numeric_cols = df_numeric.columns.values
print(numeric_cols)

['Year' 'Latitude' 'Longitude' 'Multiple' 'Success' 'Nkill' 'Nwound']
```

Veamos cuales son no numericas

```
[33] df_non_numeric = Terrorismo.select_dtypes(exclude=[np.number])
non_numeric_cols = df_non_numeric.columns.values
print(non_numeric_cols)

['Country' 'City' 'Attacktype' 'Target_type' 'Group' 'Weapon_type']
```

Ilustración 35. Registros numéricos y no numéricos.

Seguidamente vemos en la composición realizada que no tenemos datos duplicados o vacíos.

```
[45] Terrorismo.duplicated().drop_duplicates(keep=False)
Series([], dtype: bool)
```

Vemos que no hay registros duplicados

Veamos que valores estan missing

```
[37] Terrorismo.isnull().sum
```

	Year	Country	City	Latitude	Longitude	Multiple	Success	Attacktype	Target_type	Group	Weapon_type	Nkill	Nwound
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...
5012	False	False	False	False	False	False	False	False	False	False	False	False	False
5013	False	False	False	False	False	False	False	False	False	False	False	False	False
5014	False	False	False	False	False	False	False	False	False	False	False	False	False
5015	False	False	False	False	False	False	False	False	False	False	False	False	False
5016	False	False	False	False	False	False	False	False	False	False	False	False	False
...
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...
5012	False	False	False	False	False	False	False	False	False	False	False	False	False
5013	False	False	False	False	False	False	False	False	False	False	False	False	False
5014	False	False	False	False	False	False	False	False	False	False	False	False	False
5015	False	False	False	False	False	False	False	False	False	False	False	False	False
5016	False	False	False	False	False	False	False	False	False	False	False	False	False

[5017 rows x 13 columns]>

Ilustración 36. Registros duplicados o vacíos

A continuación, veamos en la siguiente composición mostrada en la ilustración adjunta cuales son los registros por países. Se denota que Rusia y Reino Unido, han sido en este periodo de estudio, los países con mayor incidencia en atentados terroristas.

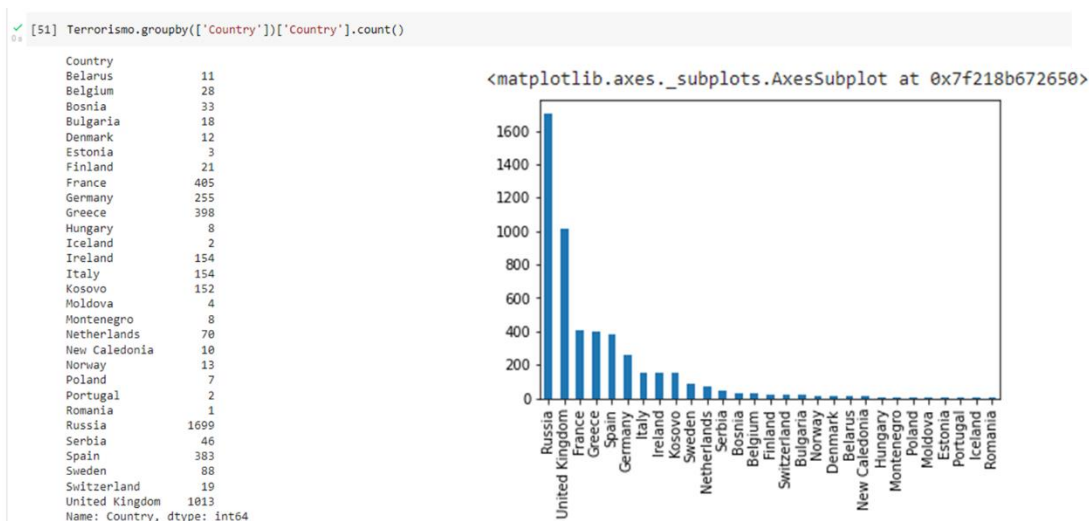


Ilustración 37. Datos repartidos por países.

Con posterioridad estudiamos las variables para ver las que tienen una correlación lineal superior al 70%. Vemos que hay dos variables numéricas que están muy correlacionadas entre sí. Las dos variables que están muy correlacionadas entre sí son Nkill y Nwound, que están directamente relacionadas por lo que lo tendremos en cuenta para posteriormente, cuando añadamos el resto de las variables (de tablas externas) eliminarla ya que sería redundante (esto lo hago en uno de los pasos finales, tras haber analizado más el Dataset).

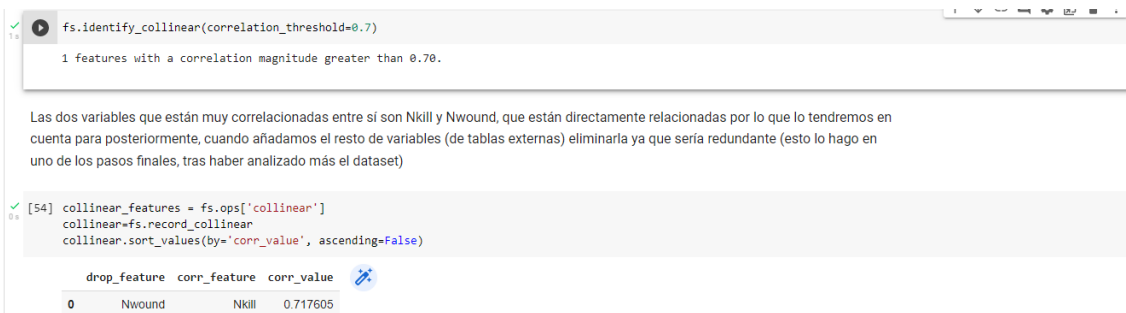


Ilustración 38. Correlación entre las variables Nkill y Nwound.

A continuación, procedemos a adjuntar los otros dos Dataset extraídos mediante Octoparse, de la página web de la OCDE, relacionados con la población y el desempleo. Una vez cargadas los tres Datasets, procedemos a fusionar las tablas por países.

En primer lugar, antes de fusionarlos, creamos índices en los Datasets de Población y Unemployed basados en “Year”.

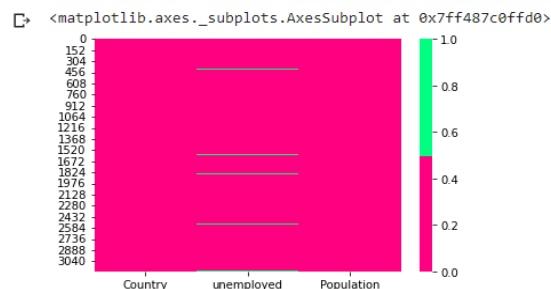
```
[ ] Población = pd.read_excel("Población.xlsx", header=0, parse_dates=False, index_col="Year")
Población
```

	Country	Population
Year		
2010	Austria	8361069.0
2010	Belgium	10895589.0
2010	Denmark	5543819.0
2010	Estonia	1331475.0
2010	Finland	5363341.0
...
2021	Switzerland	8701914.0
2021	United Kingdom	67350695.0
2021	Bulgaria	6898621.0
2021	Croatia	4024898.0
2021	Romania	19136038.0

334 rows x 2 columns

Ilustración 39. Creamos índice por Year

Comprobamos si existen en la composición realizada que no tenemos datos duplicados o vacíos.



```
[37] for col in Unemployed.columns:
      pct_missing = np.mean(Unemployed[col].isnull())
      print('{} - {}'.format(col, round(pct_missing*100)))
      plt.barh(col, round(pct_missing*100))
```

```
Country - 0%
unemployed - 2%
Population - 0%
```

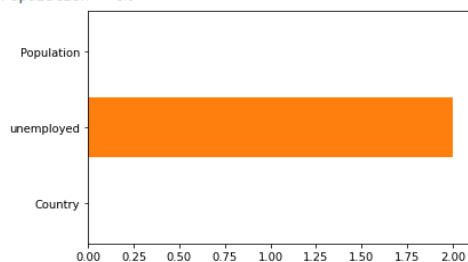


Ilustración 40. Porcentaje de valores nulos o vacíos

Comprobamos que al menos un 2% de los valores de desempleo están vacíos.

A continuación, los uno a mi Dataset principal, Terrorismo y vuelvo a analizar en búsqueda de valores “Missings” y ver si existen más correlaciones entre variables.

```
[46] Terrorismo=Terrorismo.merge(Unemployed, on=('Country' ), how='left')
Terrorismo
```

	Year	Country	City	Latitude	Longitude	Multiple	Success	Attacktype	Target_type	Group	Weapon_type	Nkill	Nwound	unemployed	Population
0	2000	Kosovo	Peje	42.659809	20.307119	0	1	Bombing/Explosion	Private Citizens & Property	Unknown	Explosives	0	1	NaN	NaN
1	2000	Kosovo	Gorazhdec	42.640556	20.369722	0	0	Armed Assault	Private Citizens & Property	Unknown	Firearms	0	0	NaN	NaN
2	2000	Spain	Galdacano	43.230556	-2.845833	0	1	Armed Assault	Military	Unknown	Incendiary	0	1	1401.967	46562483.0
3	2000	Spain	Galdacano	43.230556	-2.845833	0	1	Armed Assault	Military	Unknown	Incendiary	0	1	1504.993	46562483.0
4	2000	Spain	Galdacano	43.230556	-2.845833	0	1	Armed Assault	Military	Unknown	Incendiary	0	1	1697.113	46562483.0
...

Ilustración 41. Unión de los 3 Datasets

Veamos a continuación los valores vacíos del Dataset resultante.

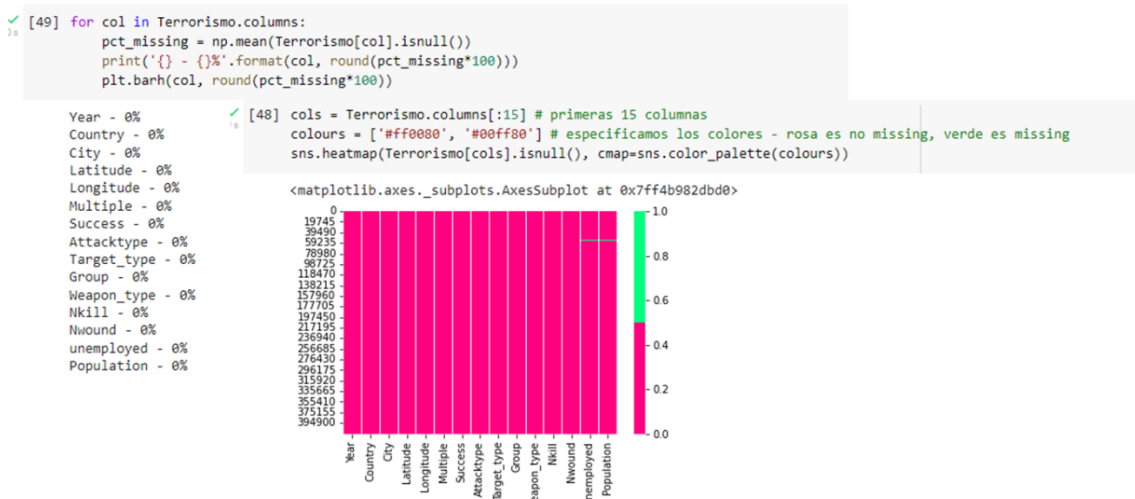


Ilustración 42. Valores vacíos

Veremos a continuación los porcentajes de nulos del Dataset resultante Terrorismo.

```
[58] Terrorismo.isnull().mean()*100
```

```
Year      0.000000
Country   0.000000
City      0.000000
Latitude  0.000000
Longitude 0.000000
Multiple  0.000000
Success   0.000000
Attacktype 0.000000
Target_type 0.000000
Group     0.000000
Weapon_type 0.000000
Nkill     0.000000
Nwound    0.000000
unemployed 0.478007
Population 0.478007
dtype: float64
```

```
[57] fs.plot_missing()
```

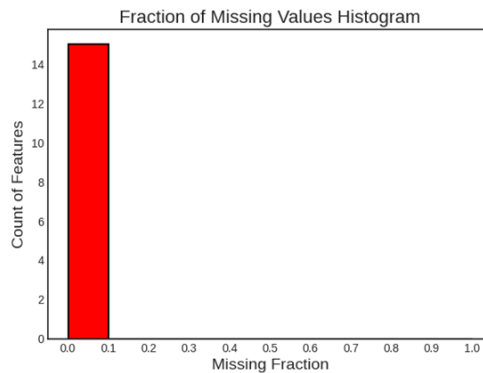


Ilustración 43. Porcentaje de valores nulos por variables

A continuación, Podemos ver la correlación lineal entre variables al 50% usando un mapa de calor. Las variables más correlacionadas son el número de heridos con el número de fallecidos. Pero también podemos observar que existen correlación en la latitud y longitud, lo que implica que los incidentes están agrupados. Las variables que aparecen en rojo en el mapa del calor inferior, son las que más relacionadas están entre sí de forma lineal. Podemos ver la correlación en valor en la tabla superior.

```
fs.identify_collinear(correlation_threshold=0.50)
fs.plot_collinear()
```

4 features with a correlation magnitude greater than 0.50.

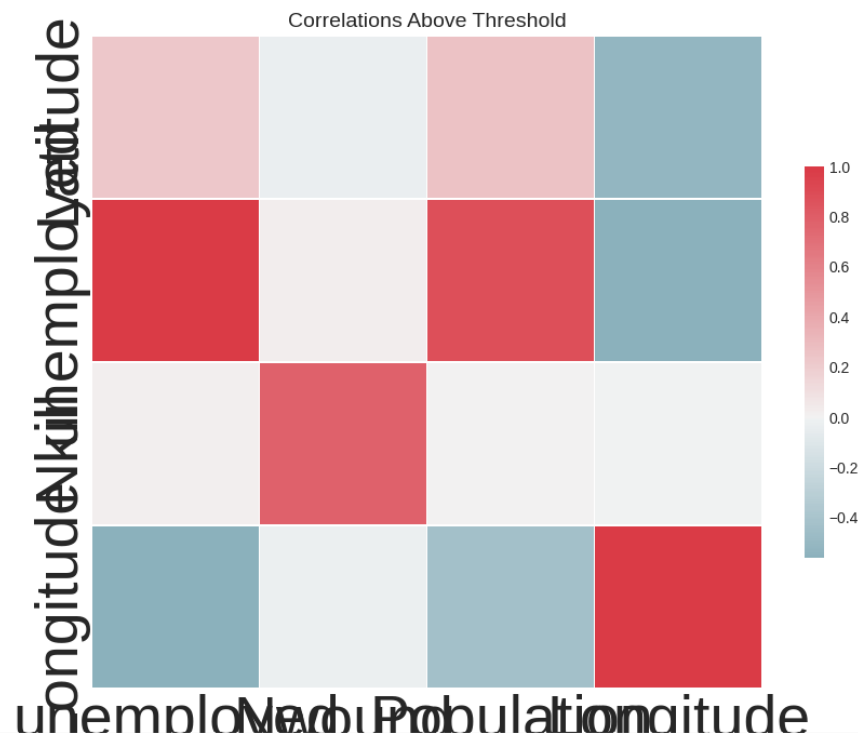


Ilustración 44. Correlación entre las variables al 50%

1.2.3. Proceso KDD con Base de datos especializada en Terrorismo utilizando RStudio con R

Para este estudio, procederé a abrir el abanico y me centraré en los datos globales desde el año 2000 hasta el año 2019. Con ello podré ver como de preciso fue mi estudio anterior y si los patrones se repiten. Por ello, los objetivos que me propongo para llevar a cabo el proceso KDD son los siguientes:

Analizar un Dataset con la temática de terrorismo haciendo uso de las herramientas estadísticas utilizando R Studio. Para ello he acudido a la página Web <https://gtd.terrorismdata.com/> la cual recopila Dataset relacionados con terrorismo a lo largo del globo.

Identificar las variables para la determinación descriptiva y posterior análisis de los modelos estadísticos a fin de vislumbrar si existe relación entre la llegada de las tecnologías de la información y el trasvase de conocimientos entre organizaciones criminales. Supondré que la llegada de las tecnologías de la información se produce en el año 2000 hacia adelante.

Al objeto de la siguiente actividad, he elegido un Dataset que recopila los atentados terroristas desde el año 1970 hasta el 2019. El Dataset se compone de 201184 registros. Para ello definiré un subconjunto del Dataset focalizándome el propósito del estudio, quedando una vez depurada en la asignatura de minería de datos 47234 registros. El Dataset se compone de 18 variables, 9 categóricas y 10 cuantitativas, como se puede apreciar en la tabla 1.

En cuanto al análisis descriptivo, podemos ver el resumen estadístico de las variables cuantitativas discretas no identificativas.

```
> summary(varNumAtaques_terr)
```

Day	Month	Year	latitude	longitude	multiple	success
Min. : 1.00	Min. : 1.000	Min. :1970	Min. : -43.533	Min. : -157.858	Min. :0.0000	Min. :0.0000
1st Qu.: 8.00	1st Qu.: 4.000	1st Qu.:1990	1st Qu.: 8.964	1st Qu.: -2.692	1st Qu.:0.0000	1st Qu.:1.0000
Median :15.00	Median : 6.000	Median :2010	Median : 26.542	Median : 40.782	Median :0.0000	Median :1.0000
Mean :15.55	Mean : 6.414	Mean :2004	Mean : 22.257	Mean : 24.333	Mean :0.1701	Mean :0.9013
3rd Qu.:23.00	3rd Qu.: 9.000	3rd Qu.:2015	3rd Qu.: 34.690	3rd Qu.: 69.147	3rd Qu.:0.0000	3rd Qu.:1.0000
Max. :31.00	Max. :12.000	Max. :2019	Max. : 67.144	Max. : 175.071	Max. :1.0000	Max. :1.0000

nkill	nwound
Min. : 0.000	Min. : 0.000
1st Qu.: 0.000	1st Qu.: 0.000
Median : 1.000	Median : 0.000
Mean : 3.854	Mean : 4.676
3rd Qu.: 3.000	3rd Qu.: 3.000
Max. :1180.000	Max. :4000.000

Tabla 2. Resumen estadístico variable cuantitativas.

Podemos ver a continuación el resumen estadístico de las variables categóricas, que aportaran datos como la localización, la autoría y el tipo de atentados entre otros, además de la tendencia a lo largo del tiempo.

```
> summary(varCatAtaques_ter)
country_txt      region_txt      provstate
India      : 4026  Middle East & North Africa:12208  Northern Ireland : 1359
Afghanistan: 3643  South Asia      :11789  Baghdad          : 776
Iraq       : 3344  South America   : 5557  Banaadir         : 775
Colombia   : 2753  Sub-Saharan Africa : 4943  Jammu and Kashmir: 711
Philippines: 2278  Western Europe   : 3928  West Bank        : 700
Peru       : 2147  Southeast Asia   : 3312  Borno            : 674
(Other)    :29043  (Other)          : 5497  (Other)          :42239

city            attacktype1_txt
Unknown      : 1782  Bombing/Explosion      :22300
Baghdad      : 776   Armed Assault          :12744
Mogadishu    : 758   Assassination          : 5776
Belfast      : 625   Facility/Infrastructure Attack : 3832
Lima         : 526   Hostage Taking (Kidnapping) : 1520
Kabul        : 408   Hostage Taking (Barricade Incident): 490
(Other)      :42359  (Other)                : 572

targtype1_txt      gname
Private Citizens & Property:11300  Taliban      : 3287
Military            :10510  Islamic State of Iraq and the Levant (ISIL): 3164
Police              : 7532  Shining Path (SL)      : 1894
Government (General) : 4512  Al-Shabaab             : 1768
Business            : 4352  Kurdistan Workers' Party (PKK) : 1425
Transportation       : 1348  New People's Army (NPA)  : 1374
(Other)              : 7680  (Other)                :34322

weaptype1_txt      weapsubtype1_txt
Chemical      : 81  Automatic or Semi-Automatic Rifle :10771
Explosives:24569  Projectile (rockets, mortars, RPGs, etc.): 6439
Firearms      :15870  Vehicle      : 5012
Incendiary: 4150  Handgun      : 4117
Melee         : 2564  Arson/Fire   : 3027
              :      Landmine      : 2870
              :      (Other)      :14998
```

Tabla 3. Resumen estadístico variables categóricas.

A continuación, mostramos la matriz de correlaciones de las variables numéricas.

```
> matriz_corr <- cor(varNumAtaques_ter[,])
> print(matriz_corr)
      Day      Month      Year      latitude      longitude      multiple
Day      1.000000000  3.770314e-03  0.006707810 -0.000764948  1.345851e-02 -0.000450856
Month    0.003770314  1.000000e+00 -0.010142767 -0.011373855  2.144575e-05 -0.013687706
Year     0.006707810 -1.014277e-02  1.000000000  0.132435528  5.257581e-01  0.220213243
latitude -0.000764948 -1.137386e-02  0.132435528  1.000000000  2.119893e-01  0.047722682
longitude 0.013458513  2.144575e-05  0.525758099  0.211989318  1.000000e+00  0.073436560
multiple -0.000450856 -1.368771e-02  0.220213243  0.047722682  7.343656e-02  1.000000000
success  -0.008113903  2.156482e-03 -0.058053937 -0.075223339 -2.245015e-02  0.024918010
nkill    -0.010592421  4.575268e-03  0.008824152 -0.030385585  7.399425e-04 -0.006469744
nwound   -0.006360617  3.161300e-03  0.047809391  0.028980520  4.296584e-02  0.028165474
      success      nkill      nwound
Day      -0.008113903 -0.0105924213 -0.006360617
Month    0.002156482  0.0045752676  0.003161300
Year     -0.058053937  0.0088241521  0.047809391
latitude -0.075223339 -0.0303855852  0.028980520
longitude -0.022450150  0.0007399425  0.042965835
multiple  0.024918010 -0.0064697444  0.028165474
success  1.000000000  0.0653041958  0.045506634
nkill    0.065304196  1.0000000000  0.308418325
nwound   0.045506634  0.3084183254  1.000000000
> correl_alta <- findCorrelation(matriz_corr, cutoff=0.75)
```

Tabla 4. Matriz de correlaciones de las variables numéricas.

Buscamos variables con correlación alta

```
> print(correl_alta)
integer(0)
```

Ilustración 45. Ninguna variable con correlación alta.

Imprimimos el nombre de las variables numéricas

```
> names(varNumAtaques_terr)
[1] "Day"      "Month"    "Year"     "latitude" "longitude" "multiple" "success"
[8] "nkill"    "nwound"
```

Ilustración 46. Variables numéricas.

A continuación, imprimimos la covarianza de la matriz de variables numéricas

```
> print(cov_matriz)
      Day      Month      Year      latitude      longitude      multiple
Day      75.550320146  0.110332264  0.8020352 -0.1191447  7.215344e+00 -0.001472436
Month     0.110332264  11.334797501 -0.4697404 -0.6861818  4.453383e-03 -0.017314815
Year       0.802035240 -0.469740389  189.2296226  32.6455256  4.460898e+02  1.138199265
latitude  -0.119144680 -0.686181780  32.6455256  321.1065697  2.343044e+02  0.321314070
longitude  7.215343837  0.004453383  446.0898454  234.3043639  3.804378e+03  1.701901554
multiple  -0.001472436 -0.017314815  1.1381993   0.3213141  1.701902e+00  0.141175916
success   -0.021033160  0.002165256 -0.2381677 -0.4020071 -4.129692e-01  0.002792224
nkill     -1.273276803  0.213026097  1.6787150 -7.5301153  6.311742e-01 -0.033618386
nwound    -1.552969766  0.298963532  18.4736774  14.5873485  7.444073e+01  0.297264431
      success      nkill      nwound
Day      -0.021033160 -1.27327680 -1.5529698
Month     0.002165256  0.21302610  0.2989635
Year      -0.238167660  1.67871502  18.4736774
latitude  -0.402007128 -7.53011535  14.5873485
longitude -0.412969151  0.63117416  74.4407338
multiple  0.002792224 -0.03361839  0.2972644
success   0.088943272  0.26934385  0.3812210
nkill     0.269343851  191.25766848  119.8105839
nwound    0.381221023  119.81058391  789.0257243
```

Ilustración 47. Covarianza de las variables numéricas.

A continuación, se realizan las gráficas de correlación entre el número de fallecidos y heridos en atentados con éxito y el año de su realización.

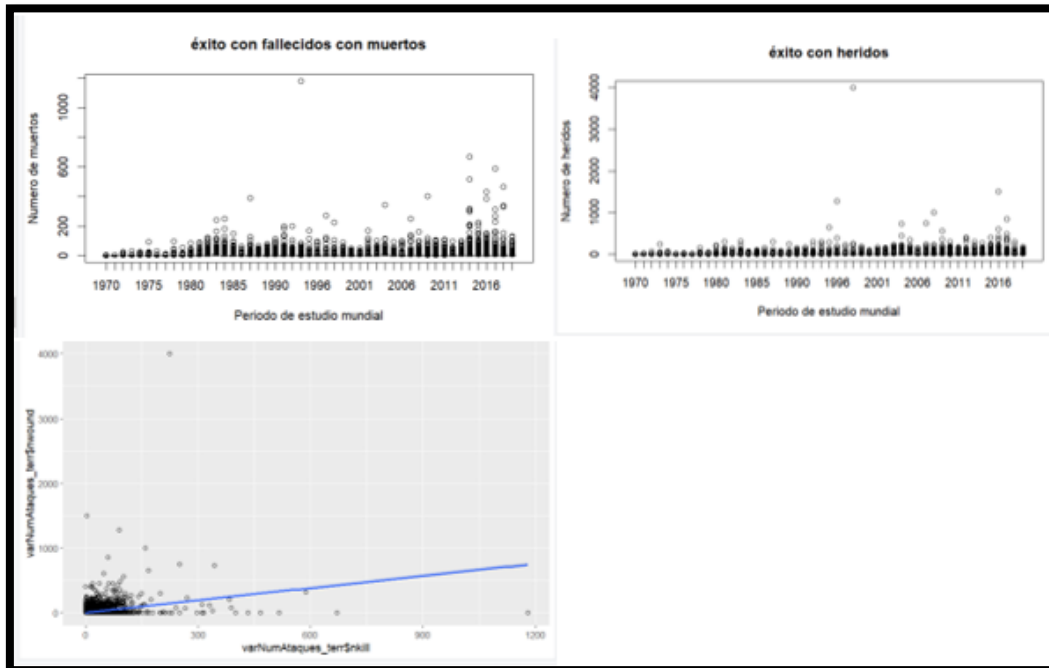


Ilustración 48. correlación entre el número de fallecidos y heridos en atentados con éxito y el año de su realización

Se verifica que variable tiene valores nulos.

```
> sapply(Ataques_ter, function(x) sum(is.na(x)))
      Day      Month      Year  country_txt  region_txt
      0          0          0          0          0
 provstate    city    latitude  longitude    multiple
      0          0          0          0          0
 success  attacktype1_txt  targtype1_txt    gname  weaptype1_txt
      0          0          0          0          0
weapsubtype1_txt    nkill    nwound
      0          0          0
```

Ilustración 49. Valores nulos.

Ninguna variable tiene valores nulos luego ya estaría suficientemente depurada las mismas.

A continuación, realizaremos un modelo de regresión. Recordemos que un modelo de regresión es un modelo matemático que busca determinar la relación entre una variable dependiente (Y), con respecto a otras variables, llamadas explicativas o independientes (X).

Caso 1. Número de fallecidos basado en el éxito del ataque.

```
> #Mostramos los datos estadísticos del modelo
> summary(lm.fit)

Call:
lm(formula = nkill ~ success, data = Ataques_ter)

Residuals:
    Min       1Q   Median       3Q      Max
-4.15  -4.15  -3.15  -1.12  1175.85

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.1244     0.2021   5.563 2.67e-08 ***
success       3.0283     0.2129  14.223 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.8 on 47232 degrees of freedom
Multiple R-squared:  0.004265, Adjusted R-squared:  0.004244
F-statistic: 202.3 on 1 and 47232 DF, p-value: < 2.2e-16
```

Ilustración 50. número fallecidos basado en el éxito de ataque

Vemos a continuación los Coeficientes de interceptación

```
> lm.fit

Call:
lm(formula = nkill ~ success, data = Ataques_ter)

Coefficients:
(Intercept)      success
      1.124         3.028
```

Ilustración 51. Coeficientes de interceptación.

Y finalmente los resultados de la predicción dan: 4.152702

Caso 2. Número de heridos basado en el el éxito del ataque.

```
> summary(lm.fit)

Call:
lm(formula = nbound ~ success, data = Ataques_ter)

Residuals:
    Min       1Q   Median       3Q      Max
   -5.1    -5.1    -5.1    -0.8   3994.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8129     0.4110   1.978   0.048 *
success       4.2861     0.4329   9.900  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.06 on 47232 degrees of freedom
Multiple R-squared:  0.002071, Adjusted R-squared:  0.00205
F-statistic: 98.01 on 1 and 47232 DF, p-value: < 2.2e-16

> data1 <-
+   data.frame(success=1, attacktype1_txt="Armed Assault")
> #la predicción una vez los condicionantes
> predict(lm.fit,data1)
      1
5.09903
> .15|
```

Ilustración 52. Número de heridos basado en el éxito de ataques.

Vemos a continuación los Coeficientes de interceptación

```
> lm.fit

Call:
lm(formula = nbound ~ success, data = Ataques_ter)

Coefficients:
(Intercept)      success
    0.8129       4.2861
```

Ilustración 53. Coeficientes de interceptación.

Y finalmente los resultados de la predicción dan: 5.09903

Mediante los datos extraídos con la predicción hecha, se estima que incrementarán el número de atentados en el área del África subsahariana, habiendo disminuido en oriente medio y Asia central.

Esta predicción es de gran utilidad para las agencias de seguridad. A su vez, en el apartado de protección frente a ataques terroristas, vemos que se estima que utilizaran en su mayor parte IEDs.

1.2.4. Machine Learning

Voy a utilizar en base a los datos y el propósito del estudio el algoritmo de árboles de decisión ya que maneja bien los datos no lineales, representando bien la complejidad en una forma compacta, dado ya realizado el proceso KDD.

El algoritmo de decisión buscado es el algoritmo de árboles de decisión que mejor se ajusta al modelo que quiero estudiar (Valenzuela.).

El problema consiste en predecir cuántos ataques se van a realizar en el futuro. Para ello utilizaremos el Dataset del proceso KDD ya analizados donde aparecen el número de ataques terroristas, el número de muertos y fallecidos, además de variables como el desempleo o la población, incluida la localización de los atentados.

Se usará análisis de regresión con el fin de capturar la relación entre características y número de fallecidos.

Para empezar, cargaremos el Dataset en Google Colab además de las librerías análogas a los apartados anteriores. A continuación, procederemos a describir el conjunto de Dataset cargado como se muestra en la tabla adjunta.

✓ [8] Terrorismo.describe()

	Latitude	Longitude	Multiple	Success	Nkill	Nwound	unemployed	Population
count	414638.000000	414638.000000	414638.000000	414638.000000	414638.000000	414638.000000	412656.000000	4.126560e+05
mean	48.209727	4.590996	0.173624	0.775211	0.348087	2.026476	1396.742902	4.823145e+07
std	6.523962	10.963521	0.378786	0.417444	3.590343	21.226690	832.046010	2.585161e+07
min	-22.336000	-21.895210	0.000000	0.000000	0.000000	0.000000	5.751000	3.180440e+05
25%	42.636510	-5.955833	0.000000	1.000000	0.000000	0.000000	284.221000	1.112134e+07
50%	50.557887	2.148641	0.000000	1.000000	0.000000	0.000000	1732.593000	6.328514e+07
75%	54.446968	11.974560	0.000000	1.000000	0.000000	0.000000	2094.850000	6.643555e+07
max	67.143672	166.912000	1.000000	1.000000	344.000000	727.000000	2565.168000	8.316087e+07

Tabla 5. Descripción del Dataset Terrorismo_ya_KDD

A continuación, estudiaremos la estadística para intentar vislumbrar alguna conclusión. Los histogramas agrupan los datos en ubicaciones y proporcionan un recuento de los números de observaciones identificando la distribución de los datos, permitiéndonos a su vez identificación valores atípicos.

```
[10] #0.2.1 Univariate Histograms
#
Terrorismo.hist()
plt.show()
```

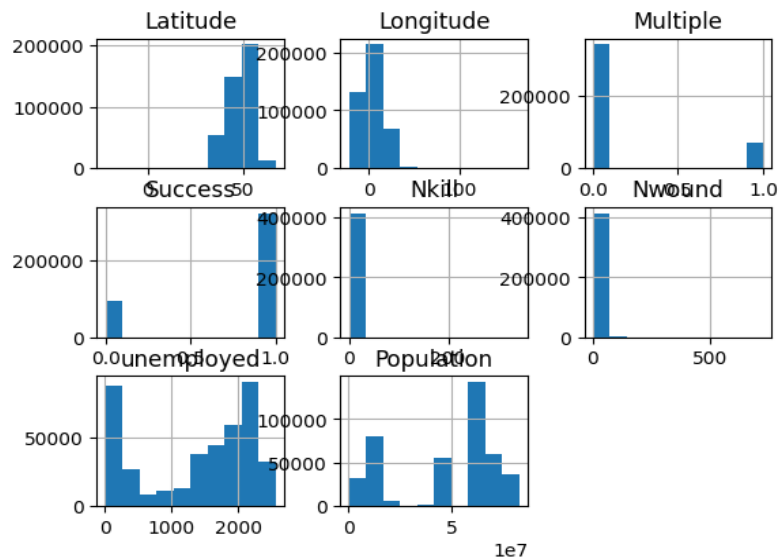


Ilustración 54. Histograma del Dataset

Posteriormente, procedemos a ver mediante los diagramas de densidad la distribución de cada atributo de una manera rápida.

```
[11] #0.2.2 Density Plots
Terrorismo.plot(kind='density', subplots=True, layout=(3,3), sharex=False)
plt.show()
```

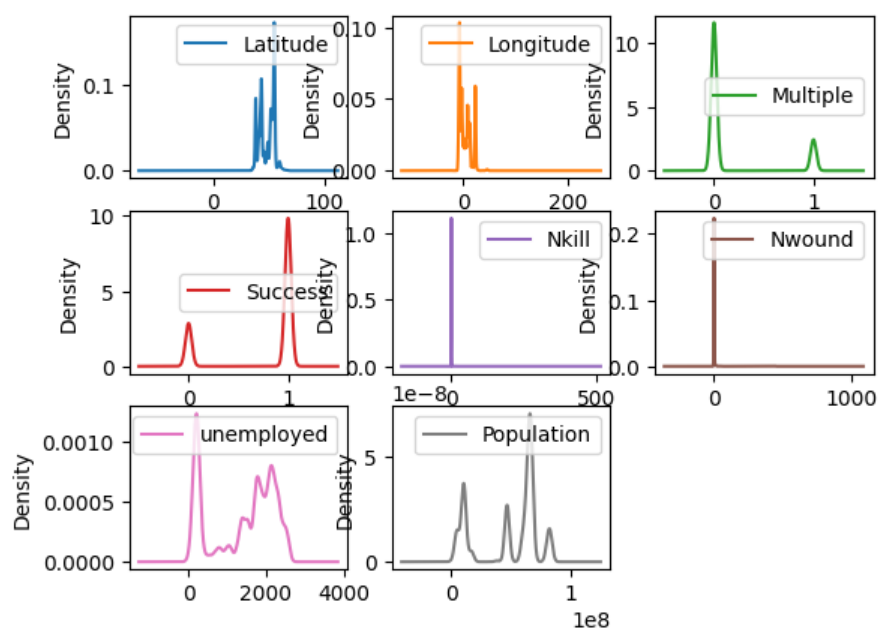


Ilustración 55. Diagrama de densidad.

A continuación, mediante un gráfico de dispersión, mostraremos las relaciones entre dos variables. Las gráficas de dispersión son útiles para detección de relaciones estructuradas entre variables.

```
[12] #0.2.3 Scatterplot matrix
from pandas.plotting import scatter_matrix
scatter_matrix(Terrorismo)
plt.show()
```

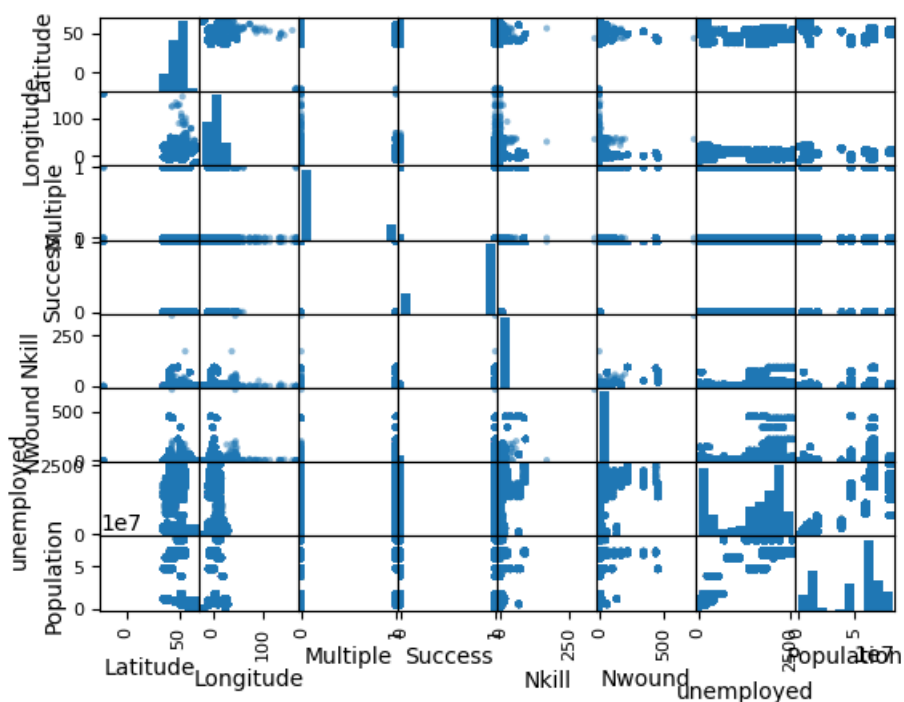


Ilustración 56. Gráfico de dispersión

Mediante los diagramas de correlación indicamos como de correlacionadas están los atributos predictores con el valor a predecir, como podemos en la siguiente ilustración.

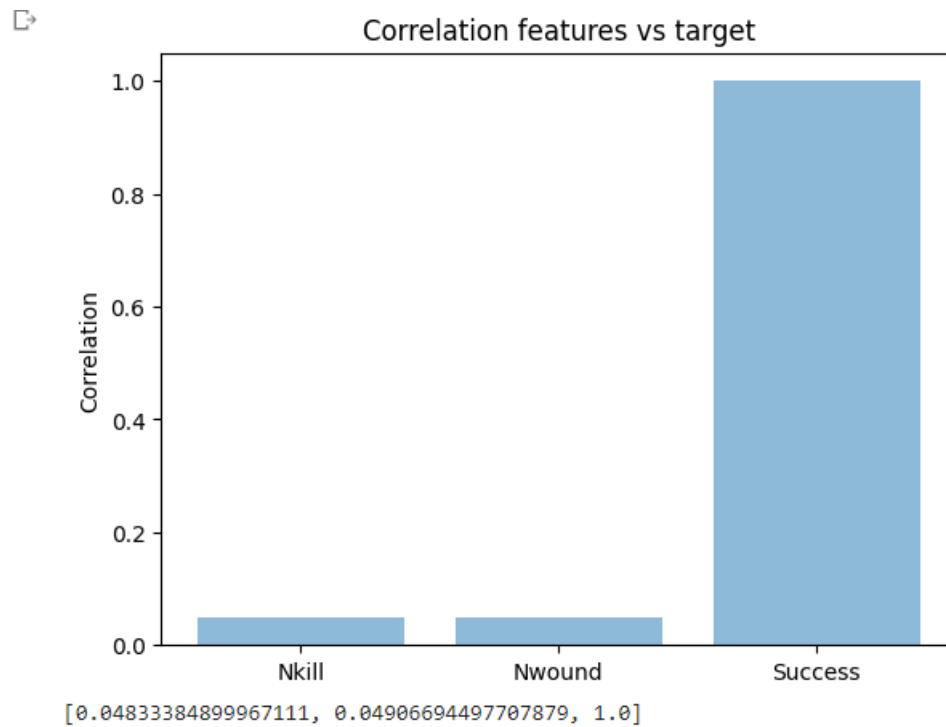


Ilustración 57. Diagrama de correlación

A continuación, veamos cual es el modelo predictivo para una sola variable.

```
[28] plt.legend()
      plt.show()

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning:
X does not have valid feature names, but DecisionTreeRegressor was fitted with feature names
```

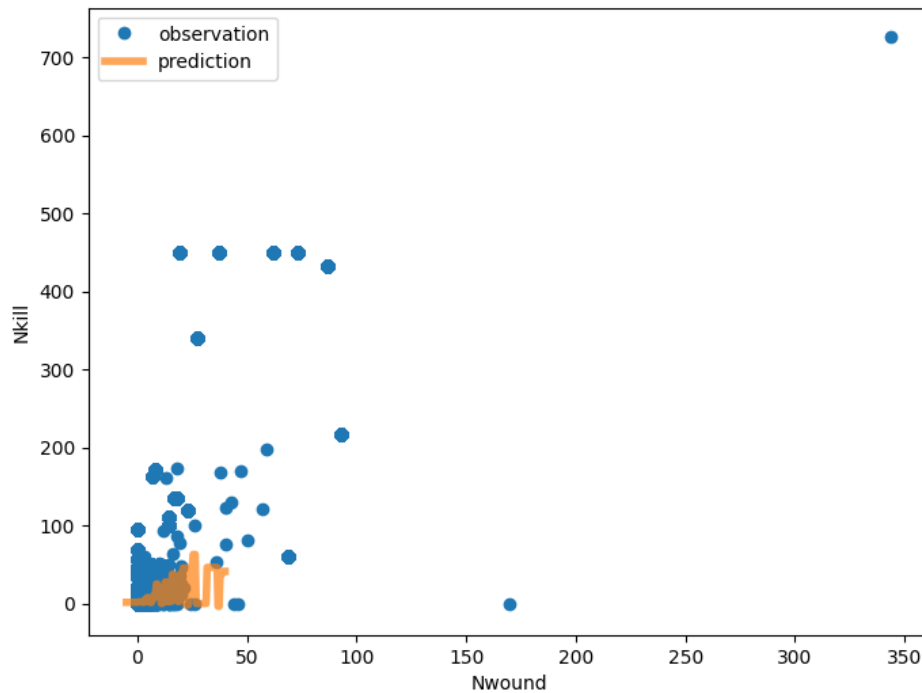


Ilustración 58. Modelo predictivo para una sola variable

Con posterioridad, procedemos a construir un modelo en base a dos variables, en nuestro caso las relacionadas con el número de muertos y el número de heridos.

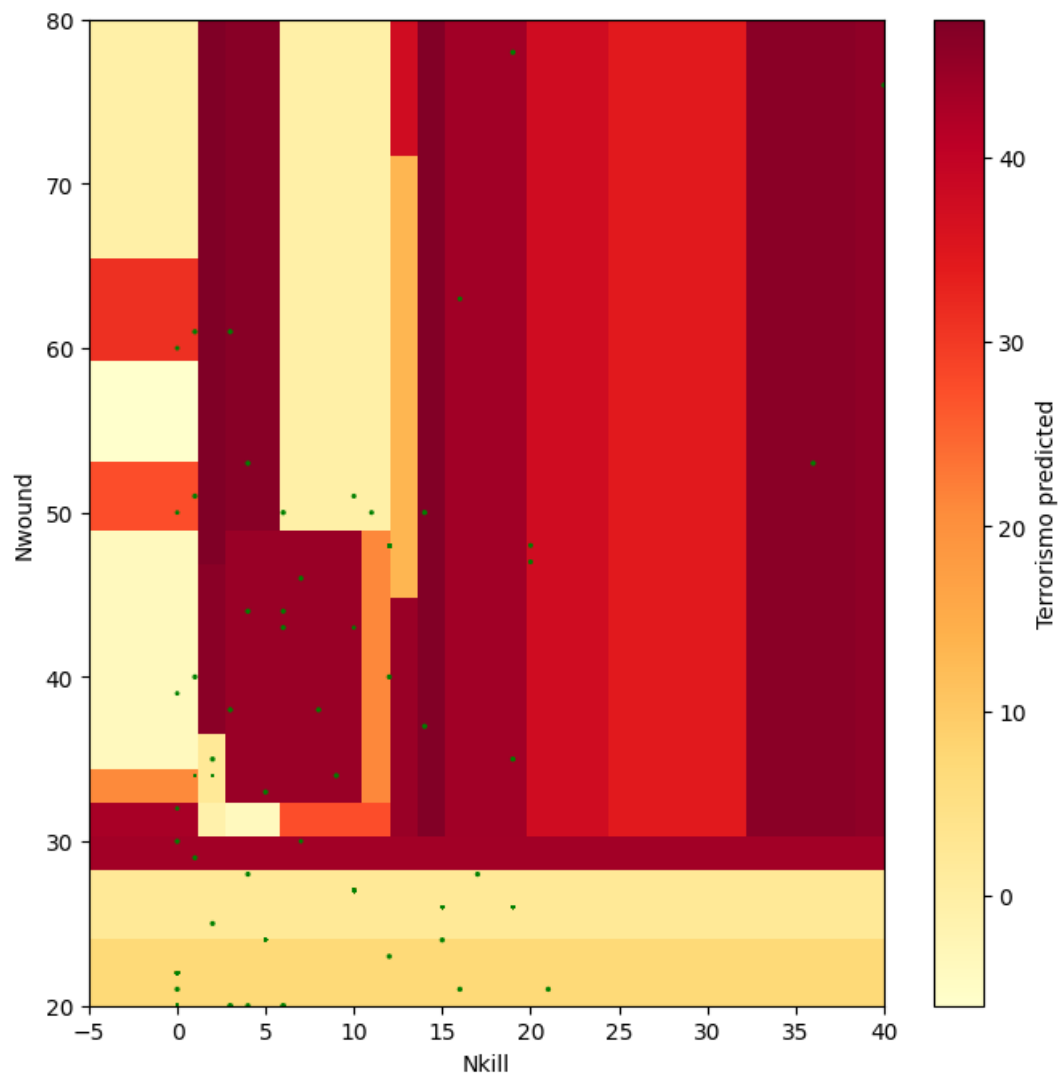


Ilustración 59. Visualización para un modelo predictivo en base a dos variables.

A continuación, procedemos a construir el modelo con las tres variables y analizamos la importancia de las características según el árbol de decisión.

```
[33] print ('Feature Relevances')
      pd.DataFrame({'Attributes': features ,
                    'Decision Tree': regressor.feature_importances_})

/usr/local/lib/python3.7/dist-packages/sklearn/tree/_classes.py:370: FutureWarning:
Criterion 'mae' was deprecated in v1.0 and will be removed in version 1.2. Use `criterion='absolute_error'` which is equivalent.
```

Feature Relevances		
	Attributes	Decision Tree
0	Nkill	0.260995
1	Nwound	0.304117
2	Longitude	0.434888

Ilustración 60. Modelo con las tres variables a analizar

Antes de proceder a crear los modelos para predecir vamos a dividir nuestro conjunto de datos en dos partes una para entrenamiento (tuning) y otra para test.

```
✓ [104] target_col=['Success']

✓ [106] X = Terrorismo.drop(['Success'], axis=1)

      y = Terrorismo['Success']

[ ]

✓ [107] from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)

✓ [108] X_train.dtypes

Country      object
City         object
Latitude     float64
Longitude    float64
Multiple     int64
Attacktype   object
Target_type  object
Group        object
Weapon_type  object
Nkill        int64
Nwound       int64
unemployed   float64
Population   float64
dtype: object
```

Ilustración 61. División de datos

Las variables elegidas para realizar la predicción del Dataset ya depurado mediante KDD para el estudio han resultado no ser lo suficientemente adecuados a la muestra de los resultados expuestos en el código adjunto.

1.2.5. Visualización en PowerBI de resultados.

En una primera instancia procedo a cargar el Dataset analizado en el apartado anterior para visualizar el objeto de estudio y proceder a continuación a compararlo con los datos extraídos del estudio de redes sociales en Twitter.

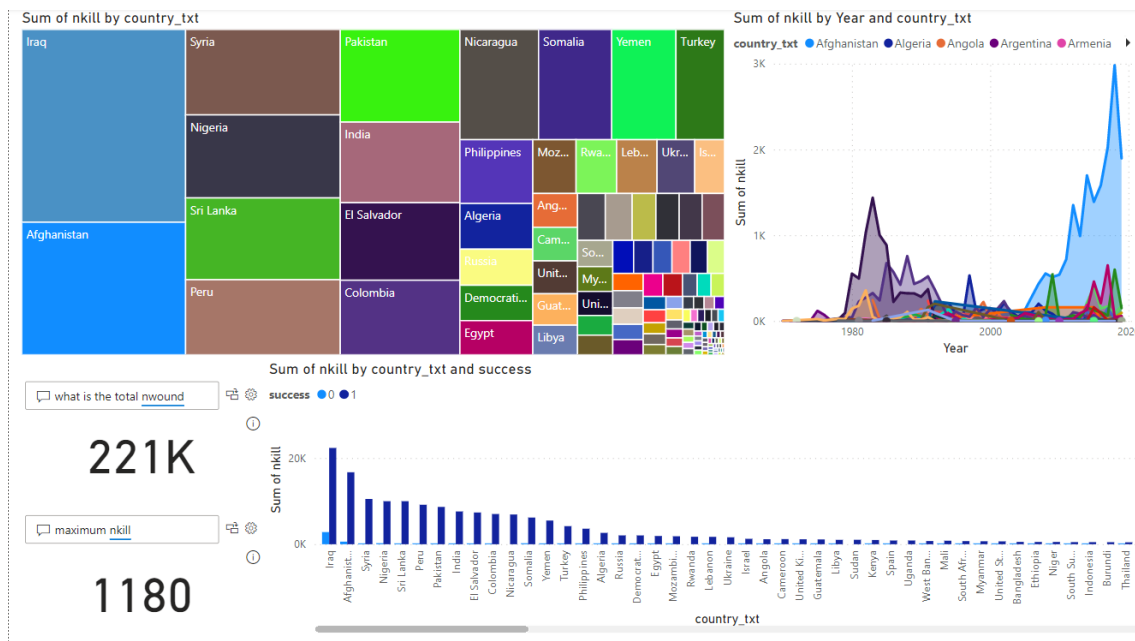


Ilustración 62. Visualización del Dataset especializado en terrorismo.

En segundo lugar, he procedido a cargar los datos una vez depurados provenientes de mongoDB.

1.3. Resultados

En relación con el primer objetivo propuesto, se ha partido de un total de 17898 tweets obtenidos de las cuentas objetivo-señaladas con anterioridad. A su vez, se han añadido un total de 2477 tweets con el método de palabras clave indicadas con anterioridad.

Todo ello se ha analizado siguiendo los conocimientos de la asignatura de Fundamentos de Big data en MongoDB y su análisis correspondiente en Mongo Compass.

A continuación, se muestra las consultas que he realizado en MongoDB:

```
[{$project: {  
  _id:0  
}}, {$count: 'id:0'}]
```

Ilustración 63. Número total de Tweets

Tras hacer la operación me da id:0: 17898 tweets.

```
[{$group: {  
  _id: "$user.screen_name",  
  suma: {$sum:1}  
}}]
```

Ilustración 64. Total de Tweets de cada cuenta

Agrupamos los documentos mediante \$group, dándole la condición de por \$user.screen_name para que nos lo sume por cuentas de Twitter.

```
[{$group: {  
  _id: "$lang",  
  lang: {  
    $sum: 1  
  }  
}}, {$sort: {  
  lang: -1  
}}, {$limit: 5}]
```

Ilustración 65. Ranking de los cinco primeros idiomas en los que se ha escrito más tweets

Agrupamos los documentos mediante \$group, dándole la condición que nos lo sume por separado, a continuación, los desplegamos de mayor a menor y finalmente lo limitamos a que nos muestre los cinco primeros.

```
[{$unwind: {
  path: "$extended_entities.media",
  includeArrayIndex: 'string',
  preserveNullAndEmptyArrays:
false
}}, {$group: {
  _id: "extended_entities_media",
  extended_entities: {
    $sum: 1

  }
}}]
```

Ilustración 66. Contar Tweets por tipo de media que lleva incrustado

En primer lugar, con \$unwind separamos los campos de los documentos para especificar que campo es el que buscamos. A continuación, los agrupamos con \$group contando los que nos interesa, en este caso los media agregados.

```
[{$group: {
  _id: "$user.screen_name",
  followers: {
    $first: "user.followers_count"
  }
}}, {$sort: {
  followers: 1
```

Ilustración 67. Ordenar las cuentas de mayor influencia a menor (de acuerdo con el número de seguidores que posee cada una

Agrupamos los documentos mediante \$group, dándole la condición de por \$user.screen_name. A continuación, le decimos que nos cuente en el campo followers el número de seguidores.

Por último, le decimos que no los ordene de mayor a menor número de seguidores.

```
[{$unwind: {  
  path: "$entities.hashtags",  
  preserveNullAndEmptyArrays:  
  false  
}}, {$group: {  
  _id: "$entities.hashtags.text",  
  hashtags:{  
    $sum: 1  
  }  
}}, {$sort: {  
  hashtags:-1  
}}, {$limit: 20}]
```

Ilustración 68. Realizar una consulta en Mongo DB para listar los 20 hashtags más utilizados

En primer lugar, con \$unwind separamos los campos de los documentos para especificar que campo es el que buscamos. A continuación, los agrupamos con \$group contando los que nos interesa, en este caso los hashtags. Luego le decimos con \$sort que nos lo ordene de mayor a menor y por último lo limitamos la salida a 20.

Una vez analizados el Dataset he procedido a exportarlo en un formato CSV para a continuación continuar con el análisis en PowerBI.

La mayor parte de los tweets no están geo-referenciados y al tratarse de información sobre ataques terroristas no existe correlación entre el origen de las cuentas que hablan sobre atentados terroristas con el lugar donde efectivamente se han llevado a cabo.

En cuanto al segundo objetivo propuesto, he procedido a utilizar Google Colab, para mediante el lenguaje de programación Python3 proceder a realizar el estudio propuesto.

Los resultados muestran, como se puede apreciar en la ilustración 29, técnicas de ataque, que los atentados se dirigen principalmente sobre los objetivos militares en primer lugar, seguidos por civiles y sus propiedades. El método principal utilizado de ataque son los proyectiles tipo morteros, granadas en primer lugar, seguidos por munición de fusilería y desde vehículo. En cuanto a la región del mundo donde se producen mayor número de atentados se concentran en oriente medio.

En relación con la plataforma Google sheet, he querido focalizarme en el número de víctimas por año, llegando a la conclusión, como se puede apreciar en la ilustración 30, fallecidos y heridos en atentados, que cada vez más los atentados producen más heridos que fallecidos, provocando el objetivo de las organizaciones terroristas de conseguir provocar terror.

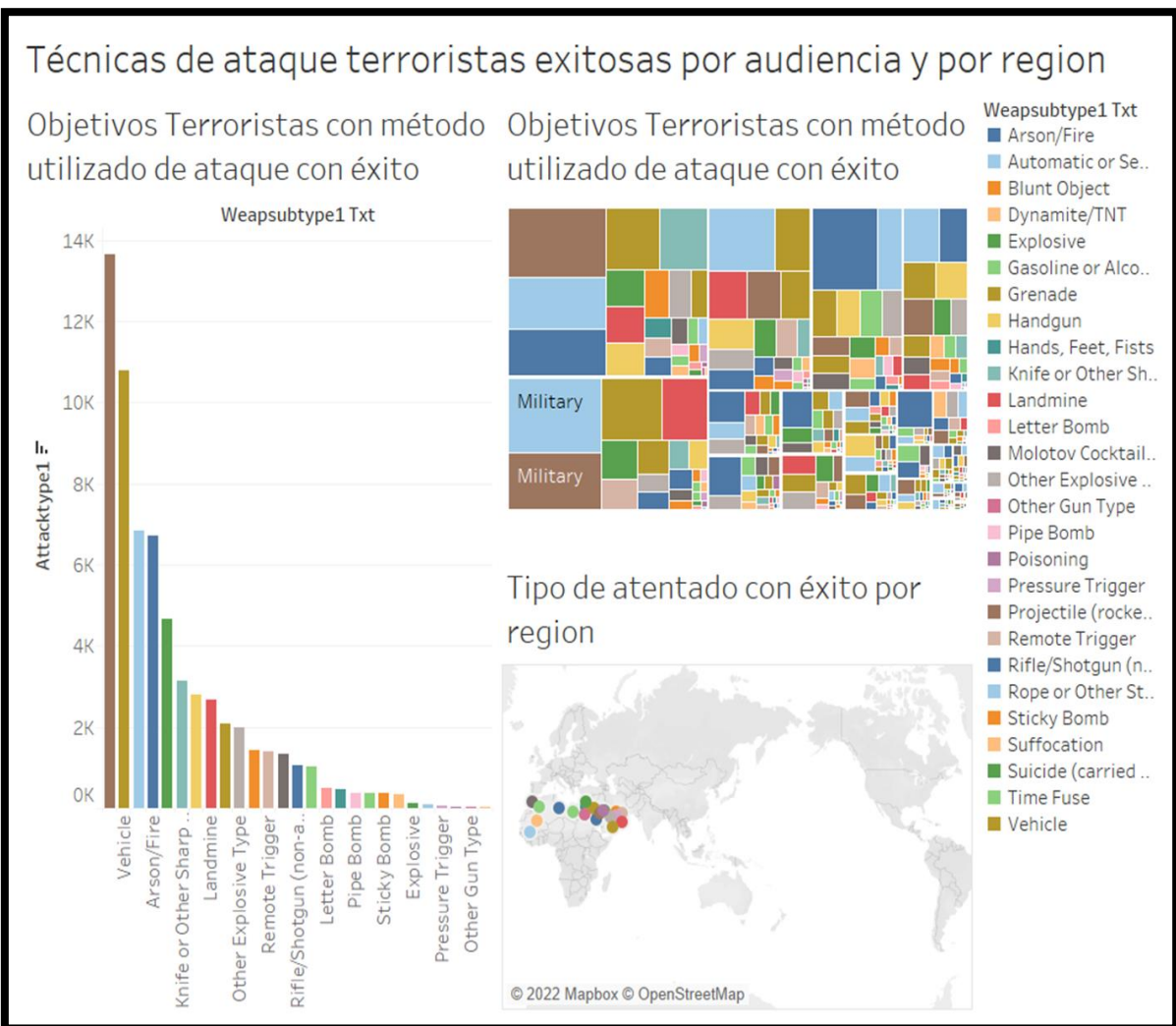


Ilustración 69. Técnicas de ataque terroristas exitosas

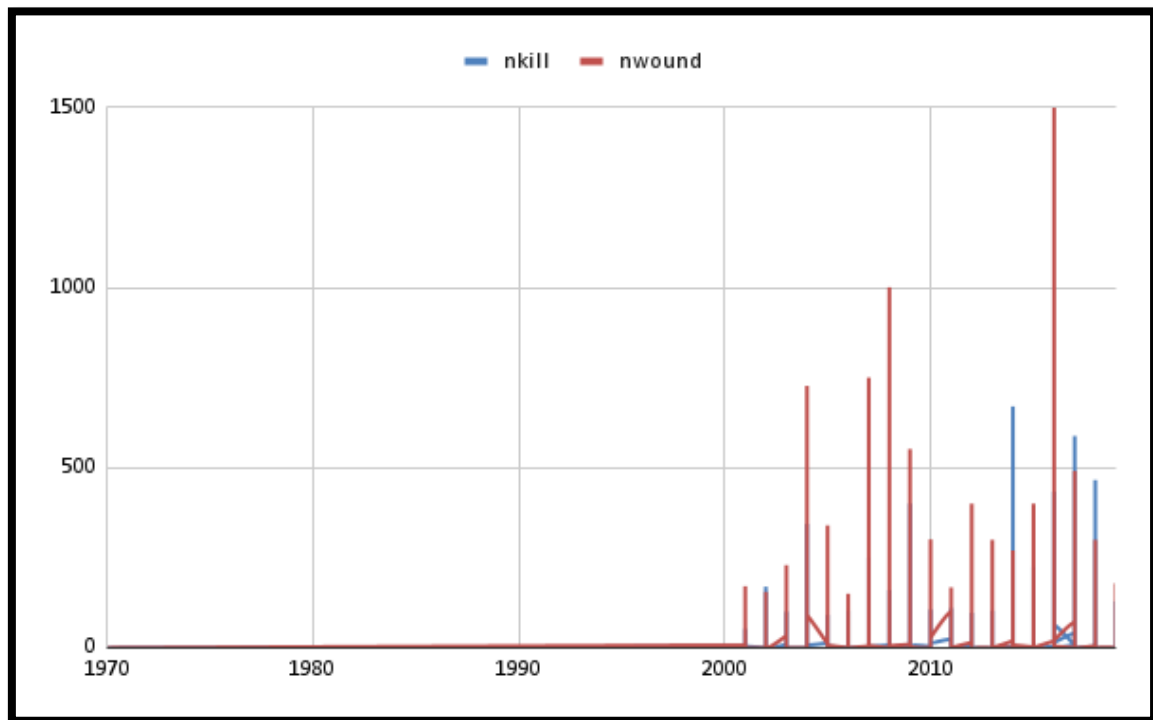


Ilustración 70. Fallecidos y heridos en atentados

En la ilustración 69, resumen de atentados por tipo, lugar y objetivo, mediante la herramienta R Visual Studio, muestra que la región de oriente medio y norte de África, junto con Asia meridional son de lejos, las regiones más conflictivas en cuanto a atentados terroristas. En relación con el método de ataque, se constata que las explosiones, tanto por bombas como por artefactos improvisados, seguido de acciones de fusilería, son los más profusos elegidos por las organizaciones terroristas. Por último, los objetivos son en primer lugar civiles, seguidos de Fuerzas armadas y fuerzas de seguridad.

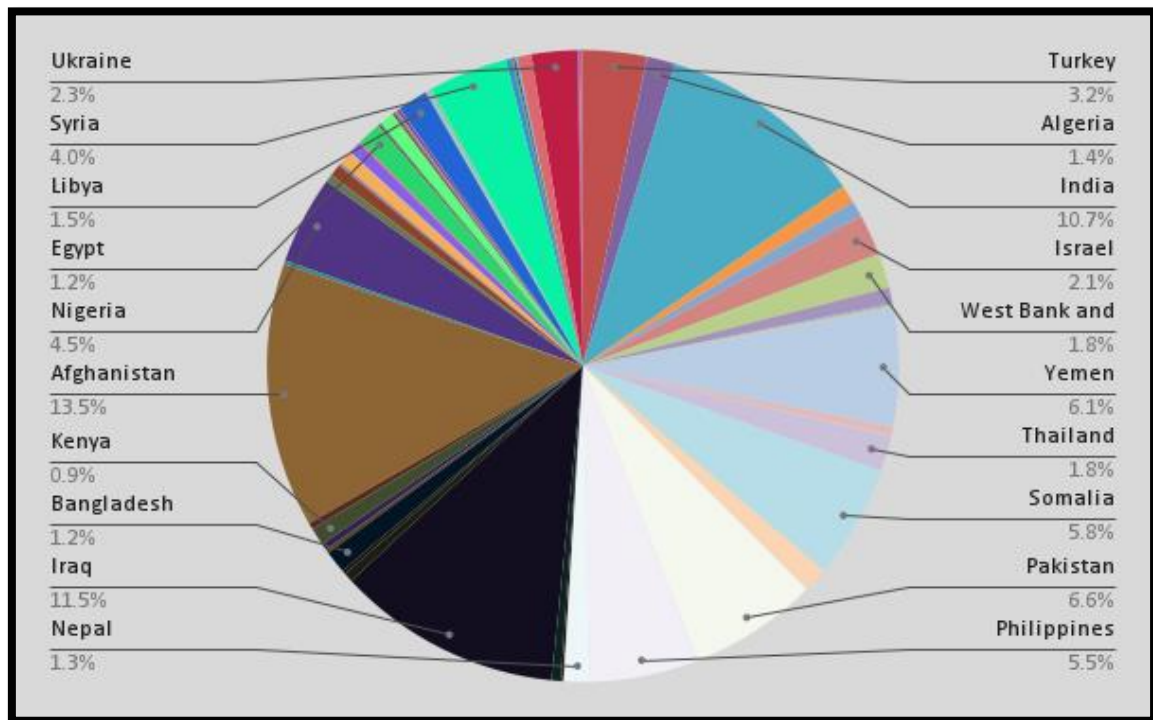


Ilustración 71. Atentados por Regiones

2. Conclusión y trabajos futuros

En relación con las premisas objeto de estudio, se han llegado a las siguientes conclusiones:

- Los Dataset obtenidos de Twitter nos dan un sentido en cuanto al impacto emocional de los atentados terroristas.
- En cambio, estos datos no son extrapolables en cuanto a la identificación de orígenes de atentados terroristas.
- En relación con los Dataset especializados, se ha precedido a realizarles su limpieza, transformación y análisis siguiendo la metodología KDD. Se ha adjuntado tanto Dataset provenientes de páginas especializadas como Dataset creados a partir de Octoparse.
- Tras haber realizado el estudio con la herramienta RStudio versión 4.1.1. la primera conclusión es que muchas de los paquetes que funcionan en unas versiones no funcionan en otras y es necesario descargarse los paquetes necesarios para poder trabajar con las librerías de estadística.

La segunda conclusión es relativa al Dataset seleccionado. Es importante seleccionar las variables clave que puedan aportar valor añadido a los estudios predictivos, afín de no tener resultados no concluyentes o débilmente justificativos como ha sido el caso de mis estudios.

En relación con los resultados obtenidos con respecto al objeto del estudio podemos concluir que no existe al menos a tenor a los estudios realizados una correlación entre el acceso a las tecnologías de la información y el mayor éxito de los atentados terroristas. Existen otros condicionantes como la adaptación al medio y el entorno donde se lleven a cabo que no son objeto de estudio. Será necesario particularizar más en un área determinada para un tipo de organización específica y compararlas unas con otras utilizando estas herramientas de análisis.

Para trabajos futuros adquiriré mayores conocimientos en la herramienta RStudio y con la ayuda de los conocimientos adquiridos en la asignatura de Estadística avanzada los pondré en práctica en mi trabajo relacionado con el asesoramiento de seguridad.

La limitación principal a la que me he encontrado ha sido el compaginar la vida laboral y privada con el estudio realizado.

3. Referencias

Corporation, R. (2022). *RAND NATIONAL SECURITY RESEARCH DIVISION*.
Obtenido de RAND Database of Worldwide Terrorism Incidents:
<https://www.rand.org/nsrd/projects/terrorism-incidents.html>

Developer Portal. (2022). Obtenido de <https://developer.twitter.com/>

Grolemund., H. W. (2020). *R for Data Science. Import, Tidy, Transform, visualize and model data*. Sebastopol, California: Ed. O'Reilly.

Guido., A. C. (2018). Supervised and Unsupervised Learning. En A. C. Guido., *Introduction to Machine learning with Python. A guide for Data Scientists*. (págs. 27-247). Ed. O'Reilly.

Gutiérrez., Á. P. (s.f.). *Python Paso a paso*. . Ed. Ra-Ma.

Lachev., T. (2022). *Applied Microsoft Power BI. Bring your data to life*. Ed. Prologika Press.

Maryland, U. o. (Mayo de 2022). *GTD Global Terrorism Database*. Obtenido de
<https://www.start.umd.edu/gtd>: <https://www.start.umd.edu/gtd>

Mitchel., R. (2021). Web Crawling models. En R. Mitchel., *Web Scrapping with Python. Collecting more data from the modern web*. (págs. 49-94). Ed. O'Reilly.

Mongodb Atlas. (2022). Obtenido de <https://cloud.mongodb.com/>

Organization for economic, cooperation and development. (2022). Obtenido de
OECD.Stats: <https://stats.oecd.org/>

Shannon Bradshaw, E. B. (2019). Creating, Updating, and Deleting Documents;
Querying. En E. B. Shannon Bradshaw, *MongoDB. The Definitive Guide. Powerful and Scalable Data Storage*. (págs. 29-67). Ed. O'Reilly.

Valenzuela., J. S. (s.f.). *Python Aplicaciones prácticas*. Ed. Ra-Ma.

Apéndice I. Análisis descriptivo

#Carga del Dataset

```
Ataques_terroristas <- read.csv2  
("C:/Users/milom/Desktop/Ataques_terroristas.csv")
```

```
View (Ataques_terroristas)
```

#Convertir variables tipo carácter a tipo factor

```
character_vars <- lapply(Ataques_ter, class) == "character"
```

```
Ataques_ter[, character_vars] <- lapply(Ataques_ter[, character_vars], as.factor)
```

#Convertir variables tipo numéricas a tipo factor

```
varNumAtaques_terr <- Ataques_ter[, sapply(Ataques_ter, is.numeric)]
```

#Resumen estadístico de las variables numéricas

```
summary(varNumAtaques_terr)
```

#Variables categóricas del Dataset

```
varCatAtaques_ter <- Ataques_ter[, sapply(Ataques_ter, is.factor)]
```

#Lista de las variables categóricas y su frecuencia

```
summary(varCatAtaques_ter)
```

```
for (i in 1:ncol(varNumAtaques_terr)) {varNumAtaques_terr  
[is.na(varNumAtaques_terr[,i]), i] <- mean (varNumAtaques_terr[,i], na.rm = TRUE)  
}
```

```
matriz_corr <- cor(varNumAtaques_terr[,])
```

```
>print(matriz_corr)
```

#Matriz de correlación de las variables numéricas

```
matriz_corr <- cor(varNumAtaques_terr[,])
```

```
print(matriz_corr)
```

#Columnas con valores superiores al umbral, superior al 0.75

```
correl_alta <- findCorrelation(matriz_corr, cutoff=0.75)
```

#Imprimimos las columnas con un valor superior a 0.75 de correlación

```
print(correl_alta)
```

#Correlación en gráficos entre Year y nkill

```
ggplot(varNumAtaques_terr, aes(x=varNumAtaques_terr$success,  
y=varNumAtaques_terr$Year)) +
```

```
  geom_point(shape=1) +
```

```
  geom_smooth(method=lm)
```

```
boxplot(varNumAtaques_terr$nkill ~ varNumAtaques_terr$Year,
```

```
  data = varNumAtaques_terr,
```

```
  main = "éxito con fallecidos con muertos",
```

```
  xlab = "Periodo de estudio mundial",
```

```
  ylab = "Número de muertos",
```

```
  col = c("pink", "pink1", "pink2", "pink3"))
```

#Correlación en gráficos entre Year y nwound

```
ggplot(varNumAtaques_terr, aes(x=varNumAtaques_terr$success,  
y=varNumAtaques_terr$Year)) +
```

```
  geom_point(shape=1) +
```

```
  geom_smooth(method=lm)
```

```
boxplot(varNumAtaques_terr$nwound ~ varNumAtaques_terr$Year,
```

```
  data = varNumAtaques_terr,
```

```
  main = "éxito con heridos",
```

```
  xlab = "Periodo de estudio mundial",
```

```
  ylab = "Número de heridos",
```

```
  col = c("pink", "pink1", "pink2", "pink3"))
```

#Correlación en gráficos entre nkill y nwound

```
ggplot(varNumAtaques_terr, aes(x=varNumAtaques_terr$nkill,  
y=varNumAtaques_terr$nwound)) +
```

```
  geom_point(shape=100) +
```

```
geom_smooth(method=lm)

boxplot(varNumAtaques_terr$nkill ~ varNumAtaques_terr$Year(2000:2019),

data = varNumAtaques_terr,

main = "fallecidos y muertos a lo largo del tiempo",

xlab = "Numero de fallecidos",

ylab = "Numero de heridos",

col = c("pink", "pink1", "pink2", "pink3"))
```

#Numero de nulos por variable

```
sapply(Notas_estudiantes, function(x) sum(is.na(x)))
```

#Caso 1. Numero de fallecidos basado en el éxito del ataque.

```
lm.fit <- lm(nkill ~ success,
```

```
data=Ataques_ter)
```

#Datos estadísticos del modelo

```
summary(lm.fit)
```

```
data1 <-
```

```
data.frame(success=1,)
```

```
#la predicción una vez los condicionantes
```

```
predict(lm.fit,data1)
```

#Resultado de la predicción

```
4.152702
```

#Caso 2. Número de heridos basado en el éxito del ataque.

```
lm.fit <- lm(nwound ~ success, data=Ataques_ter)
```

#Datos estadísticos del modelo

```
summary(lm.fit)
```

```
data1 <-
```

```
data.frame(success=1, Year=2019)
```

```
#la predicción una vez los condicionantes
```



```
predict(lm.fit,data1)
```

```
#Resultado de la predicción
```

```
5.09903
```



Anexo I. Código selector de propiedades para KDD

Anexo II. Tweets a MongoDB

Anexo III. Búsqueda de Tweets

Anexos IV. Proceso KDD Dataset Terrorismo