

Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Plataforma basada en Python para transformación e integración de datos abiertos (open data) sobre servicios de análisis y visualización

ESCUELA POLITECNICA
Autor: Marcos Navarro Juan

Tutor: Manuel De Buenaga Rodríguez

2023

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Plataforma basada en Python para transformación e integración de datos abiertos (open data) sobre servicios de análisis y visualización

Autor: Marcos Navarro Juan

Tutor/es: Manuel De Buenaga Rodríguez

TRIBUNAL:

Presidente: Luis Fernández Sanz

Vocal 1º: Francisco Javier Bueno Guillén

Vocal 2º: Manuel De Buenaga Rodríguez

FECHA: 12/09/2023

A mi familia y amigos

Agradecimientos.

Son muchas las personas a las que tengo que agradecer que hoy sea mucho mejor persona que cuando empecé este camino, cuando empecé esta bonita travesía de estudiar el grado ingeniería informática.

En primer lugar, quiero dar las gracias a mis padres y a mis dos hermanos, que gracias a su apoyo he podido crecer tanto profesionalmente como persona.

En segundo lugar, ha sido de vital importancia rodearme de mi círculo de amistades, los cuales me han dado apoyo tanto en las mejores situaciones como en la peores. Especialmente a mis compañeros de la universidad, con los cuales hemos sacado multitud de proyectos y exámenes adelante.

Quiero destacar el curso que realice con la beca Erasmus, el año que pase en Varsovia, Polonia. El cual, fue fundamental para entender quién soy a día de hoy y esto es gracias a la Politechnika Warszawska y a las amistades inolvidables que forje en ese increíble viaje.

Por último, quiero agradecer a quienes han hecho posible este recorrido, agradecer a la propia universidad de Alcalá y a los diferentes profesores que me han ido acompañando en cada una de las asignaturas y por supuesto a mi tutor de TFG Manuel de Buenaga.

Resumen.

El proyecto se centra en la utilización de fuentes de datos abiertos (open data), estas fuentes de datos se someten a una rigurosa selección según unos criterios establecidos. Posteriormente, se ejecutan los procesos de extracción, transformación y carga (ETL) buscando la máxima optimización. Estos datos resultantes se someten a análisis, que abarcan diversos modelos, incluyendo machine learning. Todos estos procesos se llevan a cabo empleando el lenguaje de programación Python.

Una vez completados los procesos ETL y el análisis de datos, se procede a la visualización de las soluciones utilizando el software PowerBi.

Palabras clave: Open data, ETL, análisis de datos, PowerBi

Abstract.

The project focuses on the use of open data sources, these data sources are subjected to a rigorous selection according to established criteria. Subsequently, extraction, transformation, and loading (ETL) processes are carried out in search of maximum optimization. The resulting data is subjected to analysis, which includes various models, including machine learning. All these processes are carried out using the Python programming language.

Once the ETL processes and data analysis have been completed, the solutions are visualized using PowerBi software.

Key words: Open data, ETL, data analytics, PowerBi

Resumen Extendido.

El propósito central del proyecto es llevar a cabo un proceso completo que involucra la selección de diversas fuentes de datos abiertas, siguiendo criterios preestablecidos. A estas fuentes de datos se le aplican a una serie de procesos de extracción, transformación y carga, con la finalidad de preparar los datos para un análisis y una posterior visualización. Tanto el proceso ETL como el análisis de datos se realiza utilizando el lenguaje de programación Python. Los resultados de ambos procesos se presentan de manera efectiva a través de la herramienta de visualización PowerBi.

La selección e investigación de las diversas fuentes de datos abiertas se rige por un conjunto de criterios fundamentales. Es esencial llevar a cabo una investigación exhaustiva para determinar cuáles de estas fuentes se adecúan mejor a los objetivos establecidos. Los criterios más importantes para evaluar si las fuentes de datos son idóneas para el proyecto son los siguientes:

- En función si la temática que abarca la fuente de datos abierta está relacionada con los proyectos europeos de “Euglooh” y “Smacite”.
- En función de la calidad de los datos.
- En función del tamaño.
- En función del interés personal y global.

Se deciden seleccionar finalmente ocho fuentes de datos abiertas, donde todas cumplen con los requisitos indicados.

Una vez que se han seleccionado las fuentes de datos abiertas para el proyecto, se procede con el proceso de extracción, transformación y carga (ETL). De las ocho fuentes de datos que se han seleccionado previamente, se ha decidido aplicar el proceso de ETL a seis de ellas. Este proceso es esencial para garantizar que los datos estén debidamente preparados de manera coherente para los procesos de análisis y visualización.

Para llevar a cabo el proceso ETL, se utiliza Python que ofrece infinidad de herramientas. Se realiza una investigación para determinar qué bibliotecas específicas se adecuan mejor a los tipos de fuentes de datos abiertas seleccionadas. Esto asegura una implementación eficiente y efectiva del ETL.

Tras el proceso ETL se procede al análisis de datos aplicado a las fuentes de datos. Se realizan diferentes modelos intentando abarcar el mayor conocimiento posible, por ello se realizan modelos desde lo mas simple hasta modelos que ya presentan una dificultad notoria. También se decide aplicar el análisis de datos a cuatro fuentes de datos de las que pasaron el proceso de ETL.

También cuantos más modelos se aplican a una fuente de datos abierta mas conclusiones y conocimientos se obtienen. Por ello en el proyecto se realizan modelos de análisis descriptivo, exploratorio y de predicción, muchos de ellos son modelos de machine Learning.

En concreto se realizan modelos descriptivos, modelos de regresión lineal, modelos de clusterización, modelos de series de tiempo y modelos de red neuronal recurrente. El proceso de análisis también se realiza utilizando el lenguaje de programación Python.

Una vez realizado tanto el proceso ETL como el de análisis se procede a visualizar los resultados obtenidos de ambos procesos utilizando el software de visualización PowerBi. Esta herramienta permite crear visualizaciones interactivas de una manera profesional. En cada modelo realizado se aporta la visualización de la solución utilizando este software.

Por último, con estas se visualizaciones se crean dashboards interactivos, cada dashboard se corresponde con todos los resultados obtenidos de una fuente de datos, creando así cuatro dashboard.

Contenido

Agradecimientos	4
Resumen.....	5
Abstract.....	6
Resumen Extendido.	7
1. Introducción.	16
1.1. Motivación	16
1.2. Objetivos	17
1.3. Línea de Trabajo.....	17
1.4. Estructura de la Memoria.	19
2. Estado del Arte.....	21
2.1. Open Data.....	21
2.2. Análisis de datos	23
2.3 Proceso ETL.....	24
2.4 Python.	25
2.5 Visualización en PowerBi.	26
3. Herramientas esenciales.	28
4. Implementación.....	29
4.1. Tipos de Fuentes de Datos y Proyectos de Investigación.....	29
4.1.1. Smacite.	29
4.1.2. EUGLOH.	31
4.2. Fuentes de Datos Abiertas.....	32
4.2.1. Incendios producidos en España entre el 2006 y el 2015.....	34
4.2.2. Concesión de Nacionalidad Española entre el 2010 y el 2019.....	35
4.2.3. Capacidad Asistencial durante la Covid-19.	36
4.2.4. Catálogo de Parques Municipales en la Ciudad de Madrid.....	37
4.2.5. Siniestralidad en la Carreteras en la Ciudad de Madrid.	38
4.2.6. Población por Provincias de España 1996-2021.....	39
4.2.7. Población por Países en la unión europea 2001-2022.....	40
4.2.7. Catálogo del Bosque Urbano de la Ciudad de Madrid.....	41
4.2.8. Clasificación de las diferentes fuentes de datos abiertas.....	42
4.3. Proceso ETL en las Fuentes de Datos.	43
4.3.1. Proceso ETL en Catálogo de Parques Municipales en la ciudad de Madrid.....	44
4.3.2. Proceso ETL en Capacidad Asistencial durante la Covid-19.....	46

4.3.3. Proceso ETL en Incendios producidos en España entre el 2006 y el 2015.....	48
4.3.4. Proceso ETL en Concesión de Nacionalidad Española entre el 2010 y el 2019.....	50
4.3.5. Proceso ETL en Catálogo del Bosque Urbano de la Ciudad de Madrid.	52
4.3.6. Proceso ETL en Población por Provincias de España 1996-2021.....	56
4.4. Comparativa entre las librerías ETL.	57
4.5. Análisis de Datos.....	59
4.5.1. Análisis Descriptivo.	61
4.5.2. Regresión Lineal.	66
4.5.3. Clustering.	69
4.5.4. Serie Temporal ARIMA.....	74
4.5.5. Red Neuronal Recurrente LSTM.....	78
5. Conclusiones y Líneas Futuras.....	84
5.1. Conclusiones.	84
5.2. Posibles Mejoras y Escalabilidad.	86
5.3. Presupuesto del Proyecto.	86
6. Bibliografía.....	88
Anexo I – Manual de Instalación.	92
Anexo II – Manual de Usuario.....	94
Anexo III – Dashboard PowerBi.	95
Anexo IV – Contenido del Repositorio.....	100

Figuras

Figura 1. Diagrama de Metodología <i>CRISP-DM</i>	19
Figura 2. Tabla de puntuaciones de los países miembros en el año 2022.....	22
Figura 3. Tabla de puntuaciones de la aportación de fuentes de datos abiertas a nivel europeo...	23
Figura 4. Esquema del funcionamiento de un proceso ETL.	25
Figura 5. Evolución de los lenguajes de Programación en función del tiempo.....	26
Figura 6. Mapa de las universidades relacionadas con Euglooh.	31
Figura 7. Campos que aborda el proyecto Euglooh.	32
Figura 8. Ejemplo de tabla extraída de la fuente de datos “Incendios producidos en España entre el 2006-2015.....	34
Figura 9. Ejemplo de tabla extraída de la fuente de datos “Concesión de Nacionalidad entre el 2010 y el 2019”.....	35
Figura 10. Ejemplo del histórico extraído de la fuente de datos “Capacidad Asistencial durante el Covid-19”.....	36
Figura 11. Ejemplo de cómo se expone la información en la fuente de datos “Catalogo de Parques Municipales de Madrid”.....	38
Figura 12. Ejemplo de la composición de la fuente de datos “Siniestralidad en las Carreteras en la Ciudad de Madrid”.....	39
Figura 13. Ejemplo de la fuente de datos “Población por Provincias de España”.....	40
Figura 14. Ejemplo de la fuente de datos “Población por Países de la unión Europa”.....	41
Figura 15. Ejemplo de la exposición de la información de la fuente de datos “Valor del Bosque Urbano de Madrid”, de la página 51 del PDF.....	42
Figura 16. Parte del data warehouse una vez realizado el proceso ETL.	46
Figura 17. Parte del resultado del Proceso ETL de la fuente de datos “Capacidad Asistencial durante la Covid-19”.....	48
Figura 18. Tabla extraída de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”, página 26.	50
Figura 19. Parte del data warehouse una vez realizado el proceso ETL.	50
Figura 20. Parte del data warehouse una vez realizado el proceso ETL de la fuente de datos “Concesiones de Nacionalidad Española entre el 2010 y el 2019”.....	52
Figura 21. Parte de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”.	55
Figura 22. Parte de la visualización del data warehouse de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”.	55
Figura 23. Parte de la visualización del data warehouse de la fuente de datos “Población por Provincias de España 1996-2021”.....	57
Figura 24. Total de camas VS Camas ocupadas por Covid-19.....	62
Figura 25. Ingresos por Covid-19.	63
Figura 26. Resultado de los parámetros del análisis Descriptivo en la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.....	63
Figura 27. Número de incendios totales vs el número de incendios > 500 hectáreas.	64
Figura 28. Resultado de los parámetros del análisis Descriptivo en la fuente de datos “Catálogo del Bosque Urbano de la Ciudad de Madrid”.....	64
Figura 29. Agua Interceptada (M3/Año) vs Producción de oxígeno (Tn/Año) por especie.	65
Figura 30. Parte del resultado de los parámetros del análisis Descriptivo en la fuente de datos “Población por Provincias de España 1996-2021”.....	65
Figura 31. Evolución de la población en la provincia de Ávila en función de los años.....	66
Figura 32. Regresión Lineal en la provincia de Ávila.....	69
Figura 33. Clustering Mean Shift en la fuente de datos “Capacidad Asistencial durante la Covid-19”.....	71
Figura 34. Clustering K-Means en la fuente de datos “Catálogo del Bosque Urbano de la Ciudad de Madrid”.....	73
Figura 35. Diagnóstico del modelo ARIMA.....	77
Figura 36. Predicción de los meses de febrero y marzo.....	78

Figura 37. Arquitectura de una red neuronal recurrente.....	79
Figura 38. Arquitectura de una unidad de una red recurrente LSTM.	80
Figura 39. Resultados de las predicciones de la superficie por % en GIF de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.	82
Figura 40. Resultados de las predicciones del número de siniestros por año de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.	83
Figura 41. Dashboard del resultado del análisis descriptivo en la fuente de datos “Capacidad Asistencial durante la Covid-19”	95
Figura 42. Dashboard del resultado de la clasterización en la fuente de datos “Capacidad Asistencial durante la Covid-19”	96
Figura 43. Dashboard del resultado de la serie temporal ARIMA en la fuente de datos “Capacidad Asistencial durante la Covid-19”	96
Figura 44. Dashboard del resultado del análisis descriptivo en la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”	97
Figura 45. Dashboard del resultado de la RNN LSTM en la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”	98
Figura 46. Dashboard del resultado del análisis descriptivo en la fuente de datos “Catálogo del Bosque Urbano de la Ciudad de Madrid”	99
Figura 47.Dashboard del resultado del análisis descriptivo y de regresión lineal en la fuente de datos “Población por Provincias de España 1996-2021”.	99

Tablas

Tabla 1. Calificación de las diferentes fuentes de datos seleccionadas.....	43
Tabla 2. Comparación entre las diferentes librerías para los procesos ETL.	58
Tabla 3. Comparación entre las librerías para la extracción de tablas.	58
Tabla 4. Modelos Estadísticos de Cada Fuente de Datos Abierta.....	60
Tabla 5. Presupuesto del proyecto.....	86

Código.

Código 1. Función get_text del proceso de extracción de la fuente de datos “Catalogo de Parques Municipales de la Ciudad de Madrid”.....	45
Código 2. Función clean del proceso de transformación de la fuente de datos “Catalogo de Parques Municipales de la Ciudad de Madrid”.....	45
Código 3. Función export_xlsx del proceso de carga de la fuente de datos “Catalogo de Parques Municipales de la Ciudad de Madrid”.....	46
Código 4. Proceso de extracción de la fuente de datos “Capacidad Asistencial durante la Covid-19”.....	47
Código 5. Proceso de carga de la fuente de datos “Capacidad Asistencial durante la Covid-19”.....	48
Código 6. Proceso de extracción y transformación de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.....	49
Código 7. Proceso de Carga de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.....	49
Código 8. Proceso de ETL de la fuente de datos “Concesiones de Nacionalidad Española entre el 2010 y el 2019”.....	51
Código 9. Proceso de extracción de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”	53
Código 10. Proceso de transformación de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”	54
Código 11. Proceso de carga de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”	54
Código 12. Proceso ETL de la fuente de datos “Población por Provincias de España 1996-2021”.....	56
Código 13. Cálculo y resultado de los parámetros del análisis Descriptivo en la fuente de datos “Capacidad Asistencial durante la Covid-19”.....	62
Código 14. Creación de los conjuntos de entrenamiento del modelo de regresión lineal.	68
Código 15. Creación del modelo de clasterización con el algoritmo K-Means.	72
Código 16. Mejores valores encontrados para p, d, q en el modelo ARIMA.....	76
Código 17. Propiedades del entrenamiento del modelo LSTM parte 1.	82
Código 18. Propiedades del entrenamiento del modelo LSTM parte 2.	83

Acrónimos y Abreviaturas.

ETL	Extract, Transform, Load
Km	Kilómetros
Ud	Unidades
M ²	Metros al Cuadrado
M ³	Metros al Cubo
TN	Toneladas
CRISP-DM	Cross-Industry Standard Process for Data Mining
EULOG	European University Alliance for Global Health
INE	Instituto Nacional de Estadística
UCI	Unidad de Cuidados Intensivos
GIF	Grandes Incendios Forestales
ML	Machine Learning
RNN	Red Neuronal Recurrente
LSMT	Long Short-Term Memory
BPTT	Backpropagation through time
ARIMA	AutoRegressive Integrated Moving Average
ACF	Función de Autocorrelación
GRU	Gated Recurrent Unit
BIC	Bayesian Information Criterion

Capítulo 1

1. Introducción.

En este primer capítulo del TFG, en primer lugar, se va a exponer diferentes argumentos del porque se ha realizado este proyecto, una serie de objetivos a cumplir, se definirán unas líneas de trabajo a seguir y finalmente se describirá brevemente la estructura del documento.

1.1. Motivación

La información, determinar qué información es la relevante y cual no lo es y exponerla de una forma adecuada en la que se pueda interpretar, tiene un increíble poder en nuestra sociedad. A lo largo de nuestra historia como seres humanos, la información siempre ha jugado un papel determinante y es en la actualidad donde más información recibidos y recabamos. Por ello, todo tipo de gobiernos, empresas, y más organismos invierten tanto tiempo y dinero en poder entender y descifrar la mayor cantidad de información posible y actuar en consecuencia.

En la actual, el papel de internet juega un papel vital. Gracias a internet y la era de las tecnologías en la que vivimos recabar esta información o datos es un proceso que se produce en cada momento que estamos conectados. No importa en que dispositivo o aplicación o en qué lugar estemos conectados que permanente se está enviando información relevante.

Al visualizar la información que se considera de valor utilizando las diferentes fuentes de datos, provoca en una mejor toma de decisiones. Al tener una plataforma que facilite la transformación y la integración de datos abiertos, los diferentes agentes realizaran un análisis más consistente de la situación, por tanto, aumentará la calidad de las decisiones, que esta soportadas por evidencias sólidas.

Los datos abiertos contienen una gran cantidad de información de gran valor que ayuda a comprender mejor diversos fenómenos, esta información esta presentada por diferentes organismos públicos o privados que garantizan la consistencia del dato. El acceso a estos datos abiertos es una manera libre, por lo que, provoca un impulso en la innovación y una colaboración ciudadana.

El concepto de Open Data es un concepto que nació en el siglo XXI el cual tiene una tendencia evolutiva exponencial, donde son cada vez más los organismos públicos o privados que ponen a disposición de aquel que lo deseé información sobre infinidad de cuestiones.

El procesamiento de datos y el análisis de datos son más eficientes con el uso del lenguaje Python y su amplia gama de bibliotecas, que se están expandiendo de forma exponencial. Utilizar e investigar las bibliotecas de Python puede permitir la automatización de tareas tediosas y repetitivas, ahorrando tiempo y recursos en el manejo de grandes volúmenes de datos.

Debido a este conjunto de ideas y de la importancia que siempre ha tenido y más en la actualidad el extraer información de valor sobre un conjunto de datos nace la motivación para la creación de este proyecto. Además, el proyecto en cuestión afectaría a infinidad de campos, donde algunos de estos suponen de una importancia notable para la sociedad. Por ejemplo, se puede destacar el campo de la medicina, el de la salud y medioambiente, campos financieros, deportivos.

1.2. Objetivos

El objetivo del proyecto es trabajar sobre un conjunto de fuentes de datos que cumplan la característica de ser open data y que pertenezcan a diferentes campos de la sociedad actual. Algunas de estas fuentes de datos estarán relacionadas con proyectos de investigación en los que ha participado el departamento (Eugloh o Smacite). Se aplicarán nuevas funcionalidades a través de Python para la extracción, transformación y carga de los datos (ETL) y para el análisis a los diferentes conjuntos de datos. El resultado de este análisis se presentará de la manera más visual apoyándose en el software de Microsoft PowerBi. Este objetivo principal está compuesto por subobjetivos más concretos. Estos subobjetivos son los siguientes:

- La utilización de fuentes de datos abiertas (Open Data), cuyos dominios están relacionados con las ciudades inteligentes (Smart Cities) y con la salud y el medioambiente. Las administraciones públicas proporcionan una serie de bases de datos abiertas de múltiples campos de información. Algunos de estos campos pueden ser de información relacionada con accidentes de tráfico, la distribución de mobiliario urbano, valores climatológicos, ... Donde cualquier tipo de agente individual u organismo puede acceder a esta información y realizar proyectos que fomentan la innovación y el desarrollo.
- Relación al proyecto internacional “Eugloh” [1], realizado por universidades europeas, incluida la Universidad de Alcalá de Henares. Eugloh es un proyecto relacionado con la salud global y Smart Cities.
- Relación con el proyecto internacional “Smacite” [2], el cual es un proyecto financiado por la unión europea en donde trabajan diferentes organizaciones, una de ellas es la universidad de Universidad de Alcalá de Henares. Es un proyecto relacionado con el concepto de ciudad inteligente y con todo lo representa.
- Realización de procesos de extracción, transformación y carga de datos a través de Python para las fuentes de datos abiertas seleccionadas, para que el dato sea consistente y heterogéneo a la hora de aplicar las diferentes técnicas de análisis. Donde incluso las fuentes de datos utilizadas tengan diferente formato, pero sean validas a la hora de integrarse en la plataforma.
- La implementación de técnicas avanzadas basadas en analítica de datos, desarrolladas en Python, sobre las bases de datos abiertas anteriormente descritas y su integración en PowerBi. A partir de las soluciones obtenidas en las diferentes técnicas de análisis se aporta una valoración de los resultados que aportan funcionalidad.
- Visualizar la información recabada tras la implementación de técnicas avanzadas basadas en analítica de datos, de una forma intuitiva y comprensible para el usuario. También se visualizará las soluciones obtenidas y la interpretación de valor de cada una de ellas.

1.3. Línea de Trabajo.

Para el desarrollo del trabajo se utiliza la metodología de *CRISP-DM*. Esta metodología consiste en dividir el trabajo en diferentes fases, en concreto, el proyecto se divide en 7 fases de *CRISP-DM*, que a su vez se divide en subtareas:

1. Definición de objetivos y alcance del proyecto. Se definen los objetivos que se quieren alcanzar y se determina si son viables o no, en función del tiempo y de los

recursos. También se define el alcance del proyecto teniendo en cuenta las funcionalidades específicas a desarrollar.

2. Investigación y selección de fuentes de datos abiertos. Se investigan y se seleccionan las fuentes de datos abiertas a utilizar en el proyecto. La investigación se realiza en diferentes páginas gubernamentales donde se exponen las diferentes bases de datos y sus características. Se seleccionan las fuentes de datos en función de la temática (en relación con “Euglooh”, ciudades inteligentes, salud y medioambiente ...) y teniendo en cuenta la cantidad de información y la consistencia de los datos.
3. Implementación de la extracción y transformación de datos. Se realizan los diferentes programas a través de Python para realizar las operaciones ETL para cada una de las fuentes de datos.
4. Implementación de las técnicas avanzadas basadas en analítica de datos. Se realizan las diferentes técnicas de analítica de datos para las diferentes fuentes de datos seleccionadas. Dependiendo de los objetivos esperados de cada fuente de datos se realiza una técnica de u otra. Se realiza el análisis de datos utilizando Python y diferentes bibliotecas que provee el lenguaje.
5. Desarrollo de las herramientas de visualización en PowerBi. En función de los resultados obtenidos en el paso anterior y de las fuentes de datos se utiliza el software PowerBi para visualizar diferentes aspectos relevantes al proyecto.
6. Pruebas y evaluación de la plataforma. A lo largo de todo el proyecto se realizan las pruebas y las evaluaciones correspondientes a cada etapa del proyecto.
7. Documentación. A medida que se va a realizando cada una de las etapas, se va explicando en el documento los pasos realizados y aportación teórica necesaria para comprender de una forma sencilla el funcionamiento del proyecto.
8. Despliegue. Una vez realizada cada una de las etapas se prepara la plataforma para su despliegue y su complemento funcionamiento.

La línea de trabajo tiene el siguiente flujo de estados:

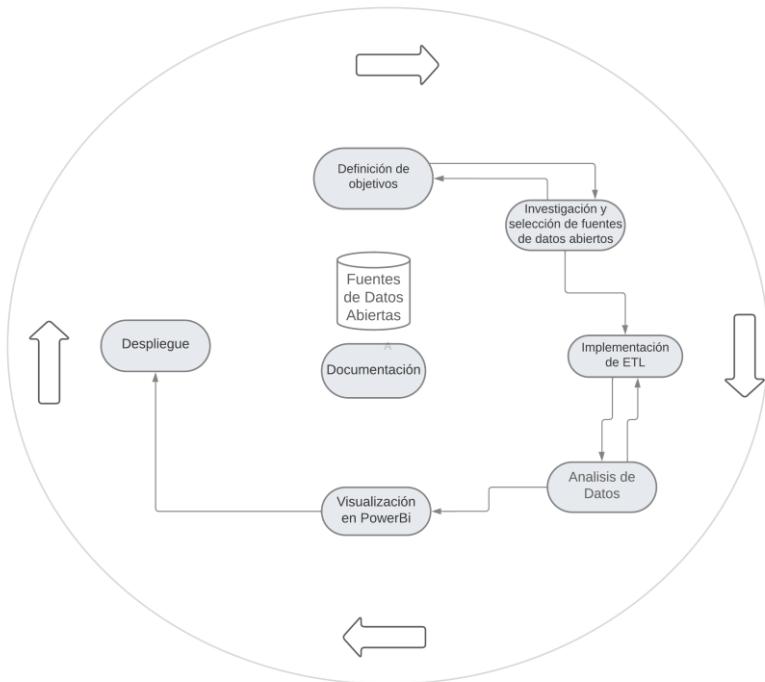


Figura 1. Diagrama de Metodología CRISP-DM

Una ventaja notoria de utilizar *CRISP-DM* [33], es que permite moverse entre las distintas fases, cuando el proyecto lo requiere. Como se puede observar en la *figura 1*, a lo largo del proyecto se realiza de forma paralela la documentación y también teniendo en todo momento acceso a las diferentes fuentes de datos.

Para facilitar la viabilidad se fija un hito, en mitad del tiempo destinado al desarrollo, donde se ha avanzado de forma de significado en todas las fases del proyecto. Completando todas las fases restantes.

1.4. Estructura de la Memoria.

La estructura de la memoria está compuesta por capítulos, la estructura es la siguiente:

- Capítulo 1. Se realiza una breve introducción del proyecto, se explica las motivaciones que se han producido para llevar a cabo el proyecto, se define un flujo de trabajo y se explica cómo va a ser la estructura de la memoria.
- Capítulo 2. Se explican teóricamente los conceptos necesarios para entender el proyecto. Los conceptos que destacan son los de: Open Data, ETL, Análisis de datos, Python y Visualización con PowerBi.
- Capítulo 3. Se explican las herramientas que se han llevado a cabo para realizar el proyecto y el porqué de la utilización de cada una.
- Capítulo 4. Se expone la implementación llevada a cabo. Se expone cada una de las fuentes de datos escogidas y el porqué de la elección, se explica todo lo correspondiente con el proceso de ETL, todo el proceso del análisis de datos, y de la visualización de resultados. Se expone como se ha desarrollado cada uno de los modelos estadísticos y de machine learning y como han aplicado a las fuentes de datos abierta, se explica las conclusiones obtenidas de los resultados y su visualización.

- Capítulo 5. Se comentan las conclusiones obtenidas al finalizar el proyecto, la posible escalabilidad del mismo y temas de consideración como posibles aplicaciones, mejoras y análisis. Por último, se realiza una estimación de presupuesto que se ha empleado para poder realizar el proyecto.
- Capítulo 6. Se indican todas las referencias utilizadas para la realización del proyecto y los anexos correspondientes.

Capítulo 2

2. Estado del Arte.

A continuación, se explican los conceptos de mayor relevancia del proyecto y las razones por las que estos conceptos son utilizados en el proyecto. Para la compresión del proyecto es necesario tener claro una serie de conceptos, entender cómo funcionan y cuáles son sus puntos fuertes y débiles. Por ello, en este punto se presentan los siguientes conceptos desde una perspectiva teórica y se explican las diferentes ventajas por las que se ha decidido que forman parte del proyecto.

2.1. Open Data

El concepto de “Open Data” se basa en la premisa proporcionar al público ciertos conjuntos de dato de una manera libre y sin restricciones excesivas. Existen diferentes organizaciones que generan o poseen datos que ponen a disposición del público en general.

El objetivo del concepto es de promover la transferencia y la apertura de la información, haciéndola accesible a todo el que lo deseé. Al liberar la información, se busca fomentar la participación ciudadana para colaborar y desarrollar soluciones innovadoras en todo tipo de ámbitos. También puede provocar un aumento de la calidad de la información presentada, ya que, al permitir que los ciudadanos e investigadores analicen y examinen la información pueden detectar errores o mejorías.

Las fuentes de datos se pueden estructurar en dos categorías principales, fuentes datos estructuradas, no estructuradas y semiestructuradas, durante el proyecto se trata con los dos primeros tipos de fuentes de datos.

Las fuentes de datos estructuradas contienen información organizada en un formato uniforme y predefinido, los datos forman parte de un esquema fijo con tablas, columnas y filas. Ejemplos de este tipo de fuente son las bases de datos relacionales, archivos CSV y XML y bases de datos SQL.

Su principal característica es que son fácilmente legibles por computadores y se pueden almacenar, consultar y analizar utilizando lenguajes de consulta estructurados.

Las fuentes de datos no estructuradas contienen la información en un formato sin una estructura predefinida y uniforme. La información se presenta de una forma libre sin seguir ningún tipo de esquema fijo. Algunos ejemplos de este formato es la información presentada en documentos de Word o archivos PDF, publicaciones de redes sociales, imágenes y videos.

La principal característica es que son más difíciles de analizar y procesos porque no siguen ningún patrón definido, lo que dificulta la extracción. El uso de técnicas más avanzadas, como el procesamiento de lenguaje natural (NLP), reconocimiento de patrones y aprendizaje automático, es necesario para su tratamiento.

Las fuentes de datos semiestructuradas tienen cierto grado de estructura, pero no cumplen los suficientes requisitos de un formato predefinido. Un ejemplo de este formato son los JSON, que proporciona una estructura flexible para almacenar y transferir datos.

Las fuentes de datos que se utilizan en este proyecto son todas de fuentes de datos abierto, ya que, nos brindan las siguientes ventajas en función a los objetivos definidos del proyecto:

- Libre acceso sin costo alguno. Debido a que la característica principal de las fuentes de datos abiertas es que tienen un acceso libre y sin restricciones provoca que el uso de las bases de datos de diferentes ámbitos no tenga coste alguno. Lo cual, es una ventaja significativa a tener en cuenta en todos tipos de proyecto y más cuando son de índole académica.
- Acceso a fuentes de datos de diversos indoles. Al utilizar open data, se tiene un acceso a una amplia variedad de bases de datos de diferentes dominios. Organizaciones como puede ser el gobierno de España o la unión europea ponen a disposición multitud de fuentes de datos abiertas de diferentes categorías como demografía, empleo, medio ambiente, economía, ...
- Calidad y confiabilidad de los datos. Como detrás de estas fuentes de datos abiertas se encuentran instituciones de reconocidas provoca un mayor grado de confiabilidad y calidad en comparación con otras fuentes de datos no verificadas. Lo que permite trabajar sabiendo que la información con la se trabaja es precisa y confiable, lo cual, es un aspecto fundamental para el análisis y visualizaciones confiables.
- Innovación. Al utilizar open data se fomenta la innovación y el desarrollo de soluciones creativas. Gracias estos conjuntos de datos se pueden descubrir nuevas conclusiones, patrones y tendencias que pueden impulsar la creación de herramientas analíticas y visuales únicas.
- Enfoque en problemas sociales. El contenido de las fuentes de datos abiertas se corresponde con temas sociales, por lo que, se pueden abordar problemas sociales o deficiencias del sistema y proporcionarle una solución de calidad que tenga un impacto real en la calidad de vida.

Cabe indicar que España se encuentra entre los líderes del fomento del Open Data a nivel europeo. Todos los gobiernos de los países pertenecientes a la unión europea tienen un sistema donde proporcionan fuentes de datos abiertas que abarcan todos los ámbitos de la vida social. El “*Open Data Maturity Report*” [3] recoge el desempeño de los países europeos en términos de aportaciones y lo califica con un sistema de puntuación de cuatro dimensiones (política, social, impacto y calidad). En los últimos años, los países miembros de la unión europea han mejorado notablemente sus aportaciones, por tanto, sus puntuaciones, en el 2015 la puntuación media era del 44% y en el 2022 se ha incrementado hasta el 81%. En la figura 2 se puede observar las puntuaciones de cada país miembro de la unión europea en función de la política, social, impacto y calidad.

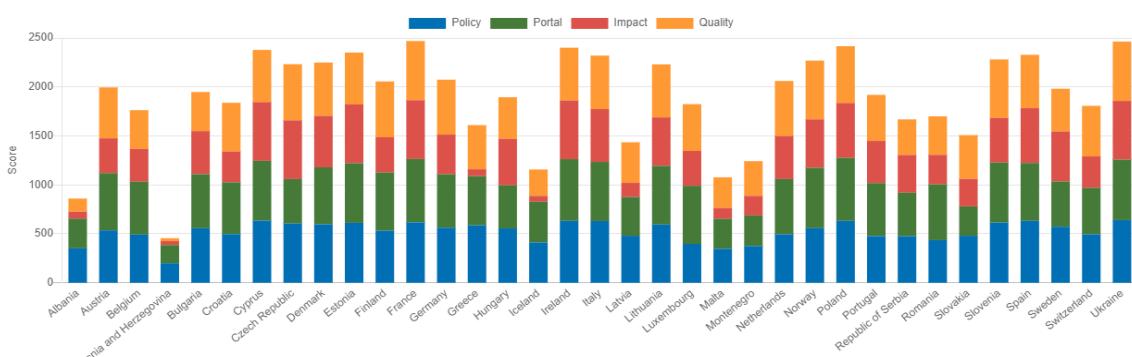


Figura 2. Tabla de puntuaciones de los países miembros en el año 2022.

España se encuentra en la tercera posición con más aportaciones y se encuentra por encima de la media europea como se puede observar en la *figura 3*.

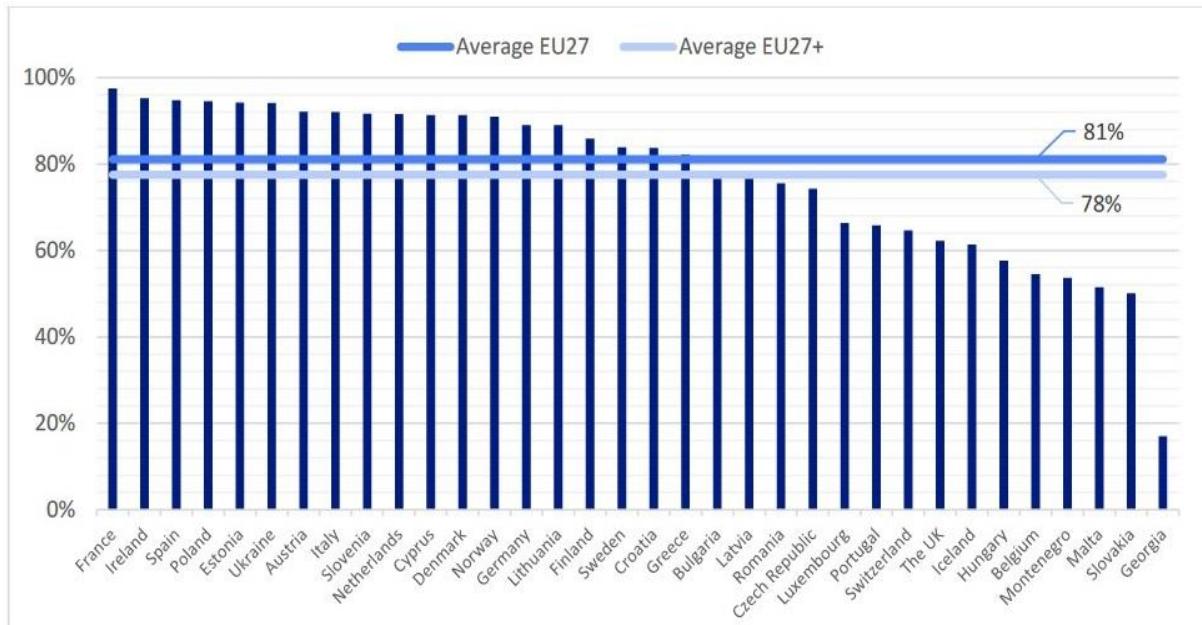


Figura 3. Tabla de puntuaciones de la aportación de fuentes de datos abiertas a nivel europeo.

Como se puede observar la utilización de fuentes de datos abiertas es una práctica que ha ido aumentado de forma notable y para el futuro se prevé que evolucione siguiendo la misma tendencia exponencial.

2.2. Análisis de datos

El análisis de datos es el proceso por el cual se examinan, se limpian, se transforman e interpretan una serie de conjuntos de datos. Tiene como objetivo encontrar patrones, relaciones para extraer conocimientos que aporten valor para tomar decisiones con información relevante.

Este concepto es fundamental a la hora de tomar decisiones relevantes. Ya que un correcto análisis de datos otorga cierta información de valor, como pueden ser tendencias, patrones o relaciones, que ayudan a comprender situaciones, evaluar opciones y respaldar la toma de decisiones. La información obtenida puede servir para optimizar procesos, al analizar los datos de diferentes procesos, sirve para identificar inefficiencias, cuellos de botella o áreas de mejora, lo cual puede aumentar la eficiencia y reducir costos.

También la información obtenida a partir del análisis de datos sirve para identificar oportunidad y riesgos. Aunque se pueden encontrar oportunidades comerciales, nichos de mercado o tendencias emergentes, también se pueden encontrar riesgos potenciales, como errores o desviaciones significativas en los datos.

Existe una multitud diferentes de tipos de análisis de datos, los cuales se pueden aplicar a un conjunto de datos, cada uno de los cuales utiliza métodos y técnicas diferentes para lograr una variedad de objetivos. A continuación, se describen los más significativos:

- **Análisis Descriptivo.** Consiste en resumir y describir los datos utilizando medidas de estadística básica como pueden ser promedios, medianas, desviaciones. Tiene como objetivo proporcionar una compresión inicial de los datos.

- **Análisis Exploratorio.** Consiste en utilizar técnicas gráficas y procesos estadísticos más avanzados, como el análisis de regresión, el análisis de conglomerados (clustering) y el análisis de componentes principales (PCA), para descubrir patrones y relaciones ocultas en los datos. El objetivo es ayudar a generar hipótesis sobre los datos y guiar la investigación futura.
- **Análisis Predictivo.** Consiste en identificar patrones históricos y tendencias para realizar una predicción que coincide con la realidad. Toda predicción tiene un error único. Se construyen modelos predictivos utilizando técnicas estadísticas y algoritmos de aprendizaje automático (Machine Learning).
Es importante destacar que cuantos más datos se procesen y mejor sea el algoritmo o el proceso utilizado, más patrones históricos y tendencias se identificarán, lo que significa que se obtendrán predicciones más reales y con menos errores. Este tipo de análisis de datos es fundamental para ámbitos como la salud, finanzas, marketing o logística.
- **Análisis Prescriptivo.** Tiene como objetivo proporcionar recomendaciones y soluciones optimas basadas en datos a través de combinar técnicas de análisis predictivo con optimización matemática con restricciones y objetivo específicos. Se utilizar para argumentar toma de decisiones y optimización de recursos.
- **Análisis de texto y Minería de Datos:** Se aplica a conjunto de datos no estructurados, como pueden ser textos, reseñas, comentarios en redes sociales. Combina las técnicas de procesamiento del leguaje natural (NLP) y la extracción de grandes conjuntos de datos. Tiene como objetivo entender patrones de comportamientos, de pensamiento u opiniones.

En el proyecto se realizan varios tipos de análisis de datos, desde soluciones más simples hasta resultados más complejas, con el objetivo de abarcar una amplia gama de modelos. El análisis predictivo y el uso de algoritmos de aprendizaje automático son los tipos de análisis que predominan en el proyecto.

El proyecto estará compuesto de tres modelos estadísticos, donde cada uno está enfocado a una fuente de datos abierta diferentes. Dependiendo de cómo este compuesta la fuente de datos abierta y de los objetivos sobre el análisis se utiliza una técnica de análisis u otra.

Los tipos de análisis realizados en el proyecto son, el análisis descriptivo, el análisis exploratorio donde destaca el análisis de regresión lineal o la clasterización con técnicas de machine learning y por último el análisis predictivo con series de tiempo o redes neuronales.

Cabe destacar que tanto los procesos de extracción, transformación y carga (ETL) como los procesos de extracción, carga y transformación (ELT) entran dentro del concepto de análisis de datos. El proceso ETL tiene una gran importancia en el proyecto.

2.3 Proceso ETL.

El proceso ETL (Extract, Transform y Load) es una metodología utilizada en el análisis de datos para la integración y la preparación de datos de diferentes fuentes de datos o data warehouse. Estos procesos se utilizan para garantizar la calidad, coherencia y disponibilidad de los datos para proceder a un análisis más profundo.

El proceso de ETL está compuesto por diferentes fases, a continuación, se explica cada una ellas:

1. **Extracción.** En esta fase se extraen los datos de las fuentes de datos, que en el caso del proyecto serán fuentes de datos abiertas. Dependiendo de cómo este presentada la fuente de datos se utiliza unas técnicas de extracción u otras, por ejemplo, la extracción

se puede realizar en fuentes de datos SQL, o de ficheros Excel o incluso de documentos PDF.

Se utiliza Python para la realización de las técnicas de extracción, ya que, Python ofrece infinidad de bibliotecas y herramientas que facilitan la extracción de datos, como puede ser Pandas [4] o BeautifulSoup [5].

2. Transformación. Una vez que ya se ha extraído todos los datos, estos someten a una serie de procesos de limpieza, normalización y enriquecimiento para garantizar su calidad y coherencia, que es fundamental a la hora de realizar el análisis de datos y la visualización de la información pertinente. Las operaciones más comunes que se suelen realizar en esta fase son los procesos de eliminación de duplicados, la conversión de formatos, la agregación de datos y la creación de nuevas variables. Para realizar la transformación también se utiliza Python, debido a sus amplias bibliotecas que facilitan la transformación de la información, algunas de estas son Pandas [4], NumPy [6] o SciPy [37].
3. Carga. Una vez que ya se ha producido la fase de extracción y la de transformación se vuelven a cargar estos datos en data warehouse. Un data warehouse es un sistema de almacenamiento de datos que está creado para un posterior análisis y generación de informes.
El objetivo de esta fase es cargar los datos ya transformados en una estructura de almacenamiento diseñada específicamente para su análisis, para su visualización o su acceso. A través de Python se producirá la carga de datos en los diferentes sistemas data warehouse.

A continuación, se presenta un esquema del funcionamiento global de un proceso ETL:

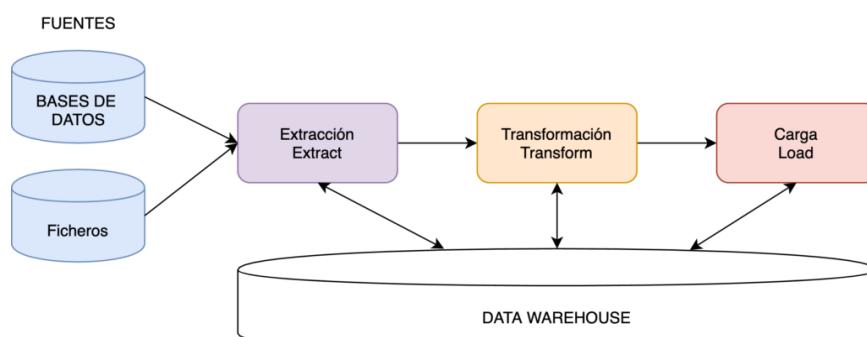


Figura 4. Esquema del funcionamiento de un proceso ETL.

2.4 Python.

Los de procesos de ETL de las diferentes fuentes de datos abiertas, tanto los diferentes tipos de análisis que se aplican a las fuentes de datos se realiza usando el lenguaje de programación Python.

Python es un lenguaje de programación de alto nivel y de propósito general, que se caracteriza por ofrecer una amplia gama de librerías y herramientas que simplifican multitud de procesos. Relacionados con el proyecto, los procesos de manipulación, procesamiento y análisis de datos.

Python destaca por su evolución en los últimos años, ya que, ha experimentado un crecimiento significativo comparado con los otros lenguajes de programación. La utilización de Python frente a otros lenguajes se debe a las ventajas que propone.

En la *figura 5* se presenta la evolución en función del tiempo, desde antes del año 2006 hasta el 2022, de los lenguajes de programación más utilizados globalmente. La información esta extraída de “PYPL” [7], donde realizan la gráfica analizando la frecuencia con la que se buscan tutoriales sobre los lenguajes de programación en Google. Como se observa Python se convierte en la más utilizada actualmente con un índice de búsquedas del 27,43% según “PYPL”.

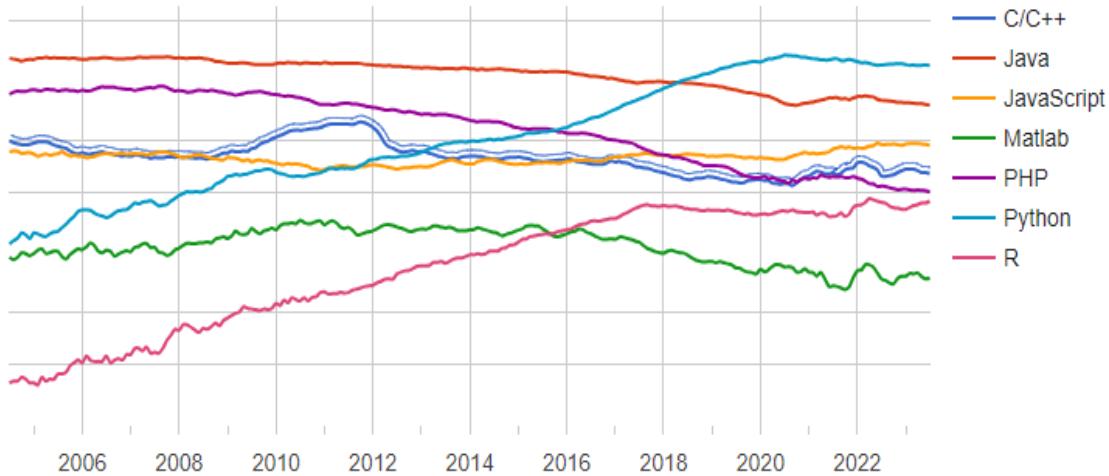


Figura 5. Evolución de los lenguajes de Programación en función del tiempo.

Se escoge este lenguaje de programación y no otro, debido a las ventajas notables que proporciona para un proyecto de este tipo, las cuales son las siguientes:

- Facilidad de uso. Python desataca por tener una síntesis clara y legible, lo cual, a la hora de manipular y transformar los diferentes datos es fundamental tener un código fácilmente comprensible.
- Amplio Ecosistema de Bibliotecas. Python tiene un amplio ecosistema de bibliotecas, que no para de incrementar. Estas bibliotecas son fundamentales a la hora de proporcionar herramientas para realizar tareas de extracción, transformación y carga (ETL) y las diferentes técnicas de análisis de datos.
- Compatibilidad con las fuentes de datos abiertos. Python puede trabajar con diferentes formatos de datos, lo cual es fundamental para el manejo de datos abiertos. No importa que los datos abiertos se encuentren almacenados en archivos de diferentes tipos como pueden ser CSV, JSON, XML que Python ofrece bibliotecas y herramientas que facilitan el proceso de extracción.
- Integración de otras tecnologías. Python se integra de forma sencilla con otras tecnologías y software. Se puede combinar con bases de datos, servicios web y otras herramientas. En concreto, es compatible con el software PowerBi, por lo que, se puede realizar la visualización de la información de una forma concisa.
- Soporte de la comunidad. Cuenta con una comunidad inmensa de documentación y de desarrolladores, por lo que, facilita el aprendizaje y desarrollo de la plataforma.

2.5 Visualización en PowerBi.

PowerBi es una herramienta de Business Intelligence desarrollada por Microsoft, la cual permite transformar y visualizar datos en informes interactivos o en paneles de control dinámicos.

Se utiliza el software PowerBi para llevar a cabo la visualización de los resultados obtenidos a través de Python y las fuentes de datos abiertas utilizadas. Durante el proceso de análisis de datos, se muestran los resultados obtenidos respectivo de cada fuente de datos.

Capítulo 3

3. Herramientas esenciales.

Para lograr los objetivos del proyecto es necesario la utilización de diferentes tecnologías y herramientas a lo largo del proceso. Cada herramienta está relacionada con las diferentes fases del proyecto. Las fases del proyecto se pueden visualizar en la *figura 1*.

Para la fase de ETL de las diferentes fuentes de datos y posterior análisis de datos se utiliza Python [8], la versión 3.7, como lenguaje de programación y Visual Studio Code [9] como entorno de desarrollo, durante todo se van utilizando diferentes librerías que sirven como apoyo. Para el proceso de extracción, transformación y carga se utilizan las siguientes librerías:

- Pandas.
- Fitz.
- XlsxWriter.
- NumPy.
- Datetime.
- Camelot.

Para el proceso de análisis de los datos, se emplean diferentes modelos estadísticos y modelos de machine learning. Estos modelos se desarrollan a partir de una serie de librerías de Python. Dependiendo del modelo se usan unas u otras. En general se utilizan las siguientes librerías:

- Pandas.
- NumPy.
- StatsModels.
- Scikit-Learn.
- SciPy.
- MatPlotLib.
- TensorFlow.
- Pdarima.

Para el almacenamiento de datos durante el análisis de datos se utiliza un almacén de datos (Data Warehouse) utilizando la aplicación Excel [10] desarrollada por Microsoft y que forma parte de la suite de Microsoft Office.

Para la fase de visualización se utiliza el software PowerBi [11]. Para la fase de documentación, se utiliza la aplicación de procesamiento de texto Microsoft Word [12], que forma parte de la suite de Microsoft Office.

Se utiliza el software en línea Lucid Chart [13] para la realización de diferentes tipos de diagramas, que posteriormente son implementados en Microsoft Word como figuras.

Para alojar, gestionar el proyecto se utiliza la plataforma en la nube GitHub [14], que es una plataforma de desarrollo colaborativa para desarrolladores que permite alojar, gestionar y colaborar con proyectos almacenados y sus posibles versiones del proyecto. El proyecto esta almacenado en un repositorio donde se van gestionando las posibles versiones y las actualizaciones, el repositorio es el siguiente [15]:

<https://github.com/MarcosNavarro00/TFG>

Capítulo 4

4. Implementación.

En el capítulo 4 se desarrolla el sistema capaz de realizar todos los procesos. En primer lugar, se definen los ámbitos de los que se van a obtener las diversas fuentes datos abiertas de la plataforma, acto seguido se definen que fuentes de datos en concreto se van a utilizar y el porqué de su elección.

Se realiza una investigación sobre las posibles herramientas ETL que Python ofrece, considerando que herramientas funcionan de una manera más optima dependiendo de cómo este constituida la fuente de datos. Dependiendo del punto anterior se aplican unas herramientas ETL u otras a las fuentes de datos, alojando en un data warehouse en el resultado del proceso.

Una vez obtenidas las fuentes de datos abiertas, en las que se han realizado el proceso de limpieza y transformación se realiza el análisis a las fuentes de datos escogidas. Se realizan 3 tipos de análisis, análisis descriptivo, análisis exploratorio y análisis predictivo.

Ya con el análisis de datos realizado se visualizan tanto las fuentes de datos abiertas seleccionadas como los resultados del análisis utilizando diferentes técnicas de visualización apoyándose en el software PowerBi.

4.1. Tipos de Fuentes de Datos y Proyectos de Investigación.

A continuación, se presenta los diferentes campos en los que se realiza la investigación de diferentes fuentes de datos abiertas, para una posterior selección. Los diferentes ámbitos son “Smacite” y “Eugloh”.

4.1.1. Smacite.

Smacite [2] es un proyecto internacional financiado por la unión europea que tiene como objetivo fundamental abordar los problemas causados por las grandes ciudades en la actualidad. El proyecto está diseñado en función de una serie de objetivos, los cuales derivan de un objetivo común dividido en tres partes:

- Abordar el déficit de competencias de los técnicos e ingenieros proporcionándoles formación y educación.
- Mejorar la accesibilidad de los profesionales de las Smart Cities a formaciones de alta calidad a través de herramientas educativas adecuadas y del desarrollo de recursos de aprendizaje.
- Unificar los resultados obtenidos en el aprendizaje y así fomentar la movilidad en los países de la unión europea a los técnicos e ingenieros de Smart Cities.

Multitud de organizaciones aportan información de valor al proyecto en cuestión, pueden ir de un nivel más educativo como son universidades u organizaciones propias de la unión europea que están dirigidas a nivel de empresa. Las universidades relacionadas con el proyecto son:

- La universidad de Patras en Grecia.
- La propia universidad de Alcalá, España.
- La universidad autónoma de Madrid, España.
- La universidad de West Attica de Atenas, Grecia.
- La politécnica Ikastegia Txorierri de Bizkaia, España.

Para poder entender el proyecto de “Smacite” y entender sus objetivos es necesario saber que es el concepto de ciudad inteligente.

Las ciudades inteligentes utilizan tecnologías TIC para mejorar la calidad de vida de sus habitantes y hacer un uso eficiente de los recursos. Consiste en aplicar soluciones tecnológicas y digitales para solventar o mejorar la eficiencia, sostenibilidad, conectividad y calidad de servicios públicos.

Una de las características fundamentales de las Smart cities y parte fundamental del proyecto es que estas ciudades se basan en la recopilación de datos en tiempo real provenientes de diferentes tipos de sensores, cámaras y recopilatorios de datos. Estos datos son una parte fundamental a la hora de tomar decisiones informadas, para la optimización del funcionamiento de la ciudad y de la mejora de la calidad de vida del ciudadano.

La relación entre el concepto de Open Data y las ciudades inteligentes se produce cuando los datos abiertos se utilizan para llenar los sistemas de información y análisis. Al recabar información y presentarla de una manera accesible los gobiernos y otras organizaciones permiten que diferentes desarrolladores, investigadores y ciudadanos utilicen estos datos para realizar análisis y crear aplicaciones.

La recopilación de información en tiempo real y básicamente el concepto de Smart Cities abarca multitud de temas, como pueden ser demografía, educación, medioambiente, salud, empleo, economía, ...

En la actualidad, hay multitud de ciudades que se pueden denominar ciudades inteligentes, incluso muchas de estas ciudades se encuentran en nuestro país. Sin embargo, el estudio realizado por Juniper Research en 2022 reveló que Barcelona es la ciudad más inteligente del mundo, ocupando el tercer puesto, mientras que Shanghái es la primera en el ranking.

Para el proyecto el concepto de las ciudades inteligentes tiene una gran importancia, ya que nos permite, investigar y estudiar infinidad de fuentes de datos abiertas sobre una multitud de campos diferentes, los cuales pueden tener objetivos muy dispares. Gracias a estas ciudades, como las que hay en España o en toda la unión Europa se han obtenido las fuentes de datos abiertas que han hecho posible la creación de este proyecto.

Las fuentes de datos abiertas que se han empleado en el proyecto se han extraído de tres organizaciones diferentes, las cuales dos de ellas se fundamental en el concepto de ciudades inteligentes. Las dos organizaciones son:

- Por el gobierno de España.
- Por la unión europea, que son proporcionados por todos los países pertenecientes.
- Por el ayuntamiento de la ciudad de Madrid.
- Por el instituto nacional de estadística (INE).

4.1.2. EUGLOH.

Eugloh [1] tiene el significado de “Alianza Universitaria Europea para la Salud Global”. Se caracteriza por ser un proyecto financiado por la Unión Europea que tiene como objetivo la creación un gran conjunto de datos que se puedan utilizar para analizar y estudiar para una posterior aplicación en el mundo real que sirva para la optimización o la finalización de problemas. Esta gestionado por un consorcio de 9 universidades europeas, en las que pertenece la propia universidad de Alcalá, de ahí el interés de contribuir al proyecto de Eugloh. Las demás universidades que participan en el proyecto son las siguientes:

- La universidad de Paris-Saclay, de Francia
- La universidad de Lund, de Suecia.
- La universidad de Szeged, de Hungría.
- La universidad de Oporto, de Portugal.
- La universidad de Múnich, de Alemania.
- La universidad del ártico, de Noruega.
- La universidad de Novi Sad, de Serbia.
- La universidad de Hamburgo, de Alemania.



Figura 6. Mapa de las universidades relacionadas con Eugloh.

El proyecto de Eugloh tiene una serie de objetivos, donde algunos de ellos presentan similitudes al proyecto realizado. Estos objetivos que destacan son los siguientes:

1. Proporcionar a los expertos en el mundo de la salud un conocimiento y unas herramientas extras que ayuden u optimicen su labor de cara al ciudadano.
2. Contribuir al espacio europeo de educación, al espacio europeo de investigación y al espacio europeo de salud.
3. Actuar como fuerza propulsora a la hora de resolver diferentes aspectos de la salud global, que van desde la salud pública, las enfermedades emergentes, el cambio climático y los peligros ambientales.

Los dominios que abarca este proyecto tienen una gran similitud entre los dominios que abarcan las Smacite. El concepto de Smacite del punto anterior y el proyecto de Eugloh están altamente relacionados, ya que, las fuentes de datos proporcionadas que se utilizan en el proyecto están suministradas por las diferentes ciudades inteligentes de la unión europea.

Los campos que abarca el proyecto de Euglooh contribuyen a una mejor compresión del desafío de social de la salud global. Estos campos se pueden ver representados en la *Figura 7*, la imagen de la *figura 7* se ha obtenido de la página oficial del Euglooh.



Figura 7. Campos que aborda el proyecto Euglooh.

Como se puede observar los campos que aborda el proyecto de Euglooh están estrechamente relacionados con la salud global. También implica la relación de la tecnología con la salud, la ingeniería, la inteligencia artificial, el Big Data y la robótica con herramientas que se utilizan para lograr sus objetivos. Tal como pasa en el proyecto que se está desarrollando.

Teniendo en cuenta los objetivos y los campos que abarca Euglooh, mi proyecto personal puede aportar gran valor y funcionalidad. Ya que los proyectos proponen objetivos similares y los campos que se abordan son los mismos. En los siguientes puntos se explica cada una de las fuentes de datos abiertas que se han escogido y sus razones, pero una de las razones de algunas de ellas por la que se han seleccionado es para abarcar los mismos temas de salud global de Euglooh.

4.2. Fuentes de Datos Abiertas.

Se seleccionan diferentes fuentes de datos abiertas, a través de las diferentes páginas web que proporcionan las organizaciones gubernamentales. Se ha realizado una investigación de multitud fuentes de datos y se han seleccionado unas pocas para la realización del proyecto, la selección se ha realizado en función del cumplimiento de una serie de propiedades, las propiedades son las siguientes:

- En función del tamaño. Para poder trabajar con un conjunto de datos debe de tener un tamaño considerable, para que a la hora de realizar el análisis se obtengan resultados de valor.
- En función de la calidad de los datos. La base de datos debe de tener datos que se ajusten a la realidad y que se estén almacenados siguiendo el formato que con el cual esta presentado en la fuente de datos. Como las fuentes de datos se obtienen a través de instituciones gubernamentales suelen presentar una alta calidad en los datos.

- En función del tipo archivo de almacenamiento. Dependiendo con qué tipo de archivo estén almacenados los datos tendrán una serie de ventajas o desventajas suponiendo una mayor facilidad de acceso de e integración con el lenguaje de programación Python, el cual va a ser el encargado de realizar la transformación y posterior análisis.
- En función de la relación con los proyectos Eugloh y Smacite. Como el proyecto que se está realizando tiene como objetivo aportar funcionalidad a los proyectos europeos de Eugloh y Smacite, es necesario que las fuentes de datos seleccionadas estén relacionadas con los temas que se tratan en Eugloh y en Smacite, que estén relacionadas con todo lo que representa la salud global y las ciudades inteligentes.
- En función del grado de interés. Un aspecto que también es importante para seleccionar cada una de las fuentes de datos abiertas, es el interés de conocimiento que producen, tanto para el autor del proyecto como para sus lectores. Durante la investigación se buscan fuentes de datos que sean atractivas.

Tambien es necesario seleccionar las fuentes de datos en función de la facilidad de acceso a los datos, en función del costo de los datos y en función de la licencia de los datos. Pero estas tres características se solventan al ser fuentes de datos abiertas, ya que, una de las principales características del “Open Data” es que son fácilmente accesibles, no tienen ningún tipo de costo y tiene una licencia libre donde se pueden usar los datos para cualquier propósito.

Siguiendo estas propiedades se seleccionan las siguientes fuentes de datos abiertas, teniendo en cuenta en que formato se almacenan la información:

1. El histórico de incendios producidos en España entre los años 2006 hasta 2015. En formato PDF.
2. El histórico de concesiones de nacionalidad española, considerando su nacionalidad de origen y el sexo (masculino o femenino) entre los años 2010 y 2019. En formato xlsx.
3. La capacidad asistencial en España durante la crisis del Covid-19 en España. En formato CSV.
4. El catálogo de parques municipales en la ciudad de Madrid. En formato PDF.
5. La siniestralidad producida en las carreteras de la ciudad de Madrid, durante el año anterior (enero del 2022 hasta diciembre del 2022). En formato xlsx.
6. El histórico de la población de las provincias de España entre los años 1996 hasta 2021 y también el histórico en función del sexo (masculino o femenino). En formato xlsx.
7. El histórico de la población en los países pertenecientes a la unión europea entre los años 2001 hasta el 2022. En formato xlsx.
8. Catalogo del bosque urbano de la ciudad de Madrid. En formato PDF.

A continuación, se explican cada una de las fuentes de datos seleccionadas, como están constituidas, el porqué de su selección y la complejidad que alberga realizar un análisis.

4.2.1. Incendios producidos en España entre el 2006 y el 2015.

Esta fuente de datos está suministrada por el gobierno de España, más concretamente por el ministerio de agricultura, pesca y alimentación. Consiste en un archivo PDF en el que en su interior se encuentran diferentes tablas con información relevante, por ejemplo, en el Cuadro 2.1 del PDF, podemos encontrar una tabla con el número de siniestros y superficies afectadas del 1968 al 2015. En el PDF se explican multitud de aspectos relacionados con los incendios en España. Este tipo de fuente de dato pertenece a la categoría de biodiversidad y la fuente de datos se puede encontrar la página oficial del ministerio de agricultura, pesca y alimentación del gobierno de España [16]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- Por el motivo de interés que presentar determinar si existe relación entre los diferentes incendios que se han producido en nuestro país, y si es posible realizar un análisis que concluya la predicción de nuevos incendios o determinar ciertas zonas como zonas rojas, ya que, presentan mayor probabilidad de presenciar un incendio en un determinado espacio de tiempo.
- Porque el formato en el que se encuentra los datos es en un archivo PDF, por lo que, supone una mayor dificultad y por lo tanto un reto, extraer la información de valor, para más tarde proceder a un proceso de transformación y carga. Es una fuente de datos no estructurada, por lo que alberga un incremento de dificultad realizar los procesos de análisis.
- Por el cumplimiento de los objetivos del proyecto europeo Eugloh. Al realizar un análisis de este tipo de fuente de datos, se está aportando información de valor al proyecto, ya que, los incendios que se producen en España es uno de los ámbitos que se trata, más concretamente en el campo de clima, medio ambiente y biodiversidad, tal como podemos observar en la figura 7.

A continuación, se muestra una tabla extraída del documento PDF que sirve como ejemplo para mostrar cómo está definida la información de valor, por tanto, la información que se necesita realizar una extracción. En la tabla se muestra un resumen de siniestros y superficies afectadas entre los años 2006 al 2015.

Año	Total de siniestros			Superficies afectadas (hectáreas)				
	Conatos < 1 ha	Incendios ≥ 1 ha	Total	Superficie arbolada (ha)	Superficie no arbolada leñosa (ha)	Superficie herbácea (ha)	Superficie no arbolada (ha)	Superficie forestal total (ha)
2006	10.741	5.593	16.334	71.064,87	72.053,29	12.226,67	84.279,96	155.344,83
2007	7.523	3.413	10.936	29.408,86	42.394,77	14.318,40	56.713,17	86.122,03
2008	7.300	4.355	11.655	8.443,49	32.847,01	9.031,59	41.878,60	50.322,09
2009	9.866	5.777	15.643	40.402,48	67.495,97	12.195,76	79.691,73	120.094,21
2010	7.811	3.910	11.721	10.184,91	39.279,26	5.305,71	44.584,97	54.769,88
2011	10.815	5.599	16.414	18.847,52	72.387,82	10.925,99	83.313,81	102.161,33
2012	10.455	5.542	15.997	83.059,85	117.118,93	18.777,81	135.896,74	218.956,59
2013	7.708	3.089	10.797	17.704,26	33.086,49	10.899,86	43.986,35	61.690,61
2014	6.610	3.196	9.806	8.283,80	32.359,33	8.074,70	40.434,03	48.717,83
2015	7.685	4.125	11.810	32.877,09	64.889,91	12.015,85	76.905,76	109.782,85
Total	86.514	44.599	131.113	320.277,13	573.912,78	113.772,34	687.685,12	1.007.962,25
Media	8.651	4.460	13.111	32.027,71	57.391,28	11.377,23	68.768,51	100.796,23

Figura 8. Ejemplo de tabla extraída de la fuente de datos “Incendios producidos en España entre el 2006-2015”

4.2.2. Concesión de Nacionalidad Española entre el 2010 y el 2019.

Esta fuente de datos está suministrada por el gobierno de España, más concretamente esta suministrada por el ministerio español de inclusión, seguridad social y migraciones. Consiste en un archivo en formato xlsx, por lo tanto, es una fuente de datos estructurada, donde los datos presentan una gran calidad. En el archivo se muestra información en función de la nacionalidad previa del ciudadano y en función del sexo (masculino o femenino). La fuente de datos abierta pertenece a la categoría de demografía y la podemos encontrar en la página oficial de ministerio español de inclusión, seguridad social y migraciones del gobierno de España [17]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- Por el motivo de interés cultural. Es interesante saber de qué países hay más inmigración y como está relacionada y si es posible encontrar algún patrón o alguna conclusión de valor al realizar en análisis.
- Al ser el formato de la fuente de datos un xlsx es más simple realizar el proceso de extracción para posteriormente realizar proceso de transformación y carga. Es interesante abarcar todo tipo de formatos a la hora de realizar la extracción, de los más simples a los más complejos.
- Entra dentro del dominio del proyecto internacional Euglooh. La realización de un estudio sobre la concesión de nacionalidad de española tiene relevancia tanto en los campos de derecho de gestión y economía.

A continuación, se muestra un ejemplo de cómo se presenta la información en el archivo xlsx, corresponde a la primera página del archivo. Como se puede observar en la figura 9, el histórico de concesiones de nacionalidad se presenta en función del país del que sea previamente residente el ciudadano, donde cada país esta dividido por continente. También se presenta la información en función del sexo. Al estar la información presentada en formato tabla, al realizar el proceso de extracción utilizando Python es un proceso sencillo.

1. Evolución de las concesiones de nacionalidad española por residencia según sexo y nacionalidad anterior. 2010-2019

índice

	2019	2018	2017(*)	2016	2015	2014	2013	2012	2011	2010
Ambos sexos										
Total	162.799	92.501	25.924	93.760	78.000	93.714	261.295	115.557	114.599	123.721
Unión Europea	8.163	2.908	833	2.794	2.084	2.611	5.360	2.149	2.086	1.734
Bulgaria	1.054	444	127	428	352	364	604	149	138	82
Francia	135	56	22	67	35	59	164	79	75	73
Italia	940	260	70	259	167	199	448	150	162	135
Polonia	505	216	59	203	106	160	328	175	145	108
Portugal	596	377	135	477	341	496	1.265	830	884	800
Reino Unido	606	35	13	30	18	28	101	40	49	56
Rumanía	3.780	1.339	350	1.171	960	1.169	2.066	528	416	319
Otros Unión Europea	547	181	57	159	105	136	384	198	217	161
AELC ¹	26	7	2	18	7	9	43	22	17	23
Resto de Europa	6.482	2.597	617	1.894	1.473	1.826	3.425	1.029	928	814
Bielorrusia	227	79	14	45	40	59	116	28	33	33
Moldavia	993	357	97	320	208	262	420	91	98	57
Rusia	1.410	625	191	522	450	569	1.184	377	327	324
Serbia	185	59	21	58	37	70	163	64	59	39
Turquía	119	43	11	33	30	51	101	38	34	22
Ucrania	3.329	1.344	257	851	629	724	1.216	318	262	221
Otros Resto de Europa	219	90	26	65	79	91	225	113	115	118
África	45.575	33.536	7.823	31.724	25.269	25.884	59.938	20.352	18.333	13.828
Angola	55	24	15	31	25	31	86	35	38	35
Argelia	1.445	1.312	513	1.340	1.059	1.187	2.342	684	544	372
Burkina Faso	66	39	12	32	20	29	54	13	13	9
Cabo Verde	118	72	15	48	24	59	153	69	66	74
Camerún	323	138	48	112	103	117	318	101	93	61
Congo	55	29	15	56	59	91	263	117	97	85
Costa de Marfil	106	41	20	58	38	41	111	46	31	9
Egipto	154	111	21	91	55	97	224	83	78	85

Figura 9. Ejemplo de tabla extraída de la fuente de datos “Concesión de Nacionalidad entre el 2010 y el 2019”.

4.2.3. Capacidad Asistencial durante la Covid-19.

Esta fuente de datos está suministrada por el gobierno de España, más concretamente por el ministerio de sanidad. Consiste en un archivo CSV, en el que se muestra un histórico de datos entre el 1 de agosto de 2020 hasta la actualidad mostrando la capacidad asistencial frente al Covid-19 en función de provincias españolas. Es una fuente de datos estructurada, donde se caracteriza por la abundancia de datos (137.851 filas) y la calidad que presentan. La fuente de datos abierta pertenece a la categoría de salud y la podemos encontrar en la página oficial del ministerio de sanidad del gobierno de España [18]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- La crisis del Covid-19 tuvo un gran impacto que abarco todos niveles de la sociedad, por lo tanto, es interesante observar cómo va evolucionando la enfermedad vista desde una perspectiva de capacidad asistencial de nuestros hospitales. Además, es importante analizar el histórico para ver relaciones ocultas o descubrir nuevos conocimientos que nos ayuden a entender que realmente paso durante esta crisis.
- Al ser el formato de la fuente de datos un CSV también es más simple el proceso extracción. Es importante poder abarcar cuantos más tipos de archivos mejor.
- El tema del Covid-19 está estrechamente relacionado con los objetivos que quiere abarcar el proyecto europeo de Euglooh, por lo que, este análisis y las soluciones obtenidas implementan un gran valor al proyecto.

A continuación, se muestra un ejemplo de cómo se presenta la información en el archivo CSV, la información se muestra en función de la provincia española y del día. Se muestra las camas ocupadas en la UCI, las camas ocupadas que tienen relación con el Covid-19 y los ingresos y altas de cada día. Esta información no se muestra como tabla, sino como histórico, por lo que su extracción es simple por ser un archivo xlsx, pero su transformación tiene un nivel más de complejidad.

Fecha	COD_CCAA	CCAA	Cod_Provinc	Provincia	TOTAL_CAM_Ocupadas	Ocupadas_Covid19	Ingresos_Covid19	Altas_24h_COVID19
01/08/2020	7	CASTILLA Y LI	37	Salamanca	34	0	18	0
01/08/2020	12	GALICIA	27	Lugo	88	0	21	0
01/08/2020	1	ANDALUCÍA	4	Almería	1303	20	582	3
01/08/2020	2	ARAGÓN	44	Teruel	393	37	180	6
01/08/2020	2	ARAGÓN	50	Zaragoza	2905	264	1520	58
01/08/2020	9	CATALUÑA	17	Girona	2105	33	1468	3
01/08/2020	6	CANTABRIA	39	Cantabria	89	0	43	0
01/08/2020	12	GALICIA	15	Coruña, A	211	0	72	0
01/08/2020	1	ANDALUCÍA	18	Granada	38	0	18	0
01/08/2020	8	CASTILLA LA	16	Cuenca	11	0	0	0
01/08/2020	8	CASTILLA LA	19	Guadalajara	30	0	0	0
01/08/2020	11	EXTREMADU	10	Cáceres	21	0	1	0
01/08/2020	12	GALICIA	32	Ourense	33	0	14	0
01/08/2020	16	PAÍS VASCO	20	Gipuzkoa	0	0	0	0
01/08/2020	2	ARAGÓN	22	Huesca	526	33	272	3
01/08/2020	6	CANTABRIA	39	Cantabria	1444	14	860	3
01/08/2020	7	CASTILLA Y LI	47	Valladolid	962	23	605	4
01/08/2020	8	CASTILLA LA	13	Ciudad Real	1284	7	702	0
01/08/2020	1	ANDALUCÍA	11	Cádiz	162	0	40	0
01/08/2020	1	ANDALUCÍA	18	Granada	132	1	45	1
01/08/2020	16	PAÍS VASCO	20	Gipuzkoa	85	1	33	0
01/08/2020	7	CASTILLA Y LI	34	Palencia	0	0	0	0
01/08/2020	8	CASTILLA LA	13	Ciudad Real	6	0	3	0

Figura 10. Ejemplo del histórico extraído de la fuente de datos “Capacidad Asistencial durante el Covid-19”.

4.2.4. Catálogo de Parques Municipales en la Ciudad de Madrid.

Esta fuente de datos está suministrada por el ayuntamiento de la Madrid, más concretamente por el ministerio de medio ambiente y movilidad. Consiste en un archivo PDF, en el que cada página de valor se muestra información correspondiente a un parque de la comunidad de Madrid. Se trata de una fuente de datos no estructurada donde la información que se expone se expone sin ningún tipo de estructura. Se trata de una fuente de datos abundante ya que nos encontramos con un PDF de 206, en donde cada parque tiene más de 60 propiedades. La fuente de datos abierta pertenece a la categoría de medio ambiente y la podemos encontrar en la página oficial del ministerio de medio ambiente y movilidad del ayuntamiento de la Madrid [19]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- El aumento de la población en las grandes ciudades y como realizar estas ciudades los más “verdes” y sostenibles posibles en un reto de gran prioridad en la actualidad. La distribución de los diferentes parques a lo largo de la ciudad y como están constituidos es fundamental para abordar este tema. Es importante realizar un análisis que pueda aportar más conocimientos para una posterior toma de decisiones enfundada u optimización de los recursos utilizados en las ciudades.
- Al ser el formato de la fuente de datos un PDF mostrando la información en forma de texto sin la utilización en forma de tabla, hace que el método de extracción sea lo más complejo posible. Esta es una razón para elección de la base de datos porque se busca mostrar la capacidad de extracción de datos de todo tipo de fuentes de datos y como se encuentren estructuradas
- El tema de los parques y como están constituidos y sus propiedades abarca dominios que se corresponden con los proyectos Euglooh y Smacite. Por lo tanto, es una razón de peso para la selección de la fuente de datos.

A continuación, se muestra un ejemplo de cómo está distribuida la información en el archivo PDF. En la *figura 11* se puede observar la página 178 del PDF, donde muestra las propiedades del parque municipal del “Jardines del Buen Retiro”. Como se puede observar la información muestrada, se expone sin utilización de ninguna estructura, por lo que, el proceso de extracción es muy complejo, pero a través de la Python y sus multitudes de bibliotecas se puede realizar el proceso de extracción.

Código	Nombre de parque		Tipología	Parques Históricos	
RETIRO	Jardines del Buen Retiro		Superficie	1.177.300 m ²	
Distrito	Retiro	Barrio	Jerónimos		
Dirección	Plaza Independencia, 7 - 28001 Madrid		Descripción general:		
			Ubicación:		
					
Transporte Metro	Retiro, Atocha, Ibiza	Cercanías RENFE	Atocha, Renfe		
Transporte Bus	1, 2, 9, 14, 15, 19, 20, 26, 28, 32, 51, 5	Aparcamiento	NO		
Usos potenciales					
Eventos deportivos	NO	Ferias temáticas	NO	Fecha Creación	Siglo XVII
Educativo/divulgativo	SI	Festejos y celebraciones .	NO	Fecha Reforma	2021
Espectáculos	NO	Rodaje audiovisual.....	SI		
Grandes espacios abiertos .	NO	Vistas panorámicas	NO		
Otros:					
 Accesibilidad parques PARCIAL					
Servicios y equipamientos		ud	Valores singulares		
Zonas infantiles	SI	12 ud	Es BIC	SI	Vegetación
Zonas de mayores	SI	2 ud	Red de miradores	SI	Pradera natural: NO
Área canina	SI	1 ud	Micro reserva biodiversidad	NO	Césped: NO
Fuente ornamental	SI	19 ud	Otros:		Nº total de unidades arbóreas: 19.859 ud
Estanque/Lámina agua ..	SI	38 ud			Especies arbóreas destacadas: Ahuehuete, ciprés de los pantanos, plátano de sombra, pinos, tejo y castaño de Indias.
Instalaciones deportivas .	SI	4 ud			Superficie de macizos arbustivos: 27.450 m ²
Auditorio	NO	0 ud			Especies arbustivas destacadas:
Senda Ecológica	NO	0 km			
Carril bici	SI	2,5 km			
Aseo público	SI	5 ud			
Elementos arquitectónicos					

Figura 11. Ejemplo de cómo se expone la información en la fuente de datos “Catalogo de Parques Municipales de Madrid”.

4.2.5. Siniestralidad en la Carreteras en la Ciudad de Madrid.

Esta fuente de datos abierta esta suministrada por el gobierno de España, más concretamente por el ministerio de asuntos económicos y transformación digital. Consiste en archivo xlsx donde se muestra un histórico de siniestros en las carreteras de la ciudad de Madrid en el año 2022, donde cada siniestro se corresponde con una fila del archivo. Se trata de una fuente de datos estructurada con una cantidad de datos importante, en concreto existen 47.054 filas que es igual a 47.054 siniestros en el año de 2022. La fuente de datos abierta pertenece a la categoría de transporte y la podemos encontrar en la página oficial del ministerio de asuntos económicos y transformación digital del gobierno de España [20]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- El interés que causa saber el por qué y donde se producen más accidentes de tráfico en la ciudad de Madrid y en un posterior análisis, encontrar relaciones ocultas o conclusiones e incluso predicciones en la que su valor obtenido puedan tener una implicación real en la vida de los ciudadanos.
- Al ser el formato de la fuente de datos un xlsx es más simple realizar el proceso de extracción para posteriormente realizar proceso de transformación y carga.
- El dominio de la fuente de datos entra dentro del proyecto “Smacite”. Aplicando un conjunto de tecnologías y una serie de análisis a la información se puede tener un estudio que tenga un impacto real sobre el ciudadano, consiguiendo una ciudad más segura tanto para los habitantes.

A continuación, se muestra un ejemplo de cómo está constituida la fuente de datos en el archivo xlsx. Cada siniestro se corresponde a una fila de fuente de datos, cada siniestro tiene una serie de propiedades donde destacan la ubicación, el tipo de vehículo, el rango de edad, el sexo y si ha dado positivo en alcohol o drogas, tal y como se puede observar en la *figura 12*.

fecha	cod_distrito	distrito	tipo_accidente	estado_meteorologico	tipo_vehicul	tipo_persona	rango_edad	sexo	positiva_alcohol	positiva_droga
01/01/2022	13	PUENTE DE V Alcance	Despejado	Turismo	Conductor	De 30 a 34 a±	Mujer	N	NULL	
01/01/2022	13	PUENTE DE V Alcance	Despejado	Turismo	Conductor	De 45 a 49 a±	Hombre	N	NULL	
01/01/2022	3	RETIRO	Colisión fror NULL	Motocicleta	Conductor	De 30 a 34 a±	Hombre	S	NULL	
01/01/2022	3	RETIRO	Colisión fror NULL	Motocicleta	Pasajero	De 35 a 39 a±	Mujer	N	NULL	
01/01/2022	3	RETIRO	Colisión fror NULL	Turismo	Conductor	De 40 a 44 a±	Hombre	N	NULL	
01/01/2022	1	CENTRO	Atropello a p Despejado	Motocicleta	Conductor	Desconocido	Desconocido	N	NULL	
01/01/2022	1	CENTRO	Atropello a p Despejado	Motocicleta	Peatón	De 18 a 20 a±	Mujer	N	NULL	
01/01/2022	20	SAN BLAS-C/	Choque cont Despejado	Turismo	Conductor	De 50 a 54 a±	Hombre	N	NULL	
01/01/2022	20	SAN BLAS-C/	Choque cont Despejado	Turismo	Pasajero	De 55 a 59 a±	Mujer	N	NULL	
01/01/2022	12	USERA	Choque cont Despejado	Turismo	Conductor	De 45 a 49 a±	Hombre	S	NULL	
01/01/2022	12	USERA	Choque cont Despejado	Turismo	Pasajero	De 35 a 39 a±	Hombre	N	NULL	
01/01/2022	7	CHAMBERÍ	Choque cont Despejado	Turismo	Conductor	De 25 a 29 a±	Hombre	N	NULL	
01/01/2022	7	CHAMBERÍ	Choque cont Despejado	Turismo	Conductor	Desconocido	Desconocido	N	NULL	
01/01/2022	7	CHAMBERÍ	Choque cont Despejado	Turismo	Conductor	Desconocido	Desconocido	N	NULL	
01/01/2022	14	MORATALAZ	Choque cont NULL	Turismo	Conductor	De 21 a 24 a±	Mujer	N	NULL	
01/01/2022	20	SAN BLAS-C/	Choque cont Despejado	Furgoneta	Conductor	Desconocido	Desconocido	N	NULL	
01/01/2022	20	SAN BLAS-C/	Choque cont Despejado	Turismo	Conductor	De 15 a 17 a±	Mujer	S	NULL	
01/01/2022	10	LATINA	Colisión late Despejado	Turismo	Conductor	De 35 a 39 a±	Hombre	S	NULL	
01/01/2022	10	LATINA	Colisión late Despejado	Turismo	Conductor	Desconocido	Desconocido	N	NULL	
01/01/2022	20	SAN BLAS-C/	Choque cont Despejado	Turismo	Conductor	Desconocido	Desconocido	N	NULL	
01/01/2022	20	SAN BLAS-C/	Choque cont Despejado	Turismo	Pasajero	De 18 a 20 a±	Hombre	N	NULL	
01/01/2022	20	SAN BLAS-C/	Choque cont Despejado	Turismo	Pasajero	De 21 a 24 a±	Hombre	N	NULL	
01/01/2022	16	HORTALEZA	Choque cont Despejado	Turismo	Conductor	De 21 a 24 a±	Hombre	S	NULL	

Figura 12. Ejemplo de la composición de la fuente de datos “Siniestralidad en las Carreteras en la Ciudad de Madrid”.

4.2.6. Población por Provincias de España 1996-2021.

Esta fuente de datos esta suministrada por el instituto nacional de estadística (INE) [21]. Consiste en un archivo xlsx, en donde cada fila se corresponde a una provincia de España, recoge la población de cada provincia desde el año 1996 hasta el 2021. También la fuente de datos está dividida en función de sexo (masculino o femenino). Es una fuente de datos estructurada por lo que, proceso de extracción no tiene mucha complejidad. La fuente de datos la podemos encontrar en la página oficial del instituto nacional de estadística [22]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- En la actualidad existe un problema muy serio en España con la despoblación de ciertas zonas rurales y como toda esta se está moviendo a las ciudades importantes, por ello es interesante realizar un análisis sobre los datos y observar cómo van evolucionando cada

una de las provincias. También a través del análisis se pueden obtener conclusiones nuevas e incluso realizar una predicción de cómo será la evolución en los siguientes años.

- Al ser el formato de la fuente de datos un xlsx es más simple realizar el proceso de extracción para posteriormente realizar proceso de transformación y carga.
- Esta fuente de datos pertenece a la categoría de demografía. No solo España es el que está sufriendo este problema de despoblación de las zonas rurales, sino, que es un problema común en algunas ciudades europeas. Por lo tanto, esta fuente de datos y su posterior análisis puede contribuir tanto al proyecto europeo de Euglooh como al proyecto Smacite porque también afecta a las grandes ciudades, siendo un problema para las Smart Cities.

A continuación, se muestra un ejemplo de cómo está constituida la fuente de datos en el archivo xlsx. Las filas de la fuente de datos se corresponden a cada una de las provincias españolas y las columnas a los años, tal y como se puede observar en la *figura 13*.

	Total	2021	2020	2019	2018	2017	2016
Total	47.385.107	47.450.795	47.026.208	46.722.980	46.572.132	46.557.008	
02 Albacete	386.464	388.270	388.167	388.786	390.032	392.118	
03 Alicante/Alacant	1.881.762	1.879.888	1.858.683	1.838.819	1.825.332	1.836.459	
04 Almería	731.792	727.945	716.820	709.340	706.672	704.297	
01 Araba/Álava	333.626	333.940	331.549	328.868	326.574	324.126	
33 Asturias	1.011.792	1.018.784	1.022.800	1.028.244	1.034.960	1.042.608	
05 Ávila	158.421	157.664	157.640	158.498	160.700	162.514	
06 Badajoz	669.943	672.137	673.559	676.376	679.884	684.113	
07 Balears, Illes	1.173.008	1.171.543	1.149.460	1.128.908	1.115.999	1.107.220	
08 Barcelona	5.714.730	5.743.402	5.664.579	5.609.350	5.576.037	5.542.680	
48 Bizkaia	1.154.334	1.159.443	1.152.651	1.149.628	1.148.302	1.147.576	
09 Burgos	356.055	357.650	356.958	357.070	358.171	360.995	
10 Cáceres	389.558	391.850	394.151	396.487	400.036	403.665	
11 Cádiz	1.245.960	1.244.049	1.240.155	1.238.714	1.239.435	1.239.889	
39 Cantabria	584.507	582.905	581.078	580.229	580.295	582.206	
12 Castellón/Castelló	587.064	585.590	579.962	576.898	575.470	579.245	
13 Ciudad Real	492.591	495.045	495.761	499.100	502.578	506.888	
14 Córdoba	776.789	781.451	782.979	785.240	788.219	791.610	
15 Coruña, A	1.120.134	1.121.815	1.119.596	1.119.351	1.120.294	1.122.799	
16 Cuenca	195.516	196.139	196.329	197.222	198.718	201.071	
20 Gipuzkoa	726.033	727.121	723.576	720.592	719.282	717.832	
17 Girona	786.596	781.788	771.044	761.947	755.716	753.576	
18 Granada	921.338	919.168	914.678	912.075	912.938	915.392	

Figura 13. Ejemplo de la fuente de datos “Población por Provincias de España”.

4.2.7. Población por Países en la unión europea 2001-2022.

Esta fuente de datos esta suministrada por la oficina estadística oficial de la unión europea (Eurostat) [23]. Consiste en un archivo xlsx en donde se muestran la población en cada país perteneciente a la unión europea desde el año 2001 hasta el 2022. Se trata de una fuente de datos estructurada, donde el proceso de extracción a través de Python tiene una complejidad sencilla. La fuente de datos la podemos encontrar en la página oficial de la oficina estadística oficial de la unión europea [24]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- Es interesante observar cómo van creciendo los diferentes países de la unión europea en términos de población.
- Al ser el formato de la fuente de datos un xlsx es más simple realizar el proceso de extracción para posteriormente realizar proceso de transformación y carga.
- El análisis y el estudio de esta fuente de datos, comparte objetivos con el proyecto europeo Euglooh, en concreto en el dominio de la demografía. Por lo tanto, este estudio sirve de implementación a este proyecto más grande.

A continuación, se muestra un ejemplo de cómo está constituida la fuente de datos en el archivo xlsx, tal y como se puede observar en la *figura 14*.

GEO (Labels)	TIME	2001	2002	2003	2004	2005	2006	2007	2008	2009
European Union - 27 countries (from)										
Belgium	429.240.746	b	429.723.142	431.190.184	432.762.039	434.416.272	435.816.236	437.227.496	438.725.386	b 440.047.692
Bulgaria	10.263.414		10.309.725	10.355.844	10.396.421	10.445.852	10.511.382	10.584.534	10.666.866	10.753.080
Czechia	8.149.468		7.868.815	7.805.506	7.745.147	7.688.573	7.629.371	7.572.673	7.518.002	7.467.119
Denmark	10.232.027		10.201.182	10.192.649	10.195.347	10.198.855	10.223.577	10.254.233	10.343.422	10.425.783
Germany (until 1990 former territory)	5.349.212		5.368.354	5.383.507	5.397.640	5.411.405	5.427.459	5.447.084	5.475.791	5.511.451
Estonia	82.259.540		82.440.309	82.536.680	82.531.671	82.500.849	82.437.995	82.314.906	82.217.837	82.002.356
Ireland	1.392.720		1.383.510	1.375.190	1.366.250	1.358.850	1.350.700	1.342.920	1.338.440	1.335.740
Greece	3.832.783		3.899.702	3.964.191	4.028.851	4.111.672	4.208.156	4.340.118	4.457.765	4.521.322
Spain	10.835.988		10.888.274	10.915.770	10.940.369	10.969.912	11.004.716	11.036.008	11.060.937	11.094.745
France	40.665.545		41.035.278	41.827.838	42.547.451	43.296.338	44.009.971	44.784.666	45.668.939	46.239.273
Croatia	60.979.315		61.424.036	61.884.088	62.292.241	62.772.870	63.229.635	63.645.065	64.007.193	64.350.226
Italy	4.295.406	b	4.305.494	4.305.384	4.305.725	4.310.861	4.312.487	4.315.530	4.311.967	4.309.796
Cyprus	56.960.692		56.987.507	57.130.506	57.495.900	57.874.753	58.064.214	58.223.744	58.652.875	59.000.586
Latvia	697.549		705.539	713.720	722.893	733.067	744.013	757.916	776.333	796.930
Lithuania	2.353.384		2.320.956	2.299.390	2.276.520	2.249.724	2.227.874	2.208.840	2.191.810	2.162.834
Luxembourg	3.486.998		3.454.637	3.431.497	3.398.929	3.355.220	3.289.835	3.249.983	3.212.605	3.183.856
Hungary	10.200.298		10.174.853	10.142.362	10.116.742	10.097.549	10.076.581	10.056.158	10.045.401	10.030.975
Malta	391.415		394.641	397.296	399.867	402.668	404.999	405.616	407.832	410.926
Netherlands	15.987.075		16.105.285	16.192.572	16.258.032	16.305.526	16.334.210	16.357.992	16.405.399	16.485.787
Austria	8.020.946		8.063.640	8.100.273	8.142.573	8.201.359	8.254.298	8.282.984	8.307.989	8.335.003
Poland	38.253.955		38.242.197	38.218.531	38.190.608	38.173.835	38.157.055	38.125.479	38.115.641	38.135.876
Portugal	10.330.774		10.394.669	10.444.592	10.473.050	10.494.672	10.511.988	10.532.588	10.553.339	10.563.014
Romania	22.430.457		21.833.483	21.627.509	21.521.142	21.382.354	21.257.016	21.130.503	20.635.460	20.440.290
Slovenia	1.990.094		1.994.026	1.995.033	1.996.433	1.997.590	2.003.358	2.010.377	2.010.269	b 2.032.362
Slovakia	5.378.783		5.378.951	5.374.873	5.371.875	5.372.685	5.372.928	5.373.180	5.376.064	5.382.401
Finland	5.181.115		5.194.901	5.206.295	5.219.732	5.236.611	5.255.590	5.276.655	5.300.484	5.326.314
Sweden	8.882.792		8.900.128	8.940.768	8.975.670	9.011.392	9.047.752	9.112.357	9.182.927	9.256.347
Iceland	283.361		286.575	288.471	290.570	293.577	299.891	307.672	315.459	319.368
Liechtenstein	32.863		33.525	33.863	34.294	34.600	34.905	35.168	35.356	35.589
Norway	4.503.436		4.524.066	4.552.252	4.577.457	4.606.363	4.640.219	4.681.134	4.737.171	4.799.252
Switzerland	7.204.055		7.255.653	7.313.853	7.364.148	7.415.102	7.459.128	7.508.739	7.593.494	7.701.856
United Kingdom	58.999.781		59.239.564	59.501.394	59.793.759	60.182.050	60.620.361	61.073.279	61.571.647	62.042.343
Montenegro	605.988		608.460	610.510	612.214	613.420	613.109	614.824	615.543	617.157

Figura 14. Ejemplo de la fuente de datos “Población por Países de la unión Europa”.

4.2.7. Catálogo del Bosque Urbano de la Ciudad de Madrid.

Esta fuente de datos está suministrada por el ayuntamiento de la ciudad de Madrid. Consiste en un archivo PDF, donde expone la situación de los bosques de la ciudad de Madrid, explicando sus propiedades, como pueden ser explicar las diferentes especies de árboles que hay, el beneficio de los árboles para la ciudad, el almacenamiento de carbono.

Esta fuente de datos destaca por tener una estructura de datos semiestructurada, ya que, los datos se presentan con tablas, gráficos y demás imágenes, pero no es estructurada porque presenta la información usando diferentes herramientas. Al final del documento se expone una tabla de gran dimensión sobre el listado de especies (página 64), que contiene gran valor para el documento. La fuente de datos la podemos encontrar en la página oficial del ayuntamiento de la ciudad de Madrid [25]:

Se ha seleccionado esta fuente de base de datos abierta por las siguientes razones:

- Tal y como se exponía en el punto 4.2.4 con el catálogo de parques municipales de Madrid, es fundamental tener en las ciudades cierta vegetación que ayude a la contaminación de la ciudad y de calidad de vida a la población. En este documento además de eso se expone cómo funcionan y que recursos necesitan cada una de las especies de los bosques, por lo tanto, al realizar un análisis se pueden lograr conclusiones y relaciones que fundamenten el tipo de especie a utilizar en cada momento para ayudar al problema de contaminación u optimizar los recursos de la población.
- Al ser el formato de la fuente de datos un archivo PDF donde tiene una estructura semiestructurada, dependiendo de que gráfico, tabla o imagen se quiera extraer la información va a tener un proceso u otro, lo que significa que van a tener diferentes grados de dificultad, por lo que, se abarca más posibilidades.
- Este tema está estrechamente relacionado tanto con el tema de Smacite como con el proyecto Eugloh. Por lo que, es una razón más de peso para seleccionar esta fuente de datos. Como se ha dicho anteriormente realizar un correcto análisis de datos que aparecen

en el documento puede suponer una optimización de los recursos y aumento de la calidad de vida de la población y este tema entra directamente en el ámbito de la salud global y medioambiente.

A continuación, se muestra un ejemplo de cómo está constituida la fuente de datos en el archivo PDF. Dependiendo de la página en donde se quiera extraer la información se usa un proceso diferente porque, puede que la información esté expuesta en gráfico, en tabla o en imagen, incluso en una misma página puede estar mostrada utilizando varios métodos, tal y como se puede observar en la *figura 15*. La *figura 15* se corresponde a la página 51 del archivo PDF

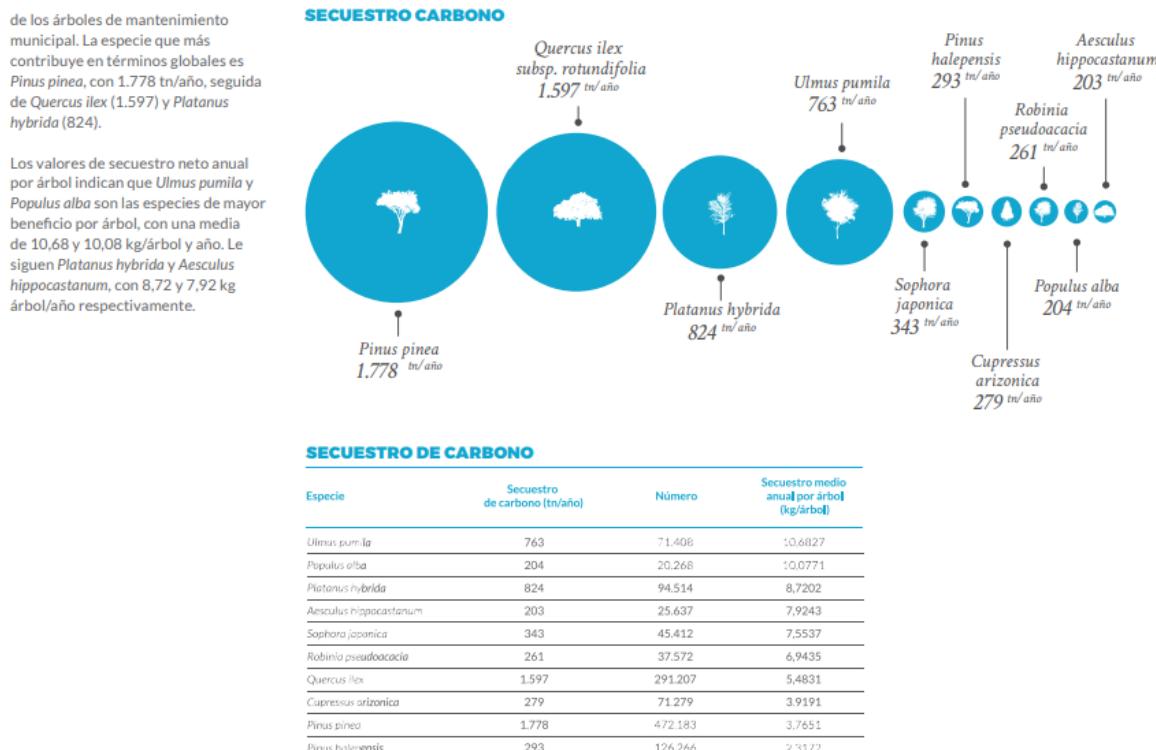


Figura 15. Ejemplo de la exposición de la información de la fuente de datos “Valor del Bosque Urbano de Madrid”, de la página 51 del PDF.

4.2.8. Clasificación de las diferentes fuentes de datos abiertas.

Como se ha puesto en los puntos anteriores, cada fuente de datos abierta seleccionada tiene una serie de propiedades por las que han sido seleccionadas para el proyecto. Donde las propiedades más características son las siguientes:

- Dependiendo si el tema de la fuente de datos está relacionado con algunos de los proyectos que se tienen en consideración (Smacite, Euglo o los dos).
- Dependiendo del tipo de estructura que tenga la fuente de datos (estructural, no estructural o semi estructural).
- Dependiendo del tipo proceso de extracción que se aplica a la fuente de datos utilizando el lenguaje de programación Python (grado de complejidad).
- Dependiendo del tipo de archivo de donde estén almacenados las diferentes fuentes de datos abiertas (PDF, xlsx, CSV).

- Dependiendo de que organismo gubernamental haya puesto la fuente de datos a disposición de todos los ciudadanos (gobierno de España, INE, Eurostat, ayuntamiento de Madrid).

Con esta serie de propiedades se puede clasificar cada una de las fuentes de datos seleccionadas, por ello, se constituye la siguiente tabla:

Nombre	Temática	Tipo de Estructura	Complejidad de Extracción	Tipo de Archivo	Suministrador
Incendios producidos en España entre el 2006 y el 2015.	Eugloh.	Semi Estructurada	Media	PDF	Gobierno de España
Concesión de Nacionalidad Española entre el 2010 y el 2019	Eugloh.	Estructurada	Baja	xlsx	Gobierno de España
Capacidad Asistencial durante la Covid-19	Eugloh.	Estructurada	Baja	CSV	Gobierno de España
Catálogo de Parques Municipales en la Comunidad de Madrid	Smacite y Eugloh	No estructurada	Alta	PDF	Ayuntamiento de Madrid
Siniestralidad en la Carreteras en la Ciudad de Madrid	Smacite	Estructurada	Baja	xlsx	Gobierno de España
Población por Provincias de España 1996-2021	Smacite y Eugloh	Estructurada	Baja	xlsx	INE.
Población por Países en la unión europea 2001-2022	Eugloh	Estructurada	Baja	xlsx	Eurostat
Catálogo del Bosque Urbano de la Ciudad de Madrid	Smacite y Eugloh	Semi Estructurada	Media	PDF	Ayuntamiento de Madrid

Tabla 1. Calificación de las diferentes fuentes de datos seleccionadas.

4.3. Proceso ETL en las Fuentes de Datos.

Una vez seleccionada las fuentes de datos abiertas, se seleccionan en este caso 6 fuentes de datos de las anteriores a las que aplicar este proceso de extracción, transformación y carga. Se seleccionan seis fuentes de datos, ya que, muchas de ellas presentan muchas similitudes a la hora de realizar el proceso, y no aportan información añadida de valor al proyecto.

Las fuentes que se seleccionan para aplicar el proceso y el porqué de su selección son las siguientes:

1. Catálogo de Parques Municipales en la Comunidad de Madrid. Se selecciona esta fuente de datos debido a la complejidad y el reto que supone aplicar el proceso ETL en una fuente de datos no estructurada, aportando conocimientos de valor al proyecto, abarcando todo tipo de estructuras de fuentes de datos.
2. Capacidad Asistencial durante la Covid-19. Se selecciona para ver, aunque tenga menos complejidad, como se produce el proceso ETL en una fuente de datos estructurada, además siendo el tipo de archivo un CSV.

3. Incendios producidos en España entre el 2006 y el 2015. Se selecciona esta fuente de datos para realizar el proceso ETL en una fuente de datos semi estructura y ya así, abarcar todas las estructuras posibles de fuentes de datos.
4. Concesión de Nacionalidad Española entre el 2010 y el 2019. Se selecciona esta fuente de datos para abarcar también más tipos de archivos, en este caso un archivo xlsx.
5. Catálogo del Bosque Urbano de la Ciudad de Madrid. Se selecciona esta fuente de datos también debido a su estructura semi estructurada y porque engloba el concepto de “Smacite” y el proyecto europeo “Euglooh”.
6. Población por Provincias de España 1996-2021. Se selecciona esta fuente de datos por el interés y la problemática actual sobre la pérdida de población en la zona rurales.

4.3.1. Proceso ETL en Catálogo de Parques Municipales en la ciudad de Madrid.

Como se ha podido observar en el punto 4.2.4 esta fuente de datos abiertas no tiene ningún tipo de estructura que defina la clasificación de la información, en cada página del PDF la información se muestra de una forma diferente y en una posición diferente por ello el proceso de ETL tiene una mayor complejidad. EL proceso ETL se ha llevado a cabo a través de Python y la utilización de las siguientes librerías:

- *Fitz* [26]. Esta librería tiene un grandísimo poder. A través de esta librería se puede manipular archivos PDF dando la posibilidad de realizar las siguientes acciones:
 - Leer texto de un archivo PDF, se puede leer todo el texto del archivo o indicar cual quieras concretamente. Para ello es necesario indicar las coordenadas de donde se quiere extraer el texto e indicar la página de donde se quiere extraer.
 - Extraer imágenes del PDF. Funciona igual que el punto anterior, indicando las coordenadas de la imagen y la pagina en cuestión.
 - Convertir archivos PDF a otros formatos.
 - Introducir imágenes o texto al archivo PDF.
 - Firmar digitalmente un archivo.
 - Encriptar el archivo con una contraseña.
- *XlsxWriter* [27]. Esta librería sirve para generar archivos de Excel en formato xlsx. Permite crear el archivo, escribir en él, crear gráficos y tablas y guardar una hoja de cálculo.

Lo primero que se hace en el programa desarrollado es abrir el archivo PDF utilizando la biblioteca *Fitz*, para poder usar todas sus ventajas. Después a partir de la página 7, que es donde empieza mostrarte la información de valor, y teniendo en cuenta que la pagina sea impar, ya que en todas las páginas impares es donde se muestra la información mientras que en las pares solo aparecen imágenes de los diferentes parques, se obtiene todo el texto de esa página.

El texto de cada página se obtiene a partir de las coordenadas de en la que se encuentra la información relevante, se sabe que coordenadas son ya que la librería *Fitz* lo indica. Utilizando

las coordenadas no se extrae todo el texto de la página, sino la información que se considera relevante. Con todo el texto relevante de la página guardado en una lista se quitan los puntos y asteriscos que son innecesarios. Esta funcionalidad se implementa con la función “`get_text()`”.

```
def get_text(words):
    lista = [658.5527954101562, 746.0987548828125, 767.6507568359375, 778.4267578125, 800.9547729492188, 735.32275390625, 756.874755859375, 674.0368041992188, 826.19970703125, 889.2657470703125, 799.8308715820312, 790.2230224609375, 770.2589758203125, 682.7987670898438, 663.8447875976562, 656.0878295898438, 684.3748168945312, 703.2327880859375, 722.0908203125, 740.948693.8038330078125, 712.6618041992188]
    text = []
    word = ""
    #
    for i in range (len(words)):
        if ( words[i][4] != '.' and words[i][4] != '..' and words[i][4] != '**' ): #quita los puntos y asteriscos
            if (words[i][1] not in lista): #comprueba que estan las coordenadas de informacion no relevante
                if (words[i][7] != 0):
                    word = word + " " + words[i][4]

            else:
                text.append(word)
                word = words[i][4]

    return text #devuelve el texto con cadenas unidas limpiado del pdf
```

Código 1. Función `get_text` del proceso de extracción de la fuente de datos “Catalogo de Parques Municipales de la Ciudad de Madrid”.

Ya extraído el texto de cada página se procede a la limpieza, se borra toda la información que no es relevante y aun así se ha extraído y se elimina multitud de elementos que se han almacenado sin que fueran parte del texto. Después se quitan multitud de caracteres que estropean los datos para poder trabajar con ellos, por ejemplo, son *km*, *ud*, *m^2*, ... Cuando se han limpiado la información de la página se repite el proceso extracción y limpieza con todas las páginas que albergan información para después guardar toda información limpia de todas las páginas en una lista.

```
def clean (texto):
    for i in range (67):#se borra lo que no es necesario
        texto.pop(0)

    texto = [texto[0],texto[2],texto[5],texto[4],texto[6],texto[1],texto[3],texto[7],texto[8],texto[9],texto[10],texto[11],
    texto[17],texto[18],texto[19],texto[21],texto[24],texto[29],texto[32],texto[35],texto[37],texto[39],
    texto[41],texto[43],texto[46],texto[48],texto[50],texto[52],texto[54],texto[22],texto[25],texto[30],texto[26],texto[27],
    ,texto[33],texto[44],
    ]

    #Es bic 23, red de miradores 26, pradera 27, Césped 28, Micro reserva biodiversidad 31, Nº total de unidades arbóreas 3
    for i in range(len (texto)):
        if ('m²' in texto[i]):#se quitan el % para poder trabajar con los datos
            texto[i]= texto[i].replace('m²', '')
        if ('ud' in texto[i]):#se quitan el % para poder trabajar con los datos
            texto[i]= texto[i].replace('ud', '')
        if ('.' in texto[i]):#se quitan el . para poder trabajar con los datos
            texto[i] = texto[i].replace('.', '')
        if (',' in texto[i]):#se quitan el , para poder trabajar con los datos
            texto[i]= texto[i].replace(',', '')
        if ('km' in texto[i]):#se quitan el , para poder trabajar con los datos
            valor= texto[i].replace('km', '')
            texto[i] = float(valor)*1000

    print (texto)
    return texto
```

Código 2. Función `clean` del proceso de transformación de la fuente de datos “Catalogo de Parques Municipales de la Ciudad de Madrid”.

Una vez ya realizado los procesos de extracción y transformación, se procede al proceso de carga de los datos en el data warehouse. Para ello se utiliza la función “`export_xlsx()`”, en la que se definen las cabeceras de las columnas del archivo xlsx en el que se va a guardar la información, a continuación, se crea el archivo xlsx y se introduce las cabeceras y posteriormente todo el texto limpio de todas las páginas del PDF. Para tratar con el archivo xlsx se usa la librería *XlsxWriter*.

```

def export_xlsx(tablas):
    csv_columns = ['Código', 'Nombre de parque', 'Distrito', 'Barrio', 'Dirección', 'Tipología', 'Superficie', 'Transporte Metro', 'Eventos deportivos', 'Ferias temáticas', 'Educavo/divulgavo', 'Festejos y celebraciones', 'Espectáculos', 'Rodaje audiovisual', 'Vistas panorámicas', 'Accesibilidad parques', 'Zonas infantiles', 'Zonas de mayores', 'Área canina', 'Fuente ornamental', 'Carril bici', 'Aseo público', 'Mirador', 'Jardín Botánico', 'Bancos', 'Papeleras', 'Es BIC', 'Red de miradores', 'Micro reserva', '# de total de unidades arbóreas', 'Superficie de macizos arbustivos']
    ]
    if (tablas[0] != csv_columns):#puede que ya esten introducida la cabecera
        tablas.insert(0,csv_columns)

    workbook = xlsxwriter.Workbook('sample_data4.xlsx')
    sheet = workbook.add_worksheet()

    for row in range (len(tablas)):
        for column in range (len(tablas[row])):
            sheet.write((row), column, tablas[row][column])

    workbook.close()

```

Código 3. Función export_xlsx del proceso de carga de la fuente de datos “Catalogo de Parques Municipales de la Ciudad de Madrid”.

Ya realizado el proceso ETL en la fuente de datos, parte del data warehouse se visualiza de la siguiente manera, donde una fila del documento se corresponde con una de las páginas como la que se puede visualizar en la *figura 16*.

Código	Nombre	d	Distrito	Barrio	Dirección	Tipología	Superficie	Transporte	Cercanías	Transporte	Aparcamiento	Eventos	Feria	de	Educavo/	Festejos	y Espectáculos	Rodaje	audiovisual	Grandes	e	Vistas	par
2001021	Parque En Arganzuel	L	Legazpi		Calle Men Parques d'400600	NO	NO	148	Avd del PISI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI
10237000	Parque de Latina	L	Las Águila	Avenida P	Parques d'452090	Aluche	Aluche	17. 117. 12	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO
13050020	Parque Lir Puente de Palomera	L	Avenida A	Parques d'394496	Miguel He No	54. 58. 103	NO	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO
12007030	Parque de Usera	L	Pradolong	Avenida R	Parques d'597000	Hospital	1 12 de Oct	6. 60 . 78	Calle del I SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	
19121040	Cuña Verc Vicálvaro	L	Ambroz	Avenida D	Parques d'41293	Vicálvaro	Vicálvaro	E3. 100	NO	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	NO	
19122020	Cuña Verc Vicálvaro	L	Ambroz	Avenida D	Parques d'115119	San Cipriá	Vicálvaro	4. E3. 100	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	
19123020	Cuña Verc Vicálvaro	L	Ambroz	Avenida C	Parques d'127835	Avenida d	Vicálvaro	4. 106. 155	Av de Can SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI	SI	SI	NO	
19124020	Cuña Verc Vicálvaro	L	Ambroz	Av de Dari	Parques d'2725	Vicálvaro	Vicálvaro	E3. 100	NO	NO	NO	SI	NO	NO	NO	SI	NO	SI	NO	NO	SI	SI	
19125020	Cuña Verc Vicálvaro	L	Ambroz	Av de Dari	Parques d'32350	San Cipriá	Vicálvaro	4. E3. 100	NO	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	
PA-JPH	Parque Ju	Hortaleza	Piovera	Avenida N	Parques d'282212	NO	NO	112. 122	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	
PA-JCI	Parque Ju Barajas	J	Corralejos	Gta de Do	Parques d'1507461	Feria de N	NO	104. 112. 1	Parking Al SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	
PA-LM	Parque Lir Usera	L	Pte San Fermí	Autovía A	Parques d'524000	Doce de C NO		23. 123. 7	San Fermí SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	
PA-MR	Madrid Rí	Varios	Varios	Paseo de	Parques d'1057100	Legazpi, P	Madrid-Pr6.	18. 25	3 SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	TOTAL	
1004032	Parque de Centro	L	Palacio	Calle Segc	Parques U46472	Príncipe P	Príncipe P	3. 31 . 41	. NO	NO	SI	SI	SI	SI	NO	SI	NO	SI	NO	SI	NO	NO	
1011031	Jardines d Calle Mon Centro	L	La Latina..	Parques U	Palacio	NO	3. 31. 41.	5 NO	NO	SI	SI	SI	SI	NO	SI	NO	SI	NO	SI	SI	PARCIAL		
1080042	Parque En Centro	L	Palacio	Cuesta Ra	Parques U3942	NO	NO	3. 31 . 41 .	NO	NO	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI		
2067030	Parque Pe Arganzuel	L	Acacias	Calle Garg	Parques U27402	Pirámides	Pirámides	18. 62	Sobre parl SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	
3002030	Parque Rc Retiro	L	Estrella	Calle Juan Parques U112052	Sainz de B NO	30. . 56 . 14	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO
3038040	Parque Lut Retiro	L	Adefnas	Calle Cerr	Parques U33671	Conde de NO	8. 10 . 24 .	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO
4004030	Parque Sa Salamanc	L	Fuente De Plaza Amé	Parques U43053	Ventas	NO	12	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO
4003030	Parque de Salamanca	L	Guindalera	Avenida B	Parques U47026	Parque de NO	53. 122. 74	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO
4002032	Parque Mi Salamanca	L	Guindalera	Calle Doct	Parques U29748	Manuel Br NO	12. 43 . 56	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO
4017044	Jardines d Salamanca	L	Recoleto	Calle Goyi	Parques U20347	Colón y Se NO	1. 5. 9. 14.	Parking PI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	
5001030	Parque de Chamartín	L	Ciad Jardí	Avenida P	Parques U49116	Concha Es NO	16. 29. 43	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	

Figura 16. Parte del data warehouse una vez realizado el proceso ETL.

4.3.2. Proceso ETL en Capacidad Asistencial durante la Covid-19.

Consiste en una fuente de datos estructura, en la cual se albergan muchísimos datos. Se realiza el proceso ETL para obtener la información que se ajuste a nuestros objetivos y así reducir el tamaño de los datos que se están tratando. En el archivo se tiene en cuenta todas las comunidades autónomas de España, pero solo se va a tener en cuenta la comunidad de Madrid. También solo se va a considerar las fechas de cada suceso, las camas totales que estén disponibles, cuantas camas están ocupadas por enfermos de la Covid-19 y cuantos ingresos se corresponden a la enfermedad.

El objetivo es extraer todos los sucesos correspondientes con la comunidad de Madrid y entre el 1 de agosto de 2020 hasta el 1 de abril del 2021. Se escoge este rango de fechas, ya que se considera que fueron los meses con mayor impacto de la enfermedad durante la pandemia.

Aunque se trate de una fuente de datos estructurada el proceso ETL, no va a ser sencillo, ya que hay multitud de datos que se van a desechar y la información de valor está escondida entre tanto dato.

Para la realización del proceso ETL ha sido necesario la implementación de las siguientes librerías:

- Pandas [4]. Es una librería famosa para el análisis de datos. Permite estructurar datos, manipularlos, analizarlos y visualizarlos usando gráficos y tablas. Se caracteriza por ser una herramienta muy flexible.
- NumPy [6]. Es una librería que se utiliza para el procesamiento numérico y científico de datos. Tiene multitud de acciones de gran relevancia, las más famosas son las siguientes:
 - La principal característica de NumPy es el “*ndarray*”, es una matriz multidimensional homogénea y de tamaño fijo, que, a diferencia de las listas de Python, los arrays de NumPy son mejores para almacenar grandes conjuntos de datos y se pueden realizar operaciones vectorizadas, lo que provoca un mejor rendimiento.
 - Tiene una amplia gama de funciones matemáticas que se pueden aplicar a los arrays, están optimizadas para trabajar sobre un gran conjunto de datos.
- Datetime [28]. Proporciona una serie de clases y funciones para trabajar con todo tipo de formatos de fechas y horas. Facilita la manipulación, formateo y cálculo de fechas y horas.

El programa desarrollado en Python esta dividido en diferentes fases, donde cada fase se corresponde con los procesos de extracción, transformación y carga.

Lo primero que se realiza, es la importación del archivo xlsx a través de la librería *Pandas*. Después se procede con la extracción de la información de valor.

Primero se extrae todos los datos correspondientes con la comunidad de Madrid y en función de un rango de fechas, para extraer la información según el rango de fechas se utiliza la librería *Datetime*. El rango de fechas es del 1 de agosto del 2020 hasta el 1 de abril del 2021. Se escoge este rango debido a que fue el punto más álgido de la pandemia, por lo tanto, el punto donde más datos interesantes existen. Este proceso se hace a través de la función “*getByCCAAByData ()*”, recorre todos los datos importados y almacena en una nueva lista los datos que estén en el rango de fechas y que cumplan la condición de ser de la comunidad de Madrid.

Después utilizando la función “*getData ()*” se extrae la información correspondiente a la fecha, al total de camas, las camas ocupadas por paciente de Covid-19 y a los ingresos asociados con la enfermedad, generando cuatro listas de igual tamaño.

```
def getByCCAAByDate(data):
    dt1 = datetime(2020, 8, 1)
    dt2 = datetime(2021, 4, 1)
    for i in range (len(data)):
        if (data[i][0] >= dt1 and data[i][0] <= dt2):
            if (data[i][1] == 13):
                CCAA.append(data[i])

    return CCAA

def getData(CCAA):
    fecha = []
    ocupadas_covid = []
    total_camas = []
    ingresos = []
    for i in range (len(CCAA)):
        fecha.append(CCAA[i][0])
        total_camas.append(CCAA[i][5])
        ocupadas_covid.append(CCAA[i][6])
        ingresos.append(CCAA[i][8])
    return fecha,total_camas,ocupadas_covid,ingresos
```

Código 4. Proceso de extracción de la fuente de datos “Capacidad Asistencial durante la Covid-19”.

Ya con toda la información relevante extraída se procede a carga en un nuevo archivo xlsx, se utiliza la función “*export_xlsx()*” para llevar esta función a cabo. Primero en la función se pasa las fechas a un formato que se pueda introducir en el archivo Excel, después se unifican las cuatro listas (fechas, camas totales, camas ocupadas e ingresos) formando una matriz con toda la información. Para ello se utiliza la librería “*Numpy*”. A continuación, se procede a cargar la matriz con las cabeceras en el archivo xlsx.

```
def export_xlsx(fechas,total_camas,ocupadas_covid,ingresos ):
    fechas_formateadas = []
    cabecera = ['fecha', 'Total de Camas', 'Camas ocupadas', 'Ingresos']

    for fecha in fechas:
        timestamp_obj = pd.Timestamp(fecha)
        # Obtener la fecha en formato 'YYYY-MM-DD'
        fecha_formateada = timestamp_obj.strftime('%Y-%m-%d')
        fechas_formateadas.append(fecha_formateada)

    # Crear la matriz utilizando numpy
    matriz = np.column_stack((fechas_formateadas, total_camas, ocupadas_covid,ingresos))

    # Nombre del archivo Excel
    nombre_archivo = "ETL\\Data\\covid.xlsx"
    df = pd.DataFrame(matriz, columns=cabecera) #Se introduce la cabecera
    df.to_excel(nombre_archivo)
```

Código 5. Proceso de carga de la fuente de datos “Capacidad Asistencial durante la Covid-19”.

Se muestra en la *figura 17* un ejemplo de cómo ha quedado almacenada parte de la información tras el proceso ETL en el data warehouse.

fecha	Total de Camas	Camas ocupadas	Ingresos
2020-08-01	13823	243	45
2020-08-01	886	37	1
2020-08-01	165	0	0
2020-08-02	165	0	0
2020-08-02	845	38	2
2020-08-02	13744	300	59
2020-08-03	165	0	0
2020-08-03	13408	351	81
2020-08-03	854	40	5
2020-08-04	13391	405	69
2020-08-04	165	1	0
2020-08-04	856	44	5
2020-08-05	165	0	0
2020-08-05	13365	367	84
2020-08-05	856	44	8
2020-08-06	866	54	11
2020-08-06	165	1	0
2020-08-06	13699	614	67
2020-08-07	165	0	0
2020-08-07	13516	459	78
2020-08-07	851	61	10
2020-08-08	13546	426	77

Figura 17. Parte del resultado del Proceso ETL de la fuente de datos “Capacidad Asistencial durante la Covid-19”.

4.3.3. Proceso ETL en Incendios producidos en España entre el 2006 y el 2015.

Se trata de una fuente de datos semi estructurada, donde en el archivo PDF se puede encontrar información expuesta en forma de texto, tablas, gráficos o incluso mapas. Al tener una estructura semi estructurada, el nivel de complejidad de extracción va a ser intermedio. Para aplicar el proceso ETL se han utilizado las siguientes librerías de Python:

- Camelot [29]. Esta librería se utiliza para extraer tablas de archivos PDF, tiene gran poder ya que puede extraer tablas incluso si no están correctamente estructuradas o si tienen múltiples categorías en una misma tabla. Otra ventaja al usar esta librería que una vez

extraída las tablas las almacena en una lista, por lo que, el proceso de extracción y carga se ve facilitado.

- Pandas [4]. Como se ha explicado en el punto anterior, es una biblioteca de gran poder que ayuda a los procesos de tratamiento de datos implementando multitud de opciones y funciones.

Lo primero que se realiza en el programa desarrollado en Python es leer el archivo PDF utilizando la librería *Camelot*. Esta librería permite que lea el archivo e indicándole la página que se quiera leer extraiga las tablas que se encuentren en esa página y almacenando las tablas en una lista de Python. Para el proyecto se está utilizando la *página 26*, donde podemos encontrar una tabla que indica la evolución de los grandes incendios entre el 1968 hasta el 2015.

Una vez extraída toda la tabla se procede a la transformación, se procede a quitar determinados caracteres entre los números que entorpecen su utilización. Por ejemplo, se quita el símbolo “%” de todos números o se quitan los puntos que en documento se utilizan como separador de miles. Para poder realizar alguna de estas transformaciones es necesario pasar la tabla extraída al formato dataframe.

```
tables = camelot.read_pdf("Fuentes de datos\Eulogh\incendios-decenio-2006-2015_tcm30-521617.pdf", flavor='stream', pages='32', strip_text='.')
tabla = tables[0].df
#print(tabla.iloc[0].tolist()) #columnas

tabla= tabla[0].df.replace(',', '.', regex=True) #cambiar los puntos por comas

tabla = tabla[5:] #elimina el numero de headers de la tabla

data = tabla.values #se pasa a dataframe

for fila in data:
    fila[-1] = fila[-1].rstrip('%')
```

Código 6. Proceso de extracción y transformación de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.

Ya completados los procesos de extracción y transformación de los datos se procede a cargar los datos resultantes en un archivo xlsx, que actúa como data warehouse. También es necesario incluir la cabecera de los datos, indicando que significa cada columna. Todo este proceso se realiza utilizando la librería *Pandas*. El código es el siguiente:

```
cabecera = ['Año', 'Número de Siniestros', 'Número de siniestros > 500 ha', 'Superficie total (ha)', 'Superficie por ha', 'Superficie por %']
df = pd.DataFrame(data, columns=cabecera)
#print (df)
df.to_excel("incendios1996-2015.xlsx")
```

Código 7. Proceso de Carga de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.

Se expone una comparación entre parte de los datos almacenados en la fuente de datos y parte de los datos almacenados en el archivo xlsx, que son frutos del resultado del proceso ETL llevado a cabo.

Año	Nº Siniestros	Nº siniestros. ≥ 500 ha	Superficie forestal total afectada (ha)	Superficie afectada por GIF	
				ha	%
1968	2.038	20	55.702,00	18.254,30	32,77%
1969	1.442	21	53.171,60	19.372,00	36,43%
1970	3.155	30	87.438,50	32.465,30	37,13%
1971	1.665	8	34.312,40	7.138,00	20,80%
1972	2.093	17	55.920,10	15.281,00	27,33%
1973	3.724	20	95.072,50	25.341,90	26,66%
1974	3.920	45	139.927,50	47.718,00	34,10%
1975	4.128	57	180.136,90	87.535,00	48,59%
1976	4.356	40	121.514,10	44.735,00	36,81%
1977	2.064	19	68.870,90	26.717,50	38,79%
1978	8.193	153	424.957,90	182.614,80	42,97%
1979	7.167	84	271.718,40	111.008,50	40,85%
1980	7.075	76	261.514,80	103.550,00	39,60%
1981	10.688	75	291.417,10	92.215,00	31,64%
1982	6.308	40	149.077,10	47.821,70	32,08%
1983	4.736	27	107.551,40	42.239,30	39,27%
1984	7.073	51	164.166,10	53.410,70	32,53%
1985	12.235	160	484.475,20	199.984,80	41,28%
1986	7.514	104	264.787,40	142.488,00	53,81%
1987	8.816	35	147.340,40	36.562,90	24,82%

Figura 18. Tabla extraída de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”, página 26.

Año	Numero de Siniestros	Numero de Siniestros ≥ 500 ha	Superficie total afectada (ha)	Superficie por %
1968	2038	20	55702	18254,3
1969	1442	21	53171,6	19372
1970	3155	30	87438,5	32465,3
1971	1665	8	34312,4	7138
1972	2093	17	55920,1	15281
1973	3724	20	95072,5	25341,9
1974	3920	45	139927,5	47718
1975	4128	57	180136,9	87535
1976	4356	40	121514,1	44735
1977	2064	19	68870,9	26717,5
1978	8193	153	424957,9	182614,8
1979	7167	84	271718,4	111008,5
1980	7075	76	261514,8	103550
1981	10688	75	291417,1	92215
1982	6308	40	149077,1	47821,7
1983	4736	27	107551,4	42239,3
1984	7073	51	164166,1	53410,7
1985	12235	160	484475,2	199984,8
1986	7514	104	264787,4	142488
1987	8816	35	147340,4	36562,9

Figura 19. Parte del data warehouse una vez realizado el proceso ETL.

En la figura 18, se corresponde con un histórico los incendios producidos en España desde el 1968 hasta el 2015, donde se tiene en cuenta en número de siniestros, el número de siniestro que ha sobrepasado las 500 hectáreas (ha), la superficie forestal afectada en hectáreas, la superficie afectada por GIF (grandes incendios forestales) en hectáreas y la superficie afectada por GIF en porcentaje.

4.3.4. Proceso ETL en Concesión de Nacionalidad Española entre el 2010 y el 2019.

Se trata de una fuente de datos totalmente estructurada, por lo que el proceso ETL, es relativamente sencillo. El archivo Xlsx consiste en una tabla en función de los años y de los diferentes países. Entonces el proceso ETL consiste en extraer todos los datos del archivo,

eliminar todos los datos menos las concesiones de nacionalidad a los hombres (se eliminan la categoría de ambos sexos y la de mujeres) y se carga estos datos en un archivo xlsx.

Solo se tienen en cuenta las concesiones de nacionalidad a los hombres, simplemente por escoger una categoría, no tiene relevancia alguna.

Para la realización del proceso ETL en esta fuente de datos se ha utilizado la siguiente librería de Python:

- Pandas [4]. Como se indicaba en el punto anterior, esta biblioteca es muy flexible. En el punto anterior se utilizaba solamente para cargar los datos en el data warehouse, mientras que en esta fuente de datos se utiliza durante todo el proceso ETL.

Lo primero que se realiza en el programa desarrollado en Python es leer el archivo xlsx utilizando la librería *Pandas*, el documento al tener más de una página es necesario indicar a qué página se está accediendo.

Cuando ya se ha extraído toda la información, se genera una nueva lista indicando a partir de las filas y columnas que información de todo el archivo va a ser parte de la nueva lista.

A continuación, se introducen las cabeceras que va a tener el archivo a la lista y se crea el archivo xlsx, con el nombre “*nacionalidad-Hombres.xlsx*” y se introduce toda la lista en el archivo. El código más relevante del programa es el siguiente:

```
# Extraer el rango de filas y columnas especificado
values = df.iloc[84:161, 0:11]
data = df.values
data = values.values #se pasa a dataframe
cabecera = ['País', '2019', '2018', '2017', '2016', '2015', '2014', '2013', '2012', '2011', '2010']
df = pd.DataFrame(data, columns=cabecera) #Se introduce la cabecera
#print (df)
df.to_excel("nacionalidad-Hombres.xlsx")
```

Código 8. Proceso de ETL de la fuente de datos “Concesiones de Nacionalidad Española entre el 2010 y el 2019”.

Ya realizado el proceso ETL, parte del data warehouse se visualiza de la siguiente manera, donde se puede hacer una comparación con la figura 9, que aunque no sea la misma categoría, porque el de la figura 9 corresponde a la categoría de ambos性, tiene el mismo tipo de estructura.

País	2019	2018	2017	2016	2015	2014	2013	2012	2011	2010
Unión Eur	3588	1264	366	1221	912	1174	2360	930	931	715
Bulgaria	479	200	54	205	172	178	286	68	55	37
Francia	71	24	10	34	20	22	77	32	37	30
Italia	481	148	41	134	90	102	238	82	88	66
Polonia	190	73	24	86	41	69	129	71	62	46
Portugal	295	211	70	227	156	261	643	409	456	369
Reino Uni	334	19	9	15	9	10	55	21	16	19
Rumanía	1496	507	139	450	377	487	764	165	130	88
Otros Unic	242	82	19	70	47	45	168	82	87	60
AELC1	10	6	0	4	3	1	24	10	7	8
Resto de E	2632	1002	238	701	562	728	1216	327	258	219
Bielorrusi	70	34	5	13	11	17	21	6	3	5
Moldavia	442	155	42	134	97	119	192	26	31	12
Rusia	464	170	48	147	123	171	322	104	72	73
Serbia	75	21	10	25	20	34	77	30	29	12
Turquía	68	25	5	22	18	38	61	24	15	17
Ucrania	1410	549	119	325	263	298	425	81	42	44
Otros Rest	103	48	9	35	30	51	118	56	66	56
África	26173	19582	4896	18677	15129	15923	36480	12039	10753	7874
Angola	30	10	7	12	11	15	49	20	19	22
Argelia	871	841	337	871	699	785	1611	460	343	234
Burkina F	44	26	8	27	14	20	44	11	8	6
Cabo Verc	55	29	9	19	9	23	49	25	22	31
Camerún	168	65	31	58	61	62	188	64	57	46
Congo	30	14	7	25	40	55	161	72	59	56
Costa de M	66	23	13	34	25	24	74	31	22	8
Egipto	88	86	15	56	37	69	160	50	56	60
Gambia	497	285	93	262	253	314	827	278	302	278
Ghana	515	367	115	259	218	269	636	148	116	66
Guinea	238	125	50	171	111	182	126	71	80	63

Figura 20. Parte del data warehouse una vez realizado el proceso ETL de la fuente de datos “Concesiones de Nacionalidad Española entre el 2010 y el 2019”.

4.3.5. Proceso ETL en Catálogo del Bosque Urbano de la Ciudad de Madrid.

Esta fuente de datos se caracteriza por tener una estructura semi estructurada, donde en el documento PDF presenta la información de diferentes formas, tal y como se podía observar en la figura 15.

De este documento se extrae la información que aparece desde la página 64 hasta la página 75. En estas páginas se muestra una tabla con el listado de especies a lo largo del bosque urbano de la ciudad de Madrid, esta información está presentada en forma de tabla, pero, aunque tenga esta estructura presenta dificultades a la hora de aplicar el proceso de extracción.

Se decide extraer esta información del PDF y no otra debido a la cantidad de datos que se presentan, es exactamente una tabla constituida por 446 filas y 19 columnas, donde cada fila es una especie en concreto.

Para la realización del proceso ETL en esta fuente de datos se ha utilizado las siguientes librerías de Python:

- Fitz [26]. Como se ha explicado anteriormente esta librería tiene un grandísimo poder y una gran flexibilidad ya que se pueden realizar multitud de acciones. Aunque ya se haya

utilizado esta librería en otro documento ETL, la funcionalidad que tiene en proceso ETL difiere totalmente con el que se describió. En esta fuente de datos la librería extrae todo el texto de la página en cuestión, da igual que este en formato de tabla o no. Por lo que, el proceso de transformación es mucho más arduo. Por ejemplo, como se puede observar en la *figura 19*, además de extraer la tabla extrae el título, el subtítulo, el numero de la página y demás elementos que carecen de valor.

- **XlsxWriter [27]**. Esta librería se utiliza para crear el archivo xlsx que actúa como data warehouse e insertar toda la información en forma de tabla.

El programa desarrollado en Python esta divido en diferentes funciones siguiendo los procesos de extracción, transformación y carga. El programa está divido en una función principal, en una función que implementa el proceso de extracción (*get_text ()*), en una función que implementa el proceso de transformación (*clean ()*) y en una función que implementa el proceso de carga (*export_xlsx ()*).

Lo primero que se realiza es importar todo el archivo PDF a través de la biblioteca *Fitz*, a continuación, se realiza un bucle para que cada página entre la 64 y 75 realice el proceso de extracción (*get_text ()*) y de transformación (*clean ()*) en cada iteración. Cuando se realiza estos dos procesos el resultado se almacena en una lista. Una vez terminado el bucle y realizado estos dos procesos en todas las páginas indicadas se procede al proceso de carga, toda la información almacenada se introduce al archivo xlsx a través de la función “*export_xlsx ()*”.

El proceso de extracción se extraen todas las palabras del archivo, sin tener en cuenta algunos caracteres, los puntos, los doble puntos, asteriscos, ... El texto se va extrayendo en función de filas, cuando es una nueva fila, se indica en el texto gracias a librería *Fitz* con un 0.

```
def get_text(words):
    text = []
    tabla = []
    word = ""

    for i in range (len(words)):
        if ( words[i][4] != '.' and words[i][4] != '...' and words[i][4] != '*' ):
            if (words[i][7] != 0):
                word = word + " " + words[i][4]
            else:
                text.append(word)
                word = words[i][4]
                if (words[i][6] == 0):
                    tabla.append(text)
                    text = []
                else:
                    tabla.append(text)
                    return tabla
```

Código 9. Proceso de extracción de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”.

Una vez obtenido todo el texto de la página en cuestión se procede a transformarlo, con la función “*clean ()*”. Primero se elimina del texto extraído toda la información que no es relevante con la tabla el titulo o el número de página del documento. Tambien se eliminan los caracteres que desvirtúan la información. Algunos de estos caracteres son los puntos, comas, el tanto por ciento, ... Cada carácter de texto del texto se extrae por separado y se extrae con formato char, por lo que hay también hay que concatenar los que son necesarios.

```

def clean (tabla):
    if (tabla[1][1] == '64'):#se borran mas elementos ya que en la primera pagina hay mas
        for i in range (4):#se borran los dos primeros que es el titulo y numero de pagina
            tabla.pop(0)
    else:
        for i in range (2):#se borran los dos primeros que es el titulo y numero de pagina
            tabla.pop(0)

    for i in range (8):#se borran los 8 ultimos que son las primeras columnas
        tabla.pop(len(tabla)-1)

    for i in range(len (tabla)):#se concatenan los nombres sueltos con los valores y se borran los valores duplicados
        if (len (tabla) > i): #como se van restando miembros de la lista el i puede ser mayor que el tamaño por lo que es un error
            if (len(tabla[i]) <= 5):
                #print (i,tabla[i])
                tabla[i] = [tabla[i][0] + " " + tabla[i][1]] + tabla[i+1]
                tabla.pop(i+1)

    for i in range(len (tabla)):
        for j in range(len (tabla[i])):
            if ('%' in tabla[i][j]):#se quitan el % para poder trabajar con los datos
                tabla[i][j] = tabla[i][j].replace('%', '')
                print (tabla[i][j], '\n')
            if ('.' in tabla[i][j]):#se quitan el . para poder trabajar con los datos
                tabla[i][j] = tabla[i][j].replace('.', '')
            if (',' in tabla[i][j]):#se quitan el , para poder trabajar con los datos
                tabla[i][j] = tabla[i][j].replace(',', '.')

    return tabla

```

Código 10. Proceso de transformación de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”.

Ya con todas las páginas extraídas y limpiadas una a una se procede a guardar la información en el data warehouse. Este proceso es similar a los procesos de carga de los puntos anteriores. Se procesa la carga a través de la librería *XlsxWriter*, lo primero que se realiza es la introducción de la cabecera de los datos en la tabla, después se introduce cada dato de la tabla, en cada casilla del archivo xlsx.

```

def export_xlsx(tablas):
    csv_columns = ['Especie', 'Cantidad', 'Porcentaje Población', 'Área Foliar (M2)', 'Porcentaje Área Foliar', 'Biomasa Foliar (Kg)', 'Dominancia', 'Almacén De Carbono (Tn)', 'Secuestro Carbono (Tn/Año)', 'Captación Contaminación (Kg/Año)', 'Producción Oxígeno (Tn/Año)', 'Agua Interceptada (M3/Año)', 'Escorrentía Evitada (M3/Año)', 'Evaporación Potencial (M3)', 'Evaporación (M3)', 'Transpiración (M3)', 'Isoprenos (G/Año)', 'Monoterpenos (G/Año)', 'Voc']
    ]
    if (tablas[0] != csv_columns):#puede que ya esten introducida la cabecera
        tablas.insert(0,csv_columns) #se meten el nombre de las columnas en la cabecera

    tablas.pop(len(tablas)-1)
    workbook = xlsxwriter.Workbook('prueba.xlsx')
    worksheet = workbook.add_worksheet()

    for row_num, row_data in enumerate(tablas):
        for col_num, col_data in enumerate(row_data):
            worksheet.write(row_num, col_num, col_data)

    workbook.close()

```

Código 11. Proceso de carga de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”.

Una vez completo todo el proceso ETL, la visualización de la información en el data warehouse corresponde con la *figura 22*. También se ofrece una comparativa de cómo está dispuesta la tabla en el documento PDF (*figura 21*) y como queda la información almacenada tras el proceso ETL.

ANEJO 02

LISTADO DE ESPECIES

Especie	Cantidad	Porcentaje Población	Área Foliar (M2)	Porcentaje Área Foliar	Biomasa Foliar (Kg)	Dominancia	Almacén De Carbono (Tn)	Secuestro Carbono (Tn/Año)	Captación Contaminación (Kg/Año)	Producción Oxígeno (Tn/Año)	Aqua Interceptada (M3/Año)	Escoorrentia Evitada (M3/Año)	Evaporación Potencial (M3)	Evaporación (M3)	Transpiración (M3)	Isoprenos (G/Año)	Monoterpenos (G/Año)	Voc (G/Año)	
Abelia	18	0.001%	630	0.000%	47.16	0.001%	1.10	0.08	0.79	0.22	4.62	0.95	64.96	4.62	23.30	0.00	0.00	0.00	
Abies	107	0.006%	9398	0.006%	1323.71	0.012%	5.58	0.35	11.97	0.94	70.13	14.42	1004.20	70.11	372.06	87.66	16543.68	16631.34	
Abies alba	233	0.013%	11320	0.007%	1594.42	0.020%	5.51	0.51	14.55	1.37	85.21	17.53	1218.06	85.17	453.86	106.01	19927.01	20.033.02	
Abies cephalonica	5	0.000%	668	0.000%	94.06	0.001%	0.40	0.02	0.85	0.05	5.04	1.04	70.01	5.03	25.23	6.06	1.175.49	1.181.55	
Abies concolor	2	0.000%	28	0.000%	3.93	0.000%	0.01	0.00	0.03	0.01	0.20	0.04	2.85	0.20	1.02	0.25	49.09	49.34	
Abies nordmanniana	81	0.005%	19.218	0.012%	2706.76	0.016%	16.18	0.60	24.52	1.59	143.82	29.57	2054.54	143.73	759.51	178.93	33.828.89	34.007.82	
Abies pinsapo	212	0.012%	24.386	0.015%	3434.61	0.027%	18.03	0.94	30.84	2.50	181.53	37.35	2556.59	181.41	928.00	222.94	42.925.51	43.148.45	
Abies procera	4	0.000%	753	0.000%	105.99	0.001%	0.43	0.02	0.94	0.05	5.54	1.14	78.64	5.54	2.85	6.89	1.324.68	1.331.57	
Abutilon	31	0.002%	928	0.001%	69.52	0.002%	0.42	0.09	1.15	0.24	6.78	1.40	94.30	6.78	33.34	0.00	0.00	0.00	
Acacia	236	0.014%	25.698	0.016%	6110.00	0.029%	19.45	1.45	32.79	3.86	192.27	39.55	2740.22	192.20	1.012.55	398.46	738.16.18	74214.64	
Acacia cyclops	166	0.010%	15.266	0.009%	3689.76	0.019%	8.85	0.79	20.15	2.10	117.92	24.22	1690.03	117.83	64.43	250.30	46.096.45	46.346.75	
Acacia dealbata	1.010	0.058%	85.431	0.052%	2064.80	0.110%	60.54	4.87	109.27	13.00	641.33	131.91	9119.54	640.99	3.368.33	1.353.76	255.159.05	256.512.81	
Acacia longifolia	4	0.000%	632	0.000%	152.71	0.001%	0.65	0.04	0.81	0.10	4.73	0.97	67.40	4.73	2.485	10.04	1897.79	1907.83	
Acacia retinodes	19	0.001%	2894	0.002%	69.94	0.003%	2.32	0.15	3.67	0.40	21.51	4.42	306.94	21.50	112.93	44.85	8.353.68	8.398.73	
Acer	1.550	0.089%	67.379	0.041%	379.25	0.130%	15.38	3.05	86.34	8.13	506.12	104.26	723.39	505.86	2.686.79	237.30	22.746.80	23.034.00	
Acer buergerianum	168	0.010%	6905	0.004%	852.43	0.014%	1.27	0.32	8.86	0.86	51.71	10.62	752.01	51.70	283.54	54.10	512.434	517.843	
Acer campestre	7.069	0.405%	450.790	0.275%	25.371.44	0.680%	178.57	25.16	564.90	67.10	332.473	684.41	46809.72	3323.29	16.897.06	1542.85	152.530.43	154.063.28	
Acer farnesii	9	0.001%	264	0.000%	14.88	0.001%	0.04	0.01	0.35	0.04	2.05	0.42	28.93	2.04	10.88	0.94	89.42	90.37	
Acer ginnala	3	0.000%	121	0.000%	6.79	0.000%	0.01	0.00	0.15	0.01	0.89	0.18	12.76	0.89	4.68	0.42	40.82	41.24	
Acer macrophyllum	1	0.000%	252	0.000%	14.18	0.000%	0.13	0.01	0.32	0.02	1.87	0.39	26.81	1.87	9.88	0.88	85.26	86.14	
Acer monspeliacum	1023	0.059%	43104	0.026%	2425.94	0.085%	10.95	2.23	54.60	5.95	320.55	65.90	4554.89	320.41	16.70.27	149.78	14583.74	14733.52	
Acer negundo	29.268	1.683%	353.240	0.215%	3221.98	0.384%	2.158	0.30	3.05	86.34	8.13	506.12	104.26	723.39	505.86	2.686.79	237.30	22.746.80	23.034.00
Acer opalus	5	0.000%	258	0.000%	14.51	0.000%	0.06	0.01	0.33	0.03	1.93	0.40	27.37	1.93	10.05	0.90	87.23	88.13	
Acer palmatum	50	0.003%	1699	0.001%	95.65	0.004%	0.62	0.10	2.17	0.26	12.75	2.62	179.69	12.74	65.64	5.90	574.97	580.86	
Acer platanoides	3.881	0.222%	357.607	0.218%	19.301.06	0.441%	128.33	14.40	452.28	38.40	2652.85	545.61	3784.21	2651.95	13892.34	1193.53	116.028.21	117.221.74	
Acer pseudoplatanus	124	0.007%	4100	0.003%	230.76	0.010%	0.61	0.19	5.36	0.50	31.27	6.43	452.25	31.27	172.50	14.80	1.387.19	1401.99	
Acer saccharinum	1143	0.065%	78.814	0.048%	4148.06	0.114%	38.67	3.03	100.32	8.07	588.67	121.09	8380.18	588.39	3.087.68	257.46	24.966.17	25.193.52	
Acer tataricum	45	0.000%	1.737	0.001%	97.76	0.004%	0.32	0.08	2.21	0.21	12.92	2.65	187.92	12.92	70.43	6.17	587.67	593.84	

64 | VALOR DEL BOSQUE URBANO DE MADRID

Figura 21. Parte de la fuente de datos “Catalogo de Bosque Urbano de la Ciudad de Madrid”.

1	Especie	Cantidad	Porcentaje	Área Folia	Porcentaje	Biomasa	F	Dominanc	Almacén	(Secuestro	Captación	Producció	Aqua	Intei	Escoorrentia	Evaporaci	Evaporaci	Transpiraci	Isoprenos	Monoterpenos	Voc
2	Abelia	18	0.001	630	0.000	47.16	0.001	1.10	0.08	0.79	0.22	4.62	0.95	64.96	4.62	23.30	0.00	0.00	0.00	0.00	
3	Abies	107	0.006	9398	0.006	1323.71	0.012	5.58	0.35	11.97	0.94	70.13	14.42	1004.20	70.11	372.06	87.66	16543.68	16631.34		
4	Abies alba	233	0.013%	11320	0.007%	1594.42	0.020%	5.51	0.51	14.55	1.37	85.21	17.53	1218.06	85.17	453.86	106.01	19927.01	20.033.02		
5	Abies cep	5	0.000%	668	0.000%	94.06	0.001%	0.40	0.02	0.85	0.05	5.04	1.04	2.85	0.20	1.02	0.25	1175.49	1181.55		
6	Abies con 2	0.000	28	0.000	3.93	0.000	0.01	0.00	0.03	0.01	0.20	0.04	2.85	0.20	1.02	0.25	49.09	49.34			
7	Abies nor 81	0.005	19.218	0.012%	2706.76	0.016%	16.18	0.60	24.52	1.59	143.82	29.57	2054.54	143.73	759.51	178.93	33.828.89	34.007.82			
8	Abies pim	212	0.012%	24386	0.015%	3434.61	0.027%	18.03	0.94	30.84	2.50	181.53	37.35	2556.59	181.41	928.00	222.94	42.925.51	43.148.45		
9	Abies pro 4	0.000	753	0.000	105.99	0.001%	0.43	0.02	0.94	0.05	5.54	1.14	78.64	5.54	2.85	6.89	1.324.68	1.331.57			
10	Abutilon	31	0.002%	928	0.001%	69.52	0.002%	0.42	0.09	1.15	0.24	6.78	1.40	94.30	6.78	33.34	0.00	0.00	0.00		
11	Acacia	236	0.014%	25698	0.016%	6211.00	0.029%	19.45	1.45	32.79	3.86	192.27	39.55	2740.22	192.20	1.012.55	398.46	73816.18	74214.64		
12	Acacia cyc	166	0.010%	15266	0.009%	3689.76	0.019%	8.85	0.79	20.15	2.10	117.92	24.22	1690.03	117.83	64.43	250.30	46096.45	46.346.75		
13	Acacia de 1010	0.058	85431	0.052%	20648.00	0.110%	60.54	4.87	109.27	13.00	641.33	131.91	9119.54	640.99	3.368.33	1.353.76	255.159.05	256.512.81			
14	Acacia lon 4	0.000	632	0.000	152.71	0.001%	0.65	0.04	0.81	0.10	4.73	0.97	67.40	4.73	2.485	10.04	1897.79	1907.83			
15	Acacia ret 19	0.001	2894	0.002%	69.94	0.003%	2.32	0.15	3.67	0.40	21.51	4.42	306.94	21.50	112.93	44.85	8353.88	8398.73			
16	Acer	1.550	0.089%	67.379	0.041%	379.25	0.130%	15.38	3.05	86.34	8.13	506.12	104.26	723.39	505.86	2.686.79	237.30	22.746.80	23.034.00		
17	Acer buer	168	0.010%	6905	0.004%	852.43	0.014%	1.27	0.32	8.86	0.84	51.71	10.62	752.01	51.70	283.54	54.10	512.434	517.843		
18	Acer campe	7.069	0.405%	450.790	0.275%	25371.44	0.680%	178.57	25.16	564.90	67.10	3324.73	684.41	46809.72	3323.29	16.897.06	1542.85	152.530.43	154.063.28		
19	Acer forre 9	0.001	264	0.000%	14.88	0.001%	0.04	0.01	0.35	0.04	2.05	0.42	28.93	2.04	10.88	0.94	89.42	90.37			
20	Acer ginnal 3	0.000	121	0.000%	6.79	0.000%	0.01	0.00	0.15	0.01	0.89	0.18	12.76	0.89	4.68	0.42	40.82	41.24			
21	Acer macr 1	0.000	252	0.000%	14.18	0.000%	0.13	0.01	0.32	0.02	1.87	0.39	26.81	1.87	9.88	0.88	85.26	86.14			
22	Acer mon 1023	0.059	43104	0.026%	2425.94	0.085%	10.95	2.23	54.60	5.95	320.55	65.90	4554.89	320.41	16.70.27	149.78	14583.74	14733.52			
23	Acer negu 29368	1.683	0.024%	3533240	0.215%	323219.80	0														

4.3.6. Proceso ETL en Población por Provincias de España 1996-2021.

Se trata de una fuente de datos estructurada, por lo que, el proceso ETL es más simple. La fuente de datos está constituida en función de las provincias españolas y los años, también esta divida en función de sexo (ambos sexos, masculino o femenino). El proceso ETL se realiza para obtener solo la categoría de ambos sexos para todas las provincias, se elige esta categoría por elegir alguna categoría, no hay ninguna razón de peso detrás de la decisión.

Para la realización del proceso ETL en esta fuente de datos se ha utilizado las siguientes librerías de Python:

- Pandas [4]. Esta librería ya se ha explicado en los puntos anteriores. La implementación desarrollada por esta librería es la misma que realizado en el punto 4.3.4 con la fuente de datos de *Concesiones de Nacionalidad Española*.

Para desarrollar el programa en Python, todo se implementa a través de la biblioteca *Pandas*. Se importa todo el archivo, a continuación, extrae de toda la información que se ha importado el rango de filas y columnas que abarcan solo a la categoría de ambos sexos.

Después se produce la limpieza de los datos a través de la función “*clean ()*”. La primera columna del archivo, donde se tiene el nombre de cada provincia tiene un numero asignado y nombre, tal y como se puede observar en la *figura 13*, se limpia el archivo para que solo se tenga en cuenta el nombre.

A continuación, se introducen las cabeceras que van a tener en el data warehouse y a través de la misma librería Pandas, se crea el archivo *xlsx* y se introduce todos los datos en el archivo.

```
# Open the Workbook
df = pd.read_excel(
    'Fuentes de datos\Smarcities\poblacionEspaña.xlsx',
    engine='openpyxl'
)
# Extraer el rango de filas y columnas especificado
values = df.iloc[8:64, 0:26]
data = values.values #se pasa a dataframe
cabecera = ['Provincia', '2021', '2020', '2019', '2018', '2017', '2016', '2015', '2014', '2013', '2012', '2011', '2010', '2009', '2008', '2007', '2006',
            '2005', '2004', '2003', '2002', '2001', '2000', '1999', '1998', '1996']

data2 = clean(data) #Funcion Clean, se procede a limpiar los datos
df = pd.DataFrame(data2, columns=cabecera) #Se introduce la cabecera
#print (df)
df.to_excel("poblacion-España.xlsx") #Se crea y se introduce al archivo xlsx
```

Código 12. Proceso ETL de la fuente de datos “Población por Provincias de España 1996-2021”.

Una vez aplicado todo el proceso ETL, parte de la visualización del resultado es la siguiente:

Provincia	2021	2020	2019	2018	2017	2016	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005	2004	2003
Albacete	386464	388270	388167	388786	390032	392118	394580	396987	400007	402837	402318	401682	400891	397493	392110	387658	384640	379448	376556
Alicante/-i	1881762	1879888	1858683	1838819	1825332	1836459	1855047	1868438	1945642	1943910	1934127	1926285	1917012	1891477	1825264	1783555	1732389	1657040	1632349
Almería	731792	727945	716820	709340	70672	704297	701211	701688	699329	704219	702819	695560	684426	667635	646633	635850	612315	580077	565310
Araba/Ála	333626	333940	331549	328868	326574	324126	323648	321932	321417	322557	319227	317352	313819	309635	305459	301926	299957	295905	294360
Asturias	1011792	1018784	1022800	1028244	1034960	1042608	1051229	1061756	1068165	1077360	1081487	1084341	1085289	1080138	1074862	1076896	1073761	1075381	
Ávila	158421	157664	157640	158498	160700	162514	164925	167015	168825	171265	172704	171896	171815	168638	167818	167032	166108	165480	
Badajoz	669943	672137	673559	676376	679884	684113	686730	690929	693729	694533	693921	692137	688777	685246	678459	673474	671299	663896	663142
Balears, II	1173008	1171543	1194460	1128908	1151599	1107220	1104479	1103442	1111674	1119439	1113114	1106049	1095426	1072844	1030650	1001062	983131	955045	947361
Barcelona	5714730	5743402	5664579	5609350	5576037	5542680	5523922	5523784	5540925	5552050	5529099	5511147	5487935	5416447	5332513	5309404	5226354	5117885	5052666
Bizkaia	1154334	1159443	1152651	1149626	1148302	1147576	1148775	1151905	1156447	1158439	1157724	1152658	1146421	1141457	1139863	1136181	1132861	1133428	
Burgos	356055	357650	356958	357070	358171	360995	364002	366900	371248	374970	375657	374826	375563	373672	365972	363874	361021	356437	355205
Cáceres	389558	391850	394151	396487	400036	403665	406267	408703	410275	413597	415446	415083	413633	412498	411531	412899	412580	411390	410762
Cádiz	1245960	1240409	1240155	1238714	1239433	1239889	1240284	1240175	1238492	1245164	1243519	1236739	1230594	1220467	1207343	1194062	1180817	1164374	1155724
Cantabria	584507	582905	581078	580229	580295	582206	585179	588656	591888	593861	593121	592250	589235	582138	572824	568091	562309	554784	549690
Castellón	587064	585590	579662	576894	575470	579245	582327	587508	601699	604564	603444	604274	602301	594915	573282	559761	543432	527345	518239
Ciudad Re	492591	495045	495761	499100	502578	506888	513713	519613	524962	530250	530175	529453	527273	522343	510122	506864	500060	492914	487670
Córdoba	776789	781451	782979	785240	788219	791610	795611	799402	802422	804498	805857	805108	803998	798822	792182	788287	784376	779870	775944
Coruña, A	1120134	1121815	1119596	1119351	1120294	1122799	1127196	1132735	1138161	1143911	1147124	1146458	1145488	1139121	1132720	1129141	1126707	1121344	1120814
Cuenca	195516	196139	196329	197222	198718	201071	203841	207449	211899	218036	219138	217716	217363	215274	211375	208616	207974	204546	202982
Gipuzkoa	726033	727121	723576	720592	719282	717832	716834	715148	713818	712097	709607	707263	705698	701056	694944	691895	688708	686513	684416
Girona	786596	781788	771044	761947	757516	753576	753054	756156	761632	761627	753046	747782	731864	706185	687331	664506	636198	619692	
Granada	921338	919168	916478	912075	912938	915392	917297	919455	919319	922928	924550	918072	907428	901220	884099	876184	860898	841687	828107

Figura 23. Parte de la visualización del data warehouse de la fuente de datos “Población por Provincias de España 1996-2021”.

4.4. Comparativa entre las librerías ETL.

Para realización de los procesos ETL aplicados en las diferentes fuentes de datos abiertas se ha utilizado una serie de librerías como ya se han indicado, donde una librería puede realizar el proceso de extracción o transformación o de carga o todos los procesos.

Al realizar cada proceso se ha realizado una investigación tanto teórica como experimental de que librería se adecuaba o funcionaba de la forma más optima teniendo en cuenta la fuente de datos abierta.

Dependiendo de cómo se presentan los datos en las diferentes fuentes de datos o cuales son los objetivos para analizar de cada una de ellas, es conveniente utilizar unas u otras. Por todo esto, se realiza una comparativa entre las diferentes librerías. Tanto las que se han llegado a implementar como las librerías que se han investigado pero que no se han utilizado, pero pueden ser útiles para otros objetivos.

La comparativa entre las diferentes librerías para la realización de los procesos ETL se visualiza en forma de tabla. La tabla está compuesta por las siguientes categorías:

- **Tipo de Proceso ETL.** Se indica que procesos ETL (extracción, transformación y carga) abarca la librería.
- **Complejidad de Uso.** Indica la complejidad de la librería a la hora del aprendizaje y de la implementación.
- **Rendimiento.** Cuantifica el rendimiento de la librería en términos de velocidad y su eficiencia en la ejecución. Califica la calidad de los resultados una vez aplicada la librería.
- **Tipo de Estructura.** Indica que tipo de estructura se presenta los datos para garantizar la mayor calidad a la hora de realizar los procesos ETL (tablas, imágenes, gráficos, grafos, texto) Cuando una información se encuentra presentada en forma de tabla, se puede indicar que tiene una tabular.
- **Documentación/Soporte.** Indica si la librería cuanta con una documentación de calidad y si la librería está siendo activamente desarrollada u actualizada.

La comparativa entre las librerías en Python utilizadas para los diferentes procesos ETL es la siguiente:

Librería	Tipo de Proceso ETL	Complejidad de Uso	Rendimiento	Tipo de Estructura (Extracción)	Documentación /Soporte
Fitz	Extracción y transformación de datos de archivos PDF.	Difícil	Medio	Texto	Baja
XlsxWriter	Carga en archivos Excel (xlsx).	Fácil	Alto	-	Buena
Pandas	Extracción, transformación y carga de datos tabulares.	Fácil	Alto	Tabular	Excelente
NumPy	Transformación en arreglos y matrices.	Medio	Alto	-	Buena
Camelot	Extracción de tablas de archivos PDF.	Fácil	Alto	Tabular	Media
Tabula-Py	Extracción de tablas de archivos PDF	Medio	Media	Tabular	Media.
PySpark SQL	Transformación de datos con SQL.	Media	Alto	-	Buena
PyPDF2	Extracción y transformación en archivos PDF.	Medio	Baja	Texto	Buena

Tabla 2. Comparación entre las diferentes librerías para los procesos ETL.

Cabe destacar que la mayoría de los procesos de transformación realizados, se han llevado a cabo a partir de las propias funciones que proporciona el lenguaje de programación Python.

La gran mayoría de la información de las fuentes de datos abiertas se presenta en formato tabular, en formato de tabla, por ello, es de importancia utilizar las librerías correctas para cada tipo de tabla.

Como se puede observar en la *tabla 2*, hay múltiples librerías para realizar el proceso de extracción cuando la información esta presentada en forma tabular. Pero no todas las tablas están constituidas de la misma forma, ni presentadas en el mismo archivo. Por ello, dependiendo de la tabla se utiliza una librería u otra. A continuación, en la *tabla 3*, se presenta que librerías son más adecuadas para que tipo de tablas.

Librería	Estructura de la Tabla	Archivo
Pandas	Tablas bien estructuradas.	CSV, Xlsx.
Camelot	Tablas bien estructuradas y simples con varios subtítulos.	PDF
Tabula-Py	Tablas más complejas con celdas fusionadas o diseños complicados.	PDF

Tabla 3. Comparación entre las librerías para la extracción de tablas.

4.5. Análisis de Datos.

Se seleccionan cuatro fuentes de datos abiertas en las que anteriormente se han aplicado el proceso ETL para proceder con los modelos estadísticos. Una vez que las fuentes de datos están transformadas con la correcta forma y la información presente tiene una buena calidad se procede con el análisis de datos.

El objetivo principal de los modelos estadísticos que se aplican es poder abarcar el mayor conocimiento de análisis posible. Se busca realizar análisis de poca complejidad, como son los modelos de descripción o de regresión lineal, y concluir en modelos de más complejidad tanto de conocimientos como de programación, como pueden ser las redes neuronales recurrentes (RNN) como puede ser LSTM.

A cada fuente de datos abierta se le aplica un modelo estadístico u otro dependiendo de los objetivos del análisis y del tipo de datos de la fuente de datos. Los objetivos del análisis pueden diferir en muchos sentidos, por ejemplo, se puede buscar clasificar los datos, predecir futuros datos o incluso entender de una mejor forma los datos actuales. También la forma en la que estén organizados los datos en el data warehouse, es significativa a la hora de aplicar un modelo u otro, por ejemplo, pueden estar organizados en históricos, categóricos, números, donde a cada organización tiene un modelo estadístico más óptimo.

Las fuentes seleccionadas a las que se le van a aplicar los diferentes modelos estadísticos son las siguientes:

- Capacidad Asistencial durante la Covid-19. Se decide seleccionar esta fuente de datos debido a alta calidad de datos que se presenta y por la gran cantidad de datos que se tienen. El interés de entender como afectó la pandemia a la sociedad española y si había algún conocimiento que se nos escapó también es una razón de peso para seleccionarla.
- Incendios producidos en España entre el 2006 y el 2015. Se decide seleccionar esta fuente de datos, más concretamente la información que aparece en la página 26 del PDF, ya que nos encontramos con un histórico de la “*Evolución de los grandes incendios, 1968-2015*”. Este histórico puede ayudarnos a entender características fundamentales de los incendios en nuestro país, incluso de como poder predecir estadísticamente el impacto de los próximos años.
- Catálogo del Bosque Urbano de la Ciudad de Madrid. Se selecciona esta fuente de datos para entender cómo se distribuyen las diferentes especies de vegetación a lo largo de la ciudad de Madrid. Con las características que presenta cada especie es de vital importancia saber distribuir de una manera eficiente las diferentes especies en la ciudad, ya que se pueden presentar ventajas tanto en la climatología de la ciudad, como en la contaminación y en la gestión de recursos. Estos son aspectos fundamentales en el concepto de “Smacite”, donde con la tecnología adecuada y esta serie de estudios pueden beneficiar a las grandes ciudades en multitud de aspectos.
- Población por Provincias de España 1996-2021. Se selecciona esta fuente de datos por el interés que suscita el problema actual de despoblación de algunas partes de la península. Se busca entender de mejor forma lo que está pasando en las provincias españolas, identificar tendencias de despoblación e influencias. Debido a la problemática de la despoblación, las grandes ciudades españolas se están viendo desbordadas, lo que está suponiendo un reto a la hora de gestionar las ciudades debidamente.

Ya sabiendo que fuentes de datos son utilizadas y los objetivos de investigación que presenta cada una de ellas, se decide que tipos de modelos se aplicaran a cada fuente de datos. Los modelos

estadísticos que se realizan abarcan cuatro tipos de análisis de datos, análisis descriptivo, análisis exploratorio, análisis predictivo.

A estas cuatro fuentes de datos se le aplicara un análisis descriptivo para poder entender de mejor forma como está constituida la fuente de datos con la que se trabaja. Después a cada fuente de datos se le aplica un modelo que abarca el análisis exploratorio o un análisis predictivo, incluso ambos. Para el análisis predictivo se utilizan las predicciones de series temporales.

Las predicciones en series temporales se enfocan en predecir comportamientos futuros o patrones en una secuencia de datos ordenados en un espacio de tiempo. Existen multitud de técnicas a la hora de realizar una predicción en una serie temporal, en el proyecto se utiliza “ARIMA”.

También se utiliza una red neuronal recurrente (RNN) para predecir comportamientos futuros y para capturar patrones en secuencias de datos como son las series temporales, a partir de un histórico de datos.

A continuación, en la siguiente tabla, *tabla 4*, se puede observar que tipo de modelos estadísticos se realiza en las cuatro fuentes de datos seleccionadas.

Fuente de Datos Abierta	Análisis Descriptivo	Análisis Exploratorio		Análisis Predictivo	
		Regresión Lineal	Clustering	Serie Temporal ARIMA	Red Neuronal Recurrente LSTM
Capacidad Asistencial durante la Covid-19	SI	-	SI	SI	-
Incendios producidos en España entre el 2006 y el 2015	SI	-	-	-	SI
Catálogo del Bosque Urbano de la Ciudad de Madrid	SI	-	SI	-	-
Población por Provincias de España 1996-2021	SI	SI	-	-	-

Tabla 4. Modelos Estadísticos de Cada Fuente de Datos Abierta

La mayoría de los modelos que se realizan en el proyecto son modelos de machine learning (aprendizaje automático). Donde el ML es una técnica de la inteligencia artificial que permite a las máquinas aprender y mejorar su rendimiento a medida que se les proporciona datos.

De los modelos estadísticos que se realizan tanto la regresión lineal, el clustering y la red neuronal, en este caso LSTM, se consideran modelos propios del machine learning.

La regresión lineal se trata de una técnica de aprendizaje supervisado.

El clustering se trata de una técnica de aprendizaje no supervisado.

LSTM se trata de una arquitectura específica de una red neuronal recurrente (RNN).

La técnica ARIMA no se considera un modelo de ML, ya que es un modelo estadístico utilizado para analizar y predecir series temporales.

Los modelos estadísticos se componen también de una parte fundamental, que es la visualización. La visualización es una herramienta crucial a la hora de realizar cualquier tipo de análisis de datos y de representar las soluciones obtenidas. Se pueden representar de manera gráfica los patrones, las tendencias y relaciones que estén presentes en los datos y en sus soluciones.

Durante el proyecto la visualización se lleva a cabo a través del software PowerBi mayormente, pero también será necesario en algunas ocasiones visualizar los datos en el propio programa de Python, usando la librería “Matplotlib” [34].

4.5.1. Análisis Descriptivo.

Es una metodología que consiste en describir las tendencias claves en los datos existentes y observar nueva información que proporcione nuevos hechos. La metodología se basa en calcular las medidas de simples de composición y distribución de variables, que sirve para proporcionar una base de conocimiento que puede servir como base a un análisis más complejo en el futuro, tal y como ocurre en el proyecto.

El análisis descriptivo consiste en realizar un resumen de estadísticas donde se calculan parámetros de los datos, como la media, la mediana, la moda, la desviación estándar, el rango y los percentiles. Estas estadísticas proporcionan una idea general de la distribución y variabilidad de los datos.

Tambien se procede a visualizar los datos, se utilizan diversas herramientas como gráficos y diagramas que ayudan a identificar de una manera visual patrones tendencias, valores atípicos y relaciones entre variables.

El procedimiento que se realiza para completar el análisis descriptivo es el mismo en las cuatro fuentes de datos, se realiza a través de Python y apoyándose en las siguientes librerías:

- Pandas [4]. Se utiliza para poder leer la información del archivo xlsx en Python y una vez obtenidos los resultados almacenarlos en otro archivo xlsx.
- NumPy [6]. Se utiliza para calcular todos los valores que aportan información al análisis descriptivo (media, mediana, moda, desviación típica, valor mínimo, valor máximo, total, primer cuartil 25% y tercer cuartil 75%)

A continuación, se explica los resultados del análisis descriptivo de cada fuente de datos con sus respectivas visualizaciones. Para realizar las respectivas visualizaciones se usa el software PowerBi. Todas estas visualizaciones que se han utilizado para realizar el análisis descriptivo de cada fuente de datos se pueden ver en el Anexo III con el dashboard completado.

4.5.1.1. Capacidad Asistencial durante la Covid-19.

Se aplica el análisis descriptivo a la fuente de datos “Capacidad Asistencial durante la Covid-19”. Se calcula los parámetros anteriormente descritos en función de tres categorías (total de camas, camas ocupadas e ingresos), que son las que se van a tener en cuenta a lo largo del análisis de datos. Los parámetros calculados se almacenan en un archivo de tipo Xlsx.

```

# Calcular estadísticas básicas
mean = np.mean(data[columna]) # Media
median = np.median(data[columna]) # Mediana
std_dev = np.std(data[columna]) # Desviación estándar
min_value = np.min(data[columna]) # Valor mínimo
max_value = np.max(data[columna]) # Valor máximo
total = np.sum(data[columna])
mode = np.argmax(np.bincount(data[columna].astype(int))) # Moda

# Calcular los cuartiles
q1 = np.percentile(data[columna], 25) # Primer cuartil (25%)
q3 = np.percentile(data[columna], 75) # Tercer cuartil (75%)

```

Estadísticas	Total de Camas	Camas ocupadas	Ingresos
media	16555.24590163932	2625.098360655738	270.5532786885246
mediana	16879.5	2499.5	236.5
Desviacion Tipi	989.333637399315	1065.4648251368367	118.47968469729953
Valor Minimo	14008	280	46
Valor Maximo	18085	5191	682
Total	4039480	640524	66015
Moda	16341	1761	188
Primer Cuartil (16306.0	1935.0	188.0
Tercer Cuartil (17076.0	3229.0	333.0

Código 13. Cálculo y resultado de los parámetros del análisis Descriptivo en la fuente de datos “Capacidad Asistencial durante la Covid-19”.

Tambien se realizan dos tipos de gráficos. El primer grafico representa en función del tiempo la cantidad de camas ocupadas por el Covid-19 y el total de camas. Se muestra los datos de la comunidad de Madrid de cada día desde que se empieza el dataset, el 01-08-2020 hasta el 01-04-2021, que fueron las fechas más determinantes del Covid-19, en total se recogen 8 meses con muestras cada día. Esta grafica se corresponde con la figura 24.

Para la visualización de las dos siguientes figuras se ha utilizado el software PowerBi, es necesario importar el dataset que se quiere visualizar 0 y a continuación, modelar a tu gusto el tipo de herramienta deseada para la visualización y configurarla.

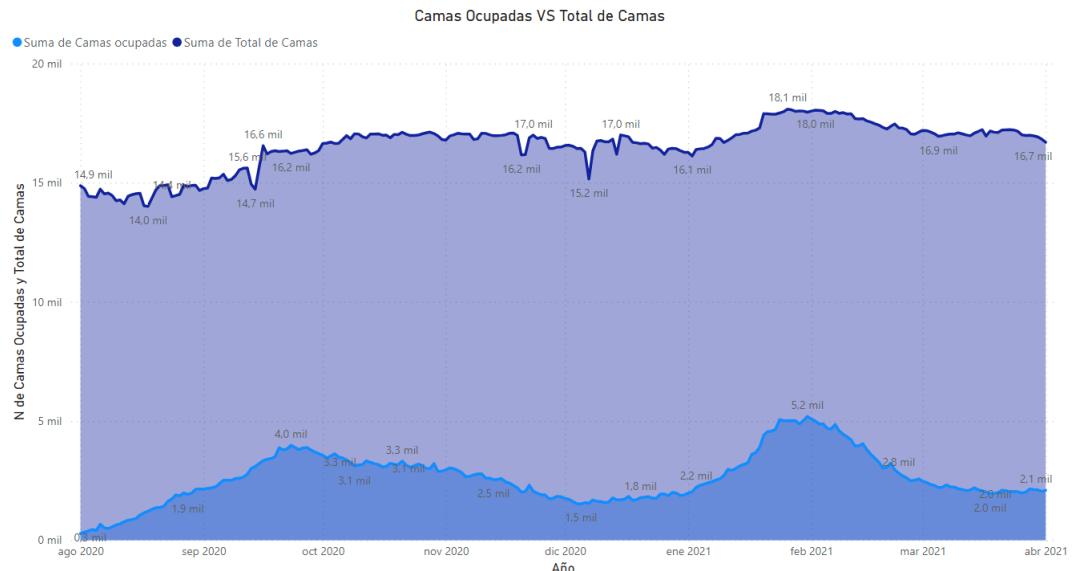


Figura 24. Total de camas VS Camas ocupadas por Covid-19.

Como se puede observar en la figura 24, las camas ocupadas por enfermos de Covid-19 durante los peores meses de la pandemia fueron insignificante respecto a todas las camas totales de los hospitales de la comunidad de Madrid. Esto se debe, a que los enfermos de Covid-19 mayormente ocupaban las camas de las UCI (Unidas de Cuidados Intensivos), siendo mucho menor la cantidad de camas de la UCI respecto las camas totales, por lo tanto, tuvo una gran presión hospitalaria respecto a las camas de la UCI.

El segundo grafico muestra la cantidad de ingresos por Covid-19 en función del tiempo (desde el 01/08/2020 hasta 01/04/2021) y en la comunidad de Madrid, que se corresponde con la figura 25.

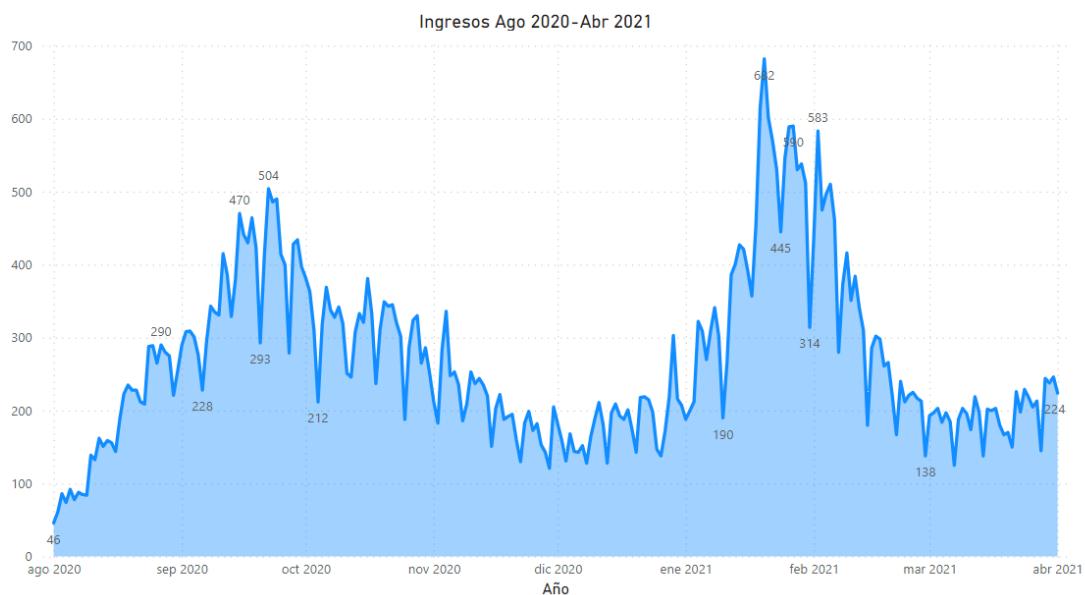


Figura 25. Ingresos por Covid-19.

En la figura 25, se puede observar que el dato de ingresos más durante la pandemia fue entre enero 2021 y febrero 2021 en la comunidad de Madrid, alcanzando los 682 pacientes ingresados.

4.5.1.2. Incendios producidos en España entre el 2006 y el 2015.

Con el mismo procedimiento anterior, pero con diferente fuente de datos se obtiene los siguientes datos estadísticos:

Estadísticas	Número de Siniestros	Número de siniestros > 500 ha	Superficie total (ha)	Superficie por %
media	12042.875	41.5625	61130.749583333345	34.89604166666667
mediana	11688.5	28.5	43138.3	33.35
Desviación Tipica	6908.763974188268	34.66440672721805	60513.423652672405	13.870104052058533
Valor Minimo	1442	6	5309.4	5.39
Valor Maximo	25557	160	335749.4	76.72
Total	578058	1995	2934275.9800000004	1675.01
Moda	1442	16	5309	32
Primer Cuartil (Q1)	6881.75	17.75	20166.43	25.315
Tercer Cuartil (Q3)	16925.25	52.25	77228.3475	42.445

Figura 26. Resultado de los parámetros del análisis Descriptivo en la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.

Utilizando la visualización también se realiza una parte fundamental del análisis descriptivo, y se puede obtener conclusiones que a primera vista son imperceptibles. Por ejemplo, a continuación, se compara el número de incendios producidos en España por año, con el número de incendios por años que superan las 500 hectáreas y por lo cual son considerados grandes incendios forestales (GIF), desde el año 1968 hasta el 2015.

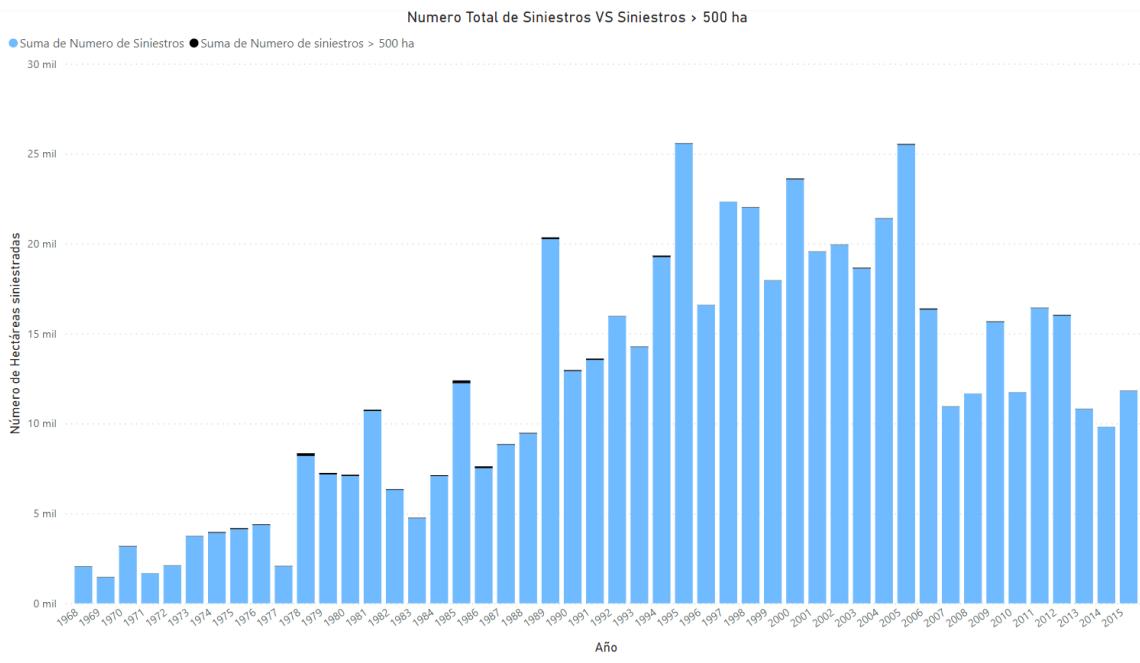


Figura 27. Número de incendios totales vs el número de incendios > 500 hectáreas.

Como se puede observar en la ilustración, el número de GIF producidos en España (representados en negro) son una pequeñísima cantidad compara con el número de incendios totales por años. Otra conclusión que se puede obtener es que en número de grandes incendios no está directamente relacionado con el número de incendios que se producen al año, ya que los GIF se producen independientemente del número de incendios.

4.5.1.3. Catálogo del Bosque Urbano de la Ciudad de Madrid.

La obtención de los valores descriptivos del análisis se realiza con el mismo procedimiento que en los puntos anteriores. Esta vez en vez de aplicar el análisis a toda la muestra, se seleccionan los campos que se consideran interesantes. Se aplica el análisis descriptivo a la cantidad de veces que aparecen cada especie, a la captación de contaminación en kg por año y al agua interceptada en m^3 por año. Tal y como se puede observar en la figura 28.

Se consideran estos campos debido al intereses que suscita una futura aplicación en las grandes ciudades. Si se conocen las ventajas e inconvenientes de cada especie sobre estos campos en cuestión, puede suponer un aumento en la calidad del aire de las ciudades y una eficiencia en la gestión de recursos.

Estadísticas	Cantidad	Captación Contaminación (Kg/Año)	Agua Interceptada (M3/Año)
media	3986.173210161662	472.0692609699769	2776.6209699769056
mediana	50.0	2.97	17.42
Desviación Tipica	28327.39540016104	3228.4989260273464	19016.09383693793
Valor Minimo	1	0.01	0.05
Valor Maximo	472183	53026.45	312878.52
Total	1726013	204405.99	1202276.8800000001
Moda	1	0	0
Primer Cuartil (Q1)	4.0	0.36	2.1
Tercer Cuartil (Q3)	536.0	34.01	199.5

Figura 28. Resultado de los parámetros del análisis Descriptivo en la fuente de datos “Catálogo del Bosque Urbano de la Ciudad de Madrid”.

Tambien aporta gran valor visualizar las diferentes propiedades de la fuente de datos, esta visualización puede aportar relaciones ocultas entre las diferentes propiedades o la obteniendo de conclusiones que serían imposible obtener a simple vista.

Por ejemplo, se quiere visualizar el agua que necesita cada especie para desarrollarse en m^3 al año comparándolo con la producción de oxígeno en toneladas al año. Se pretende realizar esta visualización para descubrir si estas dos características están relacionadas y si existe alguna conclusión que se pueda obtener que aporte un valor significativo. La siguiente visualización está realizada a partir de PowerBi.

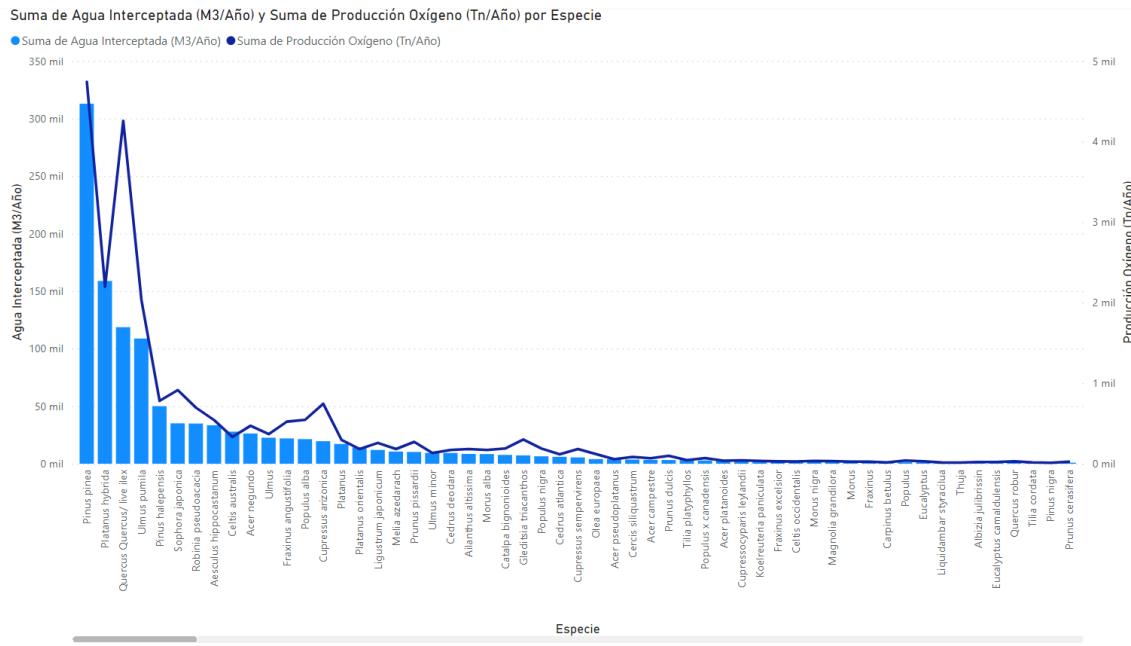


Figura 29. Agua Interceptada (M3/Año) vs Producción de oxígeno (Tn/Año) por especie.

Como se ver, sí que existe cierta relación entre la cantidad de agua necesaria y la producción de oxígeno. Se concluye que a priori las especies que necesitan notablemente más cantidad agua para subsistir producen también más oxígeno, pero, no es una regla que siempre se cumpla, sino que es una tendencia. Puede haber especies con menor captación de agua que otras, pero con mayor producción de oxígeno, tal y como se puede ver en la figura.

4.5.1.4. Población por Provincias de España 1996-2021.

Los resultados obtenidos del análisis descriptivo, a diferencia de los puntos anteriores, se ha realizado a través de la librería “*statistics*”. Esta librería es propia de Python y sirve para calcular estadísticas básicas.

A cada provincia se le ha aplicado el análisis descriptivo, obteniendo los resultados en función de histórico de datos, tal y como se puede observar en la figura 30.

Provincia	Mediana	Media	Desviacion	Rango	Maximo	Minimo	Primer Quartil (0.25%)	Tercer Quartil (0.75%)	Rango Intercuartil
Albacete	388270	385280,2	14354,57894	44240	402837	358597	376556	396987	20431
Alicante/Alacant	1836459	1748260	192580,8803	565880	1945642	1379762	1632349	1881762	249413
Almería	684426	640475,4	81336,19599	230031	731792	501761	565310	704219	138909
Araba/Álava	313819	309807,6	17301,65014	52119	333940	281821	294360	323648	29288
Asturias	1075329	1064256	23450,49448	76093	1087885	1011792	1051229	1080138	28909
Ávila	166259	165895,4	4596,652949	15064	172704	157640	163885	168825	4940
Badajoz	673559	676259,7	12017,28353	37685	694533	656848	664251	686730	22479
Baleares, Illes	1095426	1024390	125375,8588	412629	1173008	760379	947361	1113114	165753
Barcelona	5487935	5296911	360805,955	1115125	5743402	4628277	5052666	5542680	490014
Bizkaia	1147576	1145343	9103,837637	26827	1159443	1132616	1137418	1152658	15240
Burgos	358171	360786,6	9555,142219	29302	375657	346355	355205	366900	11695
Cáceres	410242	407411,6	7402,66219	25888	415446	389558	405616	412580	6964

Figura 30. Parte del resultado de los parámetros del análisis Descriptivo en la fuente de datos “Población por Provincias de España 1996-2021”.

A continuación, se selecciona únicamente la población de la provincia de Ávila, se crea un dataframe con la fecha indicada y se crea un archivo Excel, este archivo Excel es necesario para realizar la visualización con PowerBi.

Tambien se realiza, pero en el propio Python y usando la librería “Matplotlib” [34]. la visualización de cómo evoluciona la población en función de los años de todas las provincias de España.

En la figura 31 se muestra la evolución de la población de la provincia de Ávila usando un gráfico de líneas en PowerBi.

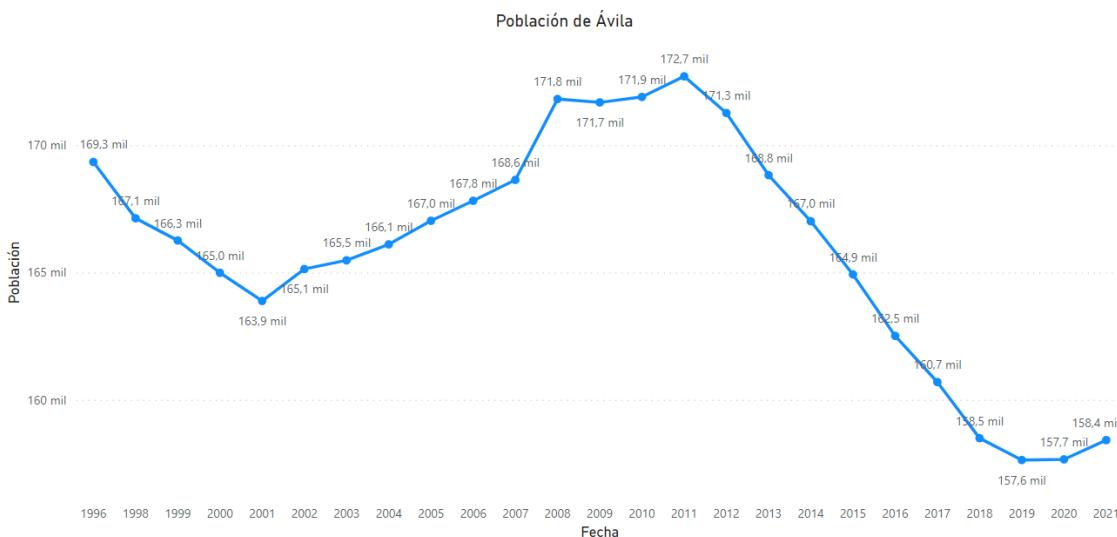


Figura 31. Evolución de la población en la provincia de Ávila en función de los años.

Como se puede observar, la población tuvo su pico en el 2011 con 172.704 habitantes y desde entonces tiene una tendencia a decrementar llegando al mínimo en 2019 con 157.640 habitantes, exponiendo el problema de la España vaciada.

Como se ha podido ver en cada uno de los modelos descriptivos, realizar una visualización de los datos es una parte fundamental del análisis descriptivo. A partir de la visualización se puede descubrir ciertas características que pueden ser fundamentales a la hora de definir los objetivos para realizar un análisis posterior de mayor complejidad.

4.5.2. Regresión Lineal.

La regresión lineal es un método estadístico que se utiliza para representar las relaciones entre una o más variables independientes y una variable dependiente. Mientras que la variable dependiente es la variable que se busca predecir, las variables independientes son las variables que se requieren para predecir la variable dependiente.

La regresión lineal está basada en la variable dependiente se representa como una función lineal de las variables independientes. La variable dependiente es representada en a partir de una combinación de la variable o de las variables independientes, más un tipo de error. Este error es el error de la variable dependiente que no puede ser representada a partir de las variables independientes, suele ser un valor aleatorio con una distribución normal.

Este modelo estadístico tiene como objetivo determinar los parámetros de la función lineal, que muestra la relación entre la variable dependiente y las variables independientes, se estiman utilizando un método que reduce la suma de los errores al cuadrado entre los valores reales y los estimados por la línea. La función lineal es la siguiente:

$$y = mx + b$$

Donde:

- y es la variable dependiente que buscamos predecir.
- x es la variable independiente o predictor.
- m es la pendiente de la línea que representa como cambia y en función los cambios en x .
- b es la coordenada origen que representa el valor de y cuando x es igual a 0.

Existen dos tipos diferentes de regresión lineal, está la regresión lineal simple y la regresión lineal múltiple.

La regresión lineal simple se caracteriza por solo utilizar una variable independiente para estimar la variable dependiente. Esto implica que la regresión lineal simple tenga una fácil interpretación, pero el poder de predicción es menos poderoso.

La regresión lineal múltiple se caracteriza por utilizar dos o más variables independientes para predecir una sola variable dependiente. Esto implica que la interpretación sea más complicada pero el poder de predicción sea mucho mayor.

4.5.2.1. Regresión Lineal Simple en Población por Provincias de España 1996-2021.

Se decide aplicar la regresión lineal simple a la fuente de datos “Población por Provincias de España 1996-2021” ya que, la información representada se caracteriza por tener una tendencia lineal, lo que significa que el valor de la variable dependiente aumenta o disminuye en función que aumenta el valor de la variable independiente. En concreto se decide realizar el análisis sobre la provincia de Ávila. La variable dependiente en este caso es la población de la provincia cada año, mientras que la variable independiente es el histórico de años.

Entonces $y = Poblacion$ y $x = Años$

La regresión lineal simple se utiliza para modelar la relación entre las dos variables y realizar predicciones sobre el valor de la población de las provincias dependiendo del nuevo año indicado.

La regresión lineal en Python se realiza a partir de las librerías “SciPy” [37] y “Scikit-Learn” [38].

El código desarrollado funciona de la siguiente manera. La librería de “SciPy” se utiliza para calcular el coeficiente de correlación entre las dos variables (población y año), que proporciona una medida de relación lineal entre las dos variables. El coeficiente de relación puede tomar valores entre -1 y 1, donde:

- 1 indica una correlación positiva perfecta. Significa que a medida que una de las variables aumenta la otra aumenta de forma lineal.
- 0 indica que no hay relación lineal, por lo tanto, no hay ningún tipo de relación lineal entre las variables.
- -1 indica una correlación negativa perfecta. Significa que a medida que una de las variables aumenta la otra disminuye de forma lineal.

A partir de los conjuntos de datos de “años” y “población” se crean los conjuntos de entrenamiento y prueba, utilizando la función “`train_test_split()`” de Scikit-learn. Los conjuntos (x_train y y_train) sirven para construir el modelo de regresión.

```

x_train, y_train = train_test_split(
    X.values.reshape(-1,1),
    y.values.reshape(-1,1),
    train_size = 0.99,
    random_state = 1234,
    shuffle = True
)

```

Código 14. Creación de los conjuntos de entrenamiento del modelo de regresión lineal.

Existen propiedades destacables de la función “*train_test_split ()*”. La propiedad *train_size* de la función indica cuanta porción de conjunto se utiliza como conjunto de entrenamiento, se le da el valor de 0.99 indicando que el 0.99 de nuestra fuente de datos sirva como modelo de entrenamiento. La propiedad *random_state* establece una semilla de números aleatorios permitiendo que cada vez que se ejecute el código con el mismo valor de *random_state* se obtenga la misma división de conjuntos y por lo tanto el mismo resultado.

A continuación, se crea el modelo de regresión lineal utilizando la función “*sm.OLS ()*”, significando OLS mínimos cuadrados ordinarios. Una vez creado el modelo se ajusta a los datos de entrenamiento.

Una vez creado el modelo se obtiene la predicción, la predicción de la población para el año 2022, y los intervalos de confianza de un 95% para los valores de entrenamiento.

Por último, se procede a almacenar los resultados en un archivo Xlsx, para después importar el archivo resultante en el software PowerBi y proceder a visualizar de forma gráfica los resultados obtenidos.

Se obtiene un coeficiente de correlación de aproximadamente -0.488, lo que significa que se obtiene una correlación negativa moderada entre las dos variables (años y población). Indica que a medida que la variable independiente (histórico de años) aumenta la variable dependiente (población) decrece.

A partir también del modelo de regresión lineal, se decide calcular se decide realizar una predicción sobre el siguiente año del cual no se tiene información, el siguiente año es el 2022. La predicción para la población en el 2022, junto el intervalo de confianza al 95% es la siguiente:

$$\text{Predición de la Población para el Año 2022} = 161.705,193$$

$$\text{Intervalo de confianza (95\%)}: [152.369,523, 171.040,864]$$

La visualización de los resultados a partir de PowerBi es la siguiente:

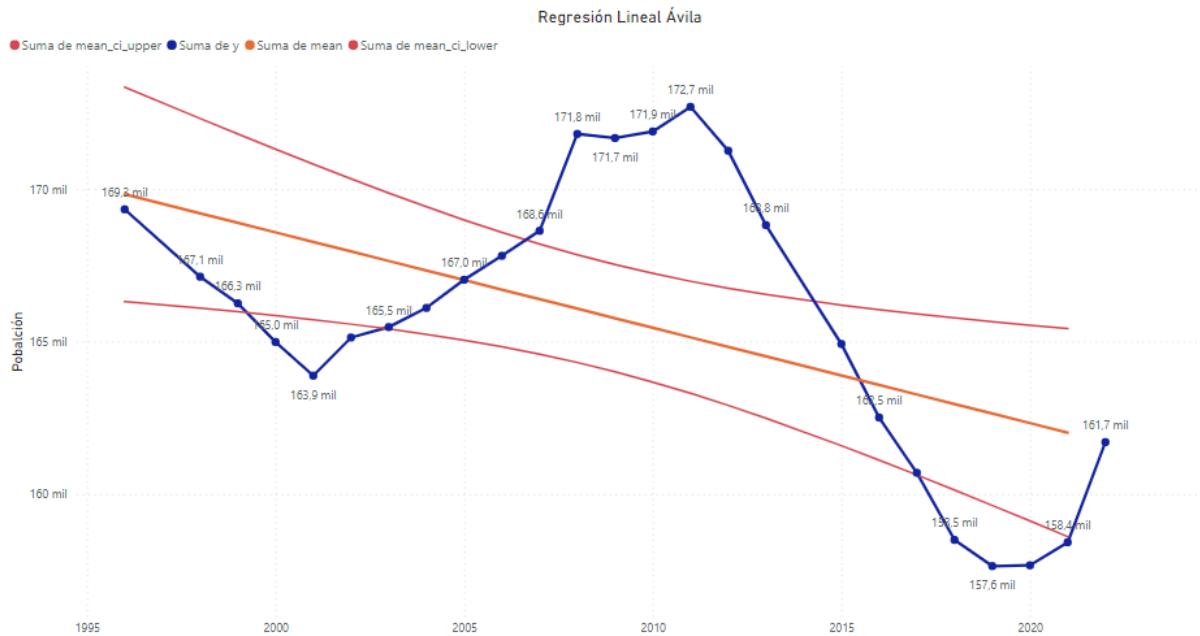


Figura 32. Regresión Lineal en la provincia de Ávila.

En la *figura 32*, se puede observar cómo evoluciona la población del Ávila en función de los años, indicando cual es la media, la media baja y alta. Tambien en la visualización se puede observar la predicción realizada para el año 2022. Esta misma visualización se encuentra en el Anexo III del documento, donde se puede el dashboard completo de todos los análisis de datos realizados.

Como el histórico de datos solo proporciona información hasta el 2021, realizar una predicción sobre el 2022, es realizar una predicción sobre el futuro. Ademas como actualmente se está en 2023, es posible determinar el error causado por la predicción buscando la población que tuvo en el 2022 la provincia de Ávila. En Wikipedia [39] encontramos que en 2022 la población de Ávila tuvo una población de 161 771 habitantes, por la tanto, se considera que el modelo es bastante preciso. Se obtiene un error en la predicción del:

$$Error = | 161.705 - 161.771 | = 66 \rightarrow 0,04\%$$

4.5.3. Clustering.

El agrupamiento (clustering) es una técnica de aprendizaje automático no supervisado que tiene como objetivo agrupar los datos que presentan alguna correlación similar en grupos, a los que se le llama clústeres. Se busca identificar patrones y estructuras intrínsecas en los datos que no son visibles en un primer análisis. Este modelo estadístico es útil para la exploración de datos, la segmentación de clientes y la detención de errores.

Existe una gran variedad de algoritmos de clustering diferentes (K-Means, Jerárquico, DBSCAN, Mean Shift), cada uno con sus propias características y enfoques diferentes, pero todos funcionan de una forma similar.

En primer lugar, el algoritmo en cuestión calcula la distancia entre cada par de dato, a continuación, el algoritmo agrupa los datos que son más cercanos entre sí y el número de clústeres queda definido por el usuario.

Los procesos de clustering en Machine Learning tiene un nivel alto de dificultad, ya que, depende de una serie de criterios y reglas que se indican a la hora de generar los clústeres. En primer lugar, el usuario debe indicar el número de clústeres que compondrá su modelo, después,

se definen las formas de los grupos similares asignando un centro (llamado centroide), desde donde se hará el agrupamiento. También se debe definir un margen de error o métrica para empezar a definir los clústeres.

Al determinar el error general, se debe incluir en el algoritmo de entrenamiento, para posteriormente generar un bucle repitiendo el proceso miles de veces para que encuentre las todas las combinaciones de errores posibles del modelo.

El clustering se decide aplicar a las fuentes de datos abiertas “*Capacidad Asistencial durante la Covid-19*” y “*Catálogo del Bosque Urbano de la Ciudad de Madrid*”, pero para la primera fuente de datos se aplica el clustering Mean Shift, mientras que para la segunda se aplica el K-Means.

4.5.3.1. Clustering Mean Shift en Capacidad Asistencial durante la Covid-19.

Se aplica el clustering a la fuente de datos “*Capacidad Asistencial durante la Covid-19*”, más concretamente en número de camas ocupadas por enfermos de Covid-19. Se decide realizar este modelo ya que, los datos a simple vista son bastante heterogéneos y es imposible clasificar o agrupar los datos.

El objetivo es clasificar en grupos las camas ocupadas durante los peores meses de la pandemia en función de la fecha producida. El agrupamiento que se aplica que se realiza a través del algoritmo Mean Shift.

El Mean Shift, se caracteriza por ser un algoritmo basado en centroide, intenta ubicar puntos centrales a cada grupo de datos, que funciona actualizando los candidatos para los puntos centrales para que sean la media de los puntos de dentro de la ventana. Los candidatos se van filtrando en la etapa de procesado en la que se eliminan duplicados, constituyendo el conjunto final de puntos centrales y sus clústeres.

Se comienza con una ventana circular con un centroide seleccionado de forma al azar y con un radio r como núcleo, donde el radio va cambiando de forma iterativa a una región con mayor densidad en cada paso, hasta que converge.

En cada iteración la ventana se mueve a regiones de mayor densidad variando el centroide a la media de los puntos dentro de la ventana. Al variar la media de los puntos de la ventana, la ventana deslizante se desplaza gradualmente hacia áreas de mayor densidad de puntos.

Se desplaza la ventana en función de la media hasta que ya no hay una dirección en la que haya más puntos, por lo tanto, aumentar la ventana de desplazamiento.

Estos tres pasos se realizan con muchas ventanas hasta que todos los puntos forman parte de una ventana. Si se superponen varias ventanas, prevalece la ventana que contiene la mayoría de los puntos. A diferencia del algoritmo K-Means no es necesario indicar antes de comenzar el proceso el número de clústeres, ya que este desplazamiento de ventanas lo descubre automáticamente. Los centroides converjan hacia los puntos de densidad máxima es bastante interesante, ya es fácil de comprender y tiene un significado dentro de los datos.

Para poder realizar el clustering por el algoritmo Mean Shift en Python se utiliza la librería “Scikit-Learn” [41]. Se agrupa la información en función de la fecha, obteniendo como resultado un día por dato. A continuación, se le aplica el clustering. El resultado del proceso se almacena en un archivo Xlsx, donde una columna es la fecha del suceso, otra el número de camas ocupadas por enfermos de la Covid-19 y otra al clúster que pertenecen. Ya con el resultado almacenado en el archivo Excel, se importa a PowerBi, donde se procede a la visualización en forma de gráfica, la cual es la siguiente:



Figura 33. Clustering Mean Shift en la fuente de datos “Capacidad Asistencial durante la Covid-19”.

Como se puede observar en la figura 33, se obtiene como resultado 4 clústeres, cada clúster está formado en función de número de camas. Los clústeres son bastante heterogéneos menos el número 3, teniendo un número similar de puntos, donde cada punto se corresponde a un día en cuestión. Los clústeres obtenidos pueden tener una gran utilidad para clasificar las categorías de emergencia a la hora de determinar la presión en la cama ocupadas estableciendo categorías y funciones en cada una de ellas.

Esta visualización también se encuentra disponible en el Anexo III del documento, pudiendo observar el dashboard completo del análisis.

4.5.3.2. Clustering K-Means en Catálogo del Bosque Urbano de la Ciudad de Madrid.

Se decide realizar un clustering a la fuente de datos abierta “Catálogo del Bosque Urbano de la Ciudad de Madrid”, ya que a simple vista es imposible clasificar la información o sacar conclusiones de valor. El objetivo de este clustering es observar que relación hay entre cada una de las especies que se tiene constancia y entre las variables de agua interceptada por m^3 al año y de la captación de contaminación de kilogramo al año.

Se tiene como objetivo realizar un modelo a partir de estas variables para obtener una mejor compresión de que especies son las mejores para las grandes ciudades. Se busca saber que especies pueden tener una gran captación de contaminación, para distribuirlas de una forma óptima para que la calidad del aire que respira la población se vea incrementada. Por el mismo lado, también se busca que especies tienen un mejor gasto de agua, ya que provoca una mejor optimización de los recursos de la ciudad.

Por último, saber que especie da el mejor rendimiento tanto en la captación de contaminación como en el consumo de agua, supondría un conocimiento de gran valor y una alta ventaja a la hora de planificar una ciudad.

Para desempeñar el clustering se utiliza el algoritmo K-Means, se caracteriza por ser un algoritmo de clasificación no supervisada que tiene como objetivo agrupar la información en clústeres en función de las propiedades. El agrupamiento se basa calculando la mínima suma de distancia entre cada punto y el centroide del clúster. Para calcular la distancia entre los puntos se suele utilizar la distancia cuadrática.

El algoritmo está formado por tres pasos:

1. Inicialización. Se selecciona un numero predefinido de clústeres, llamados k y se eligen puntos aleatorios como centroides iniciales de los clústeres. Dependiendo de cómo se seleccionen los centroides iniciales, se realiza más o menos iteraciones del algoritmo, por lo que, será más o menos eficiente.
2. Asignación de Puntos. Para cada punto de los datos se calcula la distancia desde el punto hasta todos los centroides. El punto pertenece al clúster cuyo centroide esté a menor distancia del resto de centroides.
3. Actualización de centroides. Una vez completado el paso dos y todos los centroides pertenecen a algún clúster, se calcula un nuevo centroide para cada clúster. Siendo el nuevo centroide el promedio de todas las características de los puntos asignados a ese clúster
4. Los pasos 2 y 3 se repiten constantemente hasta que no se encuentran nuevos centroides o hasta que se alcanza un número máximo de iteraciones. Cuando los clústeres convergen se considera que ya son clústeres finales. En cada iteración los puntos son reasignados a los nuevos clústeres que a su vez se han actualizado a la hora de calcular los nuevos centroides.

El algoritmo K-Means se caracteriza por ser escalable y eficiente para conjuntos de datos que son de tamaño medio a grande. Sin embargo, presenta algunas limitaciones. La selección de los centroides iniciales y su deficiencia a la hora de manejar clústeres de diferentes tamaños, densidades y formas.

Para poder realizar el clustering por el algoritmo K-Means en Python se utiliza la librería “Scikit-Learn” [40]. Primero se importan los datos resultantes del proceso ETL, se escogen las únicas columnas que se van a utilizar que son “Especie”, “Agua Interceptada (M3/Año)” y “Captación Contaminación (Kg/Año)” y se asigna un numero único a cada especie. Despues se determina el número de clústeres que va a tener, que en este modelo se van a utilizar 5 y se asigna el número de generación de números aleatorios para determinar el centroide inicial, el cual son 50 y se crea el modelo.

```
#Se crea el modelo con 5 clusters
n_clusters = 5
kmeans = KMeans(n_clusters=n_clusters, random_state=50)
```

Código 15. Creación del modelo de clasterización con el algoritmo K-Means.

A continuación, se procede a entrenar el modelo con los atributos seleccionados y se obtiene las etiquetas de clúster asignadas a cada registro. Por último, se procede a almacenar todo en un archivo Xlsx. La visualización del resultado se realiza a partir de la librería “Matplotlib” en el propio ejecutable de Python, ya el software PowerBi no se pueden visualizar gráficos en tres dimensiones.

Utilizando un gráfico de tres dimensiones, donde el eje x se corresponde con la especie en cuestión, el eje y con la cantidad de agua interceptada (M3/Año) y el eje z con la cantidad de captación contaminación (Kg/Año). Donde el grafico en cuestión es el siguiente:

Clustering de Especies por Agua Interceptada y Captación de Contaminación

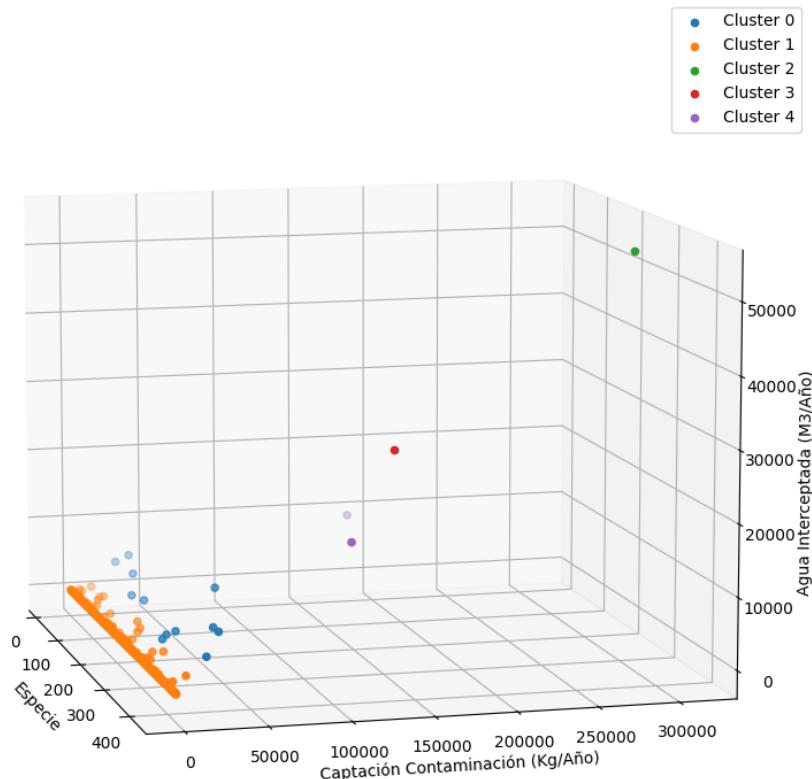


Figura 34. Clustering K-Means en la fuente de datos “Catálogo del Bosque Urbano de la Ciudad de Madrid”.

A partir de la visualización en forma de tres dimensiones se pueden sacar nuevas conclusiones y dependiendo del clúster a que pertenezca la especie tiene unas características u otras, donde cada punto de la gráfica se corresponde con una especie en concreto. Se puede observar que existe una relación lineal entre la cantidad de agua interceptada y la cantidad de captación de contaminación, ya que cuanta más agua intercepta la especie tiendo a captar más contaminación.

Las características de los clústeres más relevantes son las siguientes:

- Como se puede observar hay un clúster predominante, el clúster 1, donde se encuentran la mayoría de las especies, las cuales se caracterizan por tener una mínima cantidad de captación de contaminación y por tener un mínimo bajo consumo de agua. El resto de los clústeres tienen un tamaño mucho menor que el clúster 1.
- En el clúster 0 podemos observar que se encuentran las especies que tienen poca captación de contaminación y que también tienen un bajo consumo de agua. Estas especies de vegetación son las ideales para las grandes ciudades. Son capaces de mantener un aire limpio y optimizar de la mejor manera el consumo de agua.
- En el clúster 4 predominan las especies que tienen una capacidad media de captación de contaminación y también un gasto medio en el consumo de agua. No destacan en ninguno de los datos aspectos, pero también es una gran opción a la hora de solventar los problemas de las grandes ciudades.
- La especie del clúster 3 es una especie que tienen un gran consumo de agua comparado con el resto de las muestras y que su nivel de captación de consumo es destacable. Se puede observar claramente como esta especie es claramente un outlier.

- La especie del clúster 2 tiene un nivel alto de captación de contaminación y también tienen un gran consumo de agua. También claramente está especie es considerado un outlier.

4.5.4. Serie Temporal ARIMA.

Una serie temporal consiste en un conjunto de datos que se recopilan y se registran en intervalos regulares de tiempo. Los datos se registran secuencialmente donde reflejan cambios y variaciones a lo largo del tiempo. Las series temporales se aplican a infinitos campos, como pueden ser la economía, la meteorología, la ingeniería y más.

Existen varios objetivos a abarcar cuando se emplea una serie temporal, que son la descripción y la predicción. Las series temporales tienen una serie de componentes habituales, donde son los siguientes:

- Tendencia. La tendencia de una serie temporal a lo largo del tiempo refleja la tendencia presente en ella. Puede moverse en dirección ascendente, descendente o constante. Las alteraciones a largo plazo en los datos son capturadas por la tendencia.
- Efecto estacional. La estacionalidad describe patrones recurrentes que ocurren regularmente, como variaciones mensuales o anuales. Una serie temporal se considera estacionaria si sus propiedades estadísticas como son la media, varianza y más son constantes a lo largo del tiempo.
- Componente aleatoria. Una serie de tiempo solo queda con el componente aleatorio o residual después de identificar y eliminar la tendencia y la estacionalidad. Las tendencias a largo plazo o los patrones predecibles están ausentes de este componente. Significa fluctuaciones irrationales que pueden ser causadas por factores impredecibles o ruido de datos y que no pueden atribuirse a causas sistemáticas.

ARIMA (Autoregressive Integrated Moving Average) es un método estadístico que se utiliza para el análisis y pronóstico de series temporales. Este modelo estadístico combina tres componentes principales. El modelo autorregresivo (AR), el modelo de media móvil (MA) y la diferencia integrada (I). Lo que es lo mismo que a un modelo autorregresivo y media móvil (ARMA) integrado n veces. Se explica detalladamente cada componente del modelo.

Autorregresivo (AR). Lo más importante es determinar su orden de autorregresión, llamado p . Este orden indica hasta qué retardo una variable depende de los valores de ella misma retardada.

Donde:

y_t = Es el valor de la serie temporal en el momento actual “ t ”. Es el valor que se trata de predecir

a = Es la constante de intersección. Representa el valor base de la serie temporal cuando las demás variables son cero.

φ = Es el coeficiente de autorrelación, indica como el valor de actual de la serie (y_t), esta relacionado con su valor retardado un periodo (y_{t-1}).

y_{t-1} = Es el valor de la serie temporal retardado un periodo en el tiempo.

ϵ_t = Es el término de error producido en el tiempo “ t ”. Representa el nivel de aleatoriedad del modelo.

Un modelo AR es aquel que cuenta con una variable dependiente retardada un periodo.

$$y_t = a + \varphi y_{t-1} + \epsilon_t$$

Un modelo AR es aquel que cuenta con una variable dependiente retardada dos periodos.

$$y_t = a + \varphi y_{t-1} + \varphi y_{t-2} + \epsilon_t$$

Y así continuamente hasta que la variable dependiente está retardada en p periodos.

$$y_t = a + \varphi y_{t-1} + \dots + \varphi y_{t-p} + \epsilon_t$$

Se puede expresar el modelo AR a partir de las fórmulas anteriores y su simplificación:

$$y_t(1 - \varphi y_{t-1} - \dots - \varphi y_{t-p} L^p) = a + \epsilon_t$$

Media móvil (MA). Igual que en modelo anterior debemos tener en cuenta que lo más importante es su orden de media móvil q . La orden indica hasta qué retardo una variable depende de los errores de sus observaciones anteriores. Donde:

Θ = Es el coeficiente de la media móviles correspondientes a los retardos de los errores anteriores.

Un modelo MA es aquel que cuenta con su error de retardo de un periodo.

$$y_t = a + \epsilon_t + \theta_1 \epsilon_{t-1}$$

Un modelo MA es aquel que cuenta con su error de retardo de dos periodos.

$$y_t = a + \epsilon_t + \theta_1 \epsilon_{t-2}$$

Un modelo MA es aquel que cuenta con su error de retardo de p periodos.

$$y_t = a + \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_p \epsilon_{t-p}$$

Se puede expresar el modelo MA a partir de las fórmulas anteriores y su simplificación:

$$y_t = a + \epsilon_t (1 - \theta_1 L - \dots - \theta_d L^d)$$

Por último, tenemos la parte integrada (I) del modelo ARIMA. Lo más importante es su orden de integración, que es el número de veces que es necesario integrar el modelo para pasar de no ser estacionario a ser estacionario. Unificando cada uno de los componentes se forma la serie temporal ARIMA (p, d, q) .

El modelo ARIMA de la serie temporal debe ser estacionaria para poder realizar los pronósticos precisos, por eso, se busca transformar una serie no estacionaria en estacionaria. La no estacionariedad de una serie temporal se puede dar por la existencia de tendencia o por la existencia del componente estacional. Dependiendo si el motivo es una u otra se debe diferenciar la expresión anterior.

- La existencia de tendencia. Se debe diferenciar el modelo d veces, hasta que el modelo resultante nos dé como resultado un modelo estacionario. La expresión que varía sobre el modelo original es la siguiente:

$$(1 - L)^d$$

- La existencia de componente estacional. Cuando una serie temporal muestra patrones estacionales, como pueden ser fluctuaciones regulares en intervalos de tiempo concretos es estacionaria. Para solventar la estacionalidad se aplica una diferencia estacional, que consiste en tomar la diferencia entre observaciones separadas por el número de periodos

en la estacionalidad, llamada con “s”. Cuando al modelo ARIMA (P, D, Q) tiene un componente estacional se llama SARIMA. La expresión que varía sobre el modelo original es la siguiente

$$(1 - L^s)^d$$

Ya con todos los componentes del modelo ARIMA (p, d, q) explicados, se expresa de la forma mas general y simple posible su expresión:

$$y_t(1 - \varphi y_{t-1} - \cdots - \varphi y_{t-p} L^p)(1 - L)^d = a + \epsilon_t(1 - \theta_1 L - \cdots - \theta_d L^d)$$

4.5.4.1. Serie Temporal ARIMA en Capacidad Asistencial durante la Covid-19.

Se decide realizar un modelo ARIMA a la fuente de datos abierta “Capacidad Asistencial durante la Covid-19”, ya que se trata de una fuente de datos con datos que dependen del tiempo en los que se ha observado, donde se tiene un registro con fechas y valores asociados temporalmente. El objetivo de aplicar ARIMA a la fuente de datos es para poder ser capaces de predecir valores, se comprueba como de óptimos son los valores precedidos comparándolo con los valores actuales de la muestra.

Para la realización del modelo de serie temporal ARIMA es necesario utilizar las librerías de “Pandas” [4], “Statsmodels” [43] y “Pmdarima” [44].

Lo primero a realizar en el código es leer el archivo Excel donde se tiene almacenada toda la información de la fuente de datos y se convierte la columna de fechas como índice del dataframe ordenando los datos por fecha, se realiza a través de “Pandas” [4]. A continuación, se crea la serie temporal en función de la fecha y de la columna “Camas ocupadas”, la serie temporal ha de ser estacionaria para tener unos resultados óptimos y si no es estacionaria es necesario transformarla a estacionaria como se ha explicado anteriormente.

La serie temporal obtenida es estacionaria, por lo que, no es necesario transfórmala y es válida para el modelo. Se comprueba que la serie temporal es estacionaria a través de la prueba de Dickey-Fuller. Esta prueba consiste a grandes rasgos en determinar si la serie temporal tiene una raíz unitaria, lo que indicaría que no es estacionaria.

A continuación, es necesario seleccionar los parámetros del modelo ARIMA, encontrar los valores p, d, q . Se pueden definir de multitud de formas, pero para que el modelo sea más preciso se utiliza la librería “Pmdarima” [44], se automatiza el proceso de selección de los parámetros a través de una gama de posibles valores utilizando criterios como el AIC (Akaike Information Criterion) y el BIC (Bayesian Information Criterion) que evalúan la calidad de los modelos para cada uno de los posibles parámetros obteniendo el que mejor se ajusta a nuestra serie temporal. El nuestro modelo se obtiene que los valores óptimos para los parámetros p, d, q son:

```
Best model: ARIMA(2,2,1)(0,0,0)[0]
Total fit time: 8.517 seconds
Best p,d,q values are: 2 2 1
```

Código 16. Mejores valores encontrados para p, d, q en el modelo ARIMA

Una vez obtenidos los valores que son óptimos para nuestra serie temporal, se crea el modelo ARIMA, se crea a partir de la función “sm.tsa.ARIMA ()”. Una vez creado el modelo, se obtiene una serie de características y propiedades que indican la calidad del modelo, las características obtenidas sobre el modelo en forma de gráficas son las siguientes:

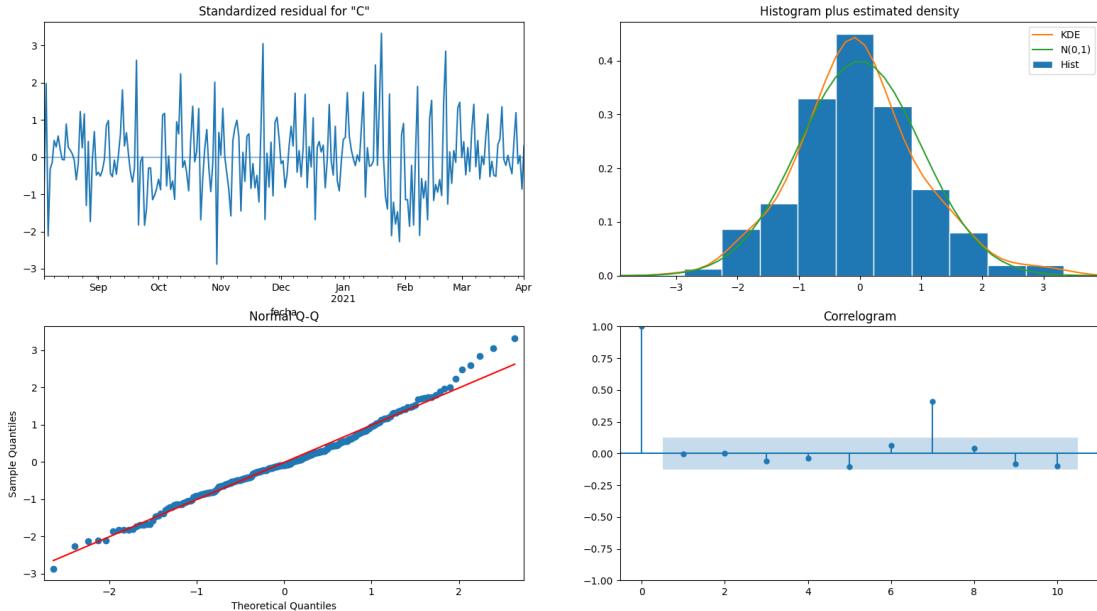


Figura 35. Diagnóstico del modelo ARIMA.

El diagnóstico se compone de cuatro gráficas, donde cada una indica las siguientes propiedades:

- La principal observación obtenida del diagnóstico del modelo es garantizar que los residuos no estén correlacionados y se distribuyan de forma que tiendan una media de cero. Tal y como se puede observar en la primera imagen a la izquierda.
- En la gráfica en la parte superior a la derecha, se observa que la línea de KDE sigue de cerca la tendencia de la línea $N(0,1)$, lo que indica que los residuos se distribuyen normalmente. La línea $N(0,1)$ indica la notación estándar para una distribución normal con media 0 y desviación estándar 1.
- En la grafico normal q-q de la parte inferior izquierda se indica que la distribución ordenada de residuos, que son los puntos azules, sigue la tendencia lineal de las muestras obtenidas. Esto significa que los residuos se distribuyen normalmente, sino los puntos no siguieran la tendencia implicaría que la distribución es sesgada.
- En el gráfico de correlación en la parte inferior a la derecha, también conocido como ACF, muestra que los errores residuales no están autocorrelacionados, si estuvieran autocorrelacionados implicaría que existe algún patrón en los errores que no se explica en el modelo, por lo que, habría que buscar más predictores para el modelo.

Este diagnóstico y las conclusiones obtenidas indican que el modelo realizado produce un ajuste satisfactorio y que es un modelo válido.

Ya con el modelo creado se procede a realizar pronósticos, se realizan los pronósticos comparándolo con valores existentes en nuestros datos. Se realiza la comparación para observar la calidad de los valores predichos y ver si se ajustan con la realidad. La predicción se realiza a través de las funciones “*get_prediction()*” donde se obtienen los valores y de la función “*conf_int()*” donde se obtiene el intervalo de confianza.

La predicción se indica que comienza el 1 de febrero del 2021, realizando la comparación con los meses de febrero y marzo. Tanto el histórico total de las camas ocupadas como las predicciones realizadas y sus intervalos se almacenan en un archivo Excel, que posteriormente se

introducen al software PowerBi para realizar la visualización de los resultados. La visualización de los resultados es la siguiente:

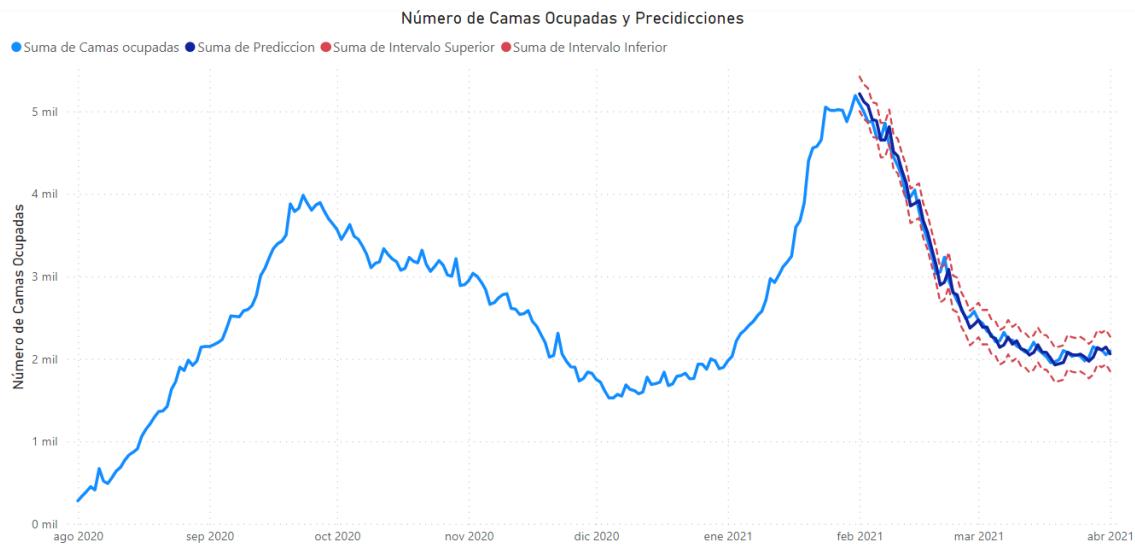


Figura 36. Prediccion de los meses de febrero y marzo.

Como se puede observar en la *figura 36*, la línea azul se corresponde con el histórico por día de las camas ocupadas por enfermos Covid-19, mientras que la línea azul oscuro es el resultado de la predicción obtenida, empezando desde el 1 de febrero y acabando el 1 de abril. Las líneas de guiones rojas se corresponden al intervalo superior e inferior. Esta visualización se encuentra disponible también en el Anexo III, donde se puede observar el dashboard completo del análisis.

Las predicciones de camas ocupadas obtenidas en general se alinean correctamente con los valores reales, tal y como se puede observar en la *figura 36*. Esto nos indica nuevamente que el modelo ARIMA realizado funciona correctamente.

4.5.5. Red Neuronal Recurrente LSTM.

Para comprender correctamente una red neuronal recurrente (RNN), primero se debe comprender qué es una red neuronal. Una red neuronal es un modelo computacional que se basa en el cerebro humano y consiste en una serie de "neuronas" artificiales conectadas que colaboran para resolver tareas específicas.

Cada una de estas neuronas artificiales dentro de una red neuronal consta de dos funciones principales, la suma ponderada y la función de activación.

La suma ponderada, las entradas se multiplican por pesos asignados a las conexiones y luego se suman.

Función de activación, la suma ponderada se pasa una función no lineal encargada de determinar si la neurona debe de activar o no.

Una red neuronal consta de varias capas, la capa de entrada, las capas ocultas y la capa de salida:

- Capa de entrada. Recibe la información inicial. Los nodos se encargan de procesar los datos, los analizan, los clasifican y se pasan a la siguiente capa.
- Capas ocultas. Estas capas intermedias realizan diferentes operaciones sobre los datos entrantes a medida que se van propagando por las diferentes neuronas de la red. Cuantas más capas ocultas tenga la red, se la considera más profunda.

- Capa de salida. Genera el resultado final del proceso, puede estar formado por varios nodos.

La estructura de la red neuronal se va ajustando a medida que se va realizando el proceso de entrenamiento. La red necesita recibir una serie de datos y una serie de respuestas anticipadas a esos datos. Luego, la red compara sus predicciones con las respuestas anticipadas y se ajusta para reducir el error de las predicciones realizadas con las respuestas anticipadas.

Las redes neuronales recurrentes es una arquitectura de red neuronal enfocada a trabajar con datos secuenciales, donde el orden y la dependencia temporal de la información es importante. No trata cada entrada a la red como un dato independiente, sino que una RNN considera la secuencia completa y procesa cada dato teniendo en constancia la información global que han aportado los datos anteriores. Lo que provoca que las neuronas puedan interaccionar y transmitir información recíprocamente y no de forma unidireccionalmente, formando bucles de retroalimentación.

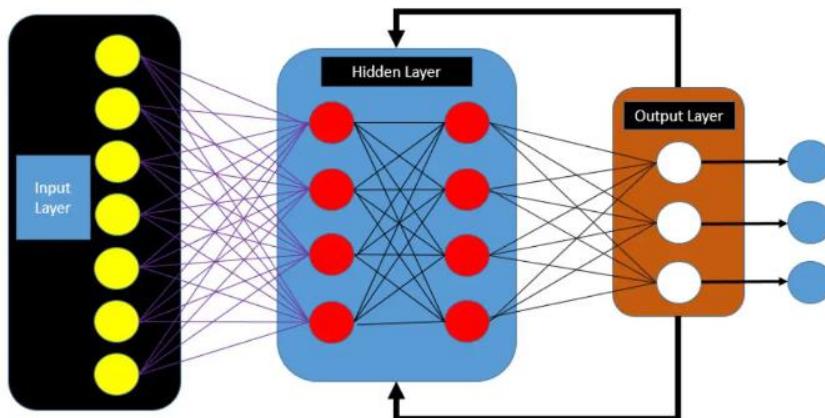


Figura 37. Arquitectura de una red neuronal recurrente.

Cada elemento que se introduce a la red neuronal se considera una unidad de tiempo en la secuencia global a analizar. Lo que diferencia a esta red neuronal del resto es su capacidad de mantener los estados ocultos y compartir la información entre las diferentes neuronas. Cada neurona consta de conexiones recurrentes que la permiten recibir datos sobre otra neurona con una unidad de tiempo anterior.

En este tipo de red neuronal su estado actual depende tanto de la entrada de información principal como del estado interno de la red inmediatamente anterior. Lo que provoca, que los patrones de excitación de la red puedan memorizar cadenas de datos de entradas.

Los algoritmos de aprendizaje en las RNN son más complejos, ya que se ha de incluir la dimensión temporal. En los años 90 se desarrolló el algoritmo denominado retro propagación a través del tiempo (BPTT), en el cual es necesario calcular el gradiente de la función de pérdida respecto al peso sináptico en cada instante de tiempo. El gradiente de total es la suma total de los valores del gradiente en cada uno de los instantes:

$$\frac{\partial E}{\partial w_{ij}} = \sum \frac{\partial E_t}{\partial w_{ij}}$$

El sumatorio tiene lugar en cada uno de los instantes de tiempo, desde el 0 hasta el T

$$w_{ij} = w_{ij} + \alpha \frac{\partial E}{\partial w_{ij}}$$

Donde E_t es la función de perdida en el momento de tiempo t.

Donde w_{ij} es el valor del peso sináptico.

Donde $\frac{\partial E}{\partial w_{ij}}$ es el gradiente en el tiempo t.

Donde α es la tasa de aprendizaje.

Pero este algoritmo de retro propagación a través del tiempo presenta un inconveniente, al introducir la nueva dimensión temporal, el problema de la desaparición del gradiente se acentúa y más en los primeros instantes de tiempo.

En las redes neuronales el gradiente de la función de perdida se propaga desde la capa de salida hasta la salida haciendo que en las primeras capas tienda a cero, provocando que sus pesos sinápticos fueran muy bajos teniendo bajos niveles de aprendizaje. Al tener en cuenta ahora la dimensión temporal provoca que los gradientes propios a los últimos instantes de tiempo se vena poco afectado matemáticamente por los primeros instantes de tiempo. Por lo cual la red, no memoriza, sino que es capaz de recordarlos instantes de tiempo más recientes. Para abordar este problema se desarrollaron arquitecturas más avanzadas como son la Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU).

Se aplica la red neuronal LSTM a una de las fuentes de datos seleccionadas, la RNN LSTM como se ha explicado anteriormente es una nueva arquitectura de red neuronal recurrente más avanzada que aborda el problema de desvanecimiento de gradientes y permite capturar dependencias a largo plazo. La red neuronal recurrente LSTM está compuesta por unidades complejas que a su vez están compuestas por cinco neuronas. Cada unidad está compuesta por las siguientes partes:

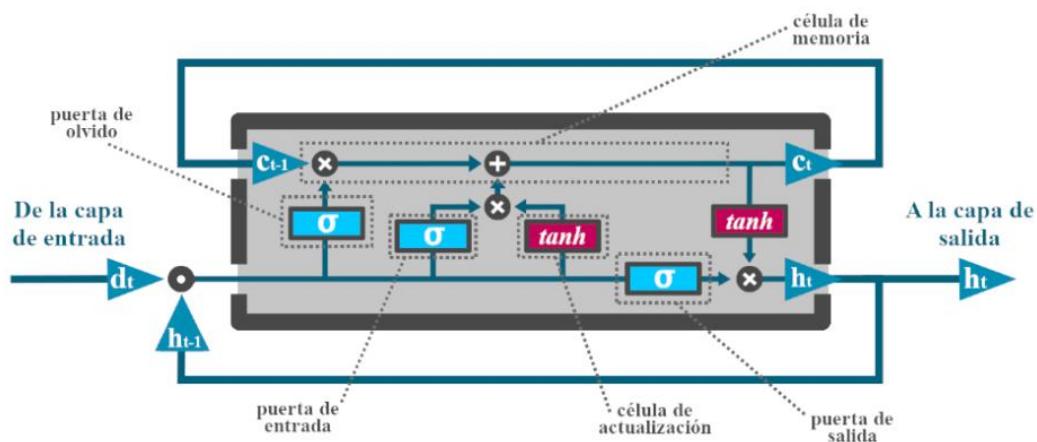


Figura 38. Arquitectura de una unidad de una red recurrente LSTM.

- Neurona de memoria (c_t). Se encarga de almacenar la información a lo largo del tiempo en los diferentes estados.
- Neurona de actualización (u_t). Se encarga de recibir la nueva información que proviene de la capa anterior.
- Puerta de entrada (i_t). Consiste en determinar cuanta información que proviene de la neurona de actualización (u_t) se almacena en la neurona de memoria (c_t).

- Puerta de salida (o_t). Determina cuanta información proveniente de la neurona de memoria se transmite a la siguiente capa. El estado de la unidad, $h(t)$, se determina multiplicando el valor de la puerta de salida (o_t) por el de la neurona de memoria(c_t).
- Puerta de olvido (f_t). Determina cuanta información debe eliminar la neurona de memoria en un instante en el tiempo t .

4.5.3.2. Red Neuronal Recurrente LSTM en Incendios producidos en España entre el 2006 y el 2015.

Se decide aplicar la red neuronal recurrente de tipo LSTM a la fuente de datos abierta “*Incendios producidos en España entre el 2006 y el 2015*”, con el objetivo de poder predecir como se van a desarrollar los diferentes incendios a lo largo de los años en España utilizando el histórico de datos.

Gracias a como se produce el funcionamiento del LSTM se puede predecir incidencias futuras a partir de parámetros anteriores, gracias a que las neuronas se comunican de una forma reciproca y no unidireccional teniendo en todo momento constancia de los estados anteriores. Cuanto mayor sea el número de datos que con lo que se entrena a la red neuronal recurrente más acertados y fiables serán los resultados. La fuente de datos abierta a utilizar tiene en total 48 filas (los incendios producidos entre el 1968 y el 2015), donde cada fila se considera una unidad de tiempo, en este caso, cada fila es un año de la muestra. Como se puede deducir la muestra utiliza para el modelo no está muy poblada, no tiene una gran cantidad de datos, por lo que los datos no serán tan fiables, ya que la RNN no estará tan entrenada.

Aunque se necesiten más datos para suministrarle a la red aplicarle este tipo de red LSTM a la fuente de datos puede producir resultados interesantes que se asemejen con la realidad y se decide aplicar este modelo ya que es un modelo matemático con un proceso muy diferente de por ejemplo el modelo usado anteriormente en otra fuente de datos, la serie temporal ARIMA, por lo que puede aportar una perspectiva totalmente diferente.

La red neuronal recurrente LSTM se va a utilizar para predecir de una manera orientativa el número de siniestros (incendios) que se van a producir en los próximos 10 años partiendo desde el 2015. Tambien se aplica la RNN desde otro enfoque, para predecir el porcentaje de superficie de superficie quemada en GIF (grandes incendios forestales) en hectáreas de los próximos 10 años.

Para desarrollar el código pertinente se han utilizado las siguientes librerías de Python: “Pandas”, “NumPy”, “TensorFlow” [47], “Scikit-Learn” [48].

Lo primero a realizar es abrir el documento utilizando Pandas y seleccionar las dos únicas columnas a utilizar, la columna de años y la columna de la superficie quemada en porcentaje de los grandes incendios forestales. Después se procede a la preparación de dato, utilizando la función “*MinMaxScaler ()*” de Scikit-Learn se transforman los datos en un rango entre el 0 y el 1, lo cual mejora el rendimiento a la hora de entrenar la RNN.

Se crean secuencias de tiempo y sus respectivas etiquetas a partir de los datos escalados y definiendo cada secuencia de tiempo con una longitud, se determina que la mejor longitud de cada secuencia es 5, las cuales se utilizan como entrada a la hora de entrenar la red neuronal.

A continuación, se crea el modelo LSTM secuencial (las capas se conectan una tras otra) que consta de 200 neuronas, con activación ReLu que activa o desactiva las neuronas dependiendo de si la entrada es negativa o positiva. Tambien se indica que la capa sea densa haciendo que todas las neuronas de la capa anterior se conecten con cada neurona de la siguiente capa. Acto seguido se compila el modelo utilizando como optimizador el “Adam” y la función de perdida “mean_squared_error”.

Una vez creado el modelo se procede a entrenarlo, se entrena a partir del histórico de datos. Para entrenar la red es necesario determinar el número de épocas (epoch), que son las iteraciones completas de todo el conjunto de dato de entrenamiento, es importante determinar el número de epochs para que las predicciones sean precisas y evitar el sobreajuste. Se le da el valor de 120 épocas.

Tambien es necesario determinar el tamaño del lote (batch_size), que son el número de entrenamientos que se realizan en una sola iteración de actualización de los pesos del modelo. Se le da un valor de 8 lo que significa que, en cada paso de actualización de los pesos, el modelo utiliza 8 ejemplos de entrenamiento.

```
# Entrenar modelo
model.fit(X_train, y_train, epochs=120, batch_size=8)
```

Código 17. Propiedades del entrenamiento del modelo LSTM parte 1.

Una vez ya entrenado el modelo se procede a realizar las predicciones, a través de un bucle realizamos las predicciones para los próximos 10 años. Cada vez que se va realizando una nueva predicción esta se va almacenando y se actualiza la secuencia para prepararla para la siguiente iteración. Esto es necesario para que la secuencia que se utiliza para la siguiente predicción incluya la información actualizada.

Cuando ya se han obtenido los resultados de las predicciones se desescalan utilizando el inverso del escalador para poder obtener los valores reales de las predicciones, ya que se almacenan con un rango comprendido entre 0 y 1.

Por último, se crea un dataframe con el histórico de datos utilizado y las predicciones, y ese dataframe se pasa a un archivo, para posteriormente visualizar los resultados obtenidos en PowerBi, obteniendo la siguiente visualización:

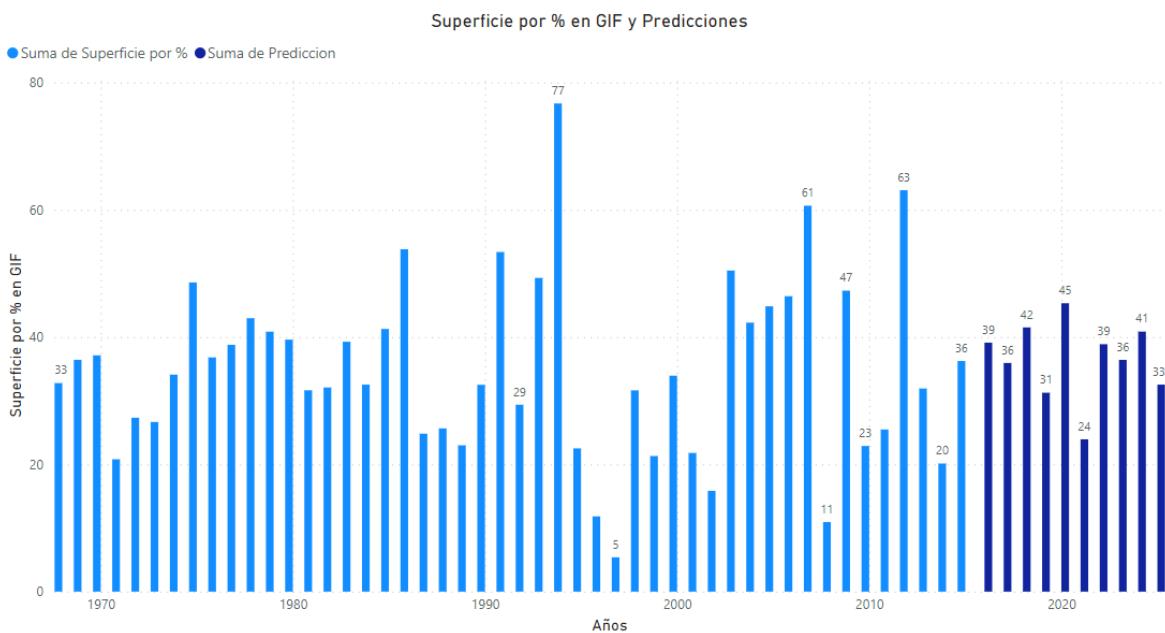


Figura 39. Resultados de las predicciones de la superficie por % en GIF de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.

Como se puede observar en la figura 39, los resultados obtenidos a partir del año 2015 corresponden a las predicciones obtenidas, donde presentan una tendencia muy similar a los datos reales previamente registrados.

Se desarrolla el mismo programa anterior, pero ahora aplicando la red neuronal recurrente a las columnas años y numero de siniestro. Se busca predecir de una manera orientativa la cantidad de siniestros que se van a producir en los próximos años. No deja de ser una aproximación, ya que la cantidad de información para entrenar la RNN no deja de ser muy escasa.

Se realiza el mismo código que el programa anterior, pero para hacer este modelo se cambia tanto el número de épocas (epochs), tanto el número de tamaño del lote (batch_size). Siendo:

```
# Entrenar modelo
model.fit(X_train, y_train, epochs=125, batch_size=16)
```

Código 18. Propiedades del entrenamiento del modelo LSTM parte 2.

Exportando las predicciones obtenidas y utilizando PowerBi obtenemos la siguiente visualización sobre el número de siniestros reales desde 1968 hasta el 2015 y el número de siniestros precedidos entre el 2016 y el 2025:

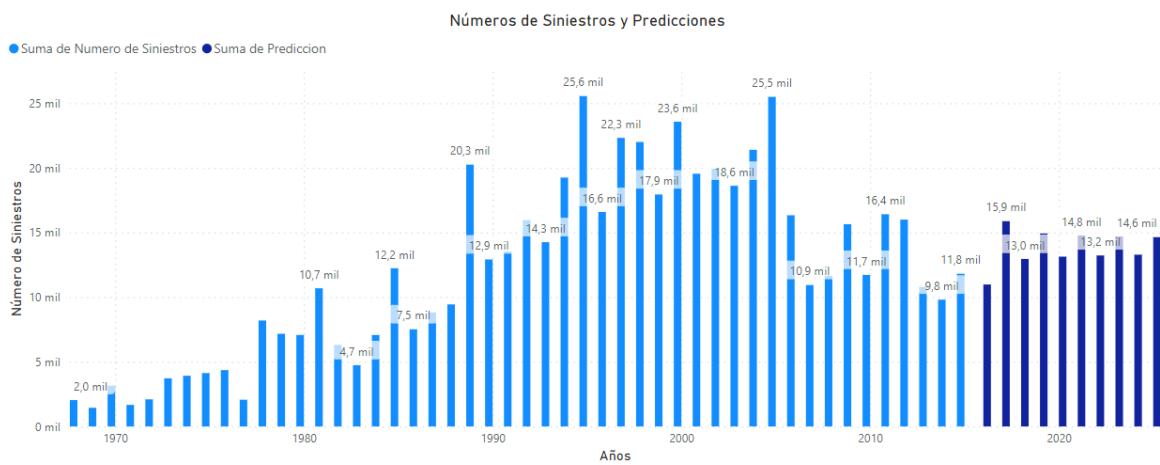


Figura 40. Resultados de las predicciones del número de siniestros por año de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”.

Como se puede ver en la *figura 40* los resultados están acordes con la información real con la que se está trabajando, pero en un primer análisis pueden que no sea una predicción muy realista, sino que es un enfoque más orientativo que puede aportar diversos beneficios.

Estas dos últimas visualizaciones, como todas las obtenidas durante el proyecto, se pueden visualizar también en el Anexo III, donde se expone el dashboard completo en PowerBi.

Capítulo 5

5. Conclusiones y Líneas Futuras.

En este capítulo se va a explicar detalladamente las conclusiones obtenidas del trabajo y sus resultados, sus posibles aplicaciones en la actualidad, sus posibles mejoras, la escalabilidad del proyecto, el análisis del negocio y el presupuesto utilizado para desarrollar el proyecto.

5.1. Conclusiones.

A lo largo de este trabajo de fin de grado se han expuesto multitud de conceptos e ideas. El proyecto se puede dividir en diferentes fases, en concreto:

- La investigación y selección de las diferentes fuentes de datos abiertas, según una serie de criterios en referencia con los objetivos.
- La realización de diferentes programas para aplicar el proceso ETL a cada una de las fuentes de datos abiertas seleccionadas, usando el lenguaje de programación Python.
- Con los resultados de los procesos ETL sobre las diferentes fases de datos abiertas, realizar y aplicar modelos de análisis de datos incluido modelos de machine Learning.
- Visualización de las fuentes de datos abiertas y de los resultados obtenidos utilizando el software PowerBi.

En la primera fase de investigación y selección de fuentes de datos, se ha realizado una selección de fuentes de datos que abarcan multitud de campos diferentes y en concreto abarcan los campos de los proyectos internacionales de la unión europea “Eugloh” y “Smacite”, el cual era un objetivo principal. Sobre todo, se querían utilizar este tipo de fuentes de datos abiertas porque cada uno de estos proyectos comparten pequeños objetivos de lo que se espera al utilizar fuentes de datos con estos dominios.

Al abarcar tantos campos diferentes como se ha hecho en el proyecto, se puede concluir que para el funcionamiento del proyecto no importa de qué tipo de campo se abarque en la fuente de datos, ya que será útil para el proyecto.

Lo que ha tenido importancia a la hora de seleccionar una fuente de datos u otra, a parte del dominio al que pertenezca, es la calidad de los datos con los que se está tratando, la cantidad de datos que hay en la fuente de datos, en qué tipo de formato están presentado los datos y el interés que existe a la hora de trabajar con ellos.

Sobre todo, se ha buscado seleccionar fuentes de datos abiertas que estuvieran constituidas de diferentes formas, para poder realizar el proyecto lo más versátil posible. Se han seleccionado fuentes de datos almacenadas en diferentes tipos de archivos y con diferentes tipos de estructura, ya sea fuentes de datos estructuradas, semi estructuradas o no estructuradas.

En la segunda fase del proyecto se realizó el proceso ETL para cada fuente de datos que seleccionada. Como se ha expuesto anteriormente, dependiendo del tipo de archivo donde estuviera almacenada la información y el tipo de estructura se ha realizado un proceso u otro.

Se ha completado el proceso en todas las fuentes de datos seleccionados, incluyendo a la que más dificultad presentaba, una fuente de datos abierta donde la información se presentaba de forma desestructurada en forma de texto y en un archivo PDF. El proceso ETL es una parte fundamental que tienen que pasar las fuentes de datos abiertas para poder luego pasar a la siguiente fase, a la fase de análisis.

Se ha intentado abarcar el mayor número de tipos de archivos diferentes y tipos diferentes de estructuración de datos. Se ha conseguido abarcar un número considerable de ambos aspectos aportando conocimiento de valor de los procesos ETL.

A partir de esta problemática, se ha realizado una comparativa entre la multitud de librerías diferentes que ofrece Python para los procesos de extracción, transformación y carga. Exponiendo cuál de ellas es más beneficiosa para que tipo de almacenamiento, ya que para la realización de los objetivos ha sido necesario realizar una investigación e ir probando poco a poco cuál de ellas ofrecía un mejor rendimiento.

En la tercera fase, se ha realizado todo tipo de análisis a las fuentes de datos abiertas que se han seleccionado para ello, esta selección ha sido en función de diversos argumentos por los cuales se cree que esas determinadas fuentes de datos eran las más idóneas para esos análisis.

Se ha conseguido realizar un análisis de datos con diferentes modelos que abarcan el mayor conocimiento posible, empezando desde un modelo poco complejo como descriptivo y terminando con modelos mucho más complejos como son las series de tiempo o las redes neuronales recurrentes. Se ha conseguido a través de una serie de fuentes de datos abiertas suministradas por diferentes organismos realizar múltiples tipos de análisis que cuyos resultados tienen un valor significativo. Los resultados obtenidos en los diferentes modelos llevados a cabo a un ámbito profesional pueden tener un impacto real sobre la vida cotidiana del ciudadano.

En la última y cuarta fase se ha conseguido realizar las visualizaciones de los resultados o de las diferentes fuentes de datos abiertas a través del software de visualización PowerBi. PowerBi es una herramienta muy poderosa con infinidad de configuraciones que permiten mostrar de una forma muy clara y concisa los resultados a destacar. Se ha conseguido obtener un cierto nivel en el software, el cual el conocimiento sobre la herramienta al inicio del proyecto era nulo.

En conclusión, se ha conseguido realizar el proceso completo. Seleccionando diferentes fuentes de datos abiertas con ámbitos relacionados con los proyectos europeos “Eugloh” y “Smacite”, aplicándole a estas fuentes de datos los determinados procesos ETL para que a continuación se puede aplicar los diferentes modelos estadísticos y de machine learning, completando el proceso aportando la visualización de los resultados y de las fuentes de datos a través de PowerBi. Por lo que se han cumplido todos los objetivos planificados desde el inicio.

Durante los estudios universitarios se ha dado temario en determinadas asignaturas en donde el temario impartido ha servido como base para la realización en parte del proyecto. Las asignaturas que se pueden destacar son fundamentos de la ciencia de datos, inteligencia artificial.

También es necesario indicar que hay conocimientos totalmente nuevos en los que ha sido necesario un estudio previo, incrementando así tanto la dificultad del proyecto como el interés personal y la curiosidad que ha suscitado realizarlo. De estos conocimientos totalmente nuevos que se han desarrollado, son los conocimientos sobre el concepto de open data, los procesos ETL y ELT y algunos de los modelos de machine learning utilizados. También nunca se había utilizado el software de visualización PowerBi.

5.2. Posibles Mejoras y Escalabilidad.

Existen multitud de mejoras que se pueden aplicar al proyecto. A continuación, se exponen una serie de mejoras que se podrían implementar en el proyecto:

- Seleccionar más cantidad de fuentes de datos abiertas. Si se seleccionan más fuentes de datos abiertas de diferentes archivos y estructuras, llegaría un punto donde se abarcaría la mayoría, llegando incluso a poder automatizar los procesos ETL, almacenando la información resultante lista para el análisis de datos. Incluso si se consiguiera realizar una automatización sobre los procesos ETL, también sería posible realizar una cierta automatización sobre el proceso de análisis de datos.
- Ampliación del análisis de datos. Incrementando el número de modelos posibles a realizar sobre las diversas fuentes de datos incrementaría la funcionalidad y valor del proyecto. Haciendo más personalizable y exacto el análisis al cliente. También al tener una mayor cantidad de modelos sobre una fuente de datos, se obtienen mayor variedad de resultados y conclusiones.
- Mayor conocimiento sobre librerías de Python. Tanto si se incrementara el número de fuentes de datos o de modelos, aumentaría el conocimiento sobre la multitud de librerías que ofrece Python. Por lo que existiría mayor conocimiento, para la comunidad, sobre qué librerías aplicar en cada momento.
- Integración con bases de datos. Se permitiría almacenar los resultados de los análisis y visualizaciones en una base de datos integrada, facilitando la gestión y acceso a la información.

En conclusión, es posible realizar multitud de mejoras al proyecto que supondrían un salto de calidad y rendimiento. Realizar estas mejoras supondrían una mayor inversión tanto en tiempo de personal como en recurso, pero se alcanzaría un nivel profesional en la plataforma.

Si se quiere proporcionar un rendimiento adecuado al proyecto ante un aumento en la carga de trabajo o en el tamaño de los conjuntos de datos es vital ir escalando el proyecto.

Se ha desarrollado el proyecto con una arquitectura independiente, dividiendo las funciones en componentes independientes, que permite aumentar y actualizar partes de la arquitectura sin involucrar otras áreas.

5.3. Presupuesto del Proyecto.

El presupuesto requerido para desarrollar el proyecto se detalla en la siguiente tabla:

Recurso	Tipo	Cantidad	Precio	Total
Desarrollador	Personal	400 horas	13 €/h	5.200 €
Ordenador	Equipamiento	1 equipo	799 €	799 €
Softwares	Equipamiento	5 softwares	0€	0€
Fuentes de datos	Equipamiento	8 fuentes de datos	0€	0€

Presupuesto Total	5.999€
-------------------	--------

Tabla 5. Presupuesto del proyecto.

Como se observar en la *tabla 5*, los únicos gastos considerables son la mano de obra (el desarrollador) y el ordenador necesario para llevarlo a cabo. Como se trata de un proyecto académico hay softwares que se han empleado que no tienen casto alguno, por ello, el precio total en este apartado es cero.

Las fuentes de datos al ser abiertas no tienen coste alguno, ya que son suministradas por organismos de actividad pública.

Capítulo 6

6. Bibliografía.

- [1] European University Alliance for Global Health. “*Eugloh*”. [en línea]. Available: <https://www.eugloh.eu/>
- [2] Enhancing Skills for Smart City Tech. “*Smacite*”. [en línea]. Available: <https://smacite.eu/>
- [3] The official portal for European Data. “*Open Data Maturity*”. [en línea]. Available: <https://data.europa.eu/en/publications/open-data-maturity/2022>
- [4] Powerful Data Structures for Data Analysis. “*Pandas*”. [en línea]. Available: <https://pandas.pydata.org/>
- [5] Beautiful Soup Documentation 4.12.2. “*Beautiful Soup*”. [en línea]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [6] The Fundamental package for Scientific Computing with Python. “*NumPy*”. [en línea]. Available: <https://numpy.org/>
- [7] Popularity of Programming Languages Index. “*PYPL*”. [en línea]. Available: <https://pypl.github.io/PYPL.html>
- [8] Python Programming Language. “*Python*”. [en línea]. Available: <https://www.python.org/>
- [9] Programming environment Visual Studio Code. [en línea]. “*Visual Studio Code*”. Available: <https://code.visualstudio.com/>
- [10] Microsoft Software Excel. “*Excel*”. [en línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/excel>
- [11] Microsoft Software PowerBi. “*PowerBi*”. [en línea]. Available: <https://powerbi.microsoft.com/es-es>
- [12] Microsoft Software Word. “*Word*”. [en línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/word?>
- [13] Visualization Software LuciChart. “*LuciChart*” [en línea]. Available: https://lucid.app/documents#/documents?folder_id=recent
- [14] “*GitHub*” [en línea]. Available: <https://github.com/>
- [15] Repositorio del Propio TFG Almacenado en GitHub. “*MarcosNavarro00.TFG*”. [en línea]. Available: <https://github.com/MarcosNavarro00/TFG>
- [16] Ministerio para la Transición Ecológica y el Reto Demográfico. “*Incendios forestales en España. Decenio 2006-2015*” [en línea]. Available: https://www.miteco.gob.es/content/dam/miteco/es/biodiversidad/temas/incendios-forestales/incendios-decenio-2006-2015_tcm30-521617.pdf

- [17] Ministerio de Inclusión, Seguridad Social y Migraciones (España). “*Estadísticas de concesiones de autorizaciones de residencia y trabajo*”. [en línea]. Available: <https://inclusion.seg-social.es/web/migraciones/homees/Estadisticas/operaciones/concesiones/index.html>
- [18] Ministerio de Sanidad (España). “*Situación actual del COVID-19*”. [en línea]. Available: <https://www.sanidad.gob.es/areas/alertasEmergenciasSanitarias/alertasActuales/nCov/situacionActual/index.html>
- [19] Ayuntamiento de Madrid. “*Catálogo de parques municipales de Madrid*”. [en línea]. Available: https://www.madrid.es/UnidadesDescentralizadas/ZonasVerdes/TodoSobre/ContenidosTemporales/Cat%C3%A1logoParques/CATALOGO%20DE%20PARQUES%20MUNICIPALES%20MADRID_BAJA%20RESOLUCI%C3%93N%2003.08.2021.pdf
- [20] Datos del Gobierno de España. “*Accidentes de tráfico de la ciudad de Madrid*”. [en línea]. Available: <https://datos.gob.es/en/catalogo/101280796-accidentes-de-trafico-de-la-ciudad-de-madrid1>
- [21] INE. “*Instituto Nacional de Estadística (España)*”. [en línea]. Available: <https://www.ine.es/>
- [22] INE (España) Tabla de Datos Estadísticos. “*Población en España por Provincia y Sexo*”. [en línea]. Available: <https://www.ine.es/jaxiT3/Tabla.htm?t=2852>
- [23] Oficina Europea de Estadística “*Eurostat*”. [en línea]. Available: <https://ec.europa.eu/eurostat>
- [24] Eurostat. “*Demographic balance and crude rates at national level*”. [en línea]. Available: https://ec.europa.eu/eurostat/databrowser/view/DEMO_GIND_custom_2733962/settings_1/tabc?lang=en&bookmarkId=7084ed24-6b91-4cf3-b90d-d47565593505
- [25] Ayuntamiento de Madrid Zonas Verdes. “*Valor Bosque Urbano de Madrid*”. [en línea]. Available: <https://www.madrid.es/UnidadesDescentralizadas/ZonasVerdes/TodoSobre/ValorBosqueUrban oMadrid/Valor%20Bosque%20Urbano%20de%20Madrid.pdf>
- [26] PyPI Fitz Documentation 1.18.9. “*Fitz Documentation*”. [en línea]. Available: <https://pypi.org/project/fitz/>
- [27] PyPI XlsxWriter Documentation 3.1.2. “*XlsxWriter Documentation*”. [en línea]. Available: <https://pypi.org/project/XlsxWriter/>
- [28] Python 3.9.7 Sotfware Foundation Datetime. “*Datetime*”. [en línea]. Available: <https://docs.python.org/3/library/datetime.html>
- [29] PyPI Camelot Documentation 0.11.0. “*Camelot Documentation*”. [en línea]. Available: <https://pypi.org/project/camelot-py/>
- [30] Características de Python. “*¿Python el lenguaje del futuro?*”. [en línea]. Available: <https://www.paradigmadigital.com/dev/es-python-el-lenguaje-del-futuro/>
- [31] TYS Magazine. “*Las 10 ciudades más inteligentes de Europa*”. [en línea]. Available: <https://tysmagazine.com/las-10-ciudades-mas-inteligentes-de-europa/>

- [32] Documentos Académicos de la Universidad de Alcalá. “*e_Bu@h*”. [en línea]. Available: <https://ebuah.uah.es/dspace/handle/10017/17681>
- [33] Wikipedia. Cross Industry Standard Process for Data Mining. “*CRISP-DM*”. [en línea]. Available: https://es.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining
- [34] Matplotlib. matplotlib.pyplot 3.5.3. “*Matplotlib documentation*”. [en línea]. Available: https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html
- [35] Maldita.es. Capacidad hospitalaria durante la pandemia. “*¿han cerrado hospitales o camas UCI?*” [en línea]. Available: <https://maldita.es/malditodato/20211018/capacidad-hospital-pandemia-camas-uci/>
- [36] Universidad de Santiago de Compostela. “*Regresión simple*”. [en línea]. Available: http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP-DPTO/MATERIALES/Mat_50140116_Regr_%20simple_2011_12.pdf
- [37] SciPy. “*SciPy Documentation*”. [en línea]. Available: <https://scipy.org/>
- [38] Scikit-Learn. Supervised Learning. “*Scikit-Learn 0.24.2 Documentation*”. [en línea]. Available: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- [39] Diputación Provincial de Ávila. “*Población en la provincia*”. [en línea]. Available: <https://www.diputacionavila.es/la-provincia/nuestros-pueblos/poblacion/#:~:text=La%20poblaci%C3%B3n%20en%20la%20provincia,en%202022%20son%20161.771%20habitantes>
- [40] Scikit-Learn. Sklearn.cluster.KMeans. “*scikit-learn KMeans Documentation*”. [en línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
- [41] Scikit-Learn. Sklearn.cluster.MeanShift. “*scikit-learn Mean Shift. Documentation*”. [en línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html>
- [42] Aitor Alberto Báez. “*Modelo ARIMA (p,d,q)*”. [en línea]. Available: <https://www.aitorbertobaez.com/modelo-arima-pdq>
- [43] Statsmodels. “*statsmodels.tsa.arima.model.ARIMA 0.12.2 Documentation*” [en línea]. Available: <https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html>
- [44] Alkaline-ml. “*pmdarima 2.0.0 Documentation*”. [en línea]. Available: <https://alkaline-ml.com/pmdarima/>
- [45] Machine Learning Plus. “*ARIMA Model - Time Series Forecasting using Python*”. [en línea]. Available: <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>
- <https://es.linux-console.net/?p=5324#gsc.tab=0>
- [47] TensorFlow API. “*tf.keras.layers.LSTM v2.7.0 Documentation*”. [en línea]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM

- [48] Scikit-learn. sklearn.preprocessing.MinMaxScaler. “scikit-learn 0.24.2 documentation”. [en línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
- [49] Helena Rodríguez Rivero. Técnicas de aprendizaje automático aplicadas a la predicción de resultados de carreras de caballos. Trabajo de Fin de Grado. Universidad de Alcalá 2021. [en línea]. Available: <https://ebuah.uah.es/dspace/handle/10017/52491>
- [50] David Moreno López. Análisis de la evolución de la opinión pública ante las vacunas de la covid-19 mediante análisis de sentimiento en Twitter. Trabajo de Fin de Grado. Universidad de Alcalá 2021. [en línea]. Available: <https://ebuah.uah.es/dspace/handle/10017/49492>
- [51] Enrique García Miravalles. Herramienta para aplicar el proceso ETL a datos de la AEMET y prueba de posibles aplicaciones sobre los datos. Trabajo de Fin de Grado. Universidad de Valladolid 2020. [en línea]. Available: <https://uvadoc.uva.es/handle/10324/44178>
- [52] Luis Miguel Garay Teja. Análisis de datos aplicado a incidentes acuáticos utilizando técnicas de Machine Learning. Trabajo de Fin de Grado. Universidad de Cantabria 2020. [en línea]. Available: <https://repositorio.unican.es/xmlui/bitstream/handle/10902/20981/Garay%20Teja%20Luis%20Miguel.pdf?sequence=1>

Anexo I – Manual de Instalación.

Requisitos para la instalación.

Todo el proyecto se ha desarrollado en un entorno Windows. Para poder realizar la instalación de la totalidad del proyecto es necesario clonar el directorio con todos los archivos necesarios desde GitHub. A través de la siguiente sentencia se obtendrá una copia del repositorio en el ordenador.

- git clone <https://github.com/MarcosNavarro00/TFG>

Instalación de las fuentes de datos abiertas.

Para acceder a las fuentes de datos abiertas originales que se han empleado en el proyecto hay dos formas. La primera, a través de los enlaces que están reflejados en la memoria y en la bibliografía, donde se te llevan a la página oficial de cada fuente de datos de donde se han obtenido.

La segunda forma es clonando el repertorio de GitHub, como se indicaba en el punto anterior, donde en la carpeta “Fuentes de Datos”, se están recogidas todas las fuentes.

Instalación de las librerías para los procesos ETL y de análisis.

El primer paso para poder ejecutar el proyecto es la instalación del lenguaje de programación Python, cualquier versión a partir de la 3.0 es compatible con el proyecto. Se instala Python a través de su página web, la cual es la siguiente:

<https://www.python.org/downloads/>

Una vez instalado Python es necesario instalar las librerías que se requieren para su correcto funcionamiento. Para poder instalar las librerías de Python es necesario ejecutar el archivo “pip3.exe”, que se encuentra en la carpeta “Scripts” una vez que se haya instalado Python.

Se instalan las librerías usando los siguientes comandos:

- Pandas.

.\pip3.exe install pandas

- Fitz.

.\pip3.exe install fitz

- XlsxWriter.

.\pip3.exe install XlsxWriter

- NumPy.

.\pip3.exe install numpy

- Datetime.

.\pip3.exe install DateTime

- Camelot.

.\pip3.exe install camelot-py

- StatsModels.
.\pip3.exe install statsmodels
- Scikit-Learn.
.\pip3.exe install scikit-learn
- SciPy.
.\pip3.exe install scipy
- MatPlotLib.
.\pip3.exe install matplotlib
- TensorFlow.
.\pip3.exe install tensorflow
- Pdarima.
.\pip3.exe install pdarima

Una vez instaladas todas las librerías, simplemente con ejecutar los programas de Python de repositorio de GitHub, se completaría tanto el proceso de ETL como el de análisis

Instalación de la herramienta de visualización.

Para la visualización se utiliza PowerBi. Los archivos generados para la visualización de las soluciones obtenidas tienen formato “*pbix*”, por lo que es necesario instalar PowerBi desktop para visualizarlas. Se descarga el software a través de su página oficial, la cual es la siguiente:

<https://powerbi.microsoft.com/es-es/desktop/>

Anexo II – Manual de Usuario.

Una vez el contenido del repositorio este presente en nuestro ordenador ya se puede desplegar el proyecto. Para ello es vital que todas las fuentes de datos estén descargadas correctamente.

Ejecutando cada uno de los programas desarrollados para el proceso ETL se obtienen los archivos Excel con cada una de las soluciones. Ya en el propio repositorio están almacenadas las soluciones de cada uno de los programas con sus respectivos nombres. Los archivos Excel generados se les puede dar cualquier propósito, no solo el que se le propone en el proyecto.

Para poder ejecutar cada uno de los programas de análisis, es necesario haber obtenido cada uno de los archivos Excel del proceso ETL. Al ejecutar cada uno de los programas se obtiene la solución del modelo indicado, obtendremos una primera edición de la visualización de los resultados que viene proporcionada por el propio Python. Las soluciones obtenidas de cada uno de los modelos se almacenar en archivos Excel, lo cuales también están presentes en el propio repositorio.

Las visualizaciones finales de cada modelo se visualizan usando PowerBi, las visualizaciones se despliegan usando los archivos *.pbix* que se encuentran en el propio repositorio. Para poder abrir los archivos es necesario hacerlo con el software PowerBi desktop.

Anexo III – Dashboard PowerBi.

En los ficheros del repositorio podemos encontrar la carpeta “PowerBi”, donde se almacenan cada una de las visualizaciones de las dashboard de los modelos de análisis.

Cada una de las dashboard se corresponde con una fuente de datos abierta. En cada una de ellas podemos encontrar tanto visualizaciones de la propia fuente de datos como las soluciones a los modelos de análisis que se han aplicado.

Capacidad Asistencial durante la Covid-19.

En la primera página del dashboard de la fuente de datos “Capacidad Asistencial durante la Covid-19” se muestra los resultados obtenidos del análisis descriptivo. A través de un gráfico de áreas se representa la comparación entre las camas ocupadas por enfermos covid-19 y el total de camas. En el otro grafico de áreas se visualiza la evolución de los ingresos desde el 1 de agosto del 2020 hasta el 1 de abril del 2021. Por último, se tiene una tabla con la totalidad de la información usada para representar las gráficas que se corresponde con la solución tras el proceso ETL.

Tambien es posible filtrar la información por la fecha indica, por las camas ocupadas por enfermos Covid-19 y por los ingresos, mostrando tantos en las dos gráficas como en la tabla, los resultados tras el filtrado. En la parte superior derecha se incluye un botón para pasar a la siguiente página, donde se encuentra almacenado el resultado del clustering.



Figura 41. Dashboard del resultado del análisis descriptivo en la fuente de datos “Capacidad Asistencial durante la Covid-19”

En la segunda página del dashboard se muestra en un gráfico de líneas el resultado del proceso de análisis de clusterización, ofreciendo la posibilidad de filtrar el grafico en función de los clústeres seleccionados.

Tambien se incluyen dos botones, uno en la parte superior izquierda que vuelve a la primera pagina donde se encuentra el análisis descriptivo y otro en la parte superior derecha que avanza a la siguiente pagina donde se encuentra el resultado del análisis de la serie temporal ARIMA.

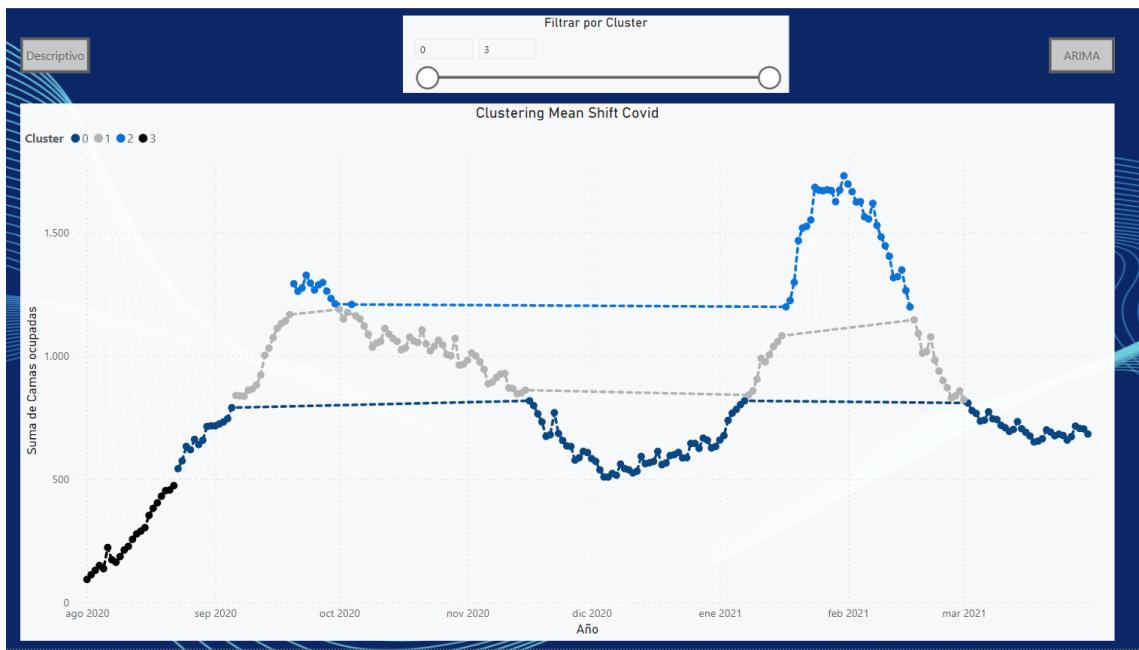


Figura 42. Dashboard del resultado de la clasterización en la fuente de datos “Capacidad Asistencial durante la Covid-19”

En la tercera página del dashboard a través de un gráfico de líneas se muestra el resultado del análisis de la serie temporal ARIMA, donde se tiene la posibilidad de filtrar el grafico por fechas y por el número de camas ocupadas por enfermos Covid-19. Tambien se incluyen un botón en la parte superior izquierda, que vuelve a la página donde se visualiza el clustering.



Figura 43. Dashboard del resultado de la serie temporal ARIMA en la fuente de datos “Capacidad Asistencial durante la Covid-19”

Incendios producidos en España entre el 2006 y el 2015.

En la primera página del dashboard de la fuente de datos “Incendios producidos en España entre el 2006 y el 2015.” se muestra los resultados obtenidos del análisis descriptivo.

A través de un gráfico de columnas apiladas se muestra la comparación entre los incendios por año y los incendios por año que se consideran GIF, por lo que superan las 500 hectáreas.

Utilizando un gráfico de líneas se muestra una comparación de la superficie total afectada por incendios por año en hectáreas y la superficie afectada por grandes incendios forestales por año en hectáreas. También con una tabla se muestra la totalidad de la fuente de datos utilizada.

Tambien existe la posibilidad de filtrar a través de los años y en función de número de siniestros mostrando tantos en las dos gráficas como en la tabla, los resultados tras el filtrado. En la parte superior derecha se añade un botón para avanzar a la siguiente página, donde se encuentra visualizado el resultado de la red neuronal recurrente LSTM.

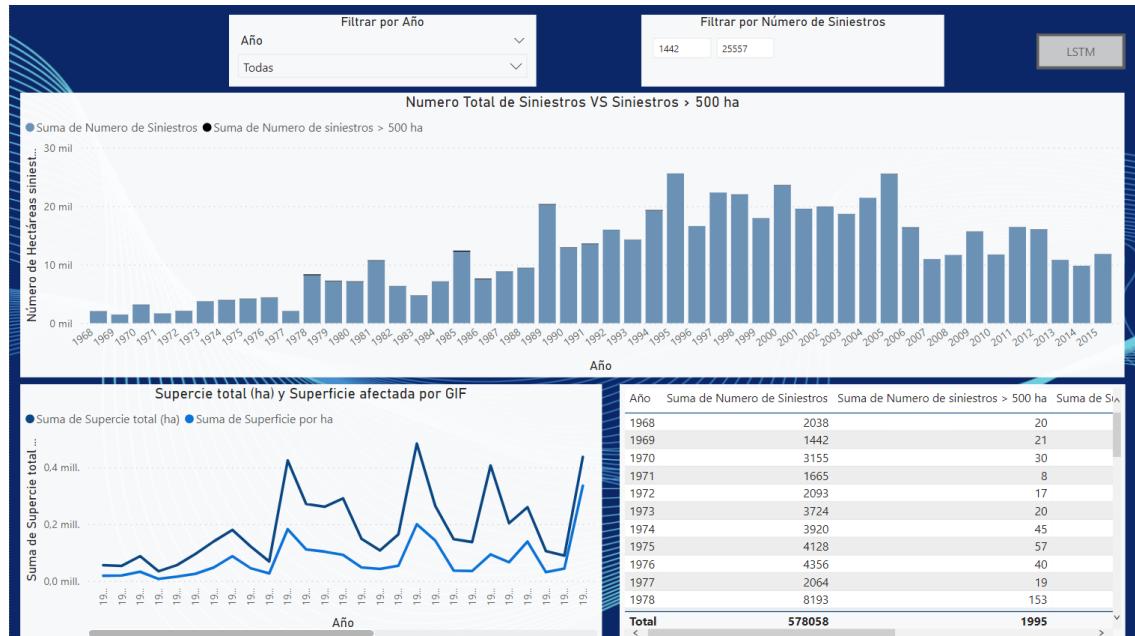


Figura 44. Dashboard del resultado del análisis descriptivo en la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”

En la segunda página del dashboard se muestran dos gráficos de columnas apiladas que visualizan el resultado de la red neuronal recurrente LSTM. Existe la posibilidad de filtrar las gráficas por años. Tambien en la parte superior izquierda se añade un botón que vuelve a la página anterior donde se visualiza en análisis descriptivo.



Figura 45. Dashboard del resultado de la RNN LSTM en la fuente de datos “Incendios producidos en España entre el 2006 y el 2015”

Catálogo del Bosque Urbano de la Ciudad de Madrid.

En la única página del dashboard de la fuente de datos “Catálogo del Bosque Urbano de la Ciudad de Madrid” se muestra los resultados obtenidos del análisis descriptivo. A través de un gráfico de columnas agrupadas y de líneas se muestra la cantidad de agua interceptada por especie y la producción de oxígeno también por especie. Tambien a través de una tabla se muestra la totalidad de la fuente de datos utilizada.

Es posible filtrar la información por una especie determinada, por el agua interceptada por especie y por la producción de oxígeno por especie. Mostrando en la gráfica como en la tabla, los resultados tras el filtrado.

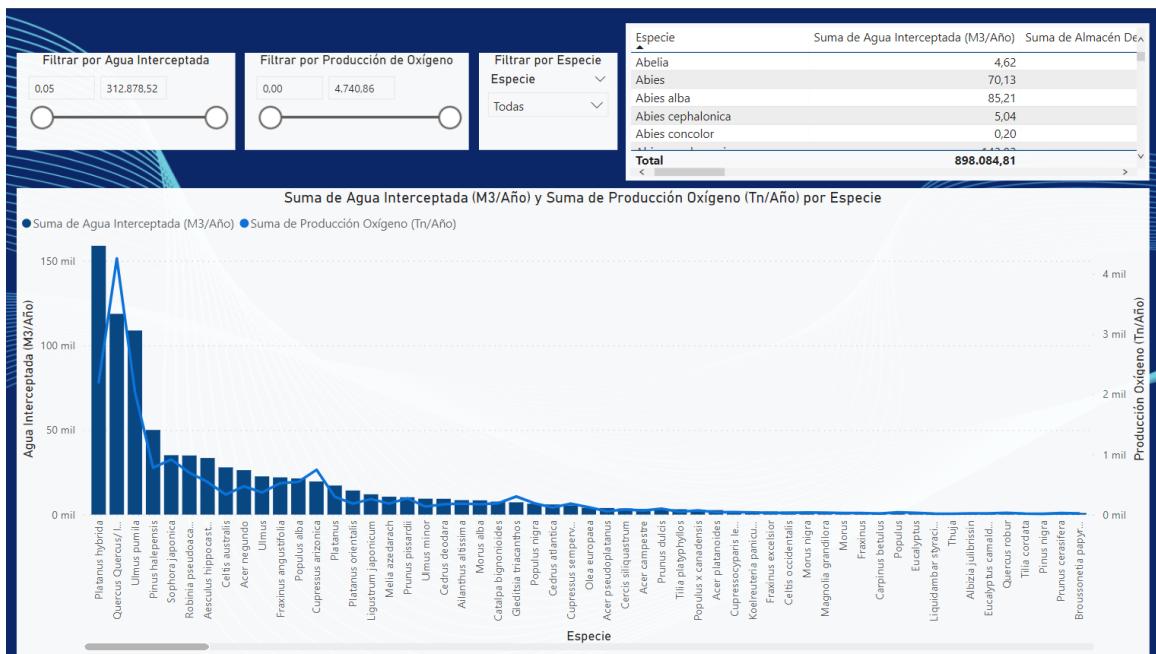


Figura 46. Dashboard del resultado del análisis descriptivo en la fuente de datos “Catálogo del Bosque Urbano de la Ciudad de Madrid”

Población por Provincias de España 1996-2021.

En la única página del dashboard de la fuente de datos “Población por Provincias de España 1996-2021” se muestra los resultados obtenidos del análisis descriptivo y los resultados de la regresión lineal. Para los dos análisis los resultados se visualizan a partir de gráficos de líneas. En el primer grafico se muestra el histórico de la población de Ávila por año mientras que en el otro grafico se muestra un histórico de la población de Ávila por año más la predicción realizada y los intervalos de confianza.

Existe la posibilidad la posibilidad de filtrar en función de la fecha y en función de la población, mostrando en los dos gráficos el resultado de la filtración

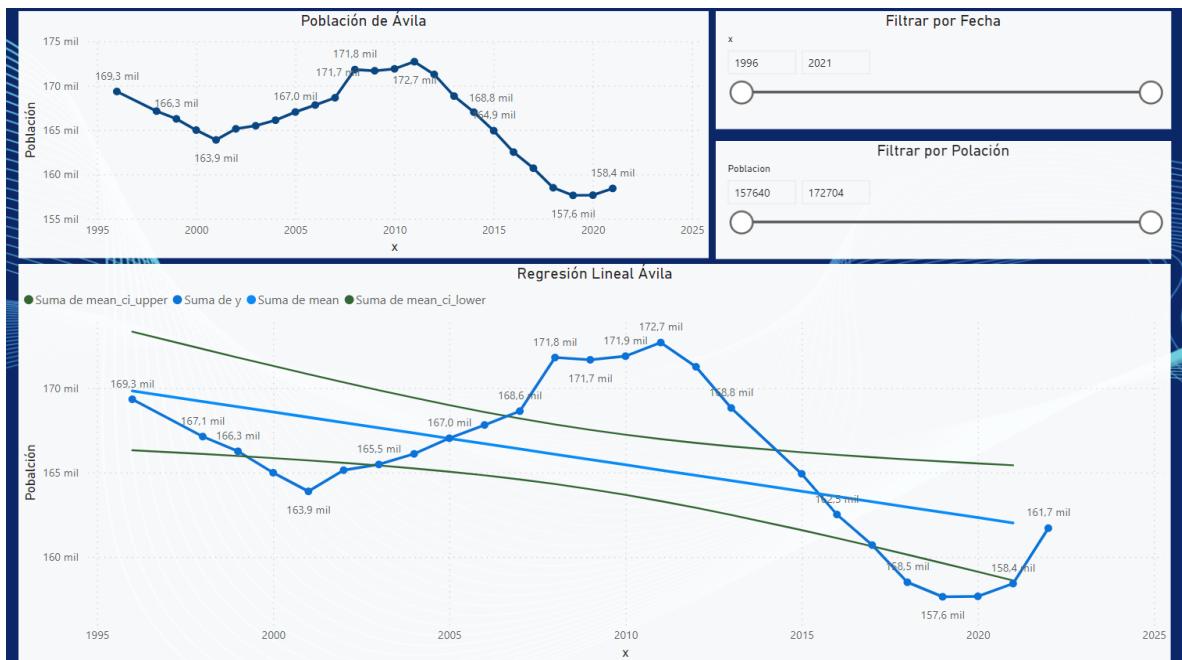


Figura 47. Dashboard del resultado del análisis descriptivo y de regresión lineal en la fuente de datos “Población por Provincias de España 1996-2021”.

Anexo IV – Contenido del Repositorio.

El repositorio de GitHub donde se almacena el proyecto tiene la siguiente estructura:

- Análisis
 - Bosque
 - clusteringK-Means-bosque.py
 - clusteringMS -bosque.py
 - descriptivo-bosque.py
 - Covid
 - ARIMA-Covid.py
 - clusteringKM-covid.py
 - clusteringMS-covid2.py
 - descriptivo-covid.py
 - Incendios
 - descriptiva-incendios.py
 - LSTM1-incendios.py
 - LSTM2- Incendios.py
 - Población
 - descriptiva-PobalcionEspaña.py
 - regresionLineal-Poblacion.py
 - Resultados
 - Bosque
 - descriptiva-bosque.xlsx
 - resultados_clustering.xlsx
 - resultados_clustering2.xlsx
 - Covid
 - ARIMA.xlsx
 - clusteringMS-covid.xlsx
 - descriptiva-covid.xlsx
 - Incendios
 - descriptiva-incendios.xlsx
 - predicciones_Siniestros.xlsx
 - predicciones_Superficie%.xlsx
 - Población
 - descriptiva-poblacion.xlsx
 - regresionLineal-Poblacion.xlsx
- ETL
 - Data
 - Covid.xlsx
 - dataBosque.xlsx
 - dataParque.xlsx
 - incendios1996-2015.xlsx
 - nacionalidad-Hombres.xlsx
 - poblacion-Espana.xlsx
 - Etl_bosque.py
 - Etl_Covid.py
 - Etl_incendios.py
 - Etl_nacionalidad.py
 - Etl_parques.py
 - Etl_posblacionE2.py
- Fuentes de datos
 - Eugloh
 - ConcesionesNacionalidad.xlsx
 - Datos_Capacidad_Asistencial_Historico_03012023.csv
 - incendios-decenio-2006-2015_tcm30-521617.pdf
 - Smacite
 - 2022_Accidentalidad (1).xlsx

- CATALOGO DE PARQUES MUNICIPALES MADRID_BAJA
RESOLUCIÓN 03.08.2021.pdf
- poblacionEspaña.xlsx
- poblacionEuropa.xlsx
- Valor Bosque Urbano de Madrid.pdf
- Memoria.
 - memoria.pdf
 - memoria.docs
- PowerBi
 - Bosque
 - Bosque.pbix
 - Covid
 - Covid.pbix
 - Incendios
 - Incendios.pbix
 - Población
 - Población.pbix
- Presentacion.pptx
- Anteproyecto.pdf

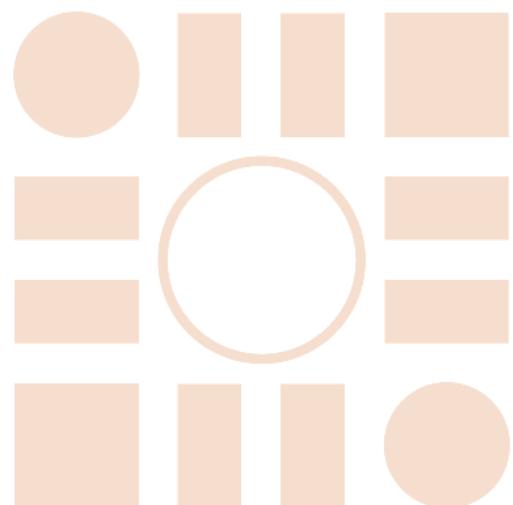
La estructura del repositorio se divide en la selección de fuentes de datos, donde se pueden encontrar todas las fuentes de datos utilizadas divididas por temática (Euglooh o Smacite), en los procesos ETL, donde en una primera instancia están todos los programas desarrollados en Python para llevar a cabo el proceso y después en la carpeta “Data” se encuentran alojados todas las soluciones de este proceso en archivos Excel.

A continuación, en la carpeta “Análisis”, se encuentran todos los archivos Python realizados para los análisis divididos en función de la fuente de datos. después en la carpeta “Resultados” se encuentran almacenados las soluciones en archivos Excel de cada proceso de análisis en función otra vez de la fuente de datos.

Después en la carpeta “PowerBi”, se encuentra todas las visualizaciones realizadas en PowerBi divididas en función de la fuente de datos. Son archivos de tipo “pbix”.

Por último, se almacena la memoria tanto en formato PDF como en Word, la presentación utilizada en la defensa con formato PowerPoint y el anteproyecto realizado previamente en formato PDF.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR

