Logotipo

Descripción generada automáticamente

**UNIVERSIDAD EUROPEA DE MADRID  
ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO**

**MÁSTER UNIVERSITARIO EN**

**ANÁLISIS DE DATOS MASIVOS (BIG DATA)**

TRABAJO FIN DE MÁSTER

**Análisis y Comparativa de las Listas de Sancionados de la Unión Europea y Estados Unidos**

**Alberto Alonso Chana**

**CURSO 2021-2022**

**TÍTULO**: Análisis y Comparativa de las Listas de Sancionados de la Unión Europea y Estados Unidos

**AUTOR**: Alberto Alonso Chana

**TITULACIÓN**: Máster Universitario en Análisis de Datos Masivos (Big Data)

**DIRECTOR DEL PROYECTO**: Jorge Luis Hita y José Javier Ruiz Cobo

**FECHA**: septiembre de 2022

# RESUMEN

El resumen tiene entre 150-250 palabras. Resumir consiste en ofrecer información exhaustiva sobre cómo, dónde, cuándo y por qué se aplica el proyecto. Se realiza al finalizar el trabajo.

El resumen

* Establece el tema tratado en el proyecto
* Resume los problemas planteados y las aportaciones más importantes del proyecto
* Resume las conclusiones más importantes del trabajo realizado

El resumen NO

* Da una información genérica
* Explica porque se realiza ese trabajo
* Se refiere a datos aportados en el texto del proyecto.

**Palabras clave:** hasta un máximo de 6

# ABSTRACT

Resumen en inglés.

**Key words:** Palabras clave en inglés

**AGRADECIMIENTOS**

En ocasiones se incluye este apartado para agradecer a aquellos que han ofrecido su ayuda en el desarrollo del trabajo, ya sea técnica o de otro tipo.

Índice

[RESUMEN 4](#_Toc112159369)

[ABSTRACT 4](#_Toc112159370)

[Capítulo 1. INTRODUCCIÓN 9](#_Toc112159371)

[1.1 Descripción del Problema 9](#_Toc112159372)

[1.1.1 Contexto Histórico 9](#_Toc112159373)

[1.1.2 Actualidad 9](#_Toc112159374)

[1.2 Motivación 9](#_Toc112159375)

[1.3 Objetivos del proyecto (pendiente) 10](#_Toc112159376)

[1.4 Estructura del proyecto (dejar para el final) 10](#_Toc112159377)

[Capítulo 2. Memoria Técnica 12](#_Toc112159378)

[2.1 Proceso ETL 12](#_Toc112159379)

[2.1.1 Extracción de Datos 12](#_Toc112159380)

[2.1.2 Transformación de Datos 16](#_Toc112159381)

[2.1.3 Concatenar Datos 21](#_Toc112159382)

[Capítulo 3. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO 23](#_Toc112159383)

[ANEXOS 24](#_Toc112159384)

[PRESUPUESTO Y TIEMPO 25](#_Toc112159385)

[BIBLIOGRAFÍA 26](#_Toc112159386)

Índice de Figuras

Este índice ha de existir tan sólo en el caso de que tengas más de 4 apoyos visuales (gráficos, figuras, ilustraciones, fotos, etc.) en tu memoria del proyecto.

Recuerda esta sección es optativa.

Índice de Tablas

Este índice ha de existir tan sólo en el caso de que tengas más de 4 tablas de datos en tu memoria del proyecto.

Recuerda esta sección es optativa.

# INTRODUCCIÓN

## Descripción del Problema

Las sanciones internacionales son medidas coercitivas que los Gobiernos y las entidades supranacionales aplican contra Estados, empresas o individuos que suponen una amenaza para la seguridad global. Abarcan restricciones económicas, diplomáticas, comerciales, militares e incluso deportivas, y sus objetivos son debilitar y reducir la capacidad de maniobra de estos actores sin hacer uso de la fuerza.

### Contexto Histórico

Las sanciones se han utilizado como herramienta diplomática desde la Antigüedad. La primera conocida fue el embargo comercial de Atenas a Mégara en el 432 a. C. Posteriormente, durante el siglo XIX, el uso de sanciones se intensificó a partir del bloqueo continental que Napoleón decretó en 1806 con el fin de excluir cualquier intercambio con el Reino Unido, su rival por el control de Europa. No obstante, el auge definitivo de las sanciones internacionales llegó con el final de la Primera Guerra Mundial. Durante este período, la Sociedad de Naciones (SDN), predecesora de la ONU, fue fundamental en el intento de imponer medidas para boicotear a los países agresores. En los años noventa, la invasión de Kuwait por el Irak de Sadam Huseín y las guerras yugoslavas llevaron a un nuevo repunte de las sanciones internacionales.

El uso de las sanciones se ha consolidado en la comunidad internacional en las últimas décadas. Su adopción ha trascendido de la ONU a la **Unión Europea**, en su Política Exterior y de Seguridad Común, o a la Organización para la Seguridad y Cooperación Europea (OSCE), entre otras. A nivel estatal, **Estados Unidos** es el país que más ha utilizado este mecanismo, a través de leyes y órdenes ejecutivas. La Ley para la Democracia Cubana de 1992, por ejemplo, reguló el embargo a la isla [1].

### Actualidad

Ya en 2022, Tanto Washington como Bruselas decidieron apostar por estas medidas para responder a la invasión rusa de Ucrania. Las sanciones, que incluyen el cierre del espacio aéreo europeo a Rusia o su desconexión parcial del sistema Swift de comunicaciones financieras, pretenden aislar la economía rusa. Las restricciones también alcanzan al ámbito deportivo, como el veto a Rusia en el Mundial de fútbol de Catar [1].

## Motivación

Para un mejor control y seguimiento de las entidades sancionadas, tanto la Unión Europea como Estados Unidos elaboran unas listas que ponen a disposición de la ciudadanía a través de internet. Las personas, grupos y compañías que aparecen en estas listas tienen sus activos bloqueados y generalmente se prohíbe a las empresas europeas y estadounidenses tratar con ellos.

Los avances en análisis y procesamiento de datos suponen un nuevo paradigma en la forma en que estas sanciones son aplicadas:

* Mediante algoritmos de Machine Learning de aprendizaje no supervisado, se pueden conocer patrones ocultos de los datos que de otra manera pasarían desapercibidos. El aprendizaje supervisado permite conocer la información faltante de algunos registros a partir de los registros en los que sí existe esa información.
* Las herramientas enfocadas a extraer información a partir de los datos mediante la generación de dashboards son cada vez más avanzadas. Estas son capaces de ingerir grandes cantidades de datos y mantenerlos siempre actualizados, además de presentarlos de forma dinámica e interconectada.

Estos avances benefician a las empresas susceptibles de tener entre sus clientes a entidades sancionadas. Existen diversos motivos por los que una compañía querría conocer cuáles son sus clientes sancionados: Mantener la reputación, evitar el pago de multas, cumplir con la normativa para la no financiación del terrorismo, etc. [2]

Este trabajo tiene como objetivo tanto realizar un proceso de investigación de los datos en busca de relaciones e informaciones desconocidas, como de generar valor a las empresas que pudieran beneficiarse del seguimiento de los sancionados mediante dashboards generados con herramientas de visualización accesibles y fáciles de usar. Esto con el objetivo de que cualquier miembro de la compañía en cualquier departamento, incluido el equipo directivo, pueda disponer fácilmente de la información e interpretarla sin necesidad de ser experto en el campo.

Se puede ampliar el dashbpard con un presupuesto de lo que se ahorra la compañía invirtiendo en la herramienta para detectar sancionados entre sus clientes (buscar información sobre leyes y multas respectivas a este campo)

## Objetivos del proyecto (pendiente)

¿Cuáles son los objetivos que persigo? Es bueno desglosarlo en un objetivo general del que se derivan varios objetivos específicos

## Estructura del proyecto (dejar para el final)

Una buena organización de tu memoria ayuda a su lectura y refleja mejor la metodología de trabajo usada. Normalmente consta de objetivos, metodología, resultados y, conclusiones y futuras líneas de trabajo. Cuenta aquí más o menos en dos líneas por cada capítulo, el contenido de dicho capítulo.

NO OLVIDES:

* Numerar las páginas.
* Numerar las figuras
* Numerar las tablas
* Justificar los márgenes.
* Destacar los apartados dentro de cada capítulo
* Cuida el FORMATO: usa siempre el mismo tipo de letra, el mismo tamaño para el texto, el mismo tamaño para los apartados, el mismo tamaño para los títulos de los capítulos,…
* Si te dan una guía de estilo con tamaños de letra, interlineados, márgenes etc, **respétala**
* Destaca lo que creas oportuno utilizando las **negritas** y las *cursivas*.

# Memoria Técnica

Los ficheros, scripts, tablas, dashboards y demás documentos empleados en este trabajo, incluida esta memoria, están publicados en el siguiente repositorio de Github:

<https://github.com/Chanalber/TFM.git>

A continuación, se exponen las tareas realizadas durante el desarrollo del trabajo y la forma en que fueron abordadas. En esta sección se tratarán los problemas, cambios e imprevistos en la consecución de estos objetivos para finalmente presentar los resultados a los que se ha llegado.

## Proceso ETL

El proceso ETL (Extract, Transform, Load) engloba el conjunto de técnicas aplicadas sobre los datos en crudo con objeto de obtener un dataset depurado y listo para aportar valor mediante diversos métodos: entrenamiento de modelos de Machine Learning, generación de visualizaciones de Inteligencia Empresarial, etc.

### Extracción de Datos

El notebook de Jupyter para código Python ha sido la aplicación utilizada en la carga de los datos de las listas de sancionados. Se generó un script de nombre *01\_extraer\_datos.ipynb* en la ubicación *TFM/Código y datos/* con los siguientes propósitos generales:

* Acceder a las listas de sancionados de EEUU y la UE vía URL y cargar los datos en Python transformándolos a formato de DataFrame.
* Almacenar los datos en crudo en formato .csv en la ubicación *TFM/Código y datos/UE\_EEUU\_data\_crudo* para tenerlos localizados en caso de que se quieran revisar. Cada vez que este proceso es ejecutado, se borran los archivos antiguos del directorio local y se vuelven a cargar desde la web.
* Selección de los datos de los DataFrame que sean de interés. Habrá dos tablas para la lista de EEUU relacionadas por clave y una para la UE. Los datos seleccionados se centran en tres dimensiones de la información de sancionados: **Nombre, ubicación y tipo de sancionado.**
* Pasar los DataFrame con los datos cargados y seleccionados a formato .csv y almacenarlos en la ubicación *TFM/Código y datos/datos\_entrada\_script\_02.* Esto se hace para que los datos que vayan a transformarse con el script 02 (del que se hablará más adelante) puedan ser tomados directamente desde esta ubicación siempre que se quiera, sin ser necesario ejecutar antes el script 01. De este modo, solo será necesario ejecutar el script *01\_extraer\_datos.ipynb* antes del script 02en caso de que se quiera tener la última versión de los datos de entrada al script 02*.*

Para cargar en el script de Python las url’s con las listas de sancionados de EEUU se utiliza wget.download() dentro de un bucle for que se ejecuta una vez por cada url. Cada vez que este proceso es ejecutado, se borran los archivos antiguos en el directorio local *TFM/Código y datos/UE\_EEUU\_data\_crudo* y se vuelven a cargar desde la web [3]. Al principio se intentó cargar la lista de EEUU desde un único .xml con toda la información de los sancionados. Pero esto no fue posible porque, al pasarlo a DataFrame, los elementos anidados del .xml no permitían su correcta representación. Por ello, se optó por acceder a la misma información, pero dividida en varios .csv, que se encontraban en la misma página web que el .xml. De entre estos archivos .csv, los que llevan información relevante para su estudio son: *sdn.csv* y *add.csv.* Estos archivos se relacionan por una columna llamada ‘uid’ por lo que, tras ponerlos en formato DataFrame y seleccionar las columnas relevantes, se hace un inner join a las tablas. Las columnas de los .csv no vienen etiquetadas con un nombre. Es necesario buscar el nombre correspondiente a cada columna en el archivo .xml con la información general y escribirlo.

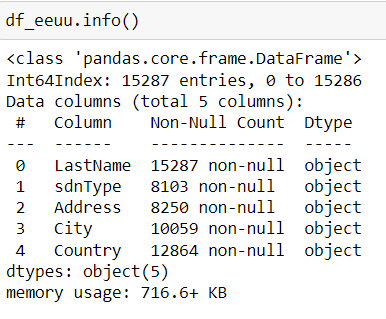


Figura 1: Información de metadatos de la tabla df\_eeuu.

Una vez se tiene un único DataFrame para la lista de EEUU, al que se llama *df\_eeuu*, se estudian los valores de cada variable en busca de posibles incidencias (fig. 1). Se encuentra qué para la columna con la información del tipo de sancionado, llamada ‘sdnType’, hay demasiados valores nulos. El tipo de sancionado puede deducirse a partir del nombre del sancionado (columna de nombre ‘LastName’). Para ello, se crea un bucle que devuelve el nombre de todos los sancionados cuya columna ‘sdnType’ es nula. Una vez hecho, se observa que la mayoría de sancionados se corresponden a una empresa o asociación (fig. 2).

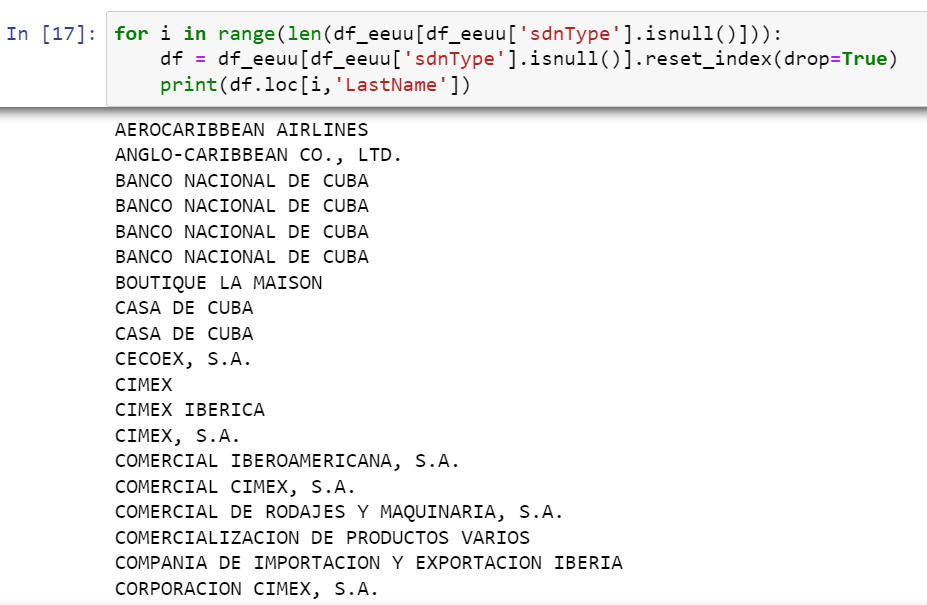


Figura 2: Bucle for para conocer los valores de la columna 'LastName' en los que 'sdnType' tiene valor nulo.

No obstante, existe un número minoritario de nombres en los que no queda claro que tipo de sancionado puede ser. Se optará por pasar todos los valores nulos a valor ‘enterprise’. Haciendo esto se introduce un cierto error en la clasificación de los registros por tipo de sancionado. Pero a cambio se recupera una gran cantidad de información que antes aparecía como nula.

La columna ‘Address’ también presenta gran cantidad de valores nulos. Como los nulos de esta columna no van a tener especial relevancia en los estudios posteriores, podemos dejarla como está.

Para las listas de la UE, la petición a la URL se hace con un token. No sirve el código utilizado con las listas de Estados Unidos porque no acepta el método wget.download(). Es necesario utilizar un proceso alternativo que no almacena la tabla en local, si no que genera directamente en Python un archivo con los datos [4]. Una vez se tiene este archivo, le aplicamos diversas modificaciones para pasarlo a un DataFrame al que se llama *df\_ue*. Cuando se tienen los datos en crudo de la lista de la UE como DataFrame, se pasan como .csv a la ubicación local *TFM/Código y datos/UE\_EEUU\_data\_crudo.* Cada vez que se ejecuta el proceso, se elimina el .csv con los datos antiguos y se cargan los nuevos datos*.*

El DataFrame con los datos da la Unión Europea presenta muchas columnas. Por ello, el mejor método para identificar cuáles de estas son relevantes para nuestro estudio es dividir el DataFrame en grupos de 15 columnas y analizar separadamente las columnas de estas subtablas. La información sobre las variables seleccionadas tras realizar este análisis se encuentra en la figura 3.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

*Figura 3: Información de metadatos de la tabla df\_ue.*

Los valores correspondientes a la ubicación (‘Address\_City’ y ‘Address\_CountryDescription’) no aparecen en el mismo registro que el nombre correspondiente a dicha ubicación. La información de nombres viene dada en la columna ‘NameAlias\_WholeName’. Para poder identificar los valores para la ubicación y moverlos al registro con su correspondiente nombre, es necesario agrupar por la columna 'Entity\_EU\_ReferenceNumber'. Esta columna engloba bajo el mismo identificador a un grupo de sancionados que comparten la misma ubicación. Lo que se pretende agrupando por cada valor único 'Entity\_EU\_ReferenceNumber' es generar un nuevo DataFrame, llamado *df*, en el que a cada número de referencia se le asocie una ubicación descrita en términos de país y ciudad (fig. 4). Una vez definido *df*, se hace join con *df\_ue* para trasladar su información de la ubicación a la tabla de sancionados principal. Hecho esto se elimina la columna 'Entity\_EU\_ReferenceNumber', pues ya no es necesaria.

Tabla

Descripción generada automáticamente

Figura 4: Tabla con la información de ubicación para cada número de referencia. Se observa que algunos de los sancionados de la UE no van a presentar información sobre su ubicación o va a estar incompleta.

En el momento en que fue realizado este análisis sobre la lista de la UE, la columna con la información sobre el tipo de entidad, 'Entity\_SubjectType\_ClassificationCode', presentaba 16 valores únicos contando el valor nulo. Son demasiados, ya que solo interesan rasgos más generales del tipo de sancionado como si es una empresa, una persona o una embarcación. Representamos 'Entity\_SubjectType\_ClassificationCode' en un gráfico de barras (fig. 5):

Gráfico

Descripción generada automáticamente

Figura 5: Gráfico de barras de la cardinalidad para cada valor de la columna 'Entity\_SubjectType\_ClassificationCode' de la tabla df\_ue

Como se aprecia en la figura 5, para casi todos los registros el valor no nulo de 'Entity\_SubjectType\_ClassificationCode' es 'person' o 'enterprise'. Por lo tanto, se pueden eliminar los otros registros sin que se pierda demasiada información.

Tras finalizar el proceso de extracción, los DataFrames *df\_ue* y *df\_eeuu* se llevan, en formato .csv, a la ubicación *TFM/Código y datos/datos\_entrada\_script\_02.* Que es el fichero desde donde se toman los datos de entrada al script *02\_transformar\_datos.ipynb* para continuar con la depuración de las tablas.

### Transformación de Datos

Para la transformación de los valores de las tablas obtenidas en el script 01 también se utiliza un notebook de Jupyter. Estas transformaciones buscan hacer los valores lo más adecuados posibles para su empleo en la herramienta de visualización y en los modelos de Machine Learning. El script encargado de la transformación de los datos es *02\_transformar\_datos.ipynb.* Este script toma las tablas *df\_ue.csv* y *df\_eeuu.csv* en *TFM/Código y datos/datos\_entrada\_script\_02* y las carga como DataFrame. Una vez aplicadas las transformaciones, los DataFrames resultantes se almacenan en formato .csv en *TFM/Código y datos/datos\_entrada\_script\_03,* que es la ubicación desde la que accede el script 03 a los datos transformados.

Para las primeras transformaciones sobre los valores de *df\_eeuu* se crea un nuevo DataFrame llamado *df\_eeuu\_t1.* El 't1' en el nombre implica que esta es la primera transformación hecha sobre los datos tras ser cargados. Este DataFrame:

* Presenta una nueva variable llamada 'Origin' de valor 'EEUU' para identificar cual es la lista de origen.
* Se le han eliminado los registros con 'LastName' nulo. Ya que no podemos identificar un sancionado si no tiene un nombre.
* En caso de existir, se eliminan los registros duplicados. Para que se eliminen los registros duplicados pero que se diferencian por las mayúsculas y minúsculas, se ponen todas las variables en mayúsculas.
* Se cambia el nombre de algunas columnas para darles un nombre más conciso (fig. 6). Este será el mismo nombre que se dé a las variables de la lista de la UE para que las filas puedan concatenarse sin problemas. Esta concatenación de las tablas se hará en el script 03, del que se hablará más adelante.

A continuación, se crea una nueva tabla llamada *df\_eeuu\_t2* para aplicar la segunda transformación sobre los datos: en algunos valores de la columna 'City' aparecen códigos y números que no interesan (fig. 6), pues solo interesa el nombre de la ciudad. Se aplica una transformación mediante una expresión regular para eliminar estos códigos [5]. Ha sido necesario cambiar los valores perdidos en 'City' a cadena de caracteres para poder escanearlos por la expresión regular. Una vez aplicada la expresión, se han vuelto a poner como valor ‘Not a Number’ (NaN). Se ha empleado el método re.sub() dentro de un bucle for que selecciona cada registro de la columna ‘City’ y le pasa la expresión regular. Los nuevos valores para ‘City’ se van almacenando en una lista con cada ejecución del bucle. Finalmente, esta lista se sustituye por la columna ‘City’ antigua en *df\_eeuu\_t2.* Las pruebas para crear la expresión regular que eliminase las cadenas que no son de interés se han hecho en la página:

<https://regex101.com/r/3vtVKW/1>

En esta página se puede ver un cuadro con las cadenas de los valores de ‘City’. Las subcadenas que han sido eliminadas aparecen resaltadas a color. La expresión regular que selecciona estas subcadenas aparece encima del cuadro.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 6: Tabla de Estados Unidos tras aplicar la primera transformación. Se aprecia que algunos valores de la columna ‘City’ presentan códigos tras el nombre de la ciudad y que se han cambiado los nombres de columna respecto a la tabla sin transformar.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 7: Tabla transformada final df\_eeuu\_t2

Finalmentese carga la tabla transformada final *df\_eeuu\_t2 (fig. 7)* en *TFM/Código y datos/datos\_entrada\_script\_03.*

Las primeras transformaciones sobre los valores de *df\_eu* siguen una estructura similar a las de la tabla de Estados Unidos. Se crea un DataFrame llamado *df\_eu\_t1* sobre el que se aplican las siguientes transformaciones:

* Se añade una nueva variable llamada 'Origin' de valor 'UE' para identificar cual es la lista de origen.
* Se han eliminado los registros con 'NameAlias\_WholeName' nulo. Ya que no podemos identificar un sancionado si no tiene un nombre.
* Se eliminan los registros duplicados. Para que se eliminen los registros duplicados pero que se diferencian por las mayúsculas y minúsculas, se ponen todas las variables en mayúsculas.
* Se cambia el nombre de las columnas para darles un nombre más conciso (fig. 8). Este será el mismo nombre que se ha dado a las variables de la lista de EEUU para que puedan concatenarse sin problemas en el script 03.

Echando un vistazo a los valores de la columna ‘Name’ de *df\_eu\_t1* (fig. 8). Se observan dos problemas con sus valores:

1. Aparecen nombres en la columna ‘Name’ en idiomas que utilizan alfabetos distintos al inglés o al español, dificultando su comprensión. Sería útil poder traducir, en la medida de lo posible, estos nombres al inglés para que sean más comprensibles.
2. Hay registros muy parecidos salvo por pequeñas diferencias en los valores en la columna ‘Name’ que pueden considerarse duplicados. Esto puede deberse a que el sancionado escribió mal su nombre en algún formulario del que luego se pasó esta información a la lista de sancionados, o al uso de un alias por parte del sancionado. También puede ocurrir que, aunque los nombres se parezcan mucho, correspondan a distintos sancionados. En cualquier caso, como este trabajo no busca el seguimiento individual de cada sancionado, si no una visión más general de los tipos de sancionados y sus ubicaciones, se considerarán los nombres similares que tengan el resto de variables iguales como duplicados.

Tabla

Descripción generada automáticamente

Figura 8: Tabla de la Unión Europea tras aplicar la primera transformación. En la columna ‘Name’ aparecen nombres en distintos idiomas y algunos de ellos muy similares entre sí. Se han cambiado los nombres de columna respecto a la tabla sin transformar.

Para resolver la problemática del punto 1 se han probado varios módulos de Python, como goslate y googletrans, que hacen una petición a un servicio de traducción en línea. Ambos módulos daban problemas relacionados con la versión y el número límite de peticiones que se podían hacer. Se terminó por utilizar el módulo de deepl. Esta librería de Python da acceso a la API de DeepL. DeepL es un traductor en línea que utiliza tecnología basada en Inteligencia Artificial para obtener mejores resultados. Para poder utilizar DeepL desde Python, es necesario crear una cuenta gratuita que genere una API key que permita la autentificación del cliente. El registro de cuenta se hace desde la web oficial de la API DeepL:

<https://www.deepl.com/docs-api>

Se pasó un bucle for a los registros de la columna ‘Name’ para traducirlos y se llevó cada nombre traducido a una lista. Finalmente se sustituyeron los valores en la columna ‘Name’ en el DataFrame *df\_ue\_t1* por los valores de la lista y se renombró como *df\_ue\_t2* (fig. 9)*.* Algunos de los valores, como los nombres en árabe, no pudieron ser traducidos. No obstante, el resultado general de la traducción es bueno.

Tabla

Descripción generada automáticamente

Figura 9: Tabla de la Unión Europea tras aplicar la segunda transformación. En la columna ‘Name’ se puede ver cómo los últimos registros han sido traducidos.

Todavía existe el problema expuesto en el punto 2: existen nombres escritos de forma muy similar que podrían ser duplicados. Para solucionarlo se crea una tercera transformación sobre el DataFrame y se renombra como *df\_ue\_t3*. Se emplea la función 'get\_close\_matches()' del módulo de Python llamado difflib [6].

La función se emplea dentro de un bucle for que comprueba para cada registro cuantos registros similares o iguales tiene debajo. Si se detecta que no hay ningún registro similar, este es mandado a la lista ‘new\_list\_name’ con un condicional de tipo if. Cada elemento en ‘new\_list\_name’ representa un registro con las columnas separadas por guiones. La función presenta dos parámetros ajustables:

* *n* tiene valor 4 por defecto e indica el número máximo de registros que, definido un cierto nombre, pueden ser similares a este. Es decir, para cada nombre el número máximo de nombres que pueden considerarse duplicados debido a su similitud y por tanto pueden ser eliminados es de 4 nombres.
* *sensibility* es un número entre 0 y 1 que indica el grado de sensibilidad con el que funciona el detector de matches. A mayor sensibilidad, más registros serán considerados como duplicados y, en consecuencia, menos registros aparecen en la lista final. Se hicieron varias pruebas ajustando la sensibilidad hasta encontrar el valor que eliminase la mayor cantidad de registros similares sin que se perdiera excesiva información.

Tabla

Descripción generada automáticamente

Figura 10: Tabla transformada final df\_ue\_t3

Finalmentese carga la tabla transformada final *df\_ue\_t3 (fig. 10)* en *TFM/Código y datos/datos\_entrada\_script\_03.*

Se aprecia que el resultado final de las transformaciones es más completo para la tabla de Estados Unidos que para la tabla de la Unión Europea. La tabla de la UE presenta más valores nulos y no presenta la columna ‘Address’ que sí aparece en la tabla de EEUU.

### Concatenar Datos

Para terminar con los procesos manipulación de los datos, se cargan las tablas obtenidas tras ejecutar el script *02\_transformar\_datos.ipynb* en *03\_concatenar\_datos.ipynb.* Este script simplemente unirá en una misma tabla, con la función concat() de la libraría pandas, las tablas de sancionados de la UE y de EEUU (fig. 11).

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza mediaDe este modo, se tiene toda la información centralizada en un único dataset depurado que puede cargarse al script de preprocesamiento y aplicación de modelos ML, en PowerBI o en cualquier otra herramienta de análisis de datos que se quiera emplear.

Figura 11: Tabla final de sancionados

Se pueden analizar los valores de la tabla haciendo uso de funciones de pandas como info(), describe() o unique(). Se encuentra lo siguiente:

* Existen nombres repetidos. Estos nombres se corresponden a sancionados presentes en varias ubicaciones. Por ejemplo, una empresa que desarrolle actividad comercial en varios países.
* Los sancionados de tipo 'PERSON' e 'INDIVIDUAL' realmente son el mismo tipo de sancionado. Se cambia el valor 'INDIVIDUAL' por 'PERSON' en todos los registros.
* Hay muchos valores nulos para la columna ‘Address’ porque no se tiene información de la dirección para la lista de la UE. Únicamente se conoce la dirección para algunos registros de la lista de EEUU.
* Hay bastante nulos para el país y la ciudad. La mayoría procedentes de la tabla de la Unión Europea.

Finalmente, se lleva el DataFrame concatenado a la carpeta 'datos\_entrada\_script\_04\_y\_PowerBI' desde donde será leído por el programa de aplicación y evaluación del modelo ML y cargado en PowerBI para la generación del dashboard.

## Aprendizaje no supervisado

Buscar resumen en el libro que explique en qué consiste

### Preprocesamiento

1. Selección de variables
2. Imputación

# CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

Es más que conveniente terminar un trabajo con unas conclusiones que sirvan de colofón al mismo. Caben en ellas tus impresiones, o destacar la importancia que tiene el tema, lo que has aprendido, o la trascendencia, etc.

Además las conclusiones tienen que referirse siempre a los objetivos generales y específicos empleados.

Es importante destacar en este apartado las líneas con las que se podría continuar el mismo.

* -COMENTAR EN CONCLUSIONES QUE LA TRADUCCION Y EL RESOLVEDOR DE MATCHES SON MUY LENTOS Y SE PLANTEÓ UTILIZAR SPARK PERO QUE HABRÍA QUE HABER CARGADO LOS SCRIPTS A RDD’S Y ADAPTAR LA LÓGICA DEL CÓDIGO AL DE LA APLICACIÓN DE BIG DATA.
* -COMENTAR EN FUTURAS LÍNEAS DE TRABAJO COMO EL HABER TENIDO VARIABLES NUMÉRICAS HABRÍA PODIDO MEJORAS LAPOSIBILIDADES DE LOS MODELOS ML. LAS VARIABLES NUMÉRICAS PODRÍAN GENERARSE, POR EJEMPLO, A PARTIR DE LA LATITUD Y LONGITUD DE LA UBICACIÓN.

# ANEXOS

Sirven para incluir documentación complementaria (planos, circuitos, códigos, especificaciones, hojas características, fichas explicativas, etc.)

Eso sí, cada comentario en una página independiente. Y ES QUE LOS ANEXOS SE ORGANIZAN ASÍ: CADA DOCUMENTO EN UNA PÁGINA INDEPENDIENTE.

# PRESUPUESTO Y TIEMPO

Las herramientas de software empleadas en este trabajo son open source. Para generar el código, tanto el lenguaje de programación (Python) como el entorno de desarrollo integrado o IDE (Jupyter Notebook) están disponibles a través de la web de forma gratuita. La herramienta de visualización e inteligencia empresarial empleada (PowerBI) ha sido utilizada en su versión gratuita.

También se hizo uso de un traductor en línea (DeepL) para el cual el acceso a la API tenía una opción gratuita. Esta opción para el traductor solo permite ejecutar una traducción de la tabla al mes antes de que se agoten la cantidad de caracteres mensuales. De modo que si se quiere ejecutar más de una traducción al mes hay que pagar aproximadamente 7.50 € por cada nueva ejecución en el mismo mes. La cantidad exacta a pagar dependerá de la cantidad de sancionados que haya en la tabla en el momento de la traducción. Como es un servicio bajo demanda, a más sancionados habrá más cantidad de caracteres y el precio será mayor.

Hay que tener en cuenta la inversión inicial en un ordenador con la capacidad de cómputo necesaria y un sistema operativo que permita la instalación de las últimas versiones de las aplicaciones utilizadas. También es necesario el acceso a internet para poder actualizar las tablas periódicamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Descripción | Presupuesto | Opcional |  |
| Gastos de Capital  (CAPEX) | Ordenador personal | 205 – 1100 € | No |  |
|  |  |  |  |  |
| Gastos de Operación  (OPEX) | Suscripción DeepL API Pro  Internet | 7.50 – 8.50 €/mes  31 – 33 €/mes | Sí  No |  |
|  |  |  |  |  |

FALTA DIAGRAMA DE GANTT

# BIBLIOGRAFÍA

[1] "¿Qué son las sanciones internacionales?" [Online]. Available: https://elordenmundial.com/que-son-sanciones-internacionales/ [Accessed: 2022]

[2]"Reporting de Sancionados"[Online]. Available: https://www.sas.com/content/dam/SAS/es\_es/doc/other1/sas-consulting-web.pdf [Accessed: 2022]

[3]"Overwrite existing files with Python's wget?"[Online]. Available: https://stackoverflow.com/questions/63226700/overwrite-existing-files-with-pythons-wget [Accessed: 2022]

[4]"¿Cómo descargar un archivo csv desde internet con python 3?"[Online]. Available: https://es.stackoverflow.com/questions/120084/c%C3%B3mo-descargar-un-archivo-csv-desde-internet-con-python-3 [Accessed: 2022]

[5]"Operaciones con expresiones regulares"[Online]. Available: https://docs.python.org/es/3/library/re.html [Accessed: 2022]

[6]"How to use get\_close\_matches() in Python"[Online]. Available: https://www.educative.io/answers/how-to-use-getclosematches-in-python [Accessed: 2022]