ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
(ICAI)

Máster en Big Data: Tecnología y Analítica Avanzada

ANÁLISIS GEOESPACIAL DE NECESIDADES AL CIUDADANO

Autor
Francisco Javier Gisbert Gil

Dirigido por
Alejandro Llorente Pinto

Madrid
Mayo 2025

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D.Francisco Javier Gisbert Gil **DECLARA** ser el titular de los derechos de propiedad intelectual de la obra:
ANÁLISIS GEOESPACIAL DE NECESIDADES AL CIUDADANO, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, los derechos de digitalización, de archivo, de reproducción, de distribución y de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

(a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.

(c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.

(d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.

(e) Asignar por defecto a estos trabajos una licencia Creative Commons.

(f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

(a) Que la Universidad identifique claramente su nombre como autor de la misma

(b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

(c) Solicitar la retirada de la obra del repositorio por causa justificada.

(d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

(a) El autor se compromete a:

(b) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

(c) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

(d) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

(e) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 15 de Mayo de 2025

ACEPTA

Fdo.: .....

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
(ICAI)

Máster en Big Data: Tecnología y Analítica Avanzada

DETECCIÓN DE NECESIDADES EN SERVICIOS ESENCIALES AL CIUDADANO

Autor
Francisco Javier Gisbert Gil

Dirigido por
Alejandro Llorente Pinto

Madrid
Mayo 2025

Resumen

El proyecto "ANÁLISIS GEOESPACIAL DE NECESIDADES AL CIUDADANO" tiene como objetivo determinar la escasez de servicios esenciales, como farmacias, centros educativos o puntos de venta de la tarjeta de transporte público, en la Comunidad de Madrid.

Para lograr esto, se utiliza el catálogo de datos abiertos de la Comunidad de Madrid para obtener la ubicación geográfica de los servicios, así como datos sociodemográficos de cada las secciónes censales de la Comunidad. Se realiza una transformación de las coordenadas espaciales y se realiza un conteo de los servicios por cada zona censal y sus alrededores.

Se entrena un modelo explicativo basado en la tendencia general de los datos y las variables sociodemográficas para predecir el número de servicios que debería haber en cada zona. Se calcula un score para cuantificar la escasez de servicios esenciales en cada sección censal.

Se desarrolla un Dashboard en Power BI para visualizar los resultados, mostrando las secciones censales de la Comunidad de Madrid y su score para los diferentes servicios. También se muestra información sobre variables sociodemográficas relevantes.

Se utilizan fuentes de datos como el catálogo de datos abiertos de la Comunidad de Madrid y el Instituto Nacional de Estadística para obtener la ubicación de los servicios y datos sociodemográficos de las secciones censales.

El proyecto revela las zonas censales que tienen una escasez de farmacias, centros educativos y puntos de venta de la tarjeta de transporte público. Además, se identifican diferencias en variables sociodemográficas como población, renta per cápita y porcentaje de extranjeros entre las zonas con escasez y las que tienen una mayor disponibilidad de servicios.

Además, es importante destacar que todo el proceso se lleva a cabo en un pipeline robusto y escalable, lo que permite su aplicación a nuevos conjuntos de datos, zonas geográficas o servicios que se deseen estudiar en el futuro. El modelo explicativo puede ser entrenado con nuevos datos y variables relevantes para adaptarse a diferentes contextos, brindando una mayor flexibilidad en la detección de escasez de servicios esenciales en otras áreas geográficas.

Abstract

The project "Detection of essential citizen services needs" aims to determine the scarcity of essential services, such as pharmacies, educational centers, or public transportation card outlets, in the Community of Madrid.

To achieve this, the open data catalog of the Community of Madrid is utilized to obtain the geolocation of services, as well as sociodemographic data for each census tract in the community. Spatial coordinate transformation is performed, and a count of services is conducted for each census tract and its surroundings.

An explanatory model is trained based on the general trend of the data and sociodemographic variables to predict the number of services that should be present in each zone. A score is calculated to quantify the scarcity of essential services in each census tract.

A Power BI dashboard is developed to visualize the results, displaying the census tracts of the Community of Madrid and their score for different services. Relevant sociodemographic variables are also presented.

Data sources such as the open data catalog of the Community of Madrid and the National Institute of Statistics are utilized to obtain the location of services and sociodemographic data for the census tracts.

The project reveals the census tracts that have a shortage of pharmacies, educational centers, and public transportation card outlets. Furthermore, differences in sociodemographic variables such as population, per capita income, and percentage of foreigners are identified between the areas with scarcity and those with a higher availability of services.

Moreover, it is important to highlight that the entire process is carried out in a robust and scalable pipeline, enabling its application to new datasets, geographical areas, or services that may be studied in the future. The explanatory model can be trained with new data and relevant variables to adapt to different contexts, providing greater flexibility in detecting scarcity of essential services in other geographical areas.

Contents

List of Figures

Chapter 1

Introducción

El objetivo de este trabajo se centra en detectar las necesidades en servicios esenciales de los ciudadanos de la Comunidad de Madrid.

Un servicio esencial se refiere a aquellos servicios públicos o privados que son considerados fundamentales para el bienestar y el funcionamiento adecuado de una sociedad. Estos servicios son necesarios para satisfacer las necesidades básicas de la población y garantizar su calidad de vida. Los servicios esenciales suelen incluir áreas como la salud, la educación, el transporte, la seguridad, la vivienda, el suministro de agua potable y energía, entre otros.

Por tanto, lo que queremos es construir un proceso completo, con el que sea posible detectar aquellas zonas que padecen la ausencia de alguno de estos servicios.

Este análisis es bastante complejo, ya que se es necesario un tratamiento avanzado de diferentes datos geoespaciales y socioeconómicos. Por ello elaboraremos un Pipeline que automatice desde la extracción de la información y la obtención de resultados.

El resultado final se presentará al cliente como una visualización en Power Bi donde a través de un mapa interactivo se podrán analizar aquellas zonas que presenten la ausencia del servicio requerido. Además de un análisis descriptivo de dichas regiones para comprender desde un punto de vista social, económico y geográfico las causas del problema y ayudar en la toma de decisiones.

Como esta información es muy variable, ya que nuevos establecimientos pueden abrir o pueden producirse cambios sociales, también se elaborará un Pipeline generalizable para que en un futuro puedan repetirse estos análisis o para que se puedan introducir nuevos servicios esenciales según se requiera.

Chapter 2

Estado del arte

En la era de la digitalización y la creciente urbanización, las ciudades inteligentes (*smartcities*) se han convertido en un tema de gran relevancia en el ámbito de la planificación urbana y la mejora de la calidad de vida de los ciudadanos. Estas ciudades inteligentes buscan utilizar la tecnología y la ciencia de datos para optimizar la eficiencia, la sostenibilidad y la calidad de los servicios públicos, así como para mejorar la toma de decisiones en diversos ámbitos.

El estado del arte en el campo de detección de necesidades en servicios esenciales para los ciudadanos se ha enfocado en el uso del aprendizaje automático y los datos geoespaciales. Los resultados de la búsqueda revelan que existen recursos como Esri State Local Connect, los cuales brindan herramientas y conocimientos a los funcionarios gubernamentales estatales y locales interesados en emplear Sistemas de Información Geográfica (SIG) y aprendizaje automático para mejorar los servicios ciudadanos [5]. Asimismo, se ha demostrado que el aprendizaje automático tiene aplicaciones relevantes en el análisis espacial en áreas urbanas, ofreciendo valiosas oportunidades para mejorar la calidad de los servicios prestados a los ciudadanos [6]. En ese sentido, el uso de técnicas de aprendizaje profundo y de inteligencia artificial ha sido destacado como una forma de mejorar los servicios ciudadanos al dirigir respuestas específicas y proporcionar una experiencia más personalizada [7][8]. Por otra parte, la tecnología geoespacial ha demostrado ser una herramienta efectiva para mejorar los servicios relacionados con la planificación urbana, la gestión del territorio y la reducción del riesgo de desastres, lo cual impacta positivamente en la calidad de vida de la población [9]. Estos avances en el campo del aprendizaje automático y los datos geoespaciales representan un enfoque prometedor para la detección de necesidades y la toma de decisiones más informadas en la prestación de servicios esenciales a los ciudadanos.

Chapter 3

Descripción del trabajo

3.1 Obtención de los datos

En este proyecto, utilizaremos datos proporcionados por el Instituto Nacional de Estadística (INE) [10] para obtener información sociodemográfica de la población de Madrid. Estos datos son altamente confiables y de fácil acceso, lo que los convierte en una excelente fuente de información. Además, el portal de datos abiertos de la Comunidad de Madrid [11] nos permitirá obtener de forma gratuita datos sobre farmacias, centros educativos y transporte público. También exploraremos el uso de datos privados, como Google Maps, para obtener ubicaciones de otros servicios relevantes. Al combinar estas fuentes de datos, obtendremos una visión completa de la disponibilidad de servicios esenciales en la región.

3.1.1 Datos abiertos

El primer enfoque que daremos en el proyecto será gracias a la explotación de datos abiertos. Los datos abiertos son fundamentales en el análisis de datos, ya que estos se encuentran al alcance todos, lo que hace que esta disciplina sea accesible y democratizada. Al ser accesibles, promueven la transparencia, la colaboración y la innovación, permitiendo que cualquier persona pueda utilizarlos para obtener información valiosa y tomar decisiones fundamentadas. Además impulsan el progreso y el avance al fomentar la inteligencia colectiva y la diversidad de perspectivas en el análisis de los problemas y desafíos actuales.

Portal de datos abiertos CCMM: Servicios

Contamos las siguientes fuentes de datos abiertos para nuestro análisis, incluimos el catálogo de datos abiertos de la Comunidad de Madrid, que proporciona información sobre la ubicación de farmacias, centros educativos y puntos de venta de

la TTP. También tenemos acceso a los portales de datos abiertos del CRTM [3], donde encontramos información sobre las estaciones de metro, cercanías y metro ligero. Además, utilizamos los datos sociodemográficos de las secciones censales proporcionados por el Instituto Nacional de Estadística.

- **Ubicación de oficinas de farmacia:** Catálogo de datos abiertos de la Comunidad de Madrid.
  https://datos.comunidad.madrid/catalogo/dataset/oficinas_farmacia
- **Ubicación de centros educativos:** Catálogo de datos abiertos de la Comunidad de Madrid.
  https://datos.comunidad.madrid/catalogo/dataset/centros_educativos
- **Ubicación de puntos de venta de la TTP:** Catálogo de datos abiertos de la Comunidad de Madrid - Puntos de venta.
  https://datos.comunidad.madrid/catalogo/dataset/puntos_de_venta
- **Portal de datos abiertos CRTM - Estaciones de metro:**
  https://datos.crtm.es/datasets/crtm::m4-estaciones/explore
- **Portal de datos abiertos CRTM - Estaciones de cercanías:**
  https://datos.crtm.es/datasets/crtm::m5-estaciones/explore
- **Portal de datos abiertos CRTM - Estaciones de metro ligero:**
  https://datos.crtm.es/datasets/crtm::m10-estaciones/explore
- **Datos sociodemográficos secciones censales:** Instituto Nacional de Estadística.
  https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C

En el contexto del proyecto, surge la cuestión sobre la conveniencia de utilizar Google Maps como herramienta principal para acceder a datos geoespaciales. Google Maps se ha convertido en una plataforma ampliamente reconocida por su extensa colección de información geográfica, que abarca desde ubicaciones comerciales hasta datos de transporte y mapas detallados. Esta riqueza de datos geoespaciales ofrece numerosas ventajas y oportunidades para mejorar la detección de escasez de servicios esenciales en la Comunidad de Madrid. Sin embargo, también plantea desafíos y consideraciones que deben ser evaluados cuidadosamente en el marco de este proyecto. En esta discusión introductoria, exploraremos tanto los beneficios potenciales como las posibles limitaciones de aprovechar Google Maps como fuente principal de datos geoespaciales, y analizaremos cómo estas consideraciones pueden impactar en la eficacia y alcance del proyecto.

3.1.2 API de Google Maps: Nearby Search

La API de Google Maps Nearby Search [12] permite buscar lugares cercanos a una ubicación específica. Con esta API, puedes obtener información sobre diferentes tipos de lugares, como restaurantes, hoteles, tiendas, etc., que se encuentran cerca de una coordenada geográfica determinada.

Para hacer una solicitud a la *API* de *Nearby Search*, se necesita una clave de *API* de Google Maps. A continuación, se muestra un resumen de cómo se realizan las peticiones:

Primero se construye la *URL* de la solicitud: La *URL* debe contener los parámetros necesarios, como la ubicación, el radio de búsqueda, el tipo de lugar y la clave de *API*. La estructura del *endpoint* es la siguiente:

```
https://maps.googleapis.com/maps/api/place/nearbysearch/output
```

Se envía la solicitud desde *Python* utilizando la librería *requests* mediante una petición *GET*.

La *API* de *Nearby Search* responderá con una lista de lugares cercanos que coinciden con los parámetros de búsqueda. La respuesta contiene información como el nombre del lugar, la dirección, las coordenadas geográficas y más.

A priori este parece el método más adecuado, pero el único inconveniente es su elevado coste económico, que es de unos 500 si queremos hacer una búsqueda en toda la superficie de la comunidad de Madrid. Y es por eso que proponemos la siguiente solución tecnológica.

3.1.3 Scraper Google Maps

Al explorar la opción de utilizar técnicas de *scraping* en la página web de Google Maps, se desprenden una serie de beneficios y desventajas que deben ser consideradas.

En cuanto a los beneficios, uno de los aspectos más destacados es que el *scraping* de Google Maps resulta más económico, ya que los datos extraídos de la web son gratuitos. A diferencia de utilizar la *API* de Google, donde se incurre en costos por el acceso a los datos geoespaciales. Además, al emplear técnicas de *scraping*, se tiene la flexibilidad de adaptar y personalizar la extracción de datos según las necesidades específicas del proyecto, en contraste con estar limitado a los *endpoints* y funcionalidades provistas por la *API* de Google.

Sin embargo, es importante tener en cuenta los inconvenientes asociados al *scraping* de Google Maps. En primer lugar, Google tiene políticas para proteger sus datos y desalienta activamente el *scraping* de su página web. Como resultado, nos enfrentamos a obstáculos y restricciones, y existe el riesgo de que la dirección *IP* sea bloqueada si se detecta un uso excesivo o sospechoso de la web. Para evitar

esto, hemos decidido hacer uso de *proxies*, que son intermediarios que enmascaran la dirección IP real y permiten el acceso a la web sin revelar su origen. Esto puede conseguirse si a la hora de hacer las peticiones a la web, seleccionamos cada vez una IP de manera aleatoria y realizamos la petición a través de ella.

Además, el *scraping* de Google Maps puede ser más lento en comparación con el uso de la API oficial. Esto se debe a que al realizar *scraping*, se deben cargar elementos visuales innecesarios como imágenes y estilos de diseño, lo que aumenta el tiempo necesario para obtener los datos deseados. Además, una vez obtenidos los datos, se requiere un proceso de limpieza y estructuración adicional para poder ser utilizados de manera efectiva. Otro factor a considerar es que la estructura de la página web de Google Maps puede cambiar con el tiempo. Esto significa que si se realiza *scraping* de forma regular, existe el riesgo de que el proceso se rompa si la estructura de la página es modificada. Esto requeriría ajustes y actualizaciones en el código utilizado para el *scraping*.

La solución que hemos elaborado se muestra en el apéndice A. El objetivo principal de este *scraper* es extraer información de Google Maps relacionada con un servicio específico en una determinada área y nivel de zoom.
La solución que hemos elaborado se muestra en el apéndice A. El objetivo principal de este *scraper* es extraer información de Google Maps relacionada con un servicio específico en una determinada área y nivel de zoom.

La implementación se realiza mediante una clase llamada $GoogleMapsServiceScraper$. Esta clase tiene varios métodos que realizan diferentes tareas durante el proceso de extracción de datos.

El primer método importante es test_proxies(). Este método se encarga de probar los *proxies* proporcionados para determinar cuáles de ellos están funcionando correctamente. Utiliza los *proxies* para enviar solicitudes a Google Maps y registra el tiempo de respuesta de cada uno. Luego, muestra los *proxies* que están funcionando y actualiza la lista de *proxies* utilizables.

El siguiente método clave es __get_places(). Este método se encarga de realizar la extracción real de datos de Google Maps. Utiliza las coordenadas proporcionadas y el nivel de zoom para construir la URL de búsqueda en Google Maps. Luego, utiliza *Selenium* para abrir esa URL en un navegador web simulado. Después de cargar la página, busca los elementos $HTML$ que contienen la información deseada tras hacer *scroll* y cargarlos por completo, como el nombre del lugar, la URL de Google Maps, la calificación, las reseñas, la dirección y el tipo de lugar. Estos datos se recopilan y se agregan a un $DataFrame$.

Otro método importante es __get_coordinates(). Este método se utiliza para obtener las coordenadas de los lugares extraídos. Utiliza la URL de Google Maps para cada lugar y extrae las coordenadas de latitud y longitud de esa URL. Estas coordenadas se agregan al $DataFrame$.

Una vez que se han recopilado todos los datos, se eliminan los duplicados utilizando el método __drop_duplicates().

Se ha desarrollado un *scraper* de Google Maps como alternativa, permitiendo extraer información específica de manera personalizada. En el apéndice A se encuentra el código correspondiente, brindando al cliente la opción de utilizar tanto la API de Google Maps como el *scraper* desarrollado.

De este modo, se han elaborado tanto el código para hacer uso de la API de Google como para *scrapear* la información necesaria. Así el cliente podrá tomar la decisión que más se adapte a sus necesidades y sobre todo a su presupuesto.

3.2 Tratamiento y limpieza de datos

En esta sección, nos adentraremos en el proceso esencial de limpieza y estandarización de los datos, un paso crucial al trabajar con información proveniente de diversas fuentes. Es de vital importancia prestar una atención especial a las variables relacionadas con las coordenadas geoespaciales, ya que estas pueden presentarse en diferentes proyecciones o unidades de medida. La falta de uniformidad en estos datos puede conducir a errores significativos en los cálculos y análisis posteriores si no se aborda con cautela y precisión.

3.2.1 Coordenadas geoespaciales y proyecciones

En el campo de la cartografía y la representación de la Tierra en mapas, se utiliza el concepto de proyección geoespacial. Una proyección geoespacial es una técnica que permite representar la superficie curva de la Tierra en un plano, como un mapa. Debido a la naturaleza esférica de nuestro planeta, es imposible representar de manera precisa y sin distorsiones todas las áreas y formas en un mapa plano.

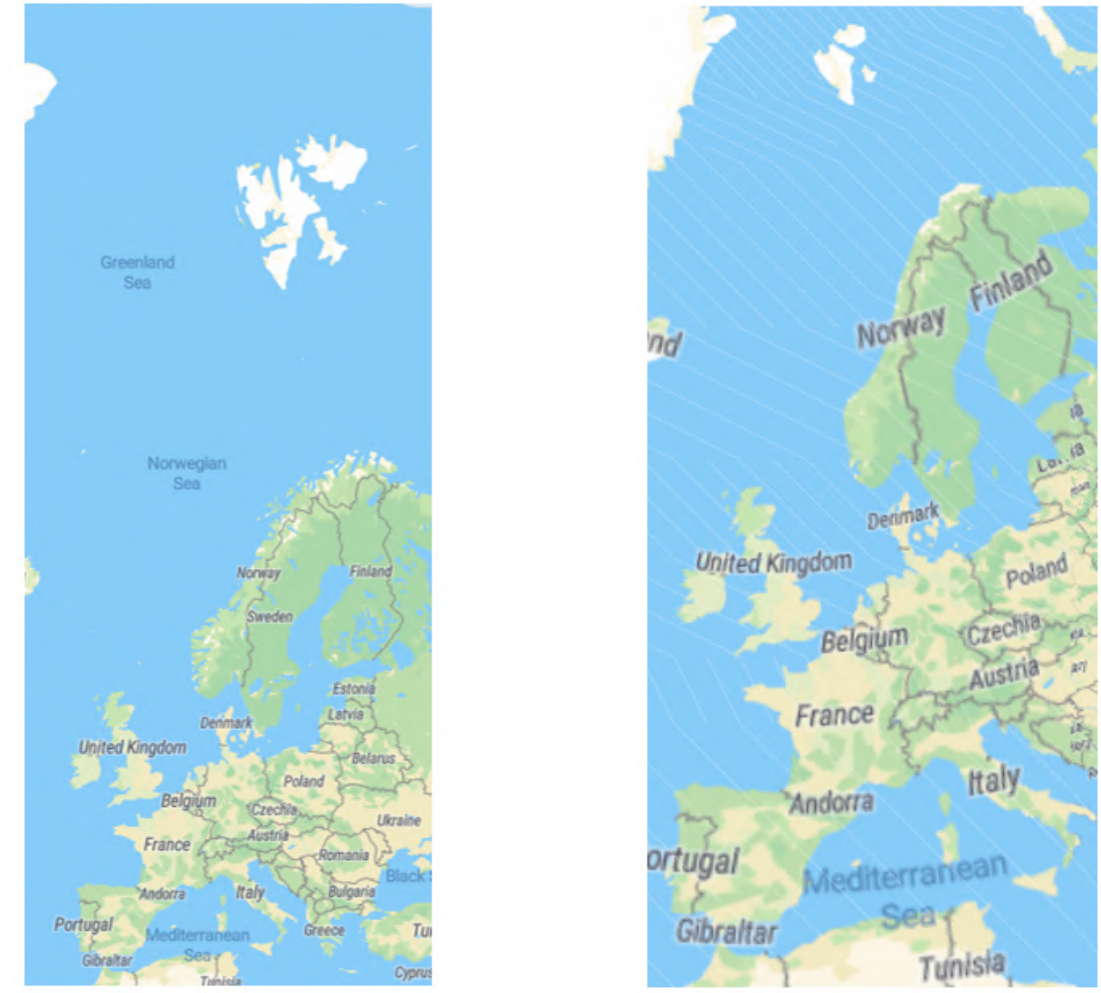
Existen numerosas proyecciones geoespaciales disponibles, cada una diseñada para cumplir con diferentes objetivos y necesidades. Estas proyecciones pueden variar en términos de su forma, área, distancia o dirección o ubicación. Algunas proyecciones se centran en conservar la forma de las áreas, mientras que otras priorizan la conservación de las distancias o direcciones.

Las coordenadas EPSG (European Petroleum Survey Group) son un sistema de referencia espacial utilizado para especificar y transformar coordenadas geoespaciales. Estas coordenadas están basadas en un catálogo de códigos numéricos que

identifican proyecciones cartográficas, sistemas de coordenadas y datos geodésicos específicos.

La designación EPSG asigna un código único a cada sistema de referencia espacial, lo que facilita la comunicación y el intercambio de datos geoespaciales entre diferentes aplicaciones y sistemas.

En el caso específico de nuestro proyecto, los datos públicos de la Comunidad de Madrid vienen expresados en la proyección lineal EPSG:25830, que es una proyección en unidades métricas lineales (metros) conocida como UTM 30N ($Universal\ Transverse\ Mercator$). Esta proyección abarca Europa desde los 6ºO hasta los 0ºO y desde los 30ºN, incluyendo toda España. La proyección UTM divide la Tierra en zonas, y la zona 30N corresponde a la región donde se encuentra la Comunidad de Madrid.



(a) UTM (b) EPSG:25830

Figure 3.1: Mapa de la zona UTM 30N (Europa) en representada en diferentes proyecciones [1].

En el contexto de nuestro proyecto, es necesario estandarizar las coordenadas geoespaciales provenientes de diferentes fuentes. Para ello, realizaremos una transformación de las coordenadas en la proyección $EPSG : 25830$ a la proyección $EPSG : 4326$. Esta última proyección nos proporciona coordenadas angulares de latitud y longitud aplicables a todos los lugares de la Tierra, permitiéndonos ubicar las localizaciones y zonas censales de manera precisa. Ya que no precisaremos el cálculo de distancias y áreas de forma precisa.

Mediante este proceso de transformación de coordenadas, podremos trabajar con datos geoespaciales coherentes y realizar análisis y visualizaciones de manera

precisa en nuestro proyecto. Aunque el trabajo con proyecciones geoespaciales puede resultar complejo y requerir un esfuerzo adicional, es fundamental para asegurar la integridad y calidad de nuestros datos y resultados geoespaciales.

3.3 Análisis exploratorio de datos

El análisis exploratorio de datos desempeña un papel fundamental en este proyecto, ya que nos permite comprender las distribuciones estadísticas y geográficas de las diferentes variables que estamos investigando. En esta sección, examinaremos de cerca las secciones censales, las variables sociodemográficas y las ubicaciones de los servicios. Este análisis nos ayudará a identificar patrones, tendencias y disparidades en la distribución de los servicios esenciales en la Comunidad de Madrid. Al comprender mejor estas distribuciones, estaremos en una posición sólida para desarrollar estrategias efectivas para detectar y abordar la escasez de servicios esenciales.

3.3.1 Secciones censales

Las secciones censales son una unidad territorial utilizada en España para realizar el censo de población. En este contexto, cada sección censal representa un área geográfica específica dentro del país. Estas secciones se establecen con el objetivo de recopilar datos sociodemográficos detallados y obtener información sobre la distribución de la población en diferentes áreas.

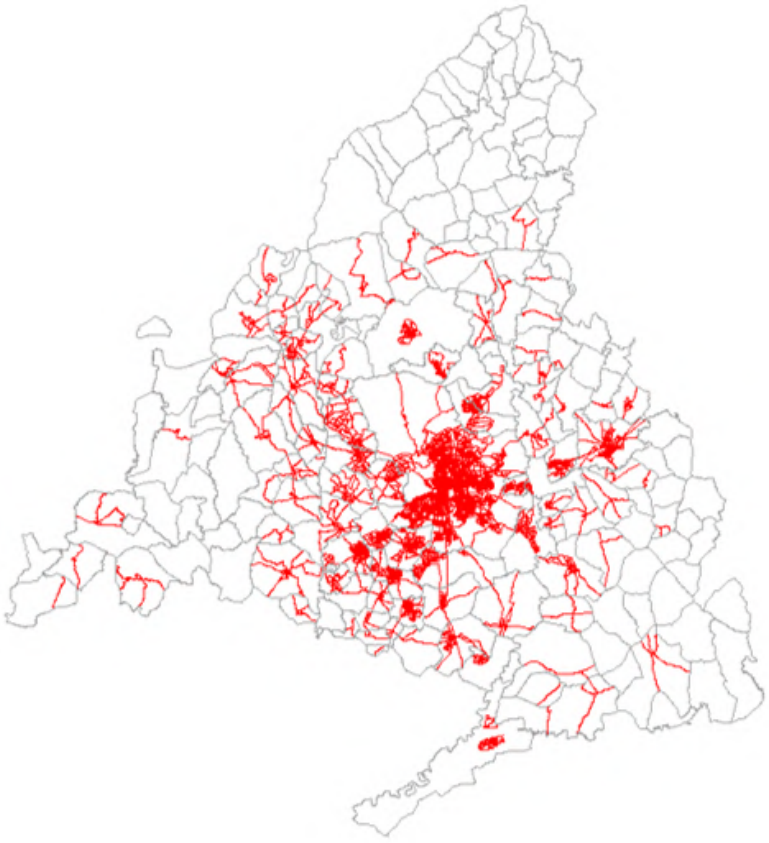


Figure 3.2: Mapa con las más de 4400 secciones censales de la Comunidad de Madrid. [2]

La población se divide en secciones censales para facilitar el análisis y la comprensión de los datos demográficos a nivel local. Al tener información sociodemográfica a nivel de sección censal, se puede obtener un panorama más detallado de la población en diferentes áreas geográficas, lo que resulta especialmente útil para estudios y análisis que requieren una granularidad más fina.

Sin embargo, una limitación de las secciones censales es que su tamaño y forma pueden variar considerablemente en diferentes partes del país. En áreas urbanas densamente pobladas, las secciones censales pueden ser relativamente pequeñas y abarcar áreas geográficas más reducidas. En contraste, en zonas rurales, las secciones censales pueden ser mucho más grandes y abarcar áreas extensas. Esto puede generar desafíos al realizar análisis comparativos, ya que las secciones censales rurales pueden ser cientos de veces más grandes que las de las zonas urbanas.

A pesar de esta variabilidad en el tamaño de las secciones censales, su uso sigue siendo valioso, ya que proporcionan una forma estructurada de obtener informa-

ción sociodemográfica detallada a nivel local. En el contexto de este proyecto, las secciones censales son una unidad de análisis relevante, ya que el Instituto Nacional de Estadística (INE) proporciona datos sociodemográficos a este nivel. Esto nos permite tener acceso a información detallada sobre la población en zonas específicas, lo que nos permite comprender mejor la distribución de los servicios esenciales y detectar posibles disparidades entre diferentes áreas geográficas.

3.3.2 Variables sociodemográficas

Las secciones censales se caracterizan utilizando diversas variables sociodemográficas que nos proporcionan una visión completa de cada zona. Estas variables nos permiten entender mejor la composición y las características de la población en cada sección censal. A continuación, se presentan las variables utilizadas en este estudio:

- Población: Representa el número total de habitantes en cada sección censal. Nos proporciona una medida del tamaño de la población en cada área y nos permite comparar las diferencias demográficas entre las zonas.

- Área: Indica el tamaño del territorio cubierto por cada sección censal. Se expresa en unidades de superficie y nos ayuda a comprender la extensión geográfica de cada zona.

- Renta familiar: Es la suma de las rentas de todos los miembros de una familia que reside en la sección censal. Esta variable nos proporciona una medida de la situación económica de las familias en cada área.

- Renta per cápita: Calculada como el promedio de las rentas personales de los habitantes de cada sección censal. Nos permite entender el nivel de ingresos individual en cada zona y comparar las disparidades económicas entre diferentes áreas geográficas.

- Porcentaje de extranjeros: Indica la proporción de habitantes extranjeros en cada sección censal. Esta variable nos ayuda a evaluar la diversidad cultural y la presencia de población extranjera en cada área.

- Porcentaje de edad 0-24: Representa la proporción de habitantes con edades comprendidas entre 0 y 24 años en cada sección censal. Nos proporciona información sobre la estructura de edad de la población y nos ayuda a comprender la distribución de la población más joven.

- Porcentaje de edad 40-49: Indica la proporción de habitantes con edades comprendidas entre 40 y 49 años en cada sección censal. Esta variable nos

permite analizar la distribución de la población en edad media y evaluar posibles patrones demográficos.

- Porcentaje de edad 70+: Representa la proporción de habitantes mayores de 70 años en cada sección censal. Nos ayuda a identificar áreas con una mayor proporción de población en edades avanzadas y a comprender las necesidades específicas de este grupo.

- Perc gasto vivienda hogar: Mide el porcentaje de gasto destinado a vivienda por parte de los hogares en cada sección censal. Esta variable nos proporciona información sobre la carga económica relacionada con la vivienda en cada área.

- Polígono geográfico de cada zona: Se refiere al conjunto de puntos en coordenadas (latitud, longitud) que delimitan la frontera de cada sección censal. Esta información espacial nos permite visualizar y analizar la distribución geográfica de las zonas censales en el mapa.

Al utilizar estas variables sociodemográficas, podemos obtener una visión detallada de cada sección censal y comprender las diferencias en términos de población, características económicas y demográficas entre las distintas áreas geográficas. Estos datos nos serán de gran utilidad para realizar un análisis exploratorio y un modelado predictivo en este proyecto de detección de necesidades en servicios esenciales.

Veamos algunos ejemplos de las distribuciones estadísticas y geográficas de algunas de estas variables. Para poder comprender de una manera visual los datos que trataremos.

Población

Como ya mencionamos, hemos escogido las secciones censales para realizar nuestro modelo, porque estas dividen a la población en grupos geográficos con una cantidad de habitantes muy similar. Y además nos brindan información muy relevante para nuestro estudio al disponer en todas ellas de una gran cantidad de datos. Veamos la distribución de población en todas ellas.

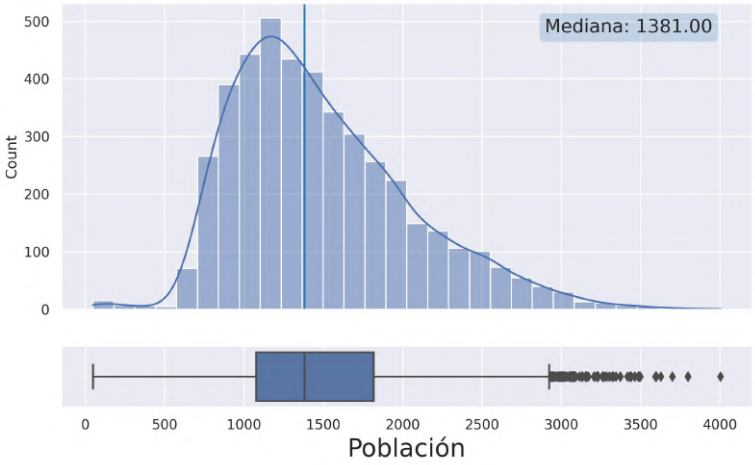


Figure 3.3: Histograma de la distribución del número de habitantes en las secciones censales.

Efectivamente vemos como la mayoría de secciones censales tienen un número de habitantes entre las 1000 y 2000 personas. Pese a esto, no vemos una distribución normal. Tanto la media como la mediana se encuentran desplazadas hacia la derecha mostrando una distribución sesgada a la derecha.

Edad media

Para ver si las secciones censales pueden caracterizarse por ser más jóvenes o de edad más avanzada, veamos como se distribuye la edad media en todas ellas.

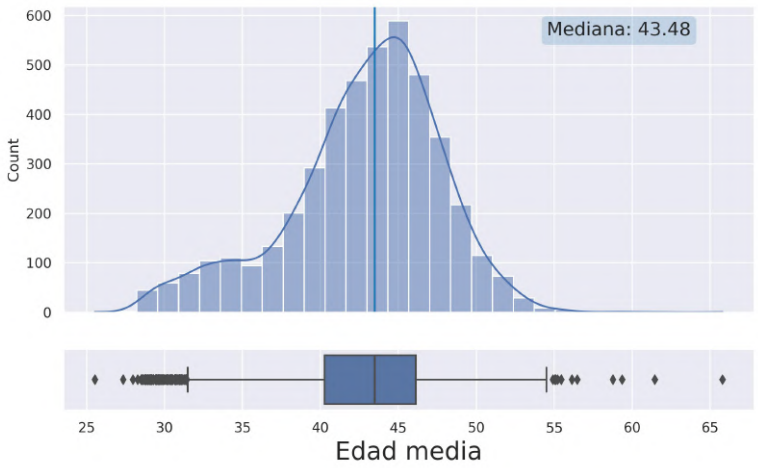
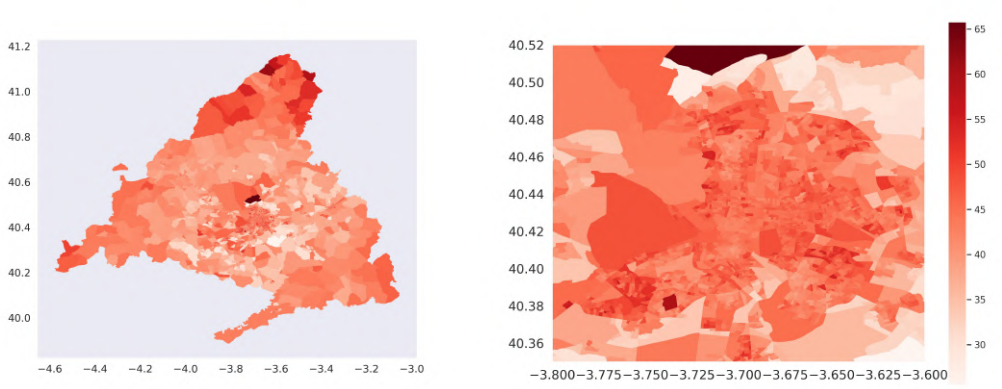


Figure 3.4: Histograma de la edad media en las secciones censales.

Esta distribución es algo más confusa, ya que pese a estar sesgada a la izquierda con la moda en 43.48 años, podría decirse que además es bimodal por la pequeña perturbación cerca de los 32 años. Lo cual podría indicar que existe una pequeña concentración de zonas censales más jóvenes.

Veamos en el mapa donde se ubican las zonas censales según la edad media para comprobar si existe dicha separación entre unas zonas especialmente jóvenes.



(a) Comunidad de Madrid (b) Ciudad de Madrid

Figure 3.5: Mapa de calor donde se representa la edad media en las secciones censales de la Comunidad de Madrid.

En general, no vemos una distinción entre zonas con edades claramente diferenciadas. Solamente podemos apreciar una pequeña concentración de lugares en la periferia de la comunidad donde la edad es mayor y la única concentración de zonas más jóvenes puede verse en la periferia de la ciudad de Madrid.

Porcentaje de extranjeros

Veamos la distribución del porcentaje de extranjeros en las zonas censales

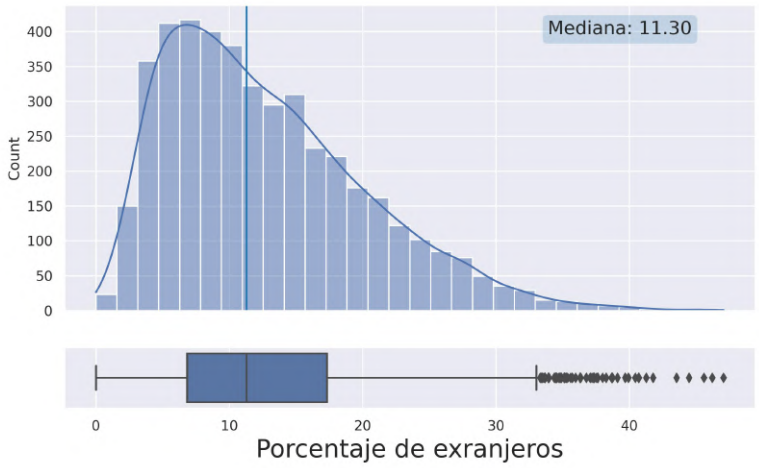
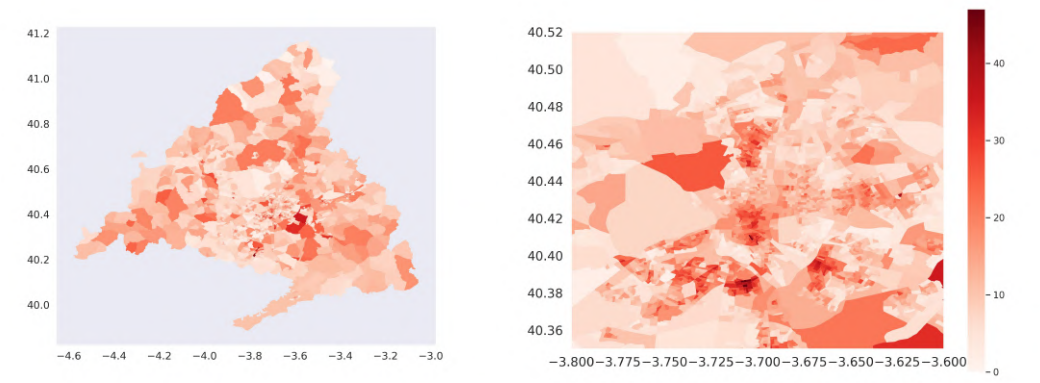


Figure 3.6: Histograma del porcentaje de extranjeros en las secciones censales.

La distribución es claramente sesgada a derechas con la mediana situada en el 11.3% de extranjeros. Par ver si esta variable tiene una distribución geográfica clara, representemos en escala de calor este porcentaje en el mapa de la Comunidad de Madrid



(a) Comunidad de Madrid (b) Ciudad de Madrid

Figure 3.7: Mapa de calor donde se representa el porcentaje de extranjeros en las secciones censales de la Comunidad de Madrid.

Es interesante ver, que en la ciudad de Madrid existen focos localizados donde el porcentaje de extranjeros asciende hasta el 30% o 40%. Pero en el resto de la comunidad no podemos apreciar una distribución concreta de dicha variable.

Renta per cápita

Este indicador económico, mide los ingresos brutos medios dentro de cada una de las zonas, un claro indicio para determinar el nivel de riqueza dentro de esta. Veamos cuál es su distribución dentro de todo del conjunto de datos.

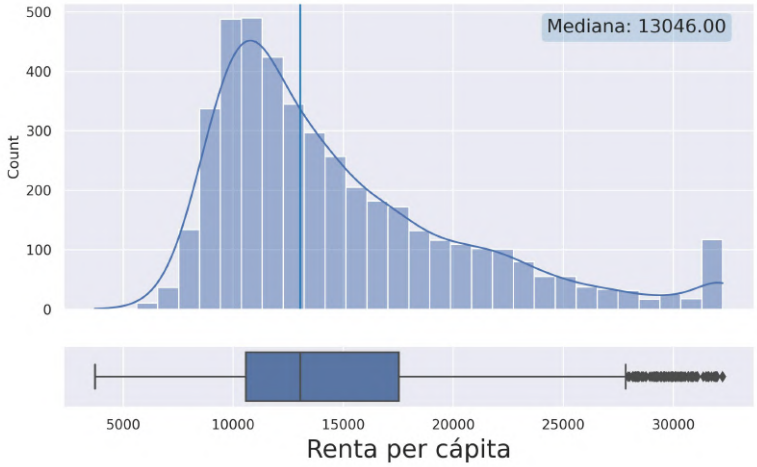
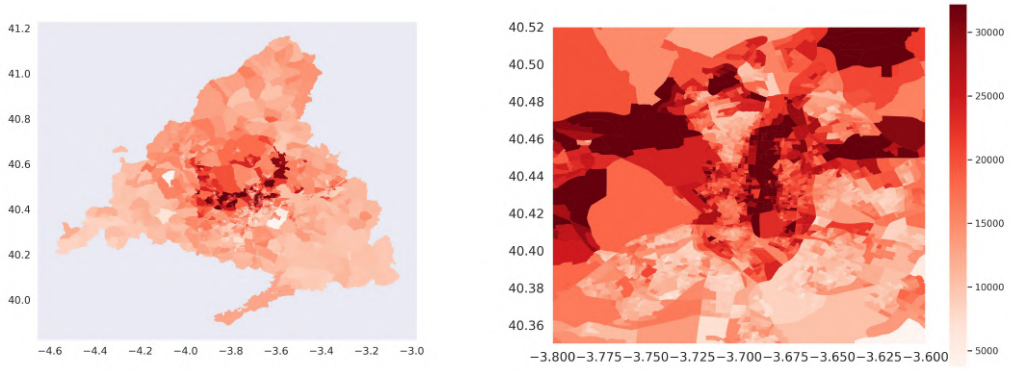


Figure 3.8: Histograma de la renta per cápita en las secciones censales.

De nuevo vemos una distribución sesgada hacia la derecha con la mediana situada en los 13.046 €. Pero lo que nos interesa realmente en este proyecto, es la distribución geográfica de este indicador, para comprender la distribución de los diferentes servicios entre las zonas censales.



(a) Comunidad de Madrid (b) Ciudad de Madrid

Figure 3.9: Mapa de color donde se representa la renta per cápita en la comunidad de Madrid

En este caso, sí que se aprecia una fuerte concentración de la riqueza en la

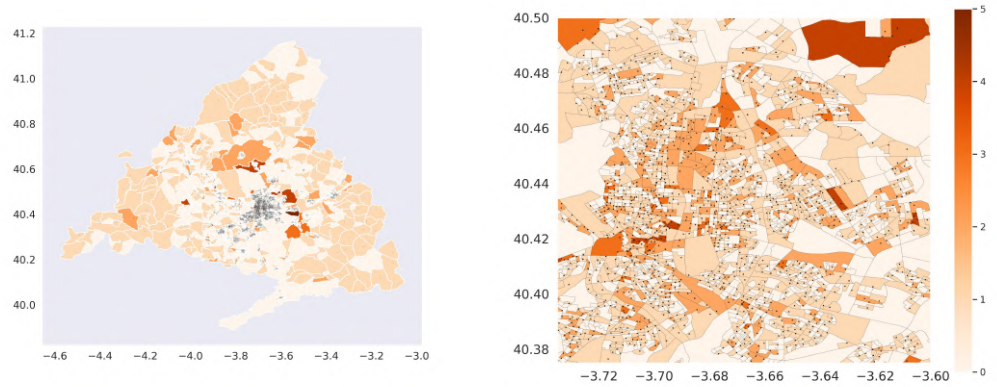
ciudad de Madrid y sus alrededores. Incluso dentro de esta, la diferencia es muy notable entre la zona sur y norte.

3.4 Obtención de la variable objetivo

Como hemos mencionado anteriormente, nuestro objetivo será, ajustar un modelo capaz de predecir la cantidad de cada uno de los servicios en las secciones censales. Para alcanzar dicha meta, lo primero será obtener el número de localizaciones que ofrecen el servicio en cada una de ellas. Partiendo de los datos que se ofrecen en el catálogo de datos abiertos de la Comunidad de Madrid. Los pasos a seguir son los siguientes. Primero descargamos el archivo .csv (o el cualquier otro formato) del servicio en cuestión, pongamos como caso el de las oficinas de farmacia. Lo primero que vemos es que las coordenadas de todas las localizaciones vienen dadas en proyección lineal EPSG:25830 (en el caso de datos abiertos de la Comunidad de Madrid). Nosotros necesitamos la latitud y longitud de todas ellas para ubicarlas dentro de una de las zonas censales. En resumidas cuentas, convertimos las columnas de coordenadas X e Y a formato POINT(X , Y), sobre la cual aplicaremos la proyección a coordenadas angulares para obtener un punto de la forma POINT (lat , lon). Una vez tenemos tanto los puntos donde se ubican los servicios, como los puntos que delimitan los polígonos de las zonas censales. Pasaremos a localizar cada servicio en la zona que le corresponde. Para ello recorreremos con un bucle todos los polígonos y todos los puntos de los servicios, detectando los puntos que caen dentro de cada polígono. Para añadir una variable contador y saber cuántos puntos de un determinado servicio, caen en cada sección censal. Ahora que hemos realizado el recuento de cada uno de los servicios de ejemplo dentro de nuestras secciones censales, podemos analizar como es su distribución geográfica otros valores.

Farmacias

Disponemos de un total de 2919 ubicaciones de oficinas de farmacia a lo largo de la Comunidad de Madrid. Puesto que nuestro objetivo es determinar si una zona censal posee un exceso o defecto de farmacias en función de sus variables sociodemográficas, realizaremos una exploración visual de los datos para explorar las ubicaciones de dicho servicio en el mapa.



(a) Comunidad de Madrid (b) Ciudad de Madrid

Figure 3.10: Mapa de la Comunidad de Madrid donde se han representado con puntos, las ubicaciones de las oficinas de farmacia y se ha pintado cada zona en función del número de farmacias albergadas con una escala de color.

A la derecha vemos una ampliación de la ciudad de Madrid, donde tenemos la mayor concentración de ellas.

Se puede ver en el mapa que la mayoría de las zonas censales presentan un número bajo (0 o 1) de farmacias. Veamos un gráfico de barras para saber concretamente el número de farmacias por cada zona.

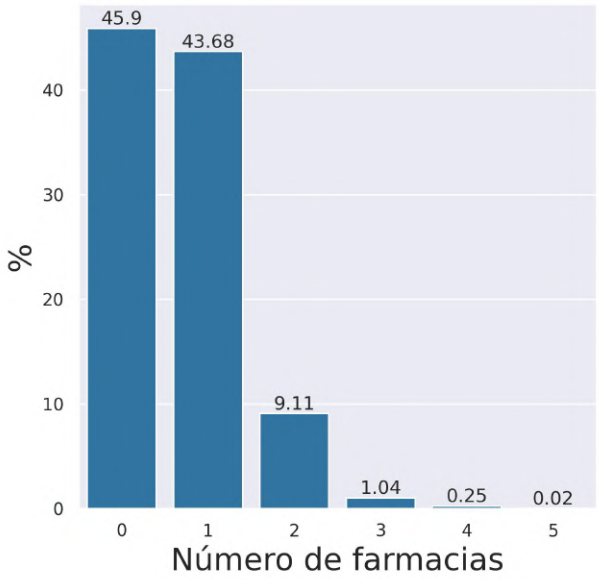
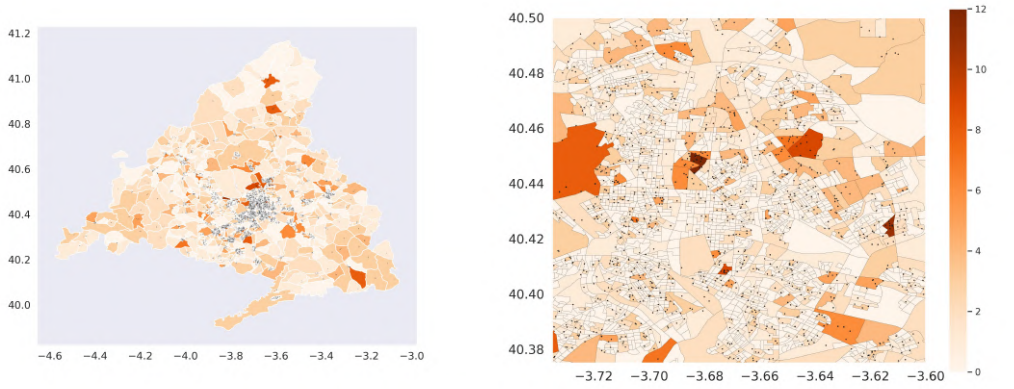


Figure 3.11: Gráfico de barras del número de farmacias por sección censal

Efectivamente vemos como prácticamente, el 90% de las zonas censales alberga una o ninguna farmacia. A posteriori, nos interesará ver especialmente, cual de estos lugares sin farmacias, muestran una mayor necesidad para dicho servicio.

Centros educativos

Disponemos de un total de 4006 ubicaciones de centros educativos repartidos por toda la Comunidad de Madrid. Del mismo modo que con las oficinas de farmacia, veamos una distribución visual de dicho servicio en el mapa.



(a) Comunidad de Madrid (b) Ciudad de Madrid

Figure 3.12: Mapa de la Comunidad de Madrid donde se han representado con puntos, las ubicaciones de los centros educativos y se ha pintado cada zona en función del número de centros albergados con una escala de color.

Podemos apreciar como algunas zonas censales presentan hasta 12 centros educativos, pero vemos como este número tan elevado no es la norma y la mayoría de las zonas tienen un número muy inferior de estos centros. Viendo una mayor densidad de ellos, como podríamos esperar, en la ciudad de Madrid Con un gráfico de barras podremos ver con mayor claridad el porcentaje de zonas con según qué número de centros educativos.

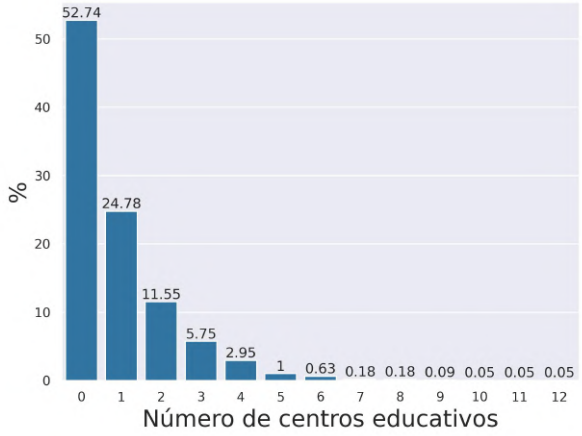


Figure 3.13: Gráfico de barras del número de centros educativos por sección censal

Efectivamente vemos como el número de zonas con más de 4 centros, es muy escaso. De hecho, el 90% de las zonas censales tiene entre 0 y 2 localizaciones de este servicio.

Puntos de venta y recarga de la Tarjeta de Transporte Público

Este servicio se refiere concretamente a los puntos de venta autorizados para la carga de la tarjeta de transporte público, habiendo hasta 1652 de ellos. De los cuales 1195 son puntos de venta autorizados y 457, estaciones de metro, cercanías y metro ligero o tranvía. Veamos primero como están distribuidas las líneas de transporte público dentro de la Comunidad de Madrid, metro, cercanías, tranvía y autobús.

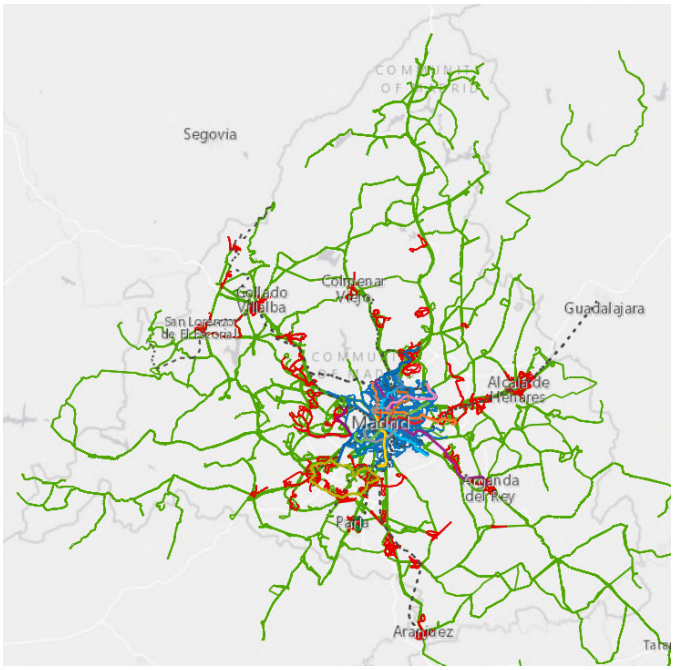
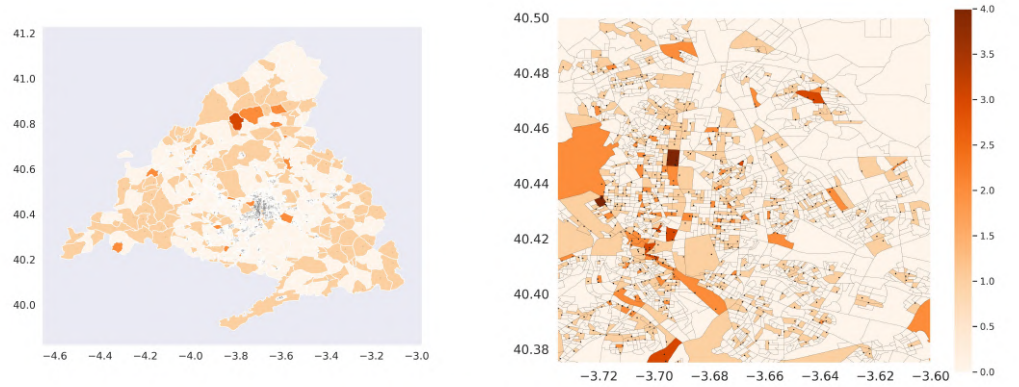


Figure 3.14: Mapa de las lineas de transporte público de la Comunidad de Madrid [3].

Vemos como en la zona urbana de Madrid hay una gran concentración de todas las líneas disponibles y en la periferia tan solo alcanzan las líneas de autobús y cercanías. Para ubicar visualmente en el mapa todos los puntos de venta de tarjeta de transporte público, representaremos de nuevo la localización de los 1195 servicios.



(a) Comunidad de Madrid (b) Ciudad de Madrid

Figure 3.15: Mapa de la Comunidad de Madrid donde se han representado con puntos, las ubicaciones de TTP y se ha pintado cada zona en función del número de lugares albergados con una escala de color.

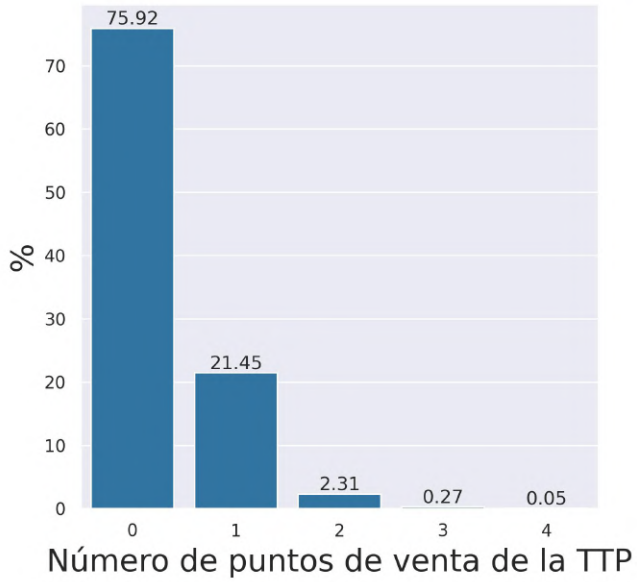


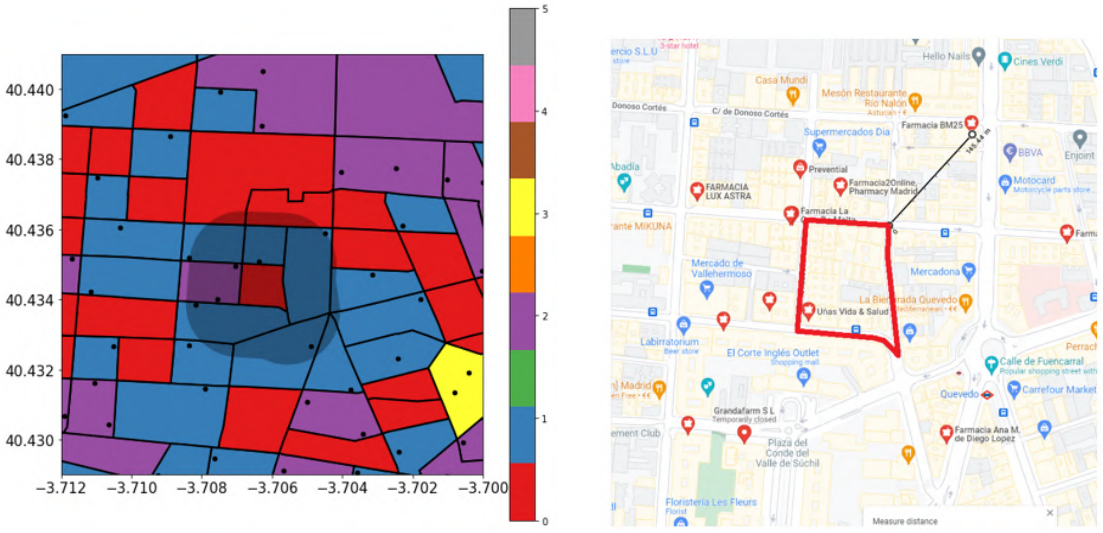
Figure 3.16: Gráfico de barras del número de centros de venta y recarga de la Tarjeta de Transporte Público por sección censal

A diferencia de los dos anteriores servicios, al disponer de un menor número de localizaciones, vemos como la mayoría de las zonas censales no presentan ningún punto de venta de la TTP. Comparando los mapas de líneas de transporte y puntos de venta de la TTP, se puede ver una correlación entre los lugares por los que pasan dichas líneas y los puntos de venta. Por ende, podremos determinar que las zonas donde no se puede adquirir la TTP son zonas con escasez de líneas de transporte y los ciudadanos que las habiten tendrán cierta necesidad en este esencial servicio. Con un gráfico de barras podremos ver con mayor claridad el porcentaje de zonas con según que número de puntos de venta de la tarjeta de transporte público.

Como habíamos afirmado anteriormente, al ser este un servicio más escaso, que además concentra localizaciones en la ciudad de Madrid. Deja a la gran mayoría de zonas censales sin cubrir.

3.4.1 Problema con servicios en las fronteras

El problema de calcular de esta manera el número de localizaciones, del servicio que deseemos estudiar, que hay en el interior cada sección, es que muchas zonas que estén rodeadas por farmacias cercanas, no van a estar reconocidas con el número veraz de estas. Sobre todo, en la zona de Madrid donde las secciones censales pueden ser tan pequeñas como un par de manzanas. A lo largo de toda la comunidad podemos ver ejemplos como el siguiente.



(a) (b) Google Maps [13]

Figure 3.17: Imagen de una sección censal ubicada en la ciudad de Madrid, en esta sección vemos como no hay ninguna farmacia en el interior. Pero en sus alrededores hay hasta 6 farmacias a menos de $150m$ de sus fronteras

Vemos como esta zona no tiene ninguna farmacia en su interior, pero si nos fijamos en sus alrededores, vemos hasta 6 que se encuentras a menos de $150m$ de sus fronteras. ¿Podríamos considerar nosotros que en esta zona hay escasez de dicho servicio? Evidentemente no. Si tratáramos de hacer un modelo de *MachineLearning* capaz de predecir las farmacias en el interior de las secciones censales, tomando como variables de entradas sus datos sociodemográficos, el modelo obtendría casi con total certeza que en este sección de la figura (3.17) habría alguna farmacia. Este precisamente sería el error que cometeríamos si contáramos el número exacto de farmacias en cada sección censal y lo usáramos como variable a predecir.

Para solucionar este problema lo que haremos será contar el número de servicios tanto en el interior como a una distancia inferior a $150m$ desde las fronteras. Tal y como vemos en la figura (3.17), la zona a menos de $150m$ es la sombreada en gris.

El único inconveniente que puede tener este proceso es que el recuento total de farmacias, será superior al número de farmacias totales. Aunque esto no supondrá un problema en nuestro caso. De este modo obtenemos nuestra variable objetivo, que será el número de farmacias interiores y cercanas a cada sección censal.

De este modo obtenemos nuestra variable objetivo, que será el **número de farmacias interiores y cercanas** a cada sección censal.

3.5 Modelos

3.5.1 Selección del modelo explicativo y funciones objetivo

A la hora de elaborar un modelo para obtener nuestra variable objetivo, procederemos de manera diferente a cuando tratamos de conseguir una predicción. En nuestro caso, no buscamos un modelo capaz de predecir el número de farmacias o centros educativos para nuevas zonas censales que pudieran aparecer en un futuro. Nuestro objetivo es, modelizar todas las zonas existentes para tratar de explicar la tendencia general que siguen e identificar las anómalas en cuanto a la cantidad de ubicaciones que ofrecen cada servicio se refiere. Estas zonas censales que presenten un mayor error en el modelo, serán identificadas como aquellas que no tengan el número de servicios correspondiente a sus características sociodemográficas. Pudiendo así detectar una escasez o un exceso de los mismos. Para lograr un modelo que satisfaga nuestras necesidades, separaremos nuestro conjunto de datos en entrenamiento (80%) y test (20%). Seguidamente aplicaremos un algoritmo de validación cruzada para prevenir un sobre entrenamiento y a la vez seleccionar el mejor conjunto de hiperparámetros.

En este proceso de validación cruzada se ha optado por minimizar la desviación de Poisson y la desviación Tweedie. Estas distribuciones de probabilidad son

adecuadas para modelar variables discretas y positivas. Precisamente como los conteos de servicios esenciales en cada sección censal de la Comunidad de Madrid, como es nuestro caso.

Desviación de Poisson

La elección de la desviación de Poisson como métrica se debe a su capacidad para manejar datos de conteo y su naturaleza discreta y no negativa. Al minimizar esta métrica en nuestro modelo, buscamos encontrar el conjunto de parámetros y características que permitan un ajuste óptimo entre los datos observados y las predicciones del modelo.

$$2 \cdot \sum_i [y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu})] \tag{3.1}$$

En esta métrica aparecen los valores reales (y_i) y predichos por el modelo ($\hat{\mu}_i$). Para ver cómo se comporta en función de los valores reales y predichos. Representaremos la desviación media de Poisson en función de los valores predichos y para diferentes valores del valor real.

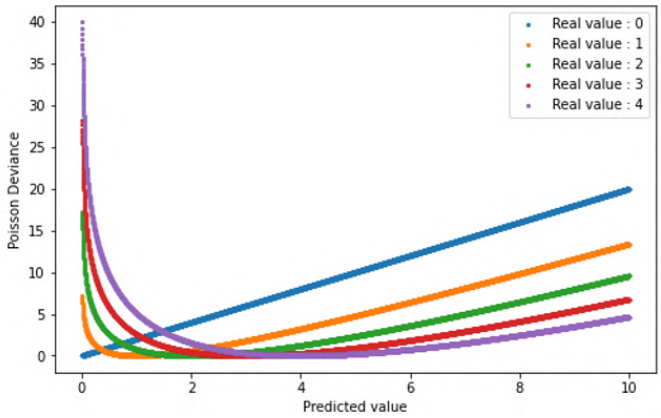


Figure 3.18: Desviación media de Poisson frente al valor predicho, representada para diferentes valores del valor real

Son destacables en la figura (3.18) tanto el comportamiento asintótico en el 0 y los mínimos alrededor del valor predicho que se corresponden con el valor real. El único problema que encontramos con esta métrica es que para el valor real = 0, la función se comporta como una recta y al no tener este comportamiento asintótico en el origen, nada impide que se predigan valores negativos en los casos cuyo valor real sea el 0. Para evitar este problema con los datos cuya cantidad de servicios sea 0, transformaremos los valores reales, añadiendo una unidad más. Evidentemente,

a la salida de las predicciones, volveremos a sustraer dicha cantidad. Cuando ya hayamos obtenido los hiperparámetros óptimos en el proceso de validación cruzada, entrenaremos un nuevo modelo con todo el conjunto de datos, sin sobre entrenamiento ni sesgo entre conjuntos de entrenamiento y test.

Desviación Tweedie

En este proceso de validación cruzada, además de minimizar la desviación de Poisson, también exploraremos la utilización de la desviación de Tweedie como métrica de evaluación. La desviación de Tweedie es una extensión natural de la desviación de Poisson y se utiliza para modelar variables con valores continuos y positivos, al igual que la desviación de Poisson.

La desviación de Tweedie presenta características adicionales que pueden ser beneficiosas en ciertos escenarios. A diferencia de la desviación de Poisson, la desviación de Tweedie puede capturar la heterogeneidad en los datos, permitiendo modelar variables con una mayor varianza y cola larga en la distribución de probabilidad. Esto resulta especialmente útil cuando se trata de datos con una alta concentración de ceros y una dispersión más amplia, podemos ver un ejemplo de este tipo de distribución en la figura (3.19).

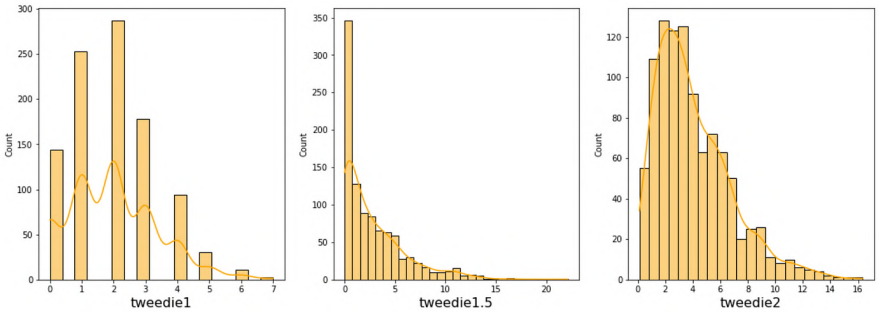


Figure 3.19: Distribución Tweedie para distintos valores del parámetro p, Cuando p toma el valor de 1, la desviación de Tweedie se reduce a la desviación de Poisson. A medida que el valor de p se acerca a 2, la distribución se asemeja más a una distribución gamma. [4]

La fórmula de la desviación de Tweedie que utilizaremos en nuestro análisis es una generalización de la desviación de Poisson y se define de la siguiente manera:

$$2 \cdot \sum_i \left[y_i \cdot \left(\frac{\hat{\mu}_i}{\phi} \right)^{p-1} - \frac{y_i}{\phi} \cdot \left(\frac{\hat{\mu}_i}{\phi} \right)^{p-2} + \frac{y_i}{\phi} \cdot \left(\frac{y_i}{\phi} \right)^{1-p} - \frac{y_i}{\phi} \right] \tag{3.2}$$

En esta fórmula, y_i representa los valores reales, $\hat{\mu}_i$ son las predicciones del modelo, ϕ es el parámetro de dispersión y p es el índice de Tweedie. Cuando p

toma el valor de 1, la desviación de Tweedie se reduce a la desviación de Poisson. A medida que el valor de p se acerca a 2, la distribución se asemeja más a una distribución gamma, como podemos ver en la figura (3.19).

Este proceso de entrenamiento ha sido ejecutado de la misma forma para diferentes modelos con distinta flexibilidad y robustez, manteniendo siempre un cierto grado de explicabilidad en los mismos.

Una vez tengamos todas las "predicciones", calcularemos de manera sencilla la métrica que nos indicará el exceso o escasez del servicio en concreto para cada zona censal ($Score$). Simplemente restaremos el valor real menos el valor predicho por el modelo.

$$Score = Y_{real} - Y_{modelo} \tag{3.3}$$

Para la predicción de cada servicio, se ha seleccionado el mejor conjunto de variables, valorando la explicabilidad y flexibilidad del modelo. Esta selección se ha llevado a cabo con el algoritmo $Shap$, el cual ordena las variables de entrada según su relevancia, pudiendo visualizar el impacto de cada una de ellas en las predicciones.

3.5.2 Entrenamiento de modelos

La peculiaridad de nuestro proyecto es que al tratarse de un modelo de explicabilidad y no de predicción, no nos interesa que este se adapte bien a nuevas entradas de datos. Lo que nosotros buscamos, es un modelo que, sin llegar a memorizar perfectamente los datos, consiga entender la tendencia general de los mismos. A priori, al tener la variable objetivo valores discretos, podemos pensar que la manera correcta de abordar este caso sea usando modelos de clasificación. Pero lo que nosotros buscamos, es que si, por ejemplo, el número de farmacias en una región es de 2, nuestro modelo prediga algo como 0.89 farmacias, si se trata de un exceso de farmacias o 2.93 farmacias si lo que tenemos es un defecto de las mismas. Puede parecer un sinsentido que el modelo prediga 2.93 farmacias en un zona censal, pero de esta forma podremos detectar las ausencias de servicios de manera más precisa. Si abordáramos el problema con modelos de clasificación. En los casos en los que el modelo predijera el valor correcto con más de 0.5 de probabilidad, no detectaríamos los errores. Y como mucho analizando esta probabilidad podríamos ver cuan seguro estaba el modelo a la hora de predecir, no si nos estamos aproximando al número de servicio con exceso o defecto. Dado que en última instancia buscamos conocer la cantidad de cada servicio esperada por nuestro modelo y sobre todo, explicar las características de cada zona con las que se ha llegado a dicho valor esperado. Haremos uso de tres modelos con complejidades diferentes, donde en todos ellos disponemos de herramientas para explicar los resultados. Los modelos utilizados son los siguientes:

Poisson regressor

Como una primera aproximación utilizaremos el modelo PoissonRegressor de la librería de Python, scikit-learn [14]. Este modelo, genera buenos resultados cuando nuestra variable objetivo expresa un conteo, es decir, una variable discreta con valores mayores que 0. Dicho de otra forma, nuestra variable objetivo presenta una distribución de Poisson. Este modelo se encuentra en la categoría de los modelos lineales generalizados (GLM). A diferencia de los modelos lineales, en estos no se espera una distribución normal en la variable respuesta con valores continuos entre menos y más infinito. Los parámetros que hemos seleccionado en el proceso de validación cruzada son los siguientes:

- **alpha**: indica el nivel de penalización de los coeficientes.
- **fit_intercept**: especifica si se aplica una constante.
- **max_iter**: indica el número máximo de iteraciones.

Decission tree

Este es el modelo clásico de árbol de decisión [15]. Donde en orden de importancia de las variables, se seccionan los datos hoja por hoja. Haciendo que las predicciones tomen un camino u otro según el valor de las entradas. Pese a su escasa complejidad, este modelo presenta una clara explicabilidad de como en cada caso se ha llegado a una predicción u otra. Para ello se puede ver el camino (rama) que ha tomado cada entrada en las diferentes bifurcaciones (hojas). Los parámetros que hemos seleccionado en el proceso de validación cruzada son los siguientes:

- **max_depth**: Profundidad máxima de un árbol.
- **min_samples_leaf**: El número mínimo de muestras para para dividir un nodo
- **min_samples_split**: El número mínimo de muestras que ha de haber en un nodo.

XGBoost

XGBoost [16] es una biblioteca de aprendizaje automático optimizada que se basa en el algoritmo de Gradient Boosting. Este enfoque utiliza un procedimiento similar al descenso de gradiente para minimizar la pérdida del modelo al agregar nuevos modelos débiles de forma secuencial.

En el proceso de Gradient Boosting, se considera el impulso como un problema de optimización numérica. El objetivo es minimizar la pérdida del modelo al agregar sucesivamente modelos débiles. Cada nuevo modelo se entrena para enfocarse en las áreas donde se ha cometido un mayor error en las predicciones anteriores. Esto permite mejorar la capacidad del modelo para capturar patrones complejos y detectar casos menos frecuentes.

A diferencia de los modelos de árboles individuales, el rendimiento del XGBoost se ve significativamente mejorado mediante el proceso de Boosting. El algoritmo identifica los errores de las predicciones anteriores y se enfoca en corregirlos a medida que se agregan más modelos débiles. Esta capacidad de corrección progresiva hace que XGBoost sea especialmente eficaz para problemas con datos complejos y desafiantes.

No obstante, es importante tener en cuenta el riesgo de sobreajuste en el proceso de Boosting. Si no se limita el número de árboles o se ajustan correctamente otros parámetros, el modelo puede memorizar los datos de entrenamiento y tener un rendimiento deficiente en datos nuevos. En nuestro caso, buscamos alcanzar un equilibrio al seleccionar el número óptimo de árboles y ajustar otros parámetros para evitar el sobreajuste. En nuestro caso queremos alcanzar un equilibrio entre, poder explicar la mayoría de casos, por anómalos que puedan llegar a ser, y no memorizarlos todos para que el Score obtenido sea 0 en la mayoría de ellos.

- **n_estimators**: Número de árboles con aumento de gradiente (Equivalente a 'num_round').
- **max_depth**: Profundidad máxima de un árbol.
- **eta (learning_rate)**: Tasa de aprendizaje de impulso.
- **gamma**: Reducción mínima de pérdidas requerida para realizar una nueva partición en un nodo hoja del árbol.
- **subsample**: Proporción de submuestra de la instancia de entrenamiento.
- **lambda**: Plazo de regularización L2 sobre ponderaciones. Incrementar este valor hará que el modelo sea más conservador.
- **min_child_weight**: Suma mínima de peso de instancia necesaria en la siguiente ronda.
- **Objective**: Indica la tarea de entrenamiento. Seleccionaremos $'count : poisson'$, ya que esta es la distribución de nuestra variable objetivo, evitando así valores negativos.

Las métricas que utilizaremos a la hora de seleccionar cual es el modelo más adecuado para nuestro cometido son las siguientes:

- **RMSE (Root Mean Square Error)**: La raíz del error cuadrático medio. Una medida de uso frecuente de las diferencias entre los valores predichos por un modelo y los valores observados.
- **MAE (Mean Absolute Error)**: El error absoluto medio es una medida de la diferencia entre dos variables continuas.
- **R2**: El coeficiente de determinación entre los valores reales y los predichos.
- **Accuracy**: El porcentaje de acierto del modelo si redondeamos las predicciones a sus valores discretos.

3.5.3 Explicabilidad del modelo con Shap

La forma más común de entender un modelo lineal es examinar los coeficientes de variables, que se dicen cuánto cambia la salida del modelo cuando las entran. Si bien los coeficientes son excelentes para saber qué sucederá cuando cambiemos el valor de una variable, por sí mismos, no son una excelente manera de medir la importancia general de una variable, porque el valor de cada coeficiente depende de su escala.

Los Shap Values son una herramienta poderosa y sofisticada para explicar las predicciones de los modelos de aprendizaje automático. Estos valores nos permiten comprender cómo cada variable de entrada contribuye a la predicción final del modelo. Proporcionan una explicación individualizada y cuantitativa de la importancia de cada característica en el resultado de la predicción.

La biblioteca Shap [17], diseñada para la explicación de modelos, utiliza la teoría de juegos para calcular los Shap Values. El enfoque principal de esta teoría es asignar una contribución justa a cada jugador en un juego cooperativo. En el contexto de los modelos de aprendizaje automático, consideramos el modelo como el "jugador" y las variables de entrada como los "jugadores" individuales. El objetivo es asignar a cada variable una contribución justa y cuantificable en la predicción final del modelo.

La idea fundamental detrás de los Shap Values es evaluar cómo cambia la salida del modelo cuando eliminamos o mantenemos cada característica en la predicción. Para hacer esto, se calcula la diferencia entre la salida del modelo con y sin la presencia de una característica en particular. Esta diferencia se conoce como el "efecto marginal" de la variable. Luego, se promedian todos los efectos marginales en diferentes combinaciones de variables, lo que nos da el valor Shap para cada variable.

El valor Shap de una variable puede interpretarse como la contribución promedio que esa variable proporciona a la diferencia entre la predicción del modelo y la predicción media. Si el valor Shap es positivo, indica que la variable aumenta la predicción en comparación con la predicción media, mientras que si es negativo, indica que la variable disminuye la predicción. De esta manera, los Shap Values nos permiten entender cómo cada variable influye en la salida del modelo y qué tan importante es para el resultado final.

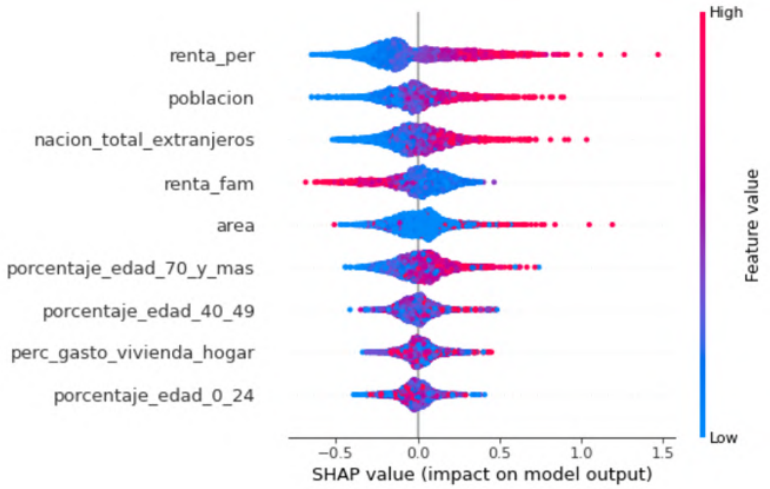


Figure 3.20: Ejemplo de summary plot de la librería Shap

Una forma común de visualizar los Shap Values es a través de un gráfico de resumen. En este gráfico, se ordenan las variables de mayor a menor importancia y se representan las contribuciones individuales de cada variable en forma de barras o puntos. Esto nos permite identificar las variables más relevantes y comprender cómo su presencia o ausencia afecta la predicción del modelo.

3.5.4 Resultados

Como primera iteración, decidimos entrenar los tres tipos de modelos que hemos mencionado anteriormente. Pero el desempeño de los dos menos complejos, Poisson Regressor y el Decission Tree era muy bajo comparado con el XGBoost, incluso sin realizar un ajuste fino de sus hiperparámetros. Podemos ver esta comparativa en el ejemplo de las farmacias. Fig.(3.21)

	Model	RMSE_train	RMSE_test	Poisson_dev_train	Poisson_dev_test	MAE_train	MAE_test	ACY_train	ACY_test	Options	Hyperparameters
0	Poisson	0.711219	0.701553	0.286495	0.280856	0.607918	0.604848	0.166667	0.200000	{'Preprocessor': None, 'K_folds': 3, 'Scoring': 'neg_mean_poisson_deviance', 'Coments': 'Y = Y + 1'}	{'Poisson__alpha': 0.1, 'Poisson__fit_intercept': True, 'Poisson__max_iter': 50}
1	DecisionTree	0.688647	0.694853	0.267675	0.275130	0.576204	0.586447	0.176529	0.203958	{'Preprocessor': None, 'K_folds': 3, 'Scoring': 'neg_mean_poisson_deviance', 'Coments': 'Y = Y + 1'}	{'DecisionTree__criterion': 'poisson', 'DecisionTree__max_depth': 3, 'DecisionTree__min_samples_leaf': 1, 'DecisionTree__min_samples_split': 2}
2	XGB	0.513114	0.704894	0.152750	0.280143	0.432496	0.573333	0.280798	0.235756	{'Preprocessor': None, 'K_folds': 3, 'Scoring': 'neg_mean_poisson_deviance', 'Coments': 'Y = Y + 1'}	{'XGB__colsample_bytree': 0.6, 'XGB__gamma': 0.5, 'XGB__max_depth': 3, 'XGB__min_child_weight': 1, 'XGB__n_estimators': 400, 'XGB__objective': 'count:poisson', 'XGB__subsample': 0.6}

Figure 3.21: Resultados comparativos de diferentes métricas en el entrenamiento para la predicción de farmacias por secciones censales. Modelos: Poisson, Decission Tree y XGBoost

Tras estos resultados, nuestro objetivo fue ajustar los modelos decantándonos entre la función Tweedie o Poisson. Como hemos mencionado anteriormente, no pretendemos que nuestro modelo sea el mejor y prediga con un error mínimo. Lo que buscamos es un modelo capaz de captar la tendencia general de los datos y poder con él, capturar aquellas secciones en las que se debería tener un mayor número de estos servicios y actualmente no se tiene.

Una vez hemos hecho el primer entrenamiento separando los datos en 80% para el entrenamiento y 20% para el test, se ha seleccionado el mejor conjunto de hiperparámetros y podemos proceder a entrenar este mismo modelo con el 100% de los datos para cada servicio. Si no hubiéramos hecho este último paso, habríamos obtenido una mejor puntuación en el 80% de las secciones censales que aleatoriamente hayan entrado en el conjunto de test. De este modo, entrenando con todos los datos, sabemos que ninguna sección está viéndose favorecida

Veamos cuales han sido los resultados obtenidos para los tres servicios esenciales de ejemplo que se hemos utilizado a lo largo del trabajo.

Farmacias

Modelo / Test Metrics	Poisson Deviance	MAE
XGBoost : Poisson	0.540	1.181
XGBoost : Tweedie	0.591	1.238

Figure 3.22: Tabla donde comparamos los resultados en test para el mejor modelo de XGBoost minimizando Tweedie o Poisson para la predicción de farmacias por sección censal

Vemos que para predecir el número de farmacias por cada sección censal, el mejor modelo ha sido obtenido minimizando la desviación de Poisson. La métrica más destacable ha sido el MAE, que pese a no ser un resultado bueno en general. En este caso concreto en el que acertar con exactitud el número de farmacias basándonos únicamente en las variables socioeconómicas es muy complicado, consideramos que un valor de 1.18 es aceptable. Recordemos de nuevo que no queremos obtener un modelo perfecto. Ya que en ese caso no detectaríamos anomalías en las secciones que no siguen la tendencia general, ya que el modelo obtendría bien esos casos menos frecuentes y no seríamos capaces de detectarlos.

Considerando ahora que el modelo ha predicho la cantidad de farmacias que cada sección debería tener, extraeremos el valor predicho por el modelo al valor que realmente tiene cada zona y de esta forma obtendremos la métrica que realmente nos importa, el *Score*.

Veamos una representación gráfica del valor real de farmacias por sección frente al valor predicho por el modelo. De este modo podremos visualizar todas aquellas que tengan exceso o ausencia en este servicio.

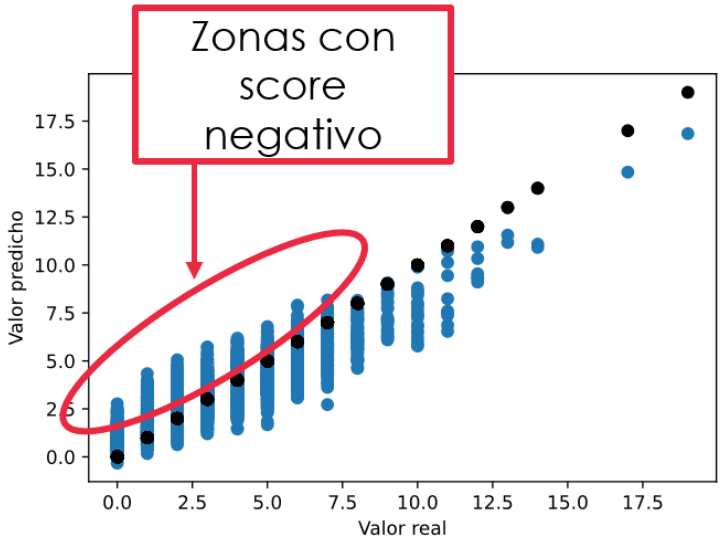


Figure 3.23: Valor real de farmacias por sección frente al valor predicho por el modelo. En negro tenemos la región donde ambos valores son iguales y en rojo los puntos de las secciones con *Score* negativo (ausencia de farmacias)

Aunque este modelo es el mejor que hemos obtenido, es evidente que no ha alcanzado un resultado perfecto, como se puede observar en la distribución no homogénea de los puntos alrededor de la región central. Para lograr un modelo más preciso, sería necesario considerar variables adicionales o información más es-

pecífica sobre cada servicio individual, ya que predecir la cantidad de farmacias únicamente basándonos en variables sociodemográficas no parece ser la solución óptima. Sin embargo, en nuestro caso, donde buscamos cuantificar la falta de servicios para los ciudadanos, en el análisis geográfico posterior veremos que estos resultados nos conducen a conclusiones válidas y significativas.

Antes de analizar zonas o secciones concretas, veamos cuales han sido las variables más importantes para el modelo. Como finalmente hemos optado por utilizar *XGBoost*, haremos uso del *summary plot* o *gráfico de resumen* de la librería *Shap*.

El gráfico de resumen muestra el impacto de cada variable en las predicciones del modelo de manera global. Para cada variable, se calcula el valor $SHAP$ para todas las instancias de datos y se representa gráficamente en un gráfico de puntos.

En el eje vertical del gráfico, se encuentran las variables del modelo ordenadas de mayor a menor importancia. La importancia se determina en función de la magnitud del valor $SHAP$ promedio de cada variable. Las variables más relevantes tendrán puntos más dispersados.

Es importante destacar que la importancia de las variables no se basa únicamente en la magnitud de sus valores $SHAP$, sino también en cómo interactúan con otras variables del modelo. Una variable puede tener un valor $SHAP$ individual alto, pero si su interacción con otras variables es baja, su importancia relativa puede ser menor en comparación con otras variables con una interacción más significativa.

En el eje horizontal vemos el valor $SHAP$ obtenido para cada entrada de la variable que corresponda. Además en escala de color vemos si las instancias de los datos se corresponden a valores grandes o pequeños en una escala diferente para cada variable.

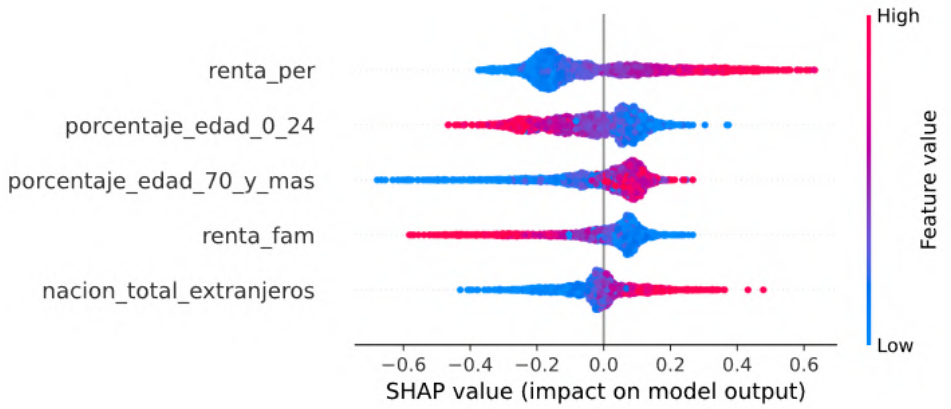


Figure 3.24: Resumen de los valores de contribución (*SHAP values*) para las variables predictoras utilizando el modelo XGBoost en el caso de farmacias. El gráfico muestra el impacto relativo de cada variable en las predicciones del modelo. Los puntos representan las observaciones individuales, mientras que la posición horizontal indica el valor de contribución (positivo o negativo)

Las conclusiones que se pueden sacar en este gráfico son que las tres variables más importante para que en una sección censal, el número de farmacias sea mayor es: que la renta per cápita sea alta, que el número de jóvenes sea bajo y que el número de personas de edad avanzada sea alto.

Esto solo habla de las predicciones del modelo, por lo que no podemos saber si por pocas farmacias que tenga una zona, esa zona tiene una escasez en este servicio. A lo mejor se trata de una sección con un número muy escaso de habitantes o está cercana a otra con un gran número de ellas. Por esta razón lo que debemos analizar es el *Score* obtenido al comparar las predicciones con el valor real.

Veamos cuales han sido los resultados obtenidos para el resto de servicios estudiados.

Centros educativos

Con el mismo criterio que tuvimos para el caso de las farmacias, esta vez el mejor modelo ha sido en el que se utilizó Tweedie como función objetivo, de nuevo el valor obtenido para el MAE es el más bajo, Fig (3.25). Pero de nuevo recalcamos que nuestro objetivo no es conseguir una predicción perfecta, sino modelar la tendencia general de los datos para ser capaces de capturar anomalías que detectaremos comparando con el valor real, en este caso el número de centros educativos en el interior de las secciones y sus alrededores.

Modelo / Test Metrics	Poisson Deviance	MAE
XGBoost : Poisson	1.219	1.789
XGBoost : Tweedie	**1.297**	**1.649**

Figure 3.25: Tabla donde comparamos los resultados en test para el mejor modelo de XGBoost minimizando Tweedie o Poisson para la predicción de centros educativos por sección censal

Si observamos el gráfico donde viene representada el valor de la variable objetivo predicha por el modelo contra el valor real, vemos que hay secciones que tienen hasta 25 centros educativos entre su interior y alrededores, por lo tanto es totalmente comprensible que hayamos obtenido un MAE de 1.84. De hecho podemos observar en la nube de puntos, Fig (3.26) que en las zonas con 17 centros educativos y más, siempre se ha cometido un error muy grande y nunca se ha predicho por encima del valor real. Esto es un buen indicativo, porque a priori parece que una sección censal con tantos centros educativos, no debería tener un valor negativo de *Score*, y esto indicará un exceso del servicio en cuestión.

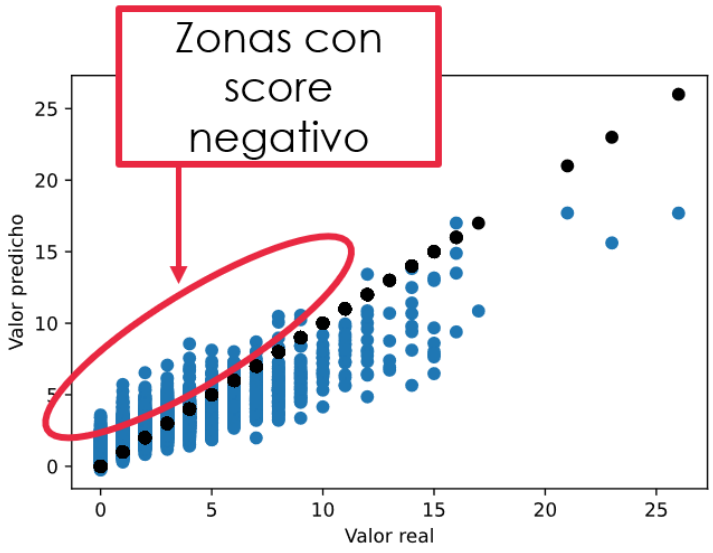


Figure 3.26: Valor real de centros educativos por sección frente al valor predicho por el modelo. En negro tenemos la región donde ambos valores son iguales y en rojo los puntos de las secciones con Score negativo (ausencia de farmacias)

Si analizamos las variables más importantes que arroja el *summary plot* de *Shap*, vemos como el area de las secciones aparece como variable más importante. El sentido que puede tener esto es que principalmente, las secciones con mayor extensión se encuentran en las zonas alejadas de Madrid, en municipios más rurales, donde suele haber menos densidad de centros educativos.

Las otras dos variables más importantes son, la renta per cápita y la renta familiar. Es curioso que cuanto mayor es la renta per cápita, mayor es la cantidad de colegios y cuanto mayor es la renta familiar, menor la cantidad de centros educativos. La única explicación que podemos encontrar de esto es que hay zonas donde las familiar tienen un mayor número de miembros y por eso la renta familiar, que es la dividida entre los miembros, es menor.

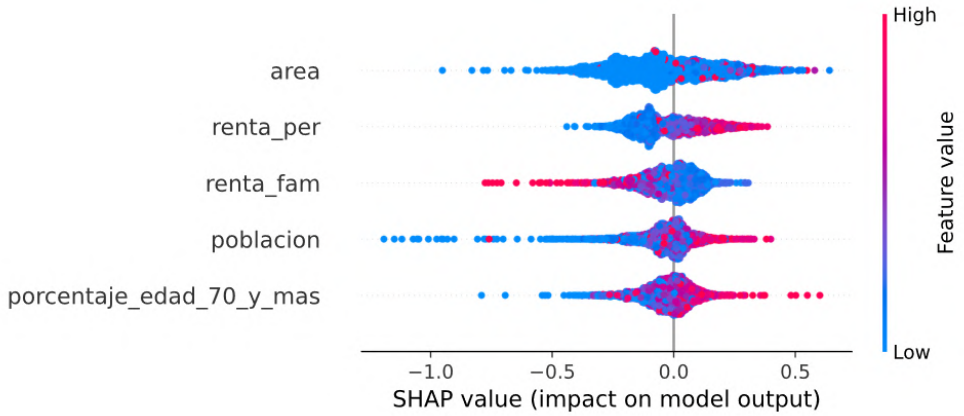


Figure 3.27: Resumen de los valores de contribución (SHAP values) para las variables predictoras utilizando el modelo XGBoost en el caso de centros educativos. El gráfico muestra el impacto relativo de cada variable en las predicciones del modelo. Los puntos representan las observaciones individuales, mientras que la posición horizontal indica el valor de contribución (positivo o negativo)

Puntos de venta y recarga de la Tarjeta de Transporte Público, TTP

Por último analizaremos los resultados obtenidos por el modelo en el caso de los puntos de venta y recarga de la tarjeta de transporte público.

El resultado obtenido ha sido muy similar al previamente comentado de los centros educativos, donde el modelo con la función Tweedie como objetivo ha sido el más satisfactorio.

A diferencia del caso anterior, ahora no tenemos secciones con más de 15 ubicaciones, por lo que deberíamos haber obtenido un error similar al de las farmacias, cercano el 1, en el error medio absoluto.

Modelo / Test Metrics	Poisson Deviance	MAE
XGBoost : Poisson	0.648	1.068
XGBoost : Tweedie	**0.716**	**1.029**

Figure 3.28: Tabla donde comparamos los resultados en test para el mejor modelo de XGBoost minimizando Tweedie o Poisson para la predicción de puntos de recarga de la TTP por sección censal

De hecho vemos en la figura 3.28 como para la gran mayoría de zonas se ha predicho un valor por encima del que realmente tienen. Podría ser porque al comparar secciones censales en el interior de Madrid, con secciones en pueblos de la comunidad autónoma, estamos comparando zonas muy distintas. Pero de nuevo, será el análisis del *Score* el que determine si en cada sección tenemos un exceso o ausencia del servicio en cuestión.

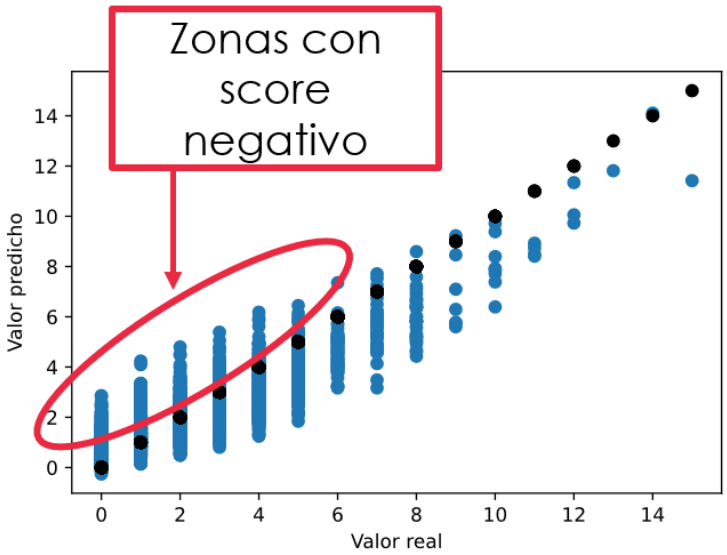


Figure 3.29: Valor real de puntos de recarga de la TTP por sección frente al valor predicho por el modelo. En negro tenemos la región donde ambos valores son iguales y en rojo los puntos de las secciones con Score negativo (ausencia de farmacias)

Esta vez, el gráfico *summary plot* de *Shap* pone al porcentaje total de ex-

tranjeros como variable más importante, seguidas de la renta per cápita y del porcentaje de jóvenes. En resumen podemos decir que las zonas con mayor cantidad de puntos de recarga de la tarjeta de transporte público son aquellas con un porcentaje alto de inmigración, renta alta y bajo porcentaje de jóvenes. Siendo esto último un tanto sorpresivo.

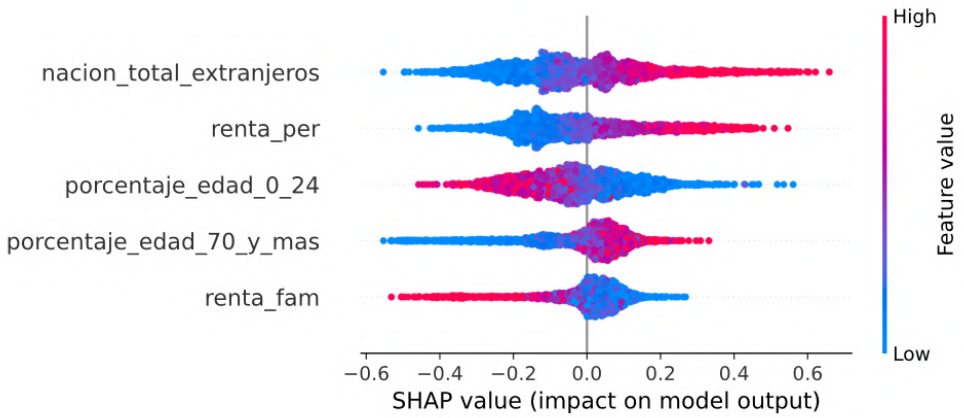


Figure 3.30: Resumen de los valores de contribución (SHAP values) para las variables predictoras utilizando el modelo XGBoost en el caso de la TTP. El gráfico muestra el impacto relativo de cada variable en las predicciones del modelo. Los puntos representan las observaciones individuales, mientras que la posición horizontal indica el valor de contribución (positivo o negativo)

3.6 Dashboard de visualización de resultados, PowerBi

Los resultados de este proyecto se presentan en un Dashboard de PowerBi. Con el fin de mostrar de forma visual e interactiva las secciones censales con su Score obtenido para los diferentes servicios. Para comprender estos resultados, también se mostrarán las características sociodemográficas más importantes de cada una de estas.

Las partes principales que componen el dashboard son las siguientes:

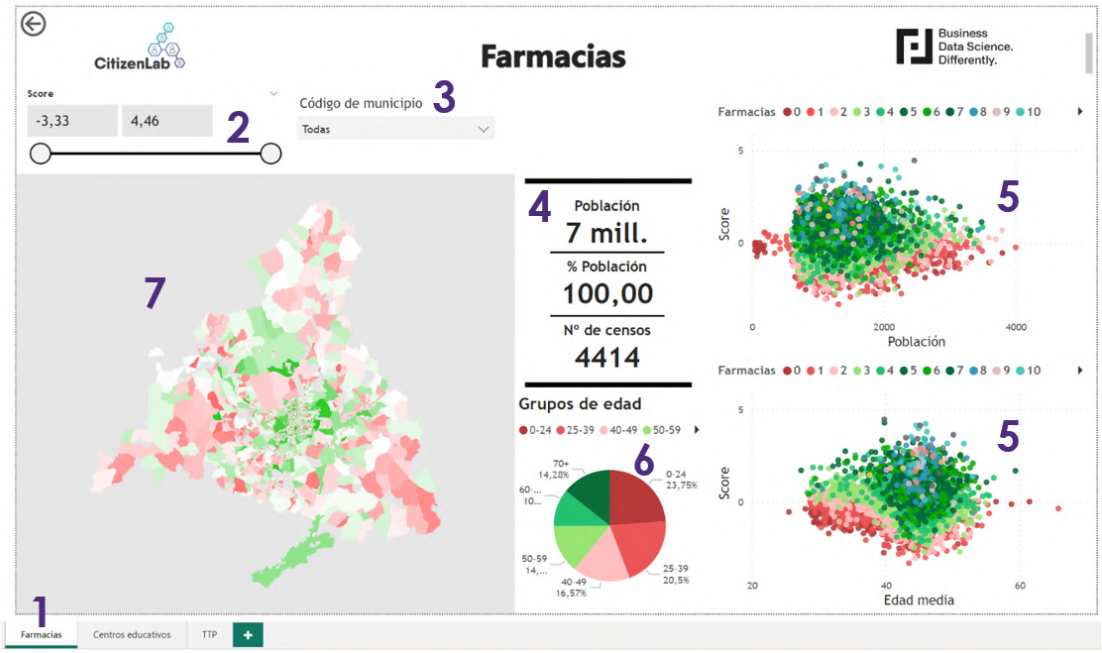


Figure 3.31: Dashboard en Power Bi

- **1. Pestañas con servicios** En la parte inferior podemos ver las diferentes ventanas. Cada una de ellas nos lleva al estudio de cada servicio: oficinas de farmacia, centros educativos o puntos de venta de la tarjeta de transporte público
- **2. Filtro de Score** En la parte superior encontramos el filtro interactivo del *Score*. En él podemos seleccionar el rango de *score* que deseamos visualizar. Recodemos que los Scores negativos indican escasez del servicio que estemos visualizando. *Score* de -2.9 indica una escasez de casi 3 ubicaciones que ofrezcan dicho servicio.
- **3. Filtro por municipio** En este desplegable podemos seleccionar un único municipio por el que filtrar. La utilización de este, aumenta considerablemente el rendimiento, ya que la cantidad polígonos a renderizar por el mapa es mucho menor.
- **4. Cantidad de afectados por los filtros** En este cuadro aparecen el número de personas afectadas por los filtros que hayamos seleccionado, el porcentaje que estos representan del total de la población de Madrid y el número de secciones censales.
- **5. Visualización de variables más importantes** Estas dos gráficas de puntos muestran la distribución del *Score* de todas las zonas censales re-

specto a las dos variables más importantes para la predicción de cada servicio. Además, el color de cada punto indica la cantidad de ubicaciones de un servicio que tiene cada zona, pudiendo clicar en la leyenda para filtrar por este valor.

- **6. Distribución de edades de zonas afectadas** En este gráfico circular podemos ver la distribución de edad promedio de todas las zonas afectadas por los filtros.

- **7. Mapa interactivo** En el mapa se muestra en escala de color de verde a rojo, pasando por blanco, el *Score* de cada sección censal. Donde los colores rojizos muestran *Score* negativo y los verdes, positivo. Además si desplazamos el ratón por encima de las diferentes zonas, se muestra información como el código identificador de la sección censal, el número de servicios en su interior, el número de servicios cercanos, la cantidad de habitantes y el valor de las dos variables más representativas, mostradas ya en las gráficas de puntos.

3.7 Análisis de casos concretos

Hasta ahora, hemos hablado de los datos que componen esta aplicación, sus fuentes, cómo se infirieron las variables significativas y cómo se los interpretaron y visualizaron. A partir de esta sección vamos a hablar de los resultados extrayendo algunos ejemplos. Oficinas de farmacia

3.7.1 Farmacias

Analizaremos el caso concreto de las oficinas de farmacia para extraer conclusiones haciendo uso del dashboard de Power Bi. Sería interesante ver por ejemplo las zonas censales que presenten escasez de farmacias, es decir, cuyo Score sea negativo. Para ello filtraremos el score para valores menores de cero.

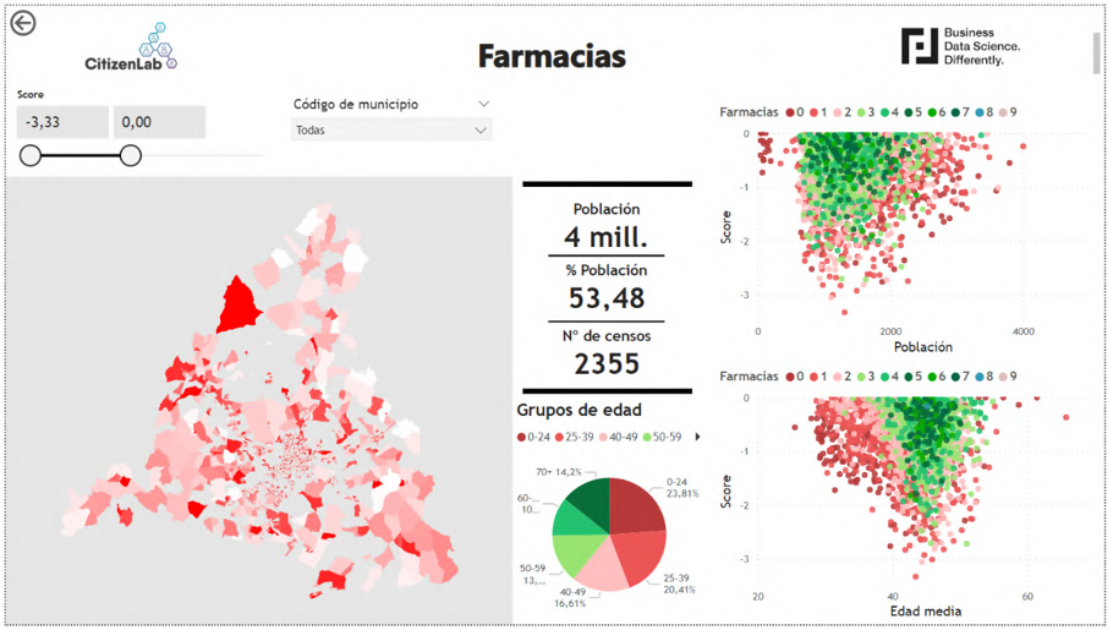


Figure 3.32: Dashboard en Power Bi filtrando las secciones con scores negativos en farmacias.

En la zona central podemos ver cuanta población se ve afectada por el filtro que hemos seleccionado. Un total de 2355 zonas censales tienen escasez de farmacias, donde habita el 53Centrándonos en el mapa vemos como no hay una clara distribución geográfica de estas zonas. Tanto fuera como en el interior de la ciudad de Madrid vemos lugares afectados. Las dos gráficas de la derecha muestran las dos variables más importantes según el explicador Shap, ambas frente al Score. En ellas podemos ver como los Scores más bajos se dan en zonas con unos 1500 habitantes y una media de edad de unos 45 años. Si reducimos aún más el filtro del Score, podemos mostrar únicamente las 20 zonas más afectadas.

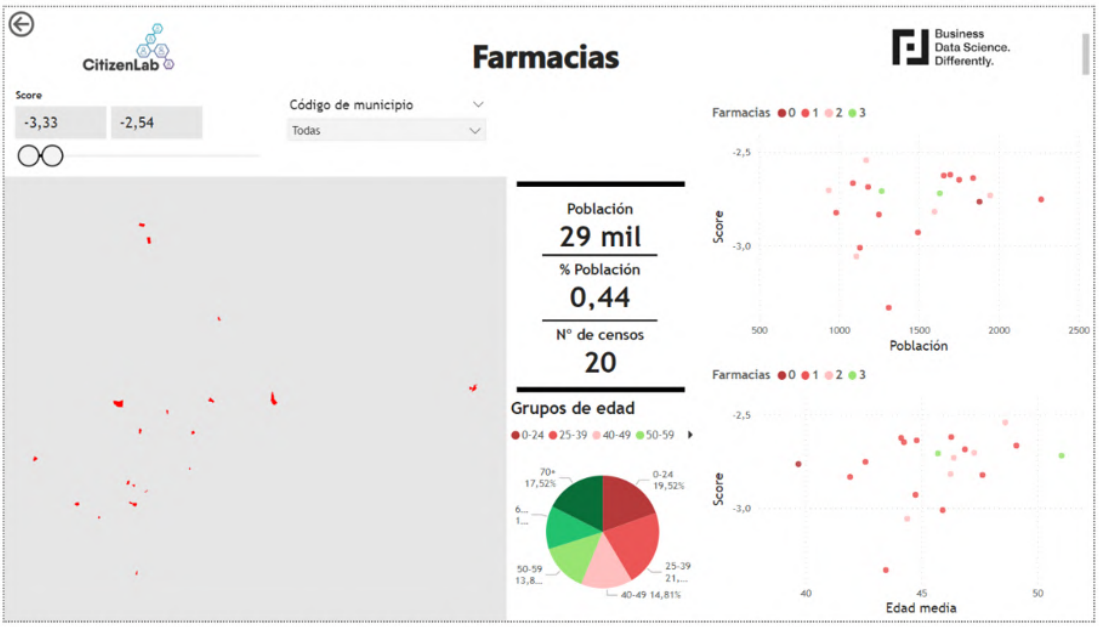


Figure 3.33: Dashboard en Power Bi filtrando las 20 secciones con menor Score en farmacias.

Ahora vemos como solo hay 20 zonas censales afectadas, con un total de 29 mil habitantes entre todas. En el filtro de las figuras vemos como casi ninguna de estas ubicaciones tiene más de 3 farmacias cercanas Esta vez el mapa sí que nos muestra como estas zonas se encuentran en la ciudad de Madrid únicamente. Si en vez de manipular el filtro del Score, analizamos visualmente concentraciones de zonas con Scores negativos (rojo), encontramos algunas agrupaciones. La agrupación de la figura inferior se corresponde al municipio de Parla y podemos ver en ella, como hacia el sur del municipio, las puntuaciones en cuanto a número de farmacias, son en general, bajas.

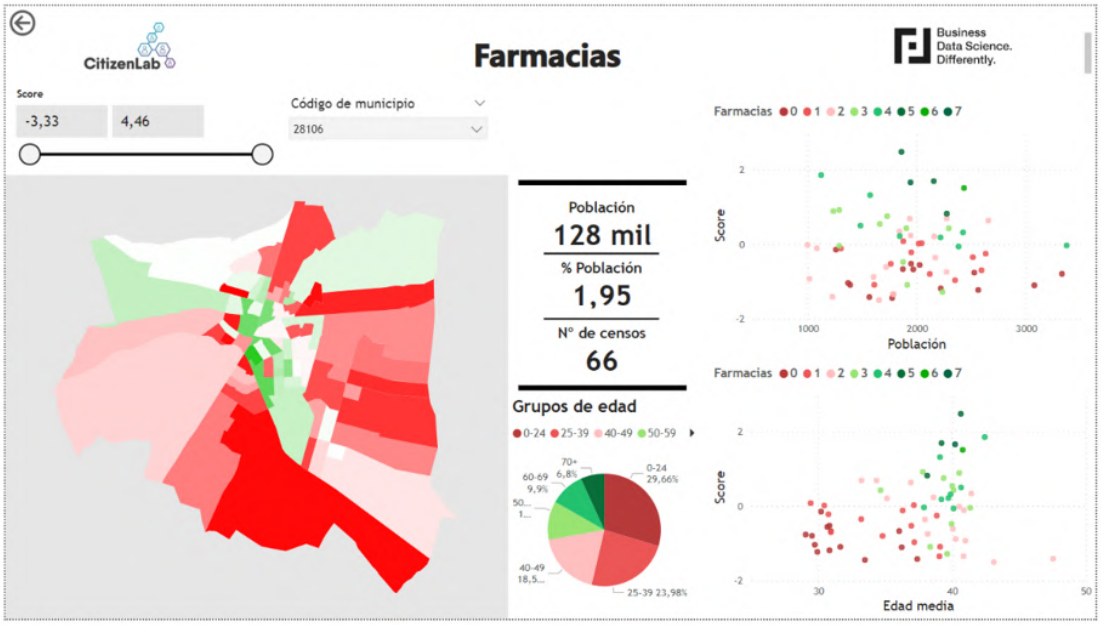


Figure 3.34: Dashboard en Power Bi filtrando el municipio de Parla

Viendo una representación, como la mostrada debajo, donde los puntos son las farmacias y el mapa de calor también indica dicha magnitud. Podemos observar cómo efectivamente, la zona sur de Parla, tiene una clara escasez de farmacias. Esto podría ser porque dicha zona, no está densamente poblada o incluso no es una zona urbanizada.

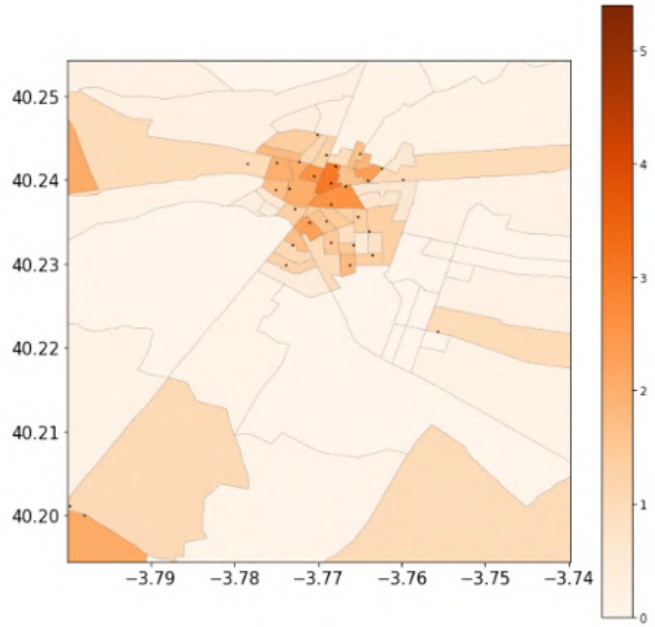


Figure 3.35: Mapa del municipio de Parla donde vemos las farmacias como puntos.

Para analizar el problema desde esta perspectiva, necesitamos una imagen satélite que muestre la verdadera naturaleza de estas secciones censales.

Figure 3.36: Imagen satélite de Parla

Todas las secciones pertenecientes a esta zona sur están constituidas en su

mayoría por urbanizaciones de bloques, adosados o casas individuales. Por lo tanto, podemos afirmar finalmente que toda esta región presenta una escasez en cuanto a la cantidad de farmacias se refiere.

Si ajustamos ahora el filtro del Score y analizamos las zonas con mayor ausencia de farmacias, podemos analizar la situación concreta de cada sección censal. Por ejemplo, la sección con código 28-07908008, tiene un Score de -2.93, lo cual indica que como norma general, una zona con sus características, necesitaría unas 3 farmacias.

Figure 3.37: Dashboard en Power Bi filtrando una de las secciones con peor Score

Si analizamos la zona desde una imagen vía satélite, efectivamente confirmamos que pese a tener una farmacia en su interior, sus alrededores presentan una mayor densidad de farmacias.

Figure 3.38: Imagen satélite de una de las secciones con peor Score en farmacias.

3.7.2 Centros educativos

De la misma manera que hemos analizado la escasez de farmacias en algunas zonas concretas de la comunidad, lo haremos ahora para los centros educativos. Seleccionaremos la sección 28-10001003, una de las que presenta un Score más negativo, -4.72. Vemos como esta zona tiene un gran número de habitantes menores de 24 años, un total de 785.

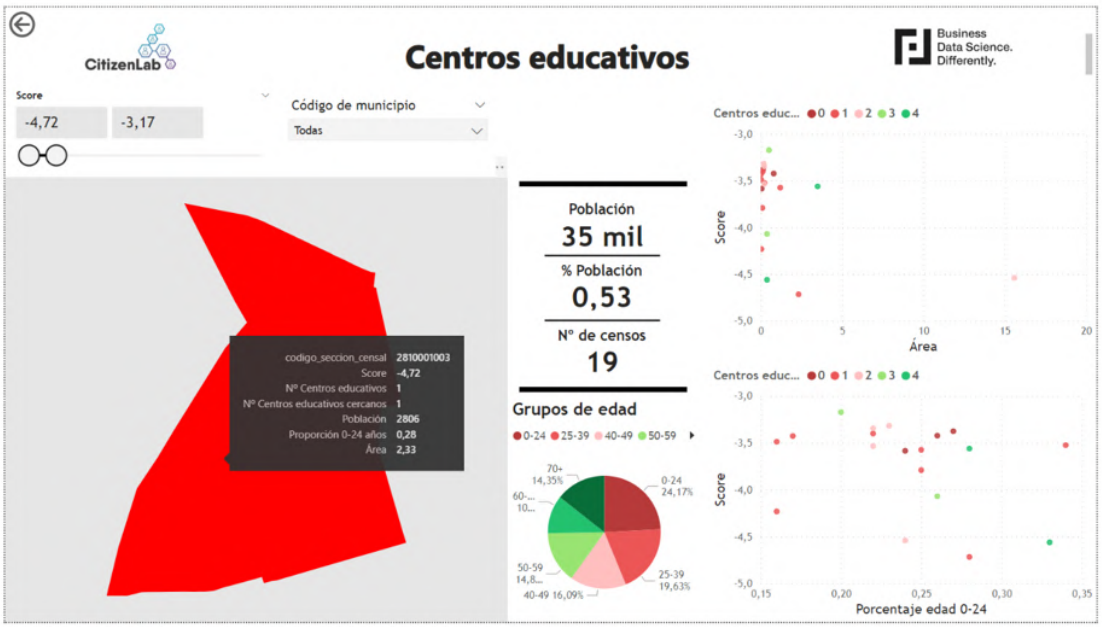


Figure 3.39: Dashboard en Power Bi filtrando una de las secciones con peor Score en centros educativos.

Si analizamos esta zona con una imagen de Google Maps, podemos ver como se trata de una zona de urbanizaciones en la que tenemos únicamente un colegio privado. El único centro público que se encuentra relativamente cerca, se ubica a más de medio kilómetro de distancia.

Figure 3.40: Imagen satélite de una de las secciones con peor Score en centros educativos.

3.7.3 Puntos de recarga de la TTP

Hagamos ahora un análisis similar para los puntos de venta y recarga de la tarjeta de transporte público. Una de las secciones con mayor escasez de dicho servicio, es la sección con código 28-13301001. Esta vez si que se trata de una zona ubicada en los límites de la comunidad autónoma. Analicemos mejor con una imagen el tipo de región que tenemos.

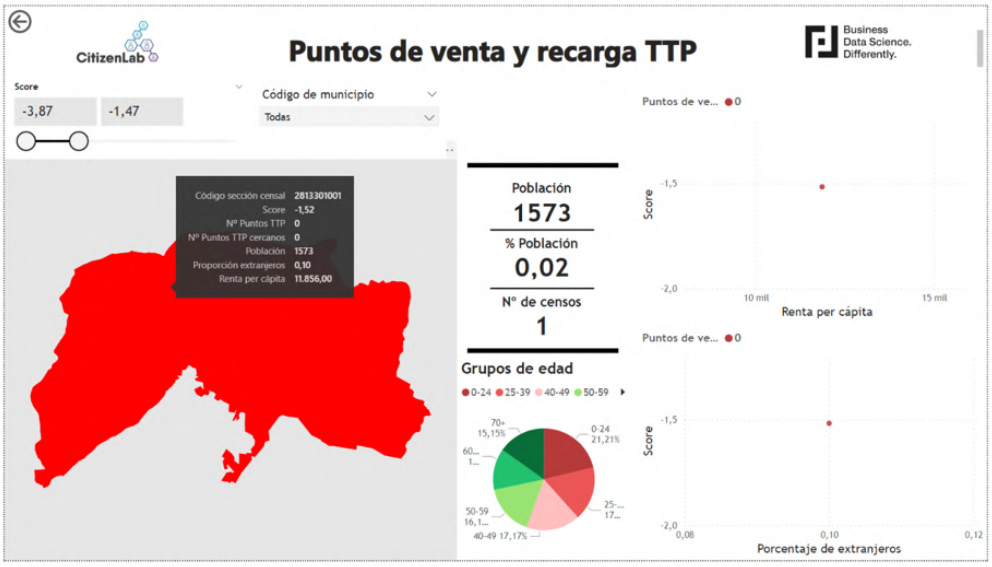


Figure 3.41: Dashboard en Power Bi filtrando una de las secciones con peor Score en puntos de recarga de la TTP.

Vemos en la imagen inferior, como la sección se compone mayoritariamente de casas individuales y zonas sin urbanizar. Salvo dos pequeñas partes en la zona sur, que recogen parte de las poblaciones de San Martín de Valdeiglesias y Pelayos de la Presa.

Figure 3.42: Imagen satélite de una de las secciones con peor Score en TTP. [2]

Si visualizamos las líneas de transporte público en el mapa, vemos como únicamente tenemos una línea de autobús que recorre dichas localidades. Con cerca de 12 paradas en las inmediaciones de la sección censal que estamos tratando.

Figure 3.43: Imagen satélite de una de las secciones con peor Score en TTP y las paradas de la linea de autobús más cercana.

Además, en las paradas de líneas de autobús no se puede adquirir ni recargar la tarjeta de transporte público. Para ello, los habitantes de esta zona han de desplazarse hasta las ubicaciones destinadas a ofrecer dicho servicio, las cuales podemos ver en el mapa inferior. Tres únicas zonas repartidas en las dos localidades anteriormente mencionadas. Todas muy distantes para los habitantes cuya vivienda se encuentra próxima al Embalse de San Juan.

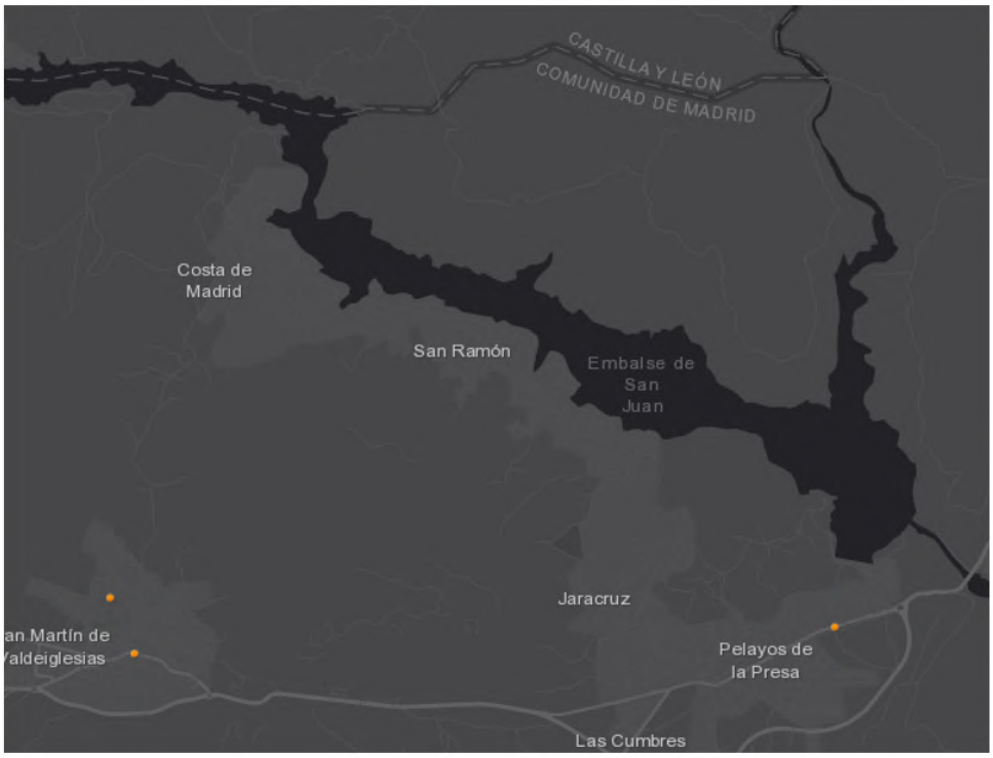


Figure 3.44: Imagen satélite de una de las secciones con peor Score en TTP y los puntos de recarga de la TTP más cercanos.

Chapter 4

Conclusiones

En este proyecto se ha abordado la detección de necesidades en servicios esenciales al ciudadano mediante el uso de técnicas de aprendizaje automático y análisis geoespacial. Se han desarrollado modelos de predicción para tres servicios: farmacias, centros educativos y puntos de venta y recarga de la Tarjeta de Transporte Público (TTP).

Los resultados obtenidos muestran que los modelos de XGBoost, utilizando las funciones de pérdida de Tweedie o Poisson, han sido los más efectivos en la predicción de los servicios estudiados. Estos modelos han demostrado ser capaces de captar la tendencia general de los datos y detectar anomalías en las secciones censales donde existe una falta o exceso de los servicios.

Es importante tener en cuenta que los modelos no buscan obtener una predicción perfecta, sino capturar la tendencia general y detectar casos atípicos. Esto es especialmente relevante en el contexto de detección de necesidades, ya que permite identificar aquellas zonas donde se debería tener un mayor número de servicios y actualmente no se tienen.

Las variables socioeconómicas han demostrado ser relevantes en la predicción de los servicios estudiados. En el caso de las farmacias, se observó que la renta per cápita, el número de jóvenes y el número de personas de edad avanzada fueron las variables más importantes. Para los centros educativos, el área de las secciones, la renta per cápita y la renta familiar fueron las variables más relevantes. Y en el caso de los puntos de venta y recarga de la TTP, el porcentaje total de extranjeros, la renta per cápita y el porcentaje de jóvenes fueron las variables más influyentes.

A partir de los resultados obtenidos, se pueden extraer conclusiones y recomendaciones para la planificación y asignación de servicios esenciales. Por ejemplo, se puede identificar la necesidad de aumentar el número de farmacias en zonas con baja renta per cápita y alta proporción de personas de edad avanzada. En el caso de los centros educativos, se puede considerar la apertura de nuevos colegios en zonas con mayor área y menor renta per cápita. Y en cuanto a los puntos de

venta y recarga de la TTP, se pueden priorizar las áreas con alta proporción de población extranjera y baja presencia de jóvenes.

Es importante tener en cuenta que estos resultados son específicos para el contexto y los datos utilizados en este proyecto. Cada localización geográfica y conjunto de datos pueden presentar características diferentes, por lo que se recomienda adaptar y validar los modelos en cada caso particular.

4.1 Trabajos futuros

Incorporación de nuevos servicios: Tal y como se ha mencionado, este trabajo muestra una forma de analizar y obtener predicciones para que en un futuro puedan incorporarse nuevos servicios esenciales y con este trabajo se tenga un punto desde el que partir para obtener resultados de la misma manera.

Inclusión de variables adicionales: En este trabajo, se han utilizado datos socioeconómicos y demográficos para predecir la demanda de servicios esenciales. Sin embargo, existen otras variables que podrían ser relevantes, como la infraestructura existente, los patrones de movilidad y las preferencias individuales.

Evaluación del impacto de las intervenciones: Una vez identificadas las áreas con déficit o exceso de servicios, sería relevante evaluar el impacto de las intervenciones realizadas para abordar estas necesidades. Se podrían realizar estudios comparativos para analizar cómo las decisiones de planificación y asignación de recursos influyen en la mejora de la calidad de vida de los ciudadanos y en la eficiencia de los servicios esenciales.

Incorporación de la participación ciudadana: La participación ciudadana desempeña un papel crucial en la planificación urbana y en la identificación de necesidades. Sería interesante explorar en futuros trabajos cómo involucrar de manera activa a los ciudadanos en la recolección de datos y en la toma de decisiones relacionadas con los servicios esenciales. Esto podría incluir el uso de aplicaciones móviles, encuestas en línea o sistemas de retroalimentación ciudadana para recopilar información y opiniones directamente de los usuarios de los servicios.

Incorporación de áreas homogéneas de análisis: Además de las secciones censales utilizadas en este trabajo, se propone considerar la incorporación de áreas más homogéneas en términos de características socioeconómicas y demográficas. Esto permitiría obtener análisis más detallados y precisos sobre las necesidades específicas de cada área, lo que facilitaría la planificación y asignación de recursos de manera más efectiva. La segmentación de la ciudad en áreas homogéneas también podría ser útil para identificar patrones y tendencias en la demanda de servicios, así como para adaptar las intervenciones de manera más localizada y personalizada.

Chapter 5

Enfoque Alternativo Basado en Malla Hexagonal

5.1 Motivación y objetivo del piloto

Como complemento al análisis tradicional basado en secciones censales, se ha desarrollado un piloto experimental con el objetivo de evaluar un enfoque más granular y espacialmente representativo para la asignación de necesidades ciudadanas. El procedimiento convencional presenta ciertas limitaciones, particularmente en áreas con baja densidad de población o geometrías censales que abarcan grandes superficies con zonas deshabitadas. Para solventar estas deficiencias, se ha diseñado una metodología que traslada la información sociodemográfica desde las secciones censales a unidades espaciales más detalladas: primero a edificios residenciales y, posteriormente, a una malla hexagonal de resolución homogénea.

5.2 Metodología del piloto

El proceso se ha estructurado en varias fases consecutivas:

1. Carga de geometrías censales

Se parte de la base de datos censal de Madrid, que proporciona indicadores como renta per cápita, edad media o población total, junto con los polígonos de las secciones censales. Esta información constituye el punto de partida del análisis sociodemográfico.

2. Identificación de edificios residenciales

Utilizando la API de OpenStreetMap [18] a través de la librería `osmnx`, se han descargado los edificios residenciales dentro del área de estudio. Posteriormente, se han filtrado aquellos etiquetados explícitamente como viviendas mediante las etiquetas `residential`, `apartments`, `house`, entre otras. Esta operación permite reducir el ruido introducido por estructuras no habitadas o infraestructuras industriales.

3. Redistribución de los datos sociodemográficos a edificios

Mediante una intersección espacial (*spatial join*) se ha asignado a cada edificio la información sociodemográfica correspondiente a la sección censal en la que se encuentra. Las variables intensivas (como renta per cápita) se han transferido directamente, mientras que las variables extensivas (como población) se han redistribuido entre los edificios proporcionalmente. Esta etapa evita asignar población a zonas donde no existen viviendas, mejorando la representación espacial de los datos.

4. Agregación sobre malla hexagonal

A partir del polígono de la Comunidad de Madrid, se ha generado una malla de hexágonos con resolución 9 del sistema H3 [19], con aproximadamente 0.2 km^2 por celda. Esta malla permite una partición uniforme del espacio, independientemente de los límites administrativos. Se ha proyectado la información de los edificios sobre los hexágonos, agregando de nuevo las variables intensivas y extensivas mediante sumas y medias ponderadas.

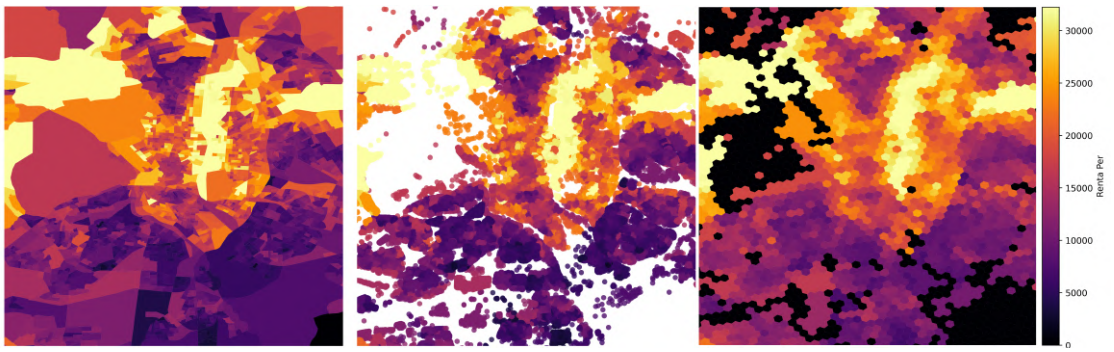


Figure 5.1: Distribución de la renta per cápita: secciones censales (izquierda), edificios residenciales (centro) y malla hexagonal (derecha). Se observa una mejora progresiva en la granularidad y definición espacial de los datos.

5. Incorporación de servicios urbanos

Mediante la misma API de OpenStreetMap se han extraído puntos de interés urbano como farmacias, supermercados, colegios o bancos. Aunque esta fuente presenta ciertas limitaciones en cuanto a cobertura y actualización, resulta útil en el contexto del piloto por ser abierta y de libre acceso. Los puntos han sido asignados a los hexágonos en función de su ubicación geográfica.

6. Modelado y estimación de necesidades

Por cada tipo de servicio se ha entrenado un modelo de regresión utilizando `XGBoost` con objetivo `count:poisson`, replicando la lógica de predicción usada anteriormente con secciones censales. Cada modelo estima el número esperado de servicios por hexágono a partir de las variables sociodemográficas. La diferencia entre el valor real y el predicho se ha utilizado para construir un *score* de carencia o suficiencia por área.

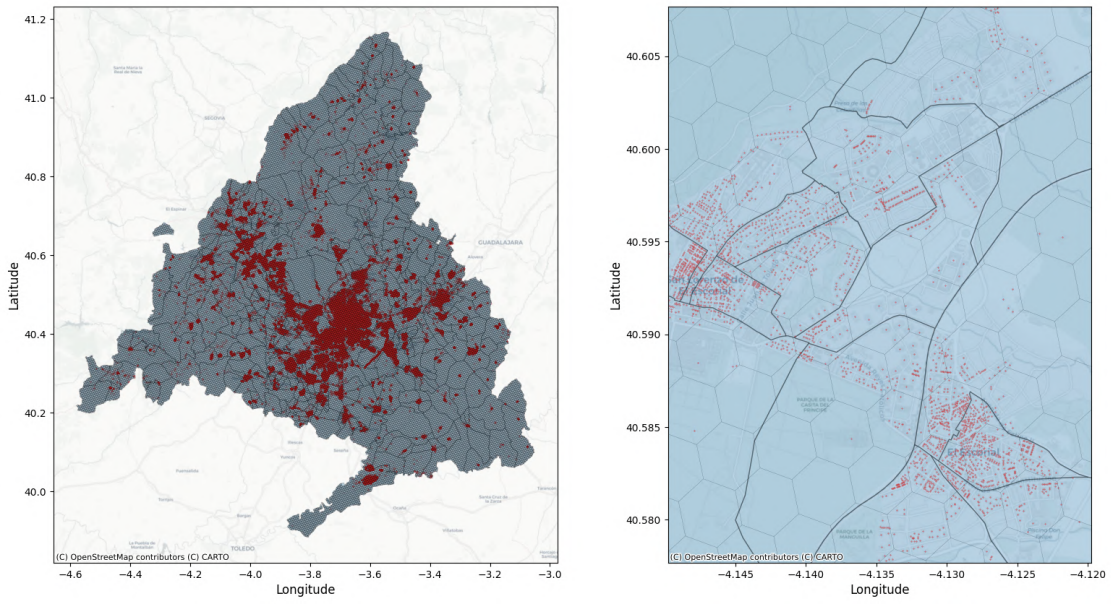


Figure 5.2: Visualización de secciones censales, edificios y hexágonos en dos escalas: Comunidad de Madrid (izquierda) y detalle en San Lorenzo del Escorial (derecha). Se aprecia cómo la malla hexagonal permite descartar zonas deshabitadas y enfocar el análisis en las áreas efectivamente residenciales.

7. Evaluación de los modelos por tipo de servicio

Para valorar el rendimiento de los modelos entrenados para cada tipo de servicio, se han analizado diversas métricas como el número medio de servicios por hexágono (*mean_count*), el número máximo de servicios en un mismo hexágono, la proporción de celdas sin presencia del servicio (*zero_percentage*) y los errores de predicción en entrenamiento y test (RMSE). En la Figura 5.3 se presenta un resumen visual con los resultados obtenidos:

place_type	mean_count	max_count	non_zero_count	zero_count	imbalance_ratio	zero_percentage	train_rmse	test_rmse
bakery	0.1002	6	528	6497	0.9248	92.4840	0.2637	0.3852
bank	0.2562	11	875	6150	0.8754	87.5445	0.6081	0.7468
dentist	0.0994	7	486	6539	0.9308	93.0819	0.3080	0.3182
fuel	0.0757	4	459	6566	0.9347	93.4662	0.2600	0.3357
greengrocer	0.0938	10	449	6576	0.9361	93.6085	0.2936	0.3443
kindergarten	0.1768	8	988	6037	0.8594	85.9359	0.4123	0.4258
optician	0.0571	5	296	6729	0.9579	95.7865	0.2117	0.2906
pharmacy	0.3601	8	1591	5434	0.7735	77.3523	0.4315	0.5057
school	0.3664	9	1638	5387	0.7668	76.6833	0.6380	0.7654
supermarket	0.2279	8	1150	5875	0.8363	83.6299	0.4405	0.5086

Figure 5.3: Resumen de métricas por tipo de servicio: distribución, desequilibrio (*imbalance*) y rendimiento del modelo (RMSE). Los colores indican mejores valores en verde y peores en rojo, en función del contexto de cada métrica.

Uno de los factores clave a tener en cuenta es la distribución espacial del servicio. Servicios como **pharmacy**, **school** y **supermarket** muestran una cobertura bastante equilibrada, con una proporción de hexágonos sin servicio inferior al 85% y una buena presencia en múltiples áreas (*non_zero_count* alto). Esta distribución homogénea facilita el aprendizaje del modelo, y se refleja en un buen desempeño predictivo tanto en entrenamiento como en test.

En cambio, servicios como **bank** y **greengrocer** presentan un patrón más concentrado, con valores máximos muy elevados (11 y 10 respectivamente) en ciertos hexágonos. Esto sugiere una alta desigualdad espacial, donde algunos núcleos urbanos concentran muchas instancias del servicio y otras zonas carecen totalmente de él. Esta heterogeneidad supone un reto para el modelo, ya que debe aprender una distribución muy sesgada.

Pese a ello, los errores de predicción en general son razonablemente bajos. Modelos como los de **optician** y **dentist** obtienen RMSE por debajo de 0.32, lo cual es notable dadas sus bajas tasas de presencia. Por otro lado, servicios como **school** y **bank** presentan errores algo mayores, con RMSE en test por encima de 0.74, lo que puede estar asociado a su mayor complejidad espacial o concentración en pocos puntos.

Un aspecto positivo es que en ninguno de los modelos se observa un sobreajuste evidente: las diferencias entre el error en entrenamiento y en test son muy pequeñas en todos los casos. Esto indica que los modelos generalizan bien y no están memorizando los datos de entrenamiento, a pesar del desequilibrio de clases existente en muchos de ellos.

En conjunto, estos resultados validan el enfoque adoptado y permiten identificar qué servicios están mejor modelados y cuáles podrían requerir ajustes metodológicos adicionales (como más datos, técnicas de balanceo o enfoques jerárquicos) en futuras versiones del sistema.

5.3 Ventajas y limitaciones del enfoque hexagonal

La principal ventaja de esta técnica es su capacidad para adaptar el análisis a la realidad espacial del territorio. Al eliminar zonas sin edificaciones del modelo, se reducen los falsos positivos de necesidad en áreas sin población. Además, la homogeneidad de tamaño de los hexágonos evita sesgos provocados por secciones censales de geometría irregular.

No obstante, también se identifican ciertas limitaciones. En zonas densamente urbanas donde las secciones censales ya eran muy pequeñas, la agregación en hexágonos puede resultar en celdas sobrepobladas. Reducir la resolución del H3 mejoraría la precisión, pero conllevaría un aumento considerable del coste computacional. Asimismo, la dependencia de OpenStreetMap introduce incertidumbre sobre la completitud de los servicios geolocalizados.

5.4 Despliegue interactivo: aplicación en Streamlit

Como parte del piloto, se ha desarrollado una aplicación interactiva en `Streamlit` que permite explorar los resultados de forma ágil y personalizada. A diferencia de soluciones como Power BI, que presentaban limitaciones de rendimiento y flexibilidad, esta implementación permite ajustar dinámicamente umbrales, visualizar áreas más necesitadas y explorar variables sociodemográficas clave por hexágono.

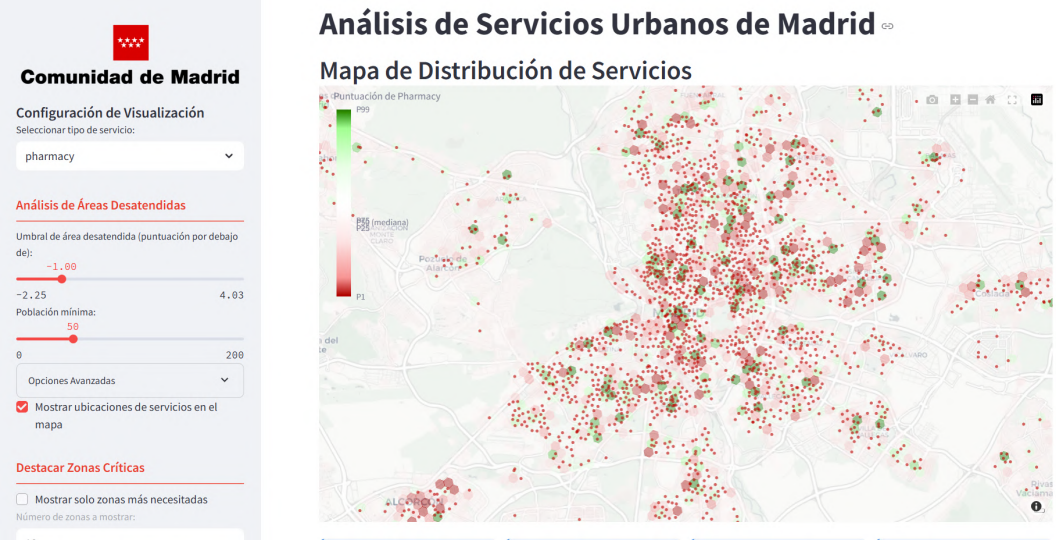


Figure 5.4: Interfaz principal de la aplicación desarrollada en Streamlit. Permite explorar la puntuación de necesidad por tipo de servicio, así como acceder a métricas, filtros y visualizaciones dinámicas.

5.5 Conclusiones del piloto

Este experimento metodológico ha demostrado el potencial de combinar geometrías geoespaciales más detalladas con técnicas de modelado predictivo. La combinación de edificios reales y malla hexagonal mejora la precisión espacial del análisis y proporciona una base más sólida para una futura planificación urbana basada en datos. Aunque aún presenta desafíos, esta aproximación abre la puerta a sistemas de evaluación de necesidades ciudadanas más precisos, adaptables y centrados en el territorio real.

Appendix A

Google Maps Scraper

```
from selenium import webdriver
#from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import time
import pandas as pd
from selenium.webdriver.common.keys import Keys
import numpy as np
from selenium.webdriver.chrome.options import Options
import warnings
warnings.filterwarnings('ignore')
import folium
from folium import Choropleth, Circle, Marker
from folium.plugins import HeatMap, MarkerCluster
import requests
from bs4 import BeautifulSoup
import logging
import datetime
from  datetime import datetime
#import selenium set proxy
#from selenium.webdriver.common.proxy import Proxy, ProxyType
import random


def __extract_phone(row):
    try:
        return '+' + str(row).split('+')[1].split("'")[0]
    except:
        return np.nan

def __get_lat(gmaps_link):
    return gmaps_link.split('!3d')[1].split('!4d')[0]

def __get_lon(gmaps_link):
    return gmaps_link.split('!4d')[1].split('!16')[0]


class GoogleMapsServiceScraper:
    """Class to scrape Google Maps for a given service in a given area and zoom level
        The information scraped is the Name, Gmaps URL, Rating, Reviews, Address,
             Type of place, Phone number and Latitude and Longitude.
    Args:

        coord_array (list): List of coordinates to scrape
        zoom (int): Zoom level
        service (str): Service to scrape
        logger (logging.Logger): Logger to log the process
        data_path (str): Path to save the scraped data
        proxy_list (list): List of proxies to use

    Returns:
        df (pd.DataFrame): DataFrame with the scraped data

    Example:

        coord_array = [[40.4201179, -3.7091249], [40.4201179, -3.7091249]]
        zoom = 15
        service = "restaurant"
```

```
        data_path = "../../data/"
        proxy_list = ['140.227.80.237:3180', '183.221.242.103:9443']

        scraper = Google_Maps_Service_Scraper(coord_array, zoom, service)
        scraper.execution()
        df = scraper.df

    """

    def __init__(self, coord_array, zoom, service, data_path, proxy_list):
        self.coord_array = coord_array
        self.zoom = zoom
        self.service = service
        self.df = None
        self.logger = logging.getLogger(__name__)
        self.data_path = data_path
        self.proxy_list = proxy_list

    def test_proxies(self,timeout=10, verbose=1):
        """Method to test the proxies provided.
            It prints the working proxies and the time taken for each one.
            It edits the proxy_list attribute to only contain the working proxies.
        Args:
            Self
            timeout (int): Timeout for the requests
            verbose (int): Verbosity level
        Returns:
            None
        """


        url = f"https://www.google.com/maps/search/{self.service}/@{self.coord_array[0][0]},{self.coord_array
              [0][1]},{self.zoom}z"

        if self.proxy_list != None:

            working_proxies = []
            for proxy in self.proxy_list:
                try:
                    response = requests.get(url, proxies={"http": proxy, "https": proxy}, timeout=timeout)
                    time_taken = response.elapsed.total_seconds()


                    if response.status_code == 200:
                        working_proxies.append(proxy)
                        if verbose > 0:
                            print(f'Proxy {proxy} working -->',f'Time taken for proxy {proxy}: {time_taken}
                                  seconds')
                    else:
                        if verbose > 0:
                            print(f'Proxy {proxy} not working. Response code: {response.status_code}')
                except:
                    if verbose > 0:
                        print(f'Could not connect to proxy {proxy}')
            if verbose > 0:
                print(f'Final proxy list: {working_proxies}')

            self.proxy_list = working_proxies
            #print the proxy list like proxie_list = ['
            print('proxie_list = [')
            for proxy in self.proxy_list:
                print(f'"{proxy}",')
            print(']')


        else:
            if verbose > 0:
                print('No proxy list provided')


    def __get_places(self,scrolls=2):
        """Method to scrape Google Maps for a given service in a given area and zoom level
            It adds the Name, Gmaps URL, Address and type of the place to a DataFrame
        Args:
            Self
        Returns:
            df (pd.DataFrame): DataFrame with the scraped data
        """

        # driver_path = "../../driver/chromedriver.exe"
        # s = Service(driver_path)
        #driver = webdriver.Chrome(service=s)

```

```
        df_places = pd.DataFrame({})
        j = 0
        print(f'Starting scraping for {self.service}')
        for coords in self.coord_array:
            scroll_element = None

            if self.proxy_list != None:
                len_proxy_list = len(self.proxy_list)
            else:
                len_proxy_list = 1

            for k in range(len_proxy_list):
                print("\r" ,f'Trying {k+1} / {len_proxy_list} proxies for search {j+1} / {len(self.
                    coord_array)}', end="")
                self.logger.info(f'Trying {k+1} / {len_proxy_list} proxies for search {j+1} / {len(self.
                    coord_array)}')

                options = webdriver.ChromeOptions()
                options.add_argument('--headless')

                if self.proxy_list != None:
                    proxy_chosen = random.choice(self.proxy_list)
                    options.add_argument('--proxy-server=%s' % proxy_chosen)
                    self.logger.info(f'Proxy chosen: {proxy_chosen}')

                driver = webdriver.Chrome(options = options)

                url = f"https://www.google.com/maps/search/{self.service}/@{coords[0]},{coords[1]},{self.zoom
                    }z"
                #driver.get(url)
                try:
                    driver.get(url)
                    self.logger.info(f'Getting url: {url}')
                except:
                    self.logger.warning(f'Error getting url: {url}')

                driver.maximize_window()


                try:
                    cookies_button_xpath = "/html/body/c-wiz/div/div/div/div[2]/div[1]/div[3]/div[1]/div[1]/
                        form[1]/div/div/button"
                    cookies_button = driver.find_elements(By.XPATH, cookies_button_xpath)
                    cookies_button[0].click()
                    #time.sleep(0.5)
                    self.logger.info('Cookies button clicked')
                except:
                    self.logger.info('Cookies button not found')
                    pass

                try:
                    scroll_element_xpath = "/html/body/div[3]/div[9]/div[9]/div/div/div[1]/div[2]/div/div[1]/
                        div/div/div[2]/div[1]"
                    scroll_element = driver.find_element(By.XPATH, scroll_element_xpath)
                except:
                    self.logger.warning('Scroll element not found')
                    pass
                if scroll_element != None:
                    for i in range(scrolls):
                        scroll_element.send_keys(Keys.SPACE)
                        scroll_element.send_keys(Keys.SPACE)
                        scroll_element.send_keys(Keys.SPACE)

                        try:
                            driver.find_element(By.CLASS_NAME, "HlvSq")
                            self.logger.info('Table final element found')
                            break
                        except:
                            pass

                        #time.sleep(0.1)


                    table_xpath = "/html/body/div[3]/div[9]/div[9]/div/div/div[1]/div[2]/div/div[1]/div/div/
                        div[2]/div[1]"
                    soup = BeautifulSoup(driver.find_element(By.XPATH, table_xpath).get_attribute('innerHTML'
                        ), 'html.parser')

                    places = soup.find_all('div' ,class_="Nv2PK THOPZb CpccDe")

                    for place in places:
                        place_html = BeautifulSoup(str(place), 'html.parser')

```

```
                        try:
                            gmaps_link = place_html.find(class_="hfpxzc").get("href")
                        except:
                            gmaps_link = np.nan
                            self.logger.error('Gmaps link not found')

                        spans = place_html.find("div", class_="Z8fK3b").find_all("span")

                        try:
                            name = spans[0].text
                        except:
                            name = np.nan
                            self.logger.warning('Name not found')

                        try:
                            rating = spans[6].text
                        except:
                            rating = np.nan
                            self.logger.warning('Rating not found')

                        try:
                            reviews = spans[7].text
                            reviews = reviews[1:-1] #remove the parenthesis
                        except:
                            reviews = np.nan
                            self.logger.warning('Reviews not found')

                        try:
                            place_type = spans[13].text
                        except:
                            place_type = np.nan
                            self.logger.warning('Place type not found')

                        try:
                            address = spans[17].text
                        except:
                            address = np.nan
                            self.logger.warning('Address not found')

                        try:
                            more_info = spans[18:35] #get the spans with the more info
                            #make a tuple with the text of the spans
                            more_info = tuple([span.text for span in more_info])
                        except:
                            more_info = np.nan
                            self.logger.warning('More info not found')

                        df_places = df_places.append({'name': name,
                                                      'gmaps_link': gmaps_link,
                                                      'rating': rating,
                                                      'reviews': reviews,
                                                      'place_type': place_type,
                                                      'adress': address,
                                                      'more_info' : more_info}, ignore_index=True)



                    if scroll_element != None:
                        j = j + 1

                        self.logger.info(f'Page {j} of {len(self.coord_array)} done')
                        break
                    if k == len(self.proxy_list)-1:
                        j = j + 1
                        self.logger.warning(f'No more proxies to try, search in coordinates {coords} stopped'
                            )
                        break

        self.df = df_places
        print('\n','Scraping done')
        return df_places

    def __get_lat(self,gmaps_link):
        try:
            return gmaps_link.split('!3d')[1].split('!4d')[0]
        except:
            self.logger.warning('Lat not found')
            return np.nan

    def __get_lon(self,gmaps_link):
        try:
            return gmaps_link.split('!4d')[1].split('!16')[0]
```

```
297             except:
298                 self.logger.warning('Lon not found')
299                 return np.nan
300 
301     def __get_coordinates(self):
302         """Get the coordinates of the places in the dataframe and add them to the dataframe
303             The coordenates are obtained from the gmaps link
304         """
305         self.df['lat'] = self.df['gmaps_link'].apply(lambda x: self.__get_lat(x))
306         self.df['lon'] = self.df['gmaps_link'].apply(lambda x: self.__get_lon(x))
307 
308         self.df['lat'] = self.df['lat'].astype(float)
309         self.df['lon'] = self.df['lon'].astype(float)
310 
311     def __extract_phone(self,row):
312         try:
313             return '+' + str(row).split('+')[1].split("'")[0]
314         except:
315             self.logger.info('Phone not found')
316             return np.nan
317 
318     def __get_phone(self):
319 
320         self.df['phone'] = self.df['more_info'].apply(lambda x: self.__extract_phone(x))
321         #change the column to string
322         self.df['phone'] = self.df['phone'].astype(str)
323 
324     def __drop_duplicates(self):
325 
326         self.df = self.df.drop_duplicates(ignore_index=True)
327 
328     def get_map(self, lat, lon):
329         """
330         Get a map with the places in the dataframe and the location of the user using folium
331 
332         Parameters
333         ----------
334         lat : float
335             Latitude of the user location
336         lon : float
337             Longitude of the user location
338 
339         Returns
340         -------
341         m : folium map
342             Map with the places in the dataframe and the location of the user
343 
344 
345         """
346         map = folium.Map(location=[lat, lon], zoom_start=13)
347         #lets add a marker to the map
348         folium.Marker([lat, lon], popup='My Location').add_to(map)
349         #lets add a marker to the map for each row in the dataframe
350         for i in range(len(self.df)):
351             folium.Marker([self.df['lat'][i], self.df['lon'][i]], popup=self.df['name'][i]).add_to(map)
352         return map
353 
354     def get_heat_map(self, lat, lon):
355         """
356         Get a heatmap with the places in the dataframe and the location of the user using folium
357 
358         Parameters
359         ----------
360         lat : float
361             Latitude of the user location
362         lon : float
363             Longitude of the user location
364 
365         Returns
366         -------
367         m : folium map
368             Heatmap with the places in the dataframe and the location of the user
369 
370 
371         """
372         map = folium.Map(location=[lat, lon], zoom_start=13)
373         #lets add a marker to the map
374         HeatMap(data=self.df[['lat', 'lon']], radius=10).add_to(map)
375         return map
376 
377 
378     def __save_data(self):
379         """
```

```
380         Save the dataframe in a csv file
381 
382         """
383 
384         now = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
385         self.df.to_csv(f'{self.data_path}/{self.service}_searches({len(self.coord_array)})_{now}.csv', index=
                False)
386 
387     def execution(self,save=True, scrolls=2):
388         """
389         Execute the search and save the data if save is True
390 
391         """
392 
393         self.__get_places(scrolls=scrolls)
394         self.__drop_duplicates()
395         self.__get_coordinates()
396         self.__get_phone()
397         if save:
398             self.__save_data()
399         return self.df
```

Bibliography

[1] Coordinate systems worldwide. https://epsg.io/.

[2] Consulta del nomenclátor de calles de la comunidad de madrid. https://gestiona.comunidad.madrid/nomecalles/.

[3] Datos abiertos - consorcio regional de transportes de madrid. https://datos.crtm.es/.

[4] Wardarahim. Modelling insurance claims data using the tweedie approach. https://medium.com/@wardarahim25/modelling-insurance-claims-data-using-the-tweedie-approach-94db8b14bfb5.

[5] Esri state & local connect. https://www.esri.com/en-us/lg/industry/government/20/esri-state-local-connect.

[6] Ylenia Casali, Nazli Yonca Aydin, and Tina Comes. Machine learning for spatial analyses in urban areas: a scoping review. *Sustainable Cities and Society*, 85:104050, 2022.

[7] StateTech Magazine. 4 ways to use deep learning to improve citizen services. https://statetechmagazine.com/article/2023/06/4-ways-use-deep-learning-improve-citizen-services, June 2023.

[8] Ash Center for Democratic Governance and Innovation. Artificial intelligence for citizen services. https://ash.harvard.edu/files/ash/files/artificial_intelligence_for_citizen_services.pdf, Fecha desconocida.

[9] The World Bank. Geospatial technology and information for development. https://www.worldbank.org/en/topic/land/brief/geospatial-technology-and-information-for-development.

[10] Datos sociodemográficos - instituto nacional de estadística. https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176992&menu=resultados&idp=1254735572981.

[11] Datos abiertos - comunidad de madrid. https://www.comunidad.madrid/gobierno/datos-abiertos.

[12] Api google maps nearby places. https://developers.google.com/maps/documentation/places/web-service/search-nearby?hl=es-419.

[13] Google maps. https://www.google.es/maps/.

[14] scikit-learn poissonregressor. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PoissonRegressor.html.

[15] scikit-learn decision trees. https://scikit-learn.org/stable/modules/tree.html.

[16] Xgboost documentation. https://xgboost.readthedocs.io/en/stable/.

[17] Shap documentation. https://shap.readthedocs.io/en/latest/index.html.

[18] Openstreetmap wiki. https://wiki.openstreetmap.org/.

[19] H3 documentation - uber's hexagonal hierarchical spatial index. https://h3geo.org/.