Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Estudio de los datos de evolución del SARS-CoV-2 en Galicia a través de diferentes fuentes de datos en el primer trimestre del año 2021

Estudiante: Diego Alvariño Arias
Dirección: Susana Ladra González

A Coruña, noviembre de 2021.

A mi familia

Agradecimientos

Gracias a mi familia por hacer que todo sea posible siempre. Hermano, mamá y papá, sois la razón de todo. A mis amigos, ellos saben quiénes son, y a Andrea.

También quiero agradecer a los compañeros y ahora muy buenos amigos que me llevo de la carrera, Ángel, Diego, Adrián, Iago y Alexandre, si se nos han pasado cuatro años volando ha sido porque nos lo hemos pasado genial juntos, gracias OG's. Agradecer también a mi tutora Susana por el apoyo.

Resumen

El objetivo de este Trabajo de Fin de Grado es analizar los datos de evolución del SARS-CoV-2 en Galicia, dividida en siete áreas sanitarias, a través de diferentes fuentes de datos investigadas, centrándonos en el primer trimestre del año 2021.
Para alcanzar el objetivo marcado, se ha seguido una metodología propia de trabajos de investigación de datos y sus respectivos pasos adaptados al estudio a realizar, la metodología CRISP-DM.

Principalmente, después de un esfuerzo por normalizar, tratar y explicar un conjunto de datos razonable y trabajado, siguiendo todos los pasos necesarios y marcados por la metodología, se han estudiado diferentes modelos de predicción. Además, se ha empleado la tarea clustering con aprendizaje no supervisado, basada en k-means para llegar a unas conclusiones y resultados finales de un análisis descriptivo.

Abstract

The objective of this Final Degree Project is to analyze the evolution data of SARS-CoV-2 in Galicia, divided into seven health areas, through different sources of data investigated, focusing on the first quarter of the year 2021.
To achieve the established objective, a methodology of data research work has been followed and its respective steps adapted to the study to be carried out, the CRISP-DM methodology.

Mainly, after an effort to normalize, treat and explain a reasonable and worked data set, following all the necessary steps and marked by the methodology, different prediction models have been studied. In addition, the clustering task with unsupervised learning, based on k-means, has been used to reach conclusions and final results of a descriptive analysis.

Palabras clave:

- COVID-19
- Pandemia
- Análisis de datos
- Método predictivo
- Clustering

Keywords:

- COVID-19
- Pandemic
- Data analytics
- Predictive method
- Clustering

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1 Motivación

Ante la época convulsa en la que vivimos, en la que un virus desconocido pone a una sociedad patas arriba, cualquier estudio que se pueda realizar es relevante. En este caso, se estudiarán los efectos causados por dicho virus, el SARS-CoV-2, a nivel de Galicia.

Para ello, se analizarán las siete áreas sanitarias principales de nuestra comunidad autónoma, Galicia: A Coruña, Lugo, Ferrol, Ourense, Pontevedra, Vigo y Santiago, en el primer trimestre de este año 2021.

Será necesario identificar los conjuntos de datos heterogéneos de interés, integrarlos y transformarlos para su posterior análisis con técnicas de minería de datos. Se plantearán diferentes tipos de dichos análisis, tanto descriptivos como predictivos.

1.2 Objetivo

El objetivo principal del proyecto es el descubrimiento de conocimiento a partir de fuentes de datos diversas sobre la pandemia del virus SARS-CoV-2 a nivel de Galicia. Para ello, se tratarán de analizar para cada área sanitaria anteriormente dicha: los casos activos de contagiados por el virus, las altas hospitalarias, los fallecidos, las diferentes pruebas realizadas... También se estudiará el comportamiento interno de los hospitales, como pueden ser las altas o bajas de pacientes en la UCI.

Investigando y haciendo un análisis completo de todas las fuentes de datos a integrar, se pueden llegar a sacar conclusiones resolutivas.

1.2.1 Objetivos concretos

Algunos objetivos concretos a los que se intentará llegar son los siguientes:

- **Analizar** las tasas de aumento de casos, letalidad y recuperación de casos de la COVID-19.
- **Estudiar** la evolución de la pandemia.
- **Estudiar** el comportamiento de las UCI a nivel gallego.
- **Encontrar** datos valiosos que se desconocían sobre el comportamiento del virus.
- En la medida de lo posible, cualquier **mejora** que se pueda aportar para ayudar a la ciudadanía.

1.3 Estructura de la memoria

La memoria está estructurada en ocho capítulos, que se describen a continuación:

- **Capítulo 1: Introducción.**
  Se introduce la motivación y el objetivo principal del proyecto, así como otros objetivos concretos y la estructura de la memoria.
- **Capítulo 2: Metodología y planificación.**
  Introducción y explicación del método de trabajo adoptado, así como la planificación estimada, seguimiento y costes.
- **Capítulo 3: Fundamentos tecnológicos.**
  Se detallan los materiales y medios empleados en el proyecto, añadiendo una breve explicación para cada uno de ellos.
- **Capítulo 4: Exploración de los datos.**
  Comenzando con un marco contextual y una exploración descriptiva de los datos, se tratará la preparación y limpieza de los mismos.
  Se mencionará la estructura y exploración descriptiva de los datos, terminando con un análisis exploratorio inicial.
- **Capítulo 5: Análisis descriptivo de los datos mediante clustering.**
  Se introduce la tarea de clustering y se explica de forma teórica y práctica el trabajo de clustering realizado. Finalmente, también se explica la extracción de resultados y se elaboran unas conclusiones.

- **Capítulo 6: Análisis predictivo.**
  Puesta en contexto y breve introducción a los modelos de predicción. Se añade una explicación teórica y práctica. Se elabora la extracción de los resultados y conclusiones.

- **Capítulo 7: Conclusiones.**
  Se hace una valoración de las lecciones aprendidas y las aportaciones que nos ha dado el proyecto. También se menciona la relación con competencias del grado y el posible trabajo futuro.

- **Capítulo 8: Anexos.**
  Se agregan los contenidos necesarios para ampliar o complementar la información del proyecto y ahondar sobre aspectos puntuales, que sirven de complemento para la investigación realizada.

Capítulo 2

Metodología

2.1 Método de trabajo adoptado

Se emplea la metodología denominada CRISP-DM [2], [3], un método probado para orientar trabajos de minería de datos. El ciclo vital por el que se rige este modelo contiene seis fases ordenadas, en nuestro caso numéricamente (como se puede observar en el cuadro 2.1).

Esta metodología fue diseñada en el 1996 por un conjunto de empresas europeas, con un objetivo común, crear un esquema que permitiera mostrar el ciclo de vida de un proyecto de este tipo, de minería de datos. En el ámbito en el que nos situamos, el ámbito de la minería de datos, CRISP-DM es una metodología puntera y una referencia a seguir, alejándose del resto de metodologías. A nivel comercial, son dos las grandes corporaciones que han sido las principales impulsoras de la aplicación de minería de datos a los procesos de negocio desde el inicio: SPSS y DaimlerChrysler [3].

1) COMPRENSIÓN DEL PROBLEMA A ESTUDIAR Y PLANIFICACIÓN.	2) COMPRENSIÓN DE LOS DATOS.	3) PREPARACIÓN DE LOS DATOS.
4) MODELADO.	5) EVALUACIÓN.	6) DESPLIEGUE DE LOS RESULTADOS Y EXTRACCIÓN DE LAS CONCLUSIONES.

Cuadro 2.1: Modelo de referencia CRISP-DM seguido.

La secuencia de las fases no es estricta. De hecho, la mayoría de los proyectos avanzan y retroceden entre fases si es necesario [2].

- **Comprensión del problema a estudiar y planificación**: en esta fase se establecen los objetivos, se evalúa la situación y se trata de obtener un plan de proyecto viable.
- **Comprensión de los datos**: la ejecución del proceso de captura de datos, la descripción del conjunto de datos, la ejecución de diversas tareas de exploración de datos y la gestión de la calidad de los datos: identificar problemas y brindar soluciones, son parte de esta etapa.
- **Preparación de los datos**: establecer el universo de datos que usaremos, realizar tareas de limpieza de datos, construir un conjunto de datos adecuado para modelos de minería de datos e integrar datos de diferentes fuentes heterogéneas.
- **Modelado**: selección de las técnicas de modelado que más se adaptan a nuestros conjuntos de datos y a nuestros objetivos, se comienza a construir un modelo a partir de la aplicación de las técnicas seleccionadas sobre el conjunto de datos y ajuste del modelo evaluando su fiabilidad, así como su impacto en los objetivos anteriormente establecidos.
- **Evaluación**: evaluación de uno o más modelos (los generados hasta el momento), y revisión de todo el proceso de minería de datos que nos ha llevado a este punto. Se determinan los próximos pasos a dar, ya sea para repetir la fase anterior o para abrir una nueva ruta o proceso en la investigación.
- **Despliegue de los resultados y extracción de las conclusiones**: se diseña un plan para implementar modelos y conocimientos sobre nuestra organización, y se revisa todo el proyecto para determinar las lecciones aprendidas y las futuras mejoras.

Se establecen objetivos dentro de unas fechas estipuladas, tratando de cumplir y seguir lo máximo posible dichas fechas, pero dejando margen de error a cualquier complicación, al tratarse además de un proyecto de investigación y desarrollo.

Además se han realizado reuniones de seguimiento (habitualmente semanales) con la directora del trabajo, para verificar y potenciar el progreso al ritmo adecuado.

2.2 Planificación del proyecto

Para la planificación del proyecto se han ajustado las tareas a realizar vistas anteriormente en la metodología de trabajo adoptada (cuadro 2.1) a un tiempo estimado para cada una de ellas.

Para ello, se han ajustado al comienzo del proyecto unas fechas con una holgura suficiente. En el cuadro de planificación 2.2, se pueden observar las fechas de inicio y de final para cada una de las tareas.

Tarea	Fecha inicio – fecha final
Reunión inicial y anteproyecto.	09/03/2021 – 09/04/2021
Comprensión del problema a estudiar y planificación.	09/04/2021 – 09/05/2021
Comprensión de los datos.	
Preparación de los datos.	09/05/2021 – 01/06/2021
Modelado.	
Evaluación.	01/07/2021 – 01/09/2021
Despliegue de los resultados y extracción de conclusiones.	
Redacción de la memoria.	09/03/2021 – 04/09/2021

Cuadro 2.2: Cuadro de planificación inicial.

2.3 Seguimiento del proyecto

Ha sido necesario recalcular las fechas en dos ocasiones, sin suponer ningún problema sobre la planificación calculada inicialmente 2.2. En el cuadro 2.3, se pueden observar las modificaciones en la planificación una vez realizado el seguimiento.

Las tareas de evaluación y despliegue de los resultados, inicialmente calculadas para que consumieran menos tiempo de lo que finalmente han consumido, han sido las tareas por las que ha sido necesario el recalcular las fechas estimadas.

Tarea	Fecha inicio – fecha final
Reunión inicial y anteproyecto.	09/03/2021 – 09/04/2021
Comprensión del problema a estudiar y planificación.	09/04/2021 – 09/05/2021
Comprensión de los datos.	
Preparación de los datos.	09/05/2021 – 01/07/2021
Modelado.	
Evaluación.	01/07/2021 – 01/10/2021
Despliegue de los resultados y extracción de conclusiones.	
Redacción de la memoria.	09/03/2021 – 13/11/2021

Cuadro 2.3: Cuadro de planificación después de haber realizado el seguimiento del proyecto.

Como se puede observar en el cuadro de seguimiento 2.4, finalmente se ha detallado el tiempo empleado para cada una de las fases, expresando dicho tiempo en horas.

El tiempo total calculado que se ha empleado en el proyecto ha sido de 386 horas, siendo la parte final: evaluación, despliegue de los resultados, extracción de conclusiones y la redacción de la memoria las partes que más tiempo han requerido.

Tarea	Tiempo
Reunión inicial y anteproyecto.	21 horas
Comprensión del problema a estudiar y planificación.	13 horas
Comprensión de los datos.	7 horas
Preparación de los datos.	30 horas
Modelado.	22 horas
Evaluación.	165 horas
Despliegue de los resultados y extracción de conclusiones.	36 horas
Redacción de la memoria.	92 horas
TOTAL	**386 horas**

Cuadro 2.4: Cuadro de seguimiento con duraciones reales.

2.4 Costes del proyecto

Todas las herramientas y el software empleado ha sido de acceso gratuito o, en algunas ocasiones, usando versiones de prueba temporales gratuitas. No obstante, se incluye una estimación del coste de dichas licencias.

Por lo tanto para la contabilidad de los costes y del presupuesto final del proyecto, se tienen en cuenta el número de horas finales trabajadas y la suma del coste estimado que supondrían las licencias que han sido usadas en sus versiones de prueba temporales gratuitas.

Consultando y analizando los datos de diversos convenios colectivos estatales de empresas de consultoría, estudios de mercados y de la opinión pública del BOE, así como diferentes webs en las que millones de personas, usuarios y trabajadores pueden compartir información de diferentes puestos de trabajo (sueldos, condiciones, beneficios...) se puede considerar que un analista de datos cobra unos 15€/h.

En su versión de prueba, se ha utilizado la tecnología Power BI Premium, la cual tiene un precio de 16,90€/mes. La versión "trial" ha durado 30 días.

En conclusión, una vez calculadas las horas empleadas en el proyecto, sin tener en imputa los costes de elaboración de la memoria, se hace la siguiente operación:

	Tiempo	**Coste**	**Total**
Analista	294 horas	15 €/h	4410 €
Power BI premium	1 mes	16,90 €/mes	16,90 €
			4426,90 €

Cuadro 2.5: Cuadro de estimación de costes del proyecto.

El coste estimado del proyecto sumando la estimación del coste de personal y la estimación del coste de licencias, ha sido de 4426,90 euros.

Capítulo 3

Fundamentos tecnológicos

3.1 Tecnologías empleadas

Se listan las diferentes herramientas y librerías que han sido necesarias para la ejecución del proyecto:

Microsoft Excel: se define como un software conocido que permite realizar tareas contables y financieras gracias a sus funciones, desarrolladas específicamente para ayudar a crear y trabajar con hojas de cálculo [4].

Microsoft Power BI: como solución integrada en Office 365, es capaz de conectarse a miles de orígenes de datos, además de poseer soluciones para la preparación de datos simplificada y la generación de análisis ad hoc o gráficos [5] y [6]. Además, ofrece una gran facilidad para encontrar y compartir conocimientos significativos a través de un sinfín de visualizaciones diferentes de datos, además de su integración perfecta con Microsoft Excel y otros conectores de datos que han ayudado a la realización del proyecto.

Anaconda Navigator: se trata de una distribución gratuita de código abierto de los lenguajes de programación Python y R ampliamente utilizada en computación científica (ciencia de datos, aprendizaje automático, ciencia, ingeniería, conjeturas de análisis predictivo...). Se ha empleado para el despliegue del entorno de trabajo con Python [7].

Python: se define como un "lenguaje de programación versátil, multiplataforma y multiparadigma que se destaca por su código legible y limpio". Tiene una licencia de código abierto que permite su uso gratuito en varios contextos [8]. Es un lenguaje de propósito general y se utiliza para aplicar métodos de ciencia de datos gracias a la gran cantidad de librerías disponibles y a la gran comunidad que las respalda. En nuestro caso, utilizado para realizar el trabajo

de análisis predictivo.

Las librerías empleadas en Python han sido las siguientes:

- Pandas [9]: se trata de la librería por excelencia del lenguaje Python, contiene diferentes herramientas que permiten leer y escribir datos en varios formatos: CSV, Microsoft Excel, bases SQL y formato HDF5. La librería da la opción de filtrar y seleccionar tablas de datos, así como fusionar y unir diferentes conjuntos de datos, aplicar funciones para transformar dichos datos, tanto de manera global como seleccionando ventanas más individualizadas, manipular series temporales, realizar y exportar gráficas… y más. Pandas dispone de tres estructuras de datos diferentes: series, DataFrame y panel.
- Numpy [10]: se trata de una librería especializada en el cálculo numérico y el análisis de datos, centrada en trabajos con un gran volumen de datos. Numpy incorpora arrays.
- Sklearn [11]: es un conjunto de rutinas escritas en Python para realizar análisis predictivo, incluyen clasificadores y algoritmos de clusterización. Está basada en las librerías NumPy, SciPy y matplotlib, con lo que es fácil reutilizar y aprovechar el código que usan estas librerías. Además, incorpora funciones para preprocesar los datos de una manera sencilla.

RStudio: un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. En nuestro caso, ha sido utilizado para unir BBDD, para el apartado de limpieza de los datos y para la realización del análisis descriptivo mediante clustering [12], [13] y [14].

Las librerías empleadas en la herramienta RStudio para el trabajo realizado de clustering han sido las siguientes:

- Tidyverse [15]: es una colección de paquetes disponibles en R y orientados a la manipulación, importación, exploración y visualización de datos, que se utiliza exhaustivamente en ciencia de datos. El uso de la librería Tidyverse nos ha facilitado el trabajo estadístico y la generación de trabajos reproducibles.
- Dplyr [16]: derivada del paquete plyr, pero con un valor añadido que no es más que la simplicidad que emplea para realizar las diferentes operaciones, gracias a las funciones de este paquete, las operaciones se hacen más rápido en comparación con las operaciones realizadas con las funciones del paquete plyr. No obstante, dplyr no tiene ninguna funcionalidad que no pueda ser llevada a cabo con el conjunto de funciones del paquete base plyr.

- Kml [17]: esta librería permite agrupar datos según patrón de sus trayectorias, es decir, es una implementación de k-means diseñada específicamente para analizar datos longitudinales.

Capítulo 4

Exploración y preparación de los datos

Comenzamos con un marco contextual necesario para conocer la problemática del momento y que ha servido de ayuda para marcar los objetivos concretos anteriormente citados en la sección 1.2.1. Después, una vez analizados y escogidos los conjuntos de datos de las diferentes fuentes obtenidas/encontradas (debidamente citadas y explicados de forma descriptiva los datos), se preparan los datos en archivos para verificar que son correctos y completos, así los análisis posteriores serán más fiables.

4.1 Marco contextual

En el comienzo de la pandemia, situado temporalmente en el primer caso de COVID-19 en España diagnosticado el 31 de enero de 2020, en La Gomera, y ante el gran desconocimiento de la mayoría de la población de dicha enfermedad, surge una urgencia a la hora de conocer información derivada de la COVID-19 [18].
Por entonces, las comunidades autónomas del país en el que vivimos se vieron ante una situación confusa en la que no se sabía la importancia que iba a tener en un futuro reciente el manejo y control de los datos de contagios, casos, fallecidos... a raíz de la pandemia, totalmente novedosa que se estaba a plantear.

Fue entonces cuando, centrándonos en Galicia, se empezó a realizar un esfuerzo, poniendo a disposición del ciudadano, teléfonos, páginas web de contacto ante una posible sospecha de infección del virus, y otros medios de contacto directo con los especialistas que lo que tenían como objetivo no era más que un propósito común, parar la expansión del virus en la comunidad y desde la administración, tratar de ir conformando una base de datos que diera información relevante a la población, para que así se pudieran tomar las medidas pertinentes,

y estar al corriente de la actualidad.

Entre los meses de septiembre y noviembre del 2020, la situación empezó a normalizarse, después de un intenso trabajo de las diferentes organizaciones y de los medios para la mejora de las diferentes vías en las que se compartían los datos en toda España, en el marco de una mejora en accesibilidad y visibilidad.

Destacar que como en todas las ocasiones en las que se siembra un cierto pánico en la sociedad, muchas veces debido a ganancias monetarias, siempre hay cuentas, webs o foros que transmiten informaciones falsas e incorrectas, por lo tanto, es importante tener desde un principio la oficialidad de las cuentas o personas en las que nos guiamos respecto a unos datos tan delicados como son los de la COVID-19. Desde el comienzo de la realización de este trabajo se han tenido muy en cuenta las diversas fuentes de datos, asegurándose así de la veracidad de las mismas en todo momento.

Los datos de la COVID-19 que se usan en este trabajo han sido sacados de diversas fuentes:

- A través de la web del Sergas que se ha creado a finales del año 2020 [19].
- **Servicios de comunicación** de diferentes áreas sanitarias del Sergas. Ya sea a través de su perfil oficial en redes sociales como Twitter [20], u otros medios.
- Instituto de Salud Carlos III [21].

4.2 Preparación de los datos

Como se ha dicho anteriormente en el apartado 4.1, los datos han sido extraídos de diferentes fuentes de manera manual y contrastados en la mayoría de lo posible entre los diferentes puntos de información en los que se ha respaldado el trabajo.

No obstante, para la extracción de datos de fuentes de datos en las que no fue posible la descarga directa a diferentes tipos de archivo, se han empleado diversas API sencillas:

- **Table Capture**: se trata de una extensión para navegadores multiplataforma, en nuestro caso empleado en Google Chrome. Es capaz de capturar fácilmente todos los datos tabulares de sitios web y exportarlos a una hoja de cálculo, de manera que si estamos viendo una tabla en HTML (como ha sido el caso para el proyecto), se exporta fácilmente el conjunto de datos a un archivo Microsoft Excel.

- **ScraperWiki**: de la misma forma que para la anterior pero para datos que se encuentran en un PDF. Se importa un PDF, y se exporta de forma sencilla al formato que se desee, además, esta API, da la oportunidad de previsualizar los datos antes de exportarlos. ScraperWiki da la opción de realizar una limpieza de datos antes de exportarlos a un archivo Microsoft Excel. Esto es interesante porque, al añadir esos datos limpios ya a la herramienta en la que se visualizarán los datos finalmente, todo es más sencillo.

Además, se han utilizado diferentes foros externos no oficiales para concluir la veracidad de los datos, para asegurarse antes del futuro análisis exhaustivo.

Una vez se han identificado los datos, se han organizado los datos de las diferentes fuentes, ordenados por fecha y por las características que nos interesan en cada momento.

4.2.1 Limpieza de los datos

El trabajo más engorroso respecto a la preparación de los datos, ha sido la eliminación de todos los datos fuera de las fechas en las que se ha centrado el estudio: 1 de enero de 2021 - 31 de marzo de 2021, teniendo así que suprimir o apartar del conjunto de datos cualquier información que no se encuentre dentro del rango de fechas anteriormente estipulado.

No obstante, aunque se han acotado los datos, no se han olvidado los datos de fechas previas y/o posteriores (diciembre/noviembre de 2020 - actualidad) ya que, en muchos casos, nos ayudan a entender ciertas tendencias y asegurarnos de que los datos son correctos en ciertas ocasiones. Además, en cierta medida, los datos antiguos nos dan una idea de lo que sucederá más adelante, de esto se hablará en el capítulo 6.

El resto de la limpieza de datos realizada ha consistido en depurar los valores escritos de forma incorrecta en la base de datos, así como, en algunas ocasiones, información errónea, ya sea numérica o de lógica, como puede ser que cierto municipio se corresponda a un área sanitaria errónea. La razón principal por la que no se han tenido problemas graves con datos incorrectos tiene que ver con lo hablado en la sección 4.1, después del trabajo intenso de verificación.

Con respecto al trabajo de limpieza de datos, los datos erróneos han sido detectados y corregidos desde el propio Microsoft Excel, debido a la pureza y limpieza de los datos recogidos desde las diferentes fuentes de datos empleadas. En los propios archivos Excel, el trabajo realizado ha consistido en:

- Revisión ortográfica: Microsoft Excel posee una herramienta de revisión ortográfica

automática que reemplaza las posibles faltas de ortografía existentes en el texto, para nuestro caso ha sido necesario su uso de una forma manual, para evitar errores.

- Quitar filas duplicadas: Microsoft Excel establece la posibilidad de eliminar filas duplicadas fácilmente a través de una interfaz interactiva. Las filas duplicadas han surgido a la hora de unificar conjuntos de datos.

- Buscar y reemplazar texto: convirtiéndose en la funcionalidad más empleada que posee Microsoft Excel, se han buscado y reemplazado la mayoría de datos erróneos encontrados en nuestros archivos derivados de la unión de las diferentes bases y fuentes de datos. Ejemplo: las diferentes formas erróneas de referirse a "A Coruña": "La Coruña" o "Coruña".

En referencia a la calidad del proyecto existe una gran cantidad de datos, conocidos como *datos sucios*, que no cumplen los criterios de calidad.

Dichos datos pueden ser de varios tipos:

- **Ruidosos:** se trata de datos que tienen valores atípicos o erróneos, habitualmente son asociados a errores de introducción y transmisión de los datos o derivados de su extracción.
  - Ejemplos encontrados en el proyecto:
    * Referentes a la letra "ñ": Bergantiños, A Coruña, O Saviñao, A Pobra do Caramiñal, Moaña…
    * Referentes a tildes: Lalín, Arzúa, Barbadás…
    * Referentes a diéresis: Plurilingüe Vila do Arenteiro…

- **Inconsistentes:** se trata de datos que deberían coincidir, pero por alguna razón de violación de dependencias, diferencias de BBDD o de nombrado, esto no ocurre. Han sido sencillos de detectar debido a los resultados extraños a la hora de realizar agrupaciones y fáciles de solucionar debido al poco volumen de los mismos.
  - Ejemplos encontrados en el proyecto:
    * Diferentes formas de referirse a "A Coruña": A Coruña, La Coruña y/o Coruña.
    * Diferentes formas de referirse a "O Grove": O Grove, Grove y/o El Grove.

- **Incompletos:** son datos que contienen, por razón humana, SW/HW, o por errores referentes a la recolección, bien sea por diferencia de criterios o por indisponibilidad en dicho momento. alguno de sus valores sin contemplar.

- No se han encontrado ejemplos de datos incompletos a lo largo del proyecto.

El esquema de normalización y de corrección de los datos para cualquiera de los tres tipos anteriores ha sido la misma, de manera análoga para cada uno: identificación del dato erróneo, búsqueda de casos repetidos, corrección y verificación de la correcta corrección finalmente.

4.3 Estructura y exploración descriptiva de datos

Se han organizado los conjuntos de datos en diferentes ficheros en función de su contenido. A continuación, se estructurarán los diferentes datos a tratar para los ficheros más relevantes del estudio, mostrando un extracto de los datos para cada uno de ellos. Con cada tabla de muestra se hará la exploración descriptiva correspondiente.

- **Áreas sanitarias:** se trata de un conjunto de datos en el cual se enlaza cada municipio, con su provincia y su área sanitaria correspondiente. Es decir, una correspondencia directa entre los diferentes municipios y áreas sanitarias. Fuentes: [19] y [22].

municipio	provincia	area_sanitaria
Abegondo	Coruña	Coruña
Ames	Coruña	Santiago
Aranga	Coruña	Coruña
Ares	Coruña	Ferrol
Arteixo	Coruña	Coruña
Arzúa	Coruña	Santiago
Baña	Coruña	Santiago
Bergondo	Coruña	Coruña
Betanzos	Coruña	Coruña

Cuadro 4.1: Cuadro de distribución áreas sanitarias.

  La importancia de este fichero recae en tener bien identificado a cada uno de los municipios en su lugar correspondiente, ya que tener algún municipio mal de los 313 que se contemplan desde el Sergas a la hora de contabilizar los casos, supondría una alteración de los datos en el área sanitaria correspondiente.

- **Conjunto de datos por área sanitaria:** se trata del fichero en el cual se tienen los datos más relevantes del estudio, donde para cada día dentro del rango temporal que estudiamos, se muestra para el área sanitaria correspondiente, el número de casos COVID-19 oficialmente diagnosticados hasta el momento, el número de altas acumuladas desde el

inicio de la pandemia, el número de fallecidos totales desde el inicio de la pandemia y el número de pruebas PCR confirmadas en las últimas 24 horas, teniendo en cuenta que los datos se toman desde las 00:00 horas del día vigente hasta las 23:59 horas del día siguiente. Por lo tanto, al tener siete ASs, tendremos siete conjuntos de datos con sus respectivas columnas recientemente explicadas. Fuentes: [21], [19] y [22].

Fecha	A_Coruña.casos.acum	A_Coruña.altas.acum	A_Coruña.fallecidos.acum	A_Coruña.confirmados_pcr_24h
2021-01-01	13776	12377	312	97
2021-01-02	13894	12434	314	116
2021-01-03	13940	12496	314	43
2021-01-04	14058	12554	315	103
2021-01-05	14144	12604	321	71

Cuadro 4.2: Cuadro de muestra del conjunto de datos en el área sanitaria de A Coruña.

- **Conjunto de datos total:** se trata de la recopilación de todos los conjuntos de datos de las áreas sanitarias anteriores recopilados en uno, de forma que nos ayuda a entender de forma general el estudio a nivel gallego. Realizado a través de una suma automática de los valores individuales de las diferentes áreas sanitarias. Fuentes: [19] y [22].

Fecha	Galicia.casos.acum	Galicia.altas.acum	Galicia.fallecidos.acum	Galicia.confirmados_pcr_24h
2021-01-01	63510	56338	1400	441
2021-01-02	63954	56724	1403	412
2021-01-03	64251	57115	1409	269
2021-01-04	64715	57401	1414	422
2021-01-05	65066	57660	1421	298
2021-01-06	65750	57925	1427	590
2021-01-07	66428	58303	1431	639
2021-01-08	67085	58766	1436	535
2021-01-09	67999	59172	1441	798
2021-01-10	68716	59483	1445	661

Cuadro 4.3: Cuadro de muestra del conjunto de datos total.

- **Conjunto de datos por área sanitaria extendido:** nuevamente se tratan nuevos datos segmentados por áreas sanitarias. A simple vista parecen menos relevantes que los datos mencionados en la tabla 4.2 pero que ayudan, en su totalidad, a realizar nuevas investigaciones o estudios. Fuentes: [20] y [22].
  - Casos activos con seguimiento a domicilio.
  - Hospitalizados totales (UCI + planta) en los hospitales del AS.
  - Pacientes en UCI en los hospitales del AS.

- El número de pruebas PCR realizadas diarias en el AS.
- El número de pruebas PCR realizadas totales en el AS.
- El número de pruebas serológicas realizadas diarias en el AS.
- El número de pruebas serológicas realizadas totales en el AS.

Fecha	A_Coruna.domicilio	A_Coruna.hospitalizados	A_Coruna.uci
2021-01-01	1004	87	20
2021-01-02	1058	92	20
2021-01-03	1039	95	18
2021-01-04	1091	102	19
2021-01-05	1126	98	17
2021-01-06	1280	97	18
2021-01-07	1366	104	19
2021-01-08	1448	109	21

Cuadro 4.4: Cuadro de muestra del conjunto de datos por AS extendido.

A_Coruna.PCR.acum	A_Coruna.serologicas.acum	A_Coruna.PCR.diaria	A_Coruna.serologicas.diaria
234308	77433	1250	205
235418	77528	1110	95
236006	77651	588	123
236832	77770	826	119
237316	77919	484	149
238580	78160	1264	241
239718	78249	1138	89
240574	78450	856	201

Cuadro 4.5: Cuadro de muestra del conjunto de datos por AS extendido.

Además, otras como:

- El número de pruebas de antígenos realizadas totales en el área sanitaria.
- Total de pruebas que no se consideran PDIA (Pruebas Diagnósticas de Infección Activa, como lo sería una prueba PCR o una prueba de antígenos) hechas hasta la fecha en el AS.

Para el área sanitaria de Ourense, la información recopilada por las diferentes fuentes ha sido mucho mayor. Ourense ha sido un área sanitaria que se desmarca en comparación con las otras áreas gallegas en lo que se refiere a la calidad y cantidad de los datos que informaba día a día, dando información relevante a través del departamento de comunicación del Sergas Ourense a través de diferentes RRSS, como los pacientes en UCI divididos por cada hospital o clínica, por ejemplo, X pacientes en la clínica el hospital “CO.SA.GA” o X pacientes en UCI en la clínica “El Carmen”.

Dichos datos se contemplan pero, al no ser relevantes a nivel gallego ya que tan solo dividen la información entre diferentes hospitales de un área sanitaria común, no se han tenido en cuenta finalmente en nuestro estudio.

Por lo tanto, al tener siete áreas sanitarias, tendremos siete conjuntos de datos con sus respectivas columnas recientemente explicadas.

Además de los ejemplos gráficos vistos, se han empleado las incidencias acumuladas por localidades y por centros educativos, los cuales serán utilizados para hacer un estudio de la transmisión y del impacto que ha tenido la COVID-19 en los diferentes municipios gallegos y en los diferentes centros escolares.

Para ello, se sitúan los datos en unas fechas que se consideran especialmente relevantes para el estudio, tendremos 4 conjuntos de datos:

- Noviembre de 2020.
- Enero de 2021.
- Marzo de 2021.
- Junio de 2021.

Dentro de cada espacio temporal que se ha seleccionado para nuestro estudio, disponemos de diferente información relevante que nos ayudará a sacar conclusiones y a realizar el trabajo de investigación y el estudio pertinente entre los diferentes municipios. Para cada uno de los meses, se disponen de los siguientes cuatro conjuntos de datos:

- **Tipo de zona:** información geográfica a cerca del municipio, para situarlo en zona costera o de interior.
- **Nivel de incidencias en 14 días por municipios:**

  Habitantes.

  Casos.

  Incidencia.

  Municipios.

fecha	codigo_municipio	municipio	habitantes	casos_14d	IA14
2021-01-01	15002	Ames	31793	55	172,99406
2021-01-01	27051	Ribadeo	9854	21	213,11143
2021-01-01	32091	Vilardevós	1846	0	0
2021-01-01	36027	Meaño	5272	16	303,49014
2021-01-01	15021	Carral	6408	39	608,61423
2021-01-01	15082	Teo	18579	42	226,06168
2021-01-01	27019	Foz	9980	11	110,22044

Cuadro 4.6: Cuadro de muestra del conjunto de datos del nivel de incidencias en 14 días por municipios.

- **Una escala de valor del PIB:** para ver el nivel económico de las zonas. Todos los datos relacionados con el PIB han sido sacados del Portal do Instituto Galego de Estatística [23]. Se definirá de la siguiente forma:

  Valor del PIB < 10.000 euros.
  Valor del PIB entre 10.000 euros y 20.000 euros.
  Valor del PIB entre 20.000 euros y 30.000 euros.
  Valor del PIB > 30.000 euros.

  Para ello, los valores del PIB para cada municipio se sacan del siguiente conjunto de datos:

codigo_municipio	municipio	PIB
32059	A Peroxa	13765
32027	Cortegada	11664
32039	Laza	15237
32048	A Mezquita	13042
36056	Valga	22358
15025	Cerdido	13671
27057	Sarria	20093
36052	Silleda	19587

Cuadro 4.7: Cuadro de muestra del conjunto de datos del PIB para los diferentes municipios gallegos.

- **Centros educativos por área sanitaria y municipio:** tipo de centro, nombre de centro, positivos, aulas cerradas y centros cerrados. Fuente: [22].

Fecha	area_sanitaria	concello	tipo_cent	nombre_centro	positivos	aulas_cerradas	centro_cerrado
2021-01-09	Coruña	A CORUÑA	E.I	Municipal Agra do Orzán	2	0	no
2021-01-09	Coruña	A CORUÑA	E.I	Trastes Los Rosales	1	0	no
2021-01-09	Coruña	A CORUÑA	E.I	A Sardiñeira	3	0	no
2021-01-09	Coruña	A CORUÑA	E.I	Ventorrillo	1	0	no
2021-01-09	Coruña	A LARACHA	E.I	A Laracha	1	1	no
2021-01-09	Coruña	CABANA DE BE	E.I	Municipal A Casa dos Títeres	1	1	no
2021-01-09	Coruña	CEE	E.I.	Municipal Vila da Xunqueira	1	1	no

Cuadro 4.8: Cuadro de muestra del conjunto de datos de centros educativos por área sanitaria y municipio.

4.4 Análisis exploratorio inicial de los datos

Una vez preparados y conocidos los datos de los que disponemos, es importante hacer un análisis exhaustivo de los mismos para así poder tener el mayor número de información posible y conocerlos a fondo. Para ello, como se ha dicho anteriormente, se hará uso de la herramienta Microsoft Power BI y sus funcionalidades a la hora de sacar las gráficas necesarias.

Nos enfrentaremos a los objetivos marcados inicialmente. Para comenzar, observaremos cómo evolucionan los casos de COVID-19 en cada área sanitaria, así nos haremos una idea de las diferencias en función de las regiones en las que nos situemos.

Para ello, se analizarán el conjunto de PCR confirmadas cada 24 horas como resultado positivo, ya que tal y como se explica en el documento de diagnóstico del virus redactado por la sociedad española de medicina de familia y preventiva [24]: "al margen de los falsos negativos de COVID-19, el motivo por el que está especialmente valorada la prueba PCR para detectar casos de COVID-19 es por su alta sensibilidad, de media un 89%, aunque con diversas técnicas (como los cebadores) puede rozar el 100% de sensibilidad y no equivocarse a la hora de identificar que es el SARS-CoV-2 y no otro virus el que está presente en el organismo." [25]

La sensibilidad de una prueba es la capacidad de un test para detectar la enfermedad, siendo la prueba PCR, la única prueba con la que, en las fechas marcadas, se contaba como persona contagiada oficialmente teniendo que pasar la cuarentena pertinente.

Partiendo de la base en la que en el mes de diciembre había comenzado una nueva "ola" de casos a nivel español, nos situamos en el primer trimestre del año, con los casos disparados en nuestra comunidad autónoma.

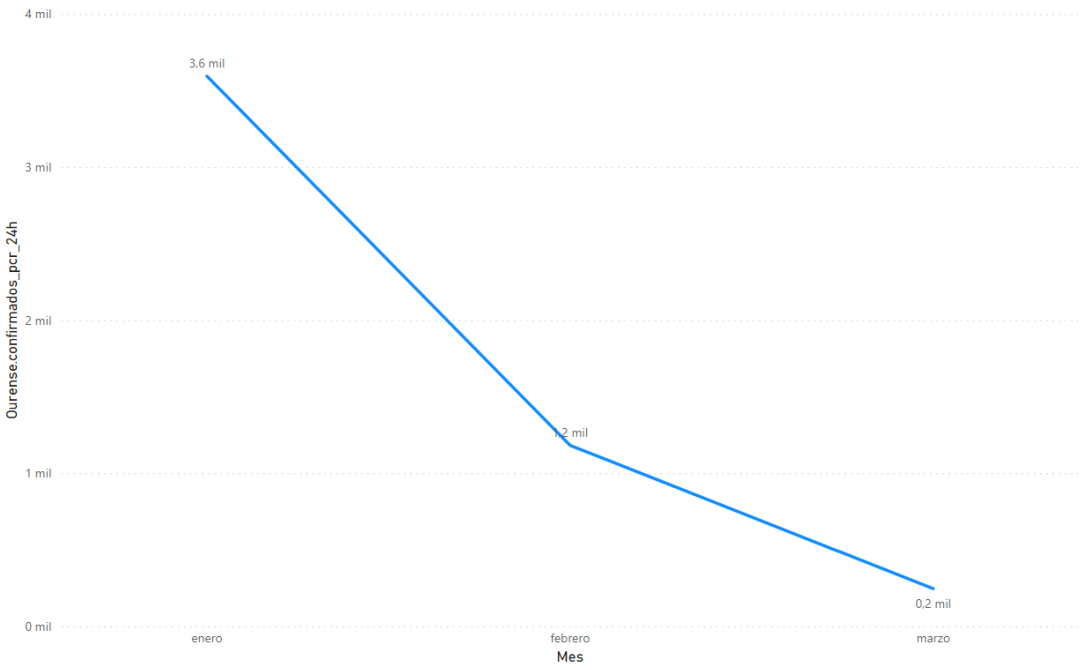

Figura 4.1: Evolución de casos de COVID-19 por PCR positiva confirmada para el área de Ourense.

En el primero de los estudios ya podemos observar la importancia de la dureza de las restricciones en el número de casos por área sanitaria. La Xunta de Galicia en enero, ante el incremento de casos abrumador, fue drástica en la toma de restricciones, [26] declarando el cierre perimetral individual de los ayuntamientos del Concello de Ourense y el cierre de la hostelería como medidas más drásticas en el impacto de los datos de casos, veremos la demostración más adelante.

Observamos como la línea de tendencia que nos enseña la gráfica 4.1 baja consecuentemente en el primer mes del trimestre, estabilizándose en valores más relajados para el segundo y tercer mes del mismo.

En cambio, en concellos en los que no se tomaron ese tipo de medidas drásticas y situados inicialmente en una situación igual o peor respecto a la incidencia acumulada de casos por habitantes, como es el caso de A Coruña o Ferrol (figura 4.2), la línea que nos marca la gráfica baja de una forma más lenta, tardando mucho más tiempo en llegar a una situación “estable” que Ourense (figura 4.1).

Analizándolo de forma cuantitativa, se puede ver esta situación comparando las pendientes de ambas gráficas, siendo la pendiente:

$$m = (\triangle y)/(\triangle x)$$

La evolución de los casos de COVID-19 por PCR positiva confirmada para el AS Ferrol ha sido de m = -23,33. En cambio para el AS de Ourense, m = -37,77. La diferencia en la evolución de casos de COVID-19 en Ferrol por PCR positiva confirmada hacia una situación "estable" es muy significativa con el resto de AS, Santiago tiene una m = -44,44 o por ejemplo, Vigo, con una pendiente = -53,33.

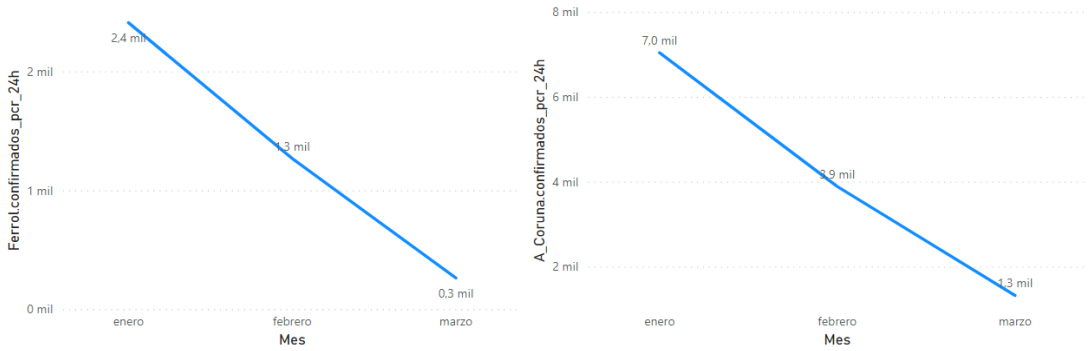

Figura 4.2: Evolución de casos de COVID-19 por PCR positiva confirmada para las áreas de A Coruña y Ferrol.

Para el resto de áreas sanitarias, en las siguientes figuras: A.1, A.2, A.3, A.4 (insertadas en el anexo) podemos observar que la situación ha sido análoga entre ellas, tardando el mismo tiempo en estabilizar los datos y siguiendo un comportamiento similar y coordinado entre los diferentes concellos.

También podemos observar que, fijándonos en el mapa gallego, Ferrol y A Coruña (ubicados cerca geográficamente), tienen a Ourense como AS más alejada en lo que se refiere a ubicación, un hecho que nos permite entender diferentes comportamientos. En el momento en el que se cierran perimetralmente los concellos del área sanitaria de Ourense, el número de casos diarios en los concellos de otras áreas sanitarias, pero geográficamente cercanas a Ourense, baja drásticamente (figuras A.1, A.2 y A.3). No se puede decir lo mismo para el caso análogo en A Coruña o Ferrol (figura 4.2).

El 8 de octubre de 2020, el Sergas declara la prueba PCR como única prueba válida para con-

tabilizar un caso más de la enfermedad. Es importante destacar que, analizando las pruebas realizadas y la pruebas PCR acumuladas a final de cada mes, la relación entre la positividad de las pruebas PCR y el tramo por el que atraviesa la variante del virus que se está expandiendo por la región en cuestión, va estrechamente ligada (figuras 4.2 y 4.3)

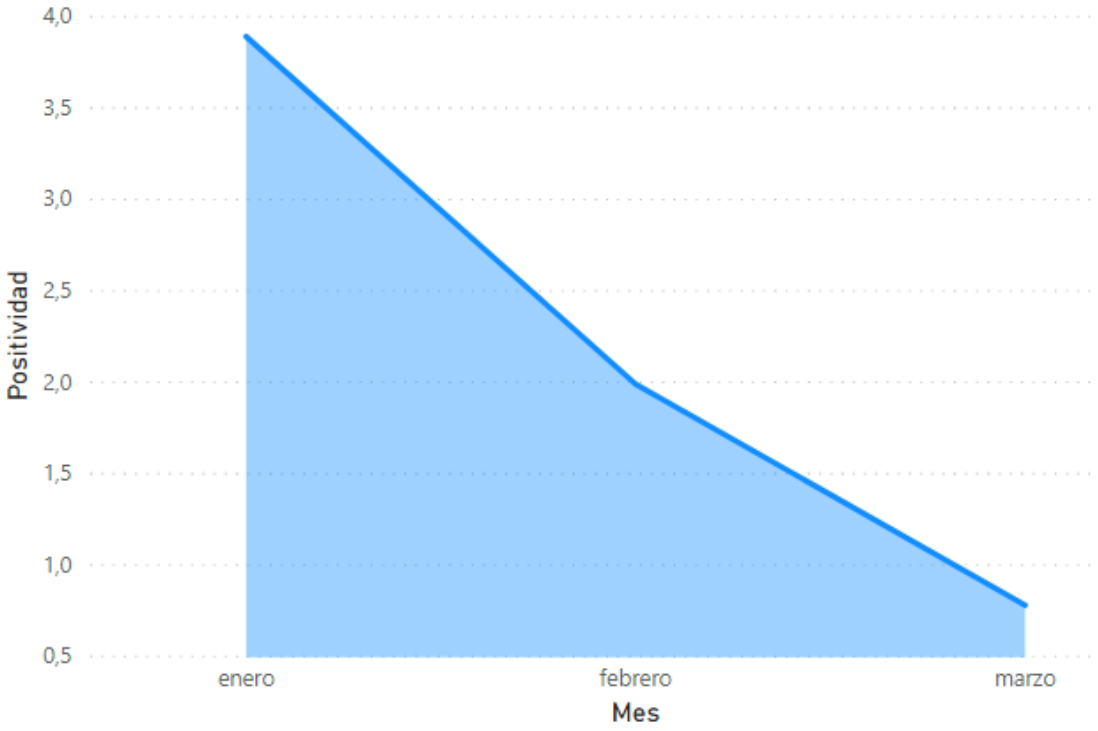

Figura 4.3: Evolución mensual de la positividad de las pruebas PCR para el área de A Coruña.

Por lo tanto, se reafirma la importancia de realizar cribados y un mayor número de pruebas PCR, debido a que en cuanto sube el número de pruebas, en el transcurso de 15-20 días posteriores, se van disminuyendo el número de casos positivos diarios. Se puede observar en la figura A.5 donde, para la fecha 13 de enero en el AS de A Coruña, se comienza a incrementar el número de pruebas PCR realizadas diariamente, obteniendo un descenso progresivo de la positividad de las mismas al transcurso de los 15-20 días posteriores.

Con respecto al número de pruebas PCR diarias realizadas en cada una de las áreas sanitarias (dejando al lado ciertos días anómalos en los que se recogen menos pruebas realizadas de lo normal), la cantidad de pruebas realizadas con relación al número de habitantes ha sido igual para todas las áreas sanitarias, excepto para Vigo y Pontevedra (ver cuadro 4.9), que ha sido menor la media de PCR por habitante.

A fecha 2021-03-31	PCRs acumuladas	Habitantes por AS	Media de PCR por habitante
A Coruña	399.674	244.850	1,632
Ferrol	127.889	66.799	1,915
Lugo	205.997	98.025	2,101
Ourense	219.734	105.505	2,083
Pontevedra	183.058	292.782	0,625
Santiago	308.367	125.405	2,459
Vigo	407.167	569.534	0,715

Cuadro 4.9: Media de pruebas PCR por habitante por AS a 2021-03-31.

Por último, como nos refleja la gráfica 4.4, en todo el trimestre se han dado en Galicia 46 mil casos positivos en el virus por medio de una prueba PCR oficial, en las áreas en las que más población vive ha habido un mayor número de positivos, como es entendible.

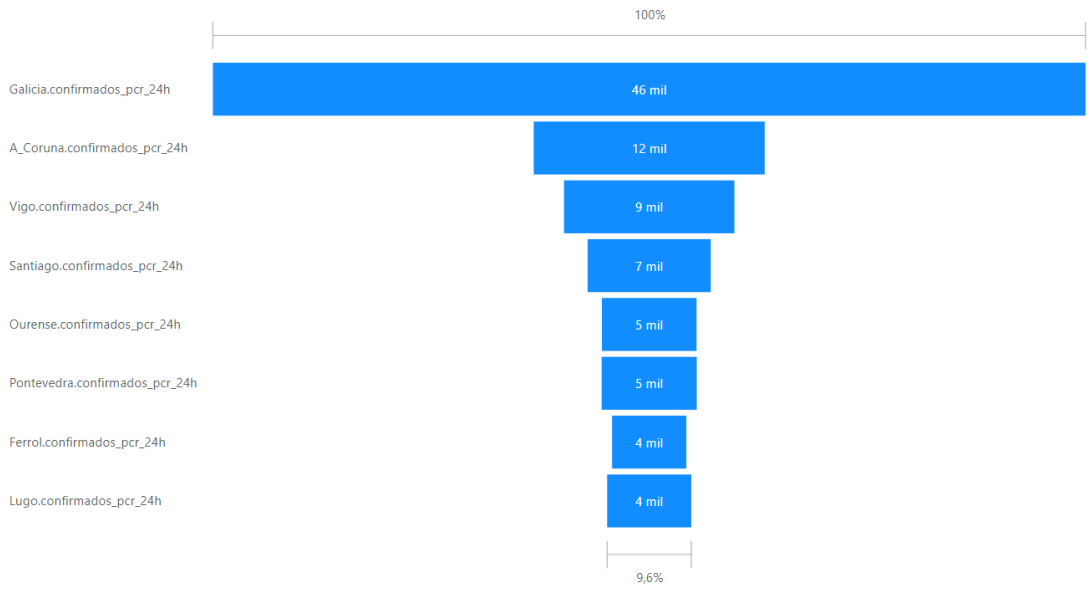

Figura 4.4: Casos totales de COVID-19 por PCR positiva en Galicia.

Analizando el comportamiento de UCIs a nivel gallego, llegamos a unos resultados interesantes que se pueden llegar a confundir a simple vista. Galicia, por líneas generales, ha seguido desde el inicio de la pandemia la misma tendencia de ocupación en lo que se refiere a las UCIs que la tendencia global española (la fuente empleada para los datos oficiales de la COVID-19 a nivel español ha sido: [27]), como podemos observar en el gráfico 4.5. Pero en cambio, como

se puede observar ligeramente en el gráfico 4.5, en el último tramo de enero del 2021, en la última semana, Galicia es la comunidad española con más ingresos en las UCIs para pacientes con la COVID-19.

Esta información se puede analizar mejor con un análisis concreto para cada una de las áreas sanitarias.

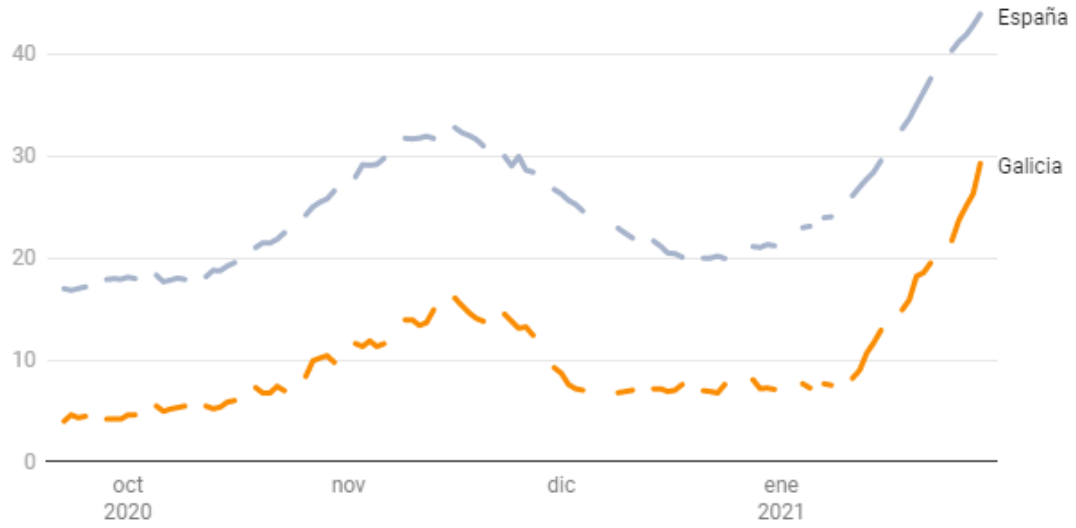

Figura 4.5: Tendencia de la situación de las UCIs gallegas y españolas. Casos diarios de ingresos en UCI secuenciados temporalmente.

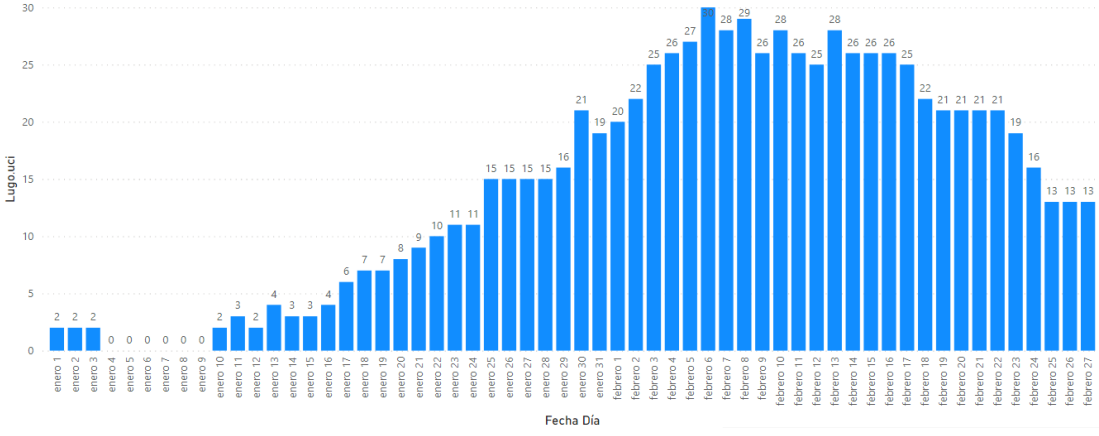

Figura 4.6: Situación de la UCI lucense.

Es para el área sanitaria lucense el ejemplo más claro (ver figura 4.6). Lugo, el día 4 de enero, presenta 0 hospitalizados en UCI por COVID-19, siendo una de las ocho UCIs españolas que no presentaba pacientes en críticos por motivo del virus, pero, en cuestión de días y en menos de una semana, se comienza a dar una tendencia al alza de ingresos de pacientes, llegando a estar el día 6 de febrero sobrepasando los límites de camas UCI que presentan los hospitales de dicho área sanitaria.

Desmarcando Vigo (área sanitaria de la cual hablaremos más adelante), el resto de áreas sanitarias gallegas también va a ser en estas fechas (1-10 de febrero) cuando lleguen a sus límites de camas UCI ocupadas, suponiendo esto uno de los mayores riesgos tratar a nivel hospitalario. Es en el comienzo del mes de marzo cuando las restricciones y la bajada de casos y hospitalizados hacen que la tendencia de todas las gráficas que se están estudiando comienza a descender.

Vigo (ver figura 4.7), ha presentado un comportamiento completamente diferente al del resto de áreas sanitarias llegando también al límite de camas UCI el 7 de febrero y teniendo que derivar pacientes a otras UCI, pero tardando en aliviar el número de ingresados y la presión hospitalaria en UCI hasta el final del mes de marzo. Más que una tendencia completamente al alza, las UCIs habían entrado ya en el primer trimestre del año con mucha presión de pacientes.

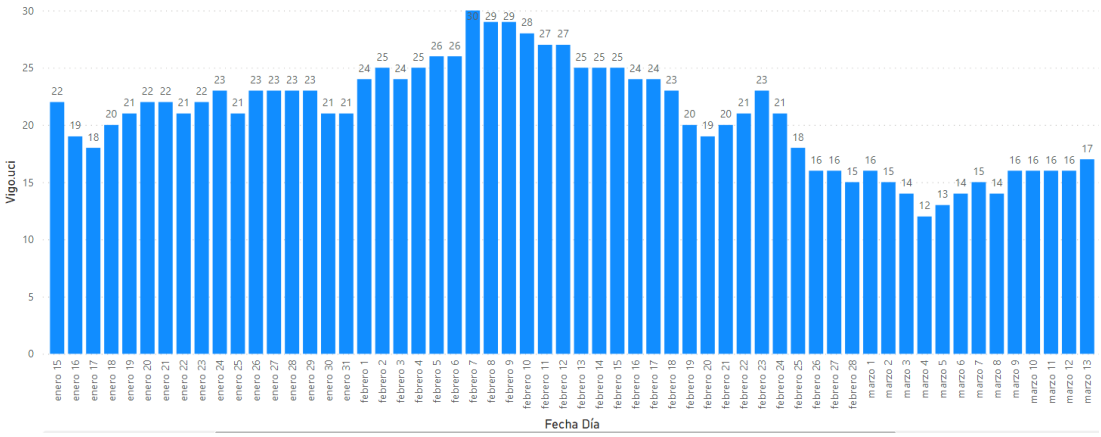

Figura 4.7: Situación de la UCI del área sanitaria de Vigo.

Ahora, una vez puesto en contexto la situación existente en las diferentes UCIs gallegas, se analizarán las tasas de letalidad y recuperación de casos de la COVID-19, un trabajo más amplio que nos permite estudiar más a fondo la enfermedad y el impacto de la misma en Galicia para las fechas marcadas, en el siguiente capítulo del trabajo.

Para comenzar, vemos la gráfica de los fallecidos por COVID-19 en el primer trimestre del año 2021.

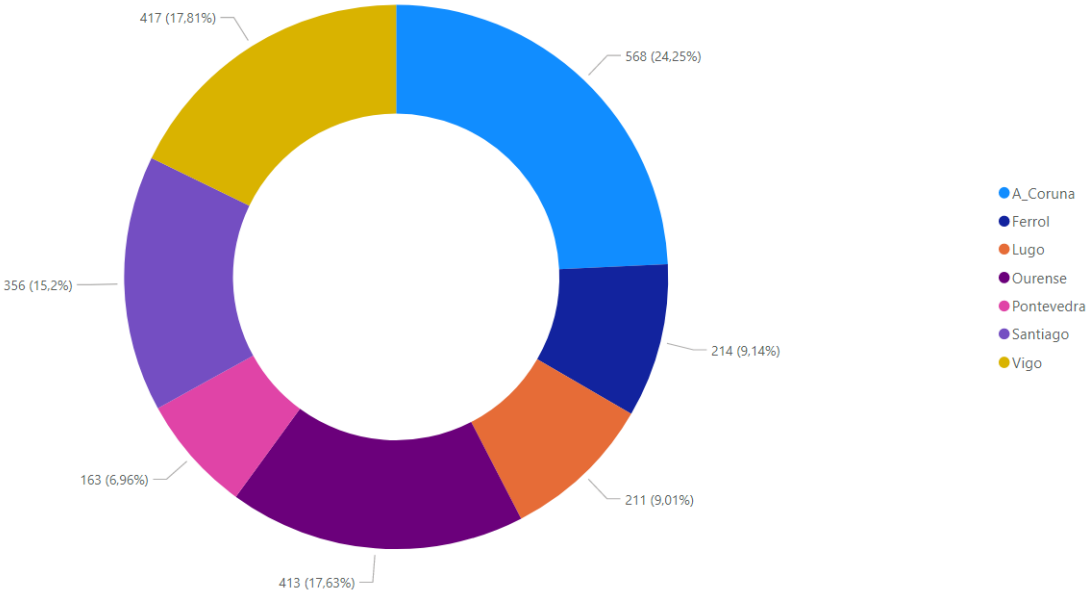

Figura 4.8: Fallecidos en el primer trimestre del año 2021 en Galicia.

Por lo tanto, una vez estudiados los casos (en este caso se tienen en cuenta todos los positivos a través de las diferentes pruebas) y los fallecidos totales, se analizará la letalidad.

La letalidad se refiere al cociente de fallecimientos en relación con las personas que se han contagiado de una enfermedad ("casos totales" en la tabla 4.10), en este caso de la COVID-19.

El área sanitaria con mayor letalidad es Ourense, por delante de Ferrol. Con mucha diferencia, el área sanitaria en la que menos fallecidos ha habido con respecto a los casos existentes por COVID-19 ha sido Pontevedra, con una diferencia clara con tan solo 163 muertos.

Área sanitaria	Fallecidos	Casos totales	Letalidad (%)
A Coruña	568	14631	0,038%
Ferrol	214	4017	0,053%
Lugo	211	4891	0,043%
Ourense	413	6230	0,066%
Pontevedra	163	5845	0,027%
Santiago	356	8157	0,043%
Vigo	417	9591	0,043%
Galicia	**2342**	**53362**	**0,043%**

Cuadro 4.10: Cuadro fallecidos/casos/letalidad.

Ahora, con objetivo concreto de revisar la evolución de la pandemia derivada de la COVID-19 en los diferentes municipios gallegos, basándonos y fijándonos en la IA, se sacan las siguientes conclusiones.

Observando y analizando la situación geográfica de las zonas, se puede apreciar un comportamiento muy diferente entre ellas, una curiosidad que, al principio del proyecto se había marcado como uno de los objetivos finales a observar y analizar.

Para ello, es importante analizar las zonas con más impacto de incidencia acumulada en los diferentes rangos temporales:

Momento	Municipio	IA14
Noviembre 2020	Ames	799,7091
Noviembre 2020	Ribadeo	771,0358
Noviembre 2020	Meaño	679,3618
Noviembre 2020	Carral	659,8680

Cuadro 4.11: Cuadro de las IA más altas en los últimos 14 días de noviembre de 2020.

Momento	Municipio	IA14
Enero 2021	Noia	774,4536
Enero 2021	O Grove	608,6142
Enero 2021	Ourense	549,5878
Enero 2021	Marín	536,3727

Cuadro 4.12: Cuadro de las IA más altas en los últimos 14 días de enero de 2021.

En los diferentes cuadros se observan diferencias; O Grove se mantiene en el top de los cuatro municipios gallegos con la incidencia acumulada más alta durante dos períodos de tiempo, en enero y en junio. El resto de municipios no se repiten para los diversos momentos temporales.

Momento	Municipio	IA14
Marzo 2021	Abadín	276,7049
Marzo 2021	Ribadumia	267,7376
Marzo 2021	A Peroxa	246,8700
Marzo 2021	Gondomar	206,2365

Cuadro 4.13: Cuadro de las IA más altas en los últimos 14 días de marzo de 2021.

Como ya se había detallado, es en noviembre de 2020 (cuadro 4.11), cuando las incidencias acumuladas en 14 días son las más altas respecto a los cuatro rangos temporales estudiados.

Son los municipios: Noia, O Grove, Marín y Ribadeo los únicos municipios puramente costeros que aparecen en el ránking de IA más alta para las diferentes fechas.

También se han estudiado los municipios con menor IA, son muchos los municipios sin ningún caso registrado en los últimos 14 días para todos los espacios temporales, con lo cual el análisis se ha enfocado en tratar de identificar algún parámetro o característica que indicara

Momento	Municipio	IA14
Junio 2021	Cerceda	530,7050
Junio 2021	Castrelo de Miño	420,3286
Junio 2021	O Grove	270,0675
Junio 2021	Baralla	209,6774

Cuadro 4.14: Cuadro de las IA más altas en los últimos 14 días de junio de 2021.

el motivo de esa incidencia acumulada nula, pero analizando tanto la zona geográfica y el nivel económico de la misma forma que se ha hecho para los municipios con mayor incidencia acumulada, no se han encontrado evidencias que demuestren un motivo en especial.

Una vez conocidos estos municipios, haciendo un estudio de la relación directa entre el nivel económico de los mismos y el impacto que el virus ha tenido sobre ellos (todos los datos relacionados con el PIB han sido sacados del Portal do Instituto Galego de Estatística [23]):

- No se encuentra ningún municipio con un valor del PIB inferior a 10.000 euros.
- Los municipios con un valor del PIB entre 10.000 euros y 20.000 euros.

  Ames, Meaño, Carral, Noia, O Grove, Ribadeo, A Peroxa, Gondomar y Baralla.
- Los municipios con un valor del PIB entre 20.000 euros y 30.000 euros.

  Marín y Ourense.
- Los municipios con un valor del PIB superior a 30.000 euros.

  Abadín, Ribadumia, Cerceda y Castrelo de Miño.

Por lo tanto, sabiendo que la media del PIB a nivel gallego es de 23.031 euros por habitante, es sorprendente que se sitúen tantos municipios (el 60% de los municipios con más incidencia acumulada vistos anteriormente) en el rango del PIB entre 10.000 euros y 20.000 euros, pudiendo afirmar que la COVID-19 ha tenido un impacto mayor en las zonas que se sitúan en este rango del PIB a nivel gallego.

Por último, respecto a los centros educativos por área sanitaria y municipios, se puede concluir que han sido espacios en donde se han respetado las medidas sanitarias, debido a que

fijándonos en los datos y en su progresión temporal, los datos han ido mejorando a lo largo del tiempo. Lo podemos observar en la figura 4.9, hasta noviembre del año 2020 se detectaron 1187 casos de COVID-19 en las aulas, desde noviembre hasta el 09/01/2021, 1033 casos. Fue para el primer trimestre del año donde los casos descendieron de manera abrupta hasta los 446 casos confirmados.

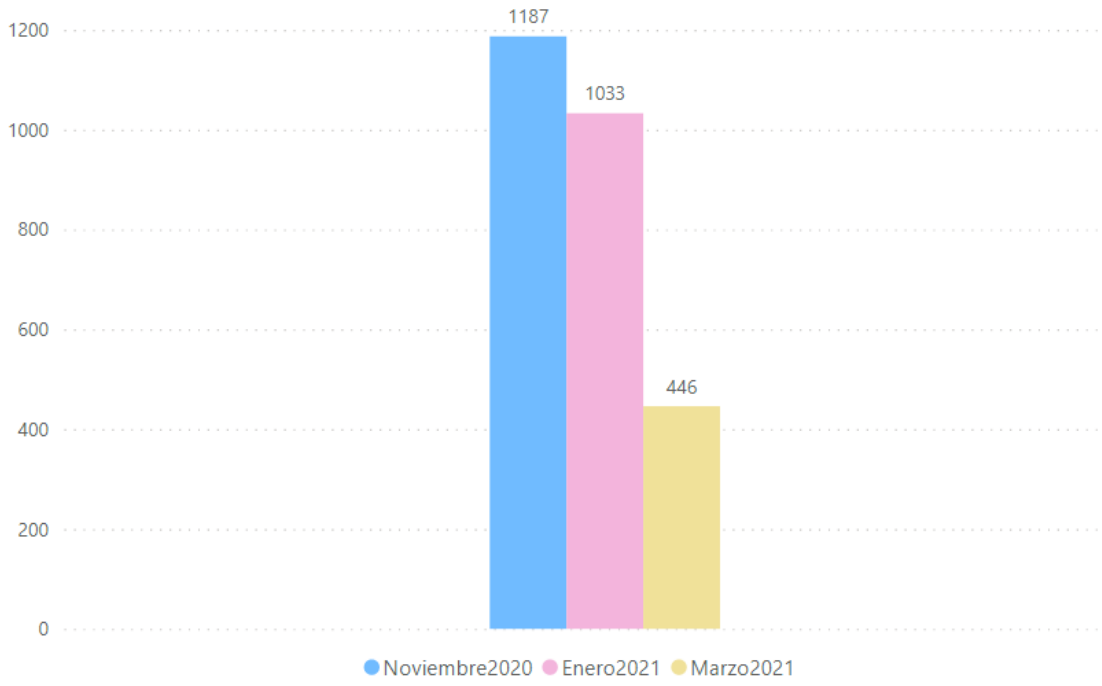

Figura 4.9: Número de positivos en centros escolares entre los diferentes períodos de tiempo.

A fecha 21/11/2020, la media de casos de COVID-19 desde el inicio del curso escolar en las aulas de todos los municipios gallegos fue de 2,1234 casos positivos por centro educativo, desmarcándose el concello de Carballo con 32 positivos en el CEIP Fogar y con ocho aulas cerradas.

En cambio, situados en el comienzo del segundo trimestre académico, a fecha 09/01/2021, la media (entre las fechas 21/11/2020 y 09/01/2021) de casos de COVID-19 en las aulas de los municipios gallegos fue de 1,8850, suponiendo una bajada respecto a la media de noviembre. Sólo 19 aulas fueron cerradas en toda Galicia, mientras que en noviembre fueron 51 aulas en total, lo que supone una bajada importante.

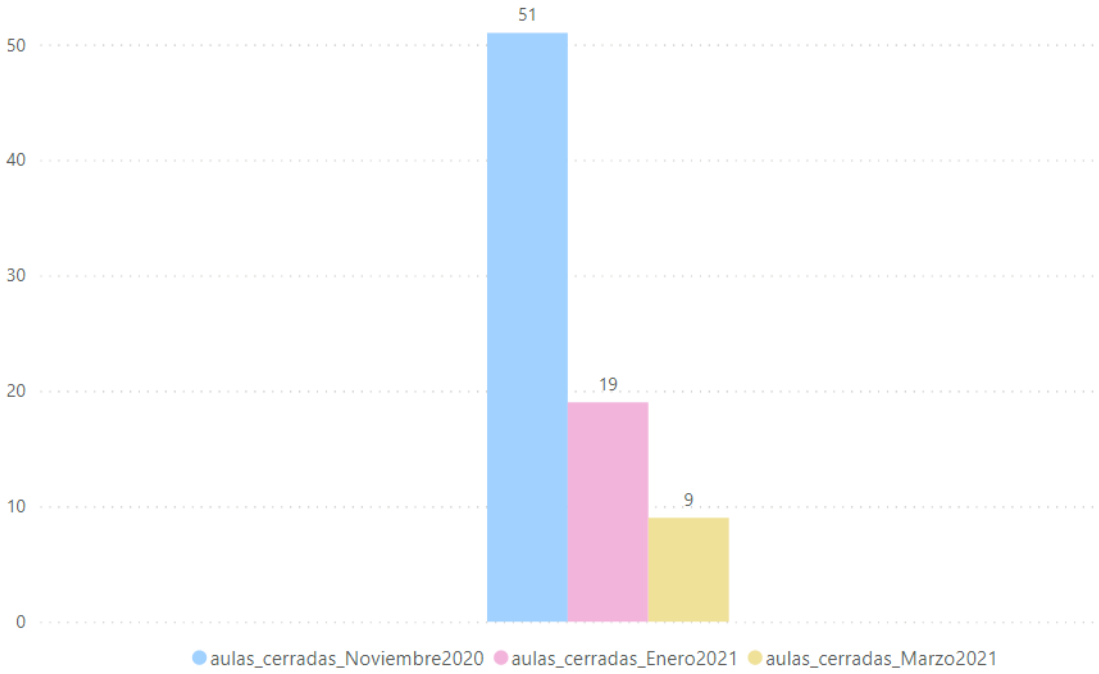

Figura 4.10: Número de aulas cerradas en centros escolares entre los diferentes períodos de tiempo.

Finalmente, en marzo, situados a finales del segundo trimestre académico para la mayoría de centros, a fecha 27/03/2021, la media (entre las fechas 09/01/2021 y 27/03/2021) de casos de COVID-19 en las aulas de los municipios gallegos fue de 1,9307, una ligera subida respecto a la media de casos positivos al comienzo del trimestre pero, fijándonos en el total de aulas cerradas, nueve, podemos afirmar que el control del virus ha sido exitoso, necesitando cerrar un menor número de aulas que en fechas anteriores. Hay que destacar que los datos para todos los municipios, tienen unos números inferiores respecto a positivos y aulas cerradas en comparación con los que tenían en meses anteriores pero, la media de casos y el total de aulas cerradas se ve alterada a razón de cuatro grandes brotes que se produjeron en el AS de A Coruña, con 21 positivos en el centro E.I. Kid's Garden y 14 en el centro CEE Nosa Señora do Rosario, por ejemplo.

En relación a los centros educativos cerrados en Galicia, antes de Navidades tan sólo se había cerrado un centro educativo, el centro E.I. Xirandola en Santiago, mientras que, al comienzo de enero los centros educativos cerrados ascendieron a seis. En adelante no volvieron a cerrarse más centros a razón de la COVID-19.

Capítulo 5

Análisis descriptivo de los datos mediante clustering

Las tareas en minería de datos pueden clasificarse en predictivas o descriptivas:

- Las tareas predictivas se basan en tratar de estimar valores futuros o desconocidos de variables de interés empleando otras variables.

  Algunos ejemplos de tareas predictivas: clasificación, regresión... Se estudian más a fondo en el capítulo 6.

- Las tareas descriptivas tratan de identificar patrones que explican o resumen los datos, son usadas para explorar las propiedades de los datos examinados y no para predecir nuevos datos.

  Algunos ejemplos de tareas descriptivas: clustering, asociación, correlaciones... Se estudian más a fondo en este capítulo 5.

Trataremos los dos tipos de tareas en minería de datos, comenzando por el análisis descriptivo.

5.1 Introducción al clustering no jerárquico

El clustering es una tarea cuya finalidad principal es conseguir el agrupamiento de diferentes conjuntos de objetos que no están etiquetados, consiguiendo así construir subconjuntos de datos, lo que se conoce como "clústers".

Cada uno de estos clústers dentro de un grafo, está formado por una colección de datos u objetos que, en términos de análisis son similares entre sí, pero en cambio contienen elementos diferenciales en comparación con otros objetos que pertenecen al conjunto de los datos y

que pueden llegar a construir un clúster independiente [28].

Se empleará una técnica clustering no jerárquico, pero antes de comenzar hay que explicar la diferencia entre los métodos.

Para entender más a fondo el sentido del nombre, se hacen llamar "jerárquicos" a los que conforman un grupo que contiene una estructura en forma de árbol, de manera que clústeres que son de niveles inferiores van englobándose en otros clústeres que pertenecen a niveles superiores. En cambio, el clustering "no jerárquico" funciona de manera diferente, los grupos se diferencian de manera automática por el análisis que se configura, sin que unos tengan dependencia con otros.

Para nuestro caso, los métodos no jerárquicos empleados en el estudio producir clústeres disjuntos (cada caso pertenece únicamente a un clúster), o clústeres solapados (un caso puede pertenecer a varios grupos), siendo este último un tipo de clúster poco utilizado.

Existen muchas técnicas variadas dentro de los métodos jerárquicos y de los no jerárquicos como se observa en la figura 5.1.

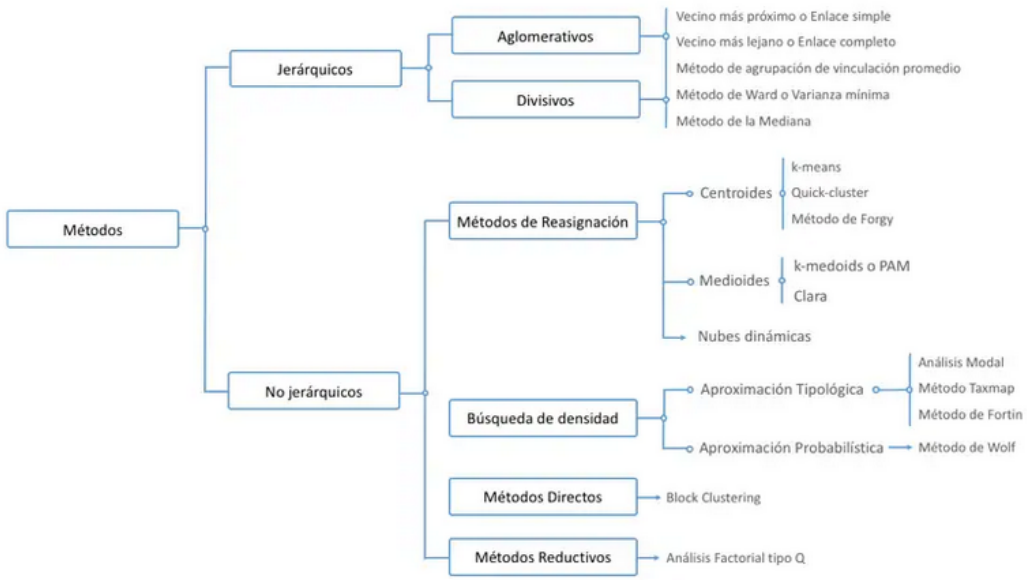

Figura 5.1: Clasificación de métodos jerárquicos y no jerárquicos. Fuente: [1]

5.1.1 Técnica clustering no jerárquico: explicación teórica de k-means

Para el estudio, la técnica empleada será k-means, que como se puede ver en el esquema 5.1, es un método no jerárquico, de reasignación y basado en centroides como se explicará a continuación.

Al contrario que para el clustering jerárquico, en este caso el número de clústeres se debe conocer previamente. En este tipo de clústeres, los clústeres no jerárquicos, los datos que tratamos se dividen en k particiones o número de grupos donde cada partición se identifica con un clúster.

Primero se comenzaría seleccionando los centroides iniciales a los que denominaremos k (número de clústeres que se desea), el siguiente paso será asignar o identificar cada observación que se aprecie, al clúster que le sea más cercano.

Después, se reasignarán o se relocalizarán las observaciones a alguno de los k clúster en función de las reglas de parada establecidas.

Por último, se debe de parar en el momento en el que no haya reasignación de los puntos o si la reasignación no satisface a dicha regla de parada. Si no, volveríamos al segundo paso, donde asignamos las observaciones a los clústeres más cercanos.

Hay que tener en cuenta que la técnica k-means necesita un conocimiento previo del número de clúster como se ha dicho, lo que explica que los algoritmos empleados sean altamente sensibles a las particiones que se hagan al inicio. Se han de identificar los centros de los clústeres antes de que la técnica pueda seguir con las observaciones.

El método k-means puede funcionar bien en una gran cantidad de problemas, pero se ha realizado un estudio previo y se ha entendido bien de dónde viene y cómo calculamos nuestro dataset. Este análisis previo es el que nos ha dado las pistas para escoger el algoritmo más apropiado para realizar una buena segmentación.

El objetivo perseguido de k-means se basa en conseguir una partición de un conjunto de N elementos (D dimensiones) en k subconjuntos o grupos disjuntos S_i, consiguiendo minimizar el error cometido.

$$\epsilon = \sum_{i=1}^{k} \sum_{\boldsymbol{x_j} \in S_i} dist(\boldsymbol{x_j}, \mu_i)$$

Donde, para la formula anterior, $dist(\boldsymbol{x_j}, \mu_i)$ es la función de distancia deseada y μ_i es el representante o centroide de cada grupo S_i [29].

Las principal ventaja del método k-means es su sencillez, así como su rapidez. Pero es necesario decidir el valor de k como ya se ha detallado anteriormente y el resultado final, depende de la inicialización de los centroides. Converge a un mínimo local y no a un mínimo global [30].

5.1.2 Conjunto de datos del clustering

El dataset que ha sido usado en el trabajo de clustering ha sido el compuesto por:

- Habitantes.
- Casos.
- Incidencia en 14 días.
- Municipios.
- Valor del PIB para cada municipio.
- Zona geográfica: costa o interior.

Situándose los datos dentro de las mismas fechas que se han usado en el análisis exploratorio de los datos de municipios (sección 4.4):

- Noviembre de 2020.
- Enero de 2021.
- Marzo de 2021.
- Junio de 2021.

fecha	codigo_municipio	municipio	habitantes	casos_14d	IA14
2021-01-01	15002	Ames	31793	55	172,99406
2021-01-01	27051	Ribadeo	9854	21	213,11143
2021-01-01	32091	Vilardevós	1846	0	0
2021-01-01	36027	Meaño	5272	16	303,49014
2021-01-01	15021	Carral	6408	39	608,61423
2021-01-01	15082	Teo	18579	42	226,06168
2021-01-01	27019	Foz	9980	11	110,22044

Cuadro 5.1: Cuadro de muestra del conjunto de datos del nivel de incidencias en 14 días por municipios.

5.1.3 Código del clustering en RStudio

Una vez comprendida y analizada la parte teórica del clustering con el método no jerárquico, en especial k-means, se ha procedido a la práctica de un pequeño estudio empleándolo.

En el anexo B se detalla el código empleado en RStudio y la ejecución de las gráficas que nos ayudan a obtener el número de k.

El paquete kml es un paquete ofrecido por R que nos proporciona diferentes técnicas variadas para tratar con los valores perdidos en las trayectorias. La interfaz gráfica que aporta el paquete es de ayuda ya que nos permite elegir el número adecuado y apropiado de clústeres cuando los criterios clásicos no son lo suficientemente eficiente. Se ejecuta el algoritmo en numerosas ocasiones, empleando diferentes tipos de condiciones de partida y agrupaciones a buscar, que se irán viendo, ayudándonos del código empleado, evitando así tener que hacer remuestreos manuales. Además, el paquete nos proporciona la opción de exportar representaciones gráficas resultantes de los agrupamientos.

A continuación, se hace una breve explicación del código paso a paso:

Para comenzar, se importan las librerías y se realiza la carga de datos desde los diferentes ficheros (ficheros ya secuenciados en los diferentes rangos de meses).

Se hace la unificación de los cuatro dataframes, se reemplazan los valores que no aplican por ceros.

A la hora de trabajar con los tres modelos (sección 5.1.3), la metodología ha sido la misma

en todos:

- Creación de un objeto clusterLongData para usarlo en kml(). ClusterLongData necesita saber que columnas son de serie temporal (timeInData).
- Creación del modelo.
- Lanzamiento del modelo.
- Visualización de resultados.

Se utiliza el paquete kml que permite agrupar datos según patrón de sus trayectorias, es decir, es una implementación de k-means diseñada específicamente para analizar datos longitudinales. [17]

Se realizan e implementan tres modelos diferentes con la idea de comparar los resultados incluyendo variados parámetros en cada uno, analizaremos los resultados obtenidos.

Calinski-Harabasz [31] es un análogo multivariado del estadístico F de Fisher. Reconoce cualquier grupo convexo. Al tratarse de un dispositivo heurístico, la forma correcta de usar Calinski-Harabasz es comparar las soluciones, existen tres tipos de soluciones:

- Soluciones de agrupación obtenidas con los mismos datos.
- Soluciones que difieren en la cantidad de agrupaciones.
- Soluciones que difieren en el método de agrupación utilizado.

En nuestro caso compararemos las soluciones de agrupación obtenidas con los mismos datos.

Modelos estudiados

Para los tres modelos, se analizarán los resultados mediante dos tipos de gráficas, en las que es necesario comprender sus leyendas:

- Rerolling: indica el número de clúster que debe de contener. Siendo el eje y = trayectorias y el eje x = agrupaciones.
- Times: indica las trayectorias de los diferentes clúster. Siendo el eje y = tiempos y el eje x = trayectorias.

- 3 criterios estimados por el algoritmo Calinski-Harabasz: Calinski-Harabasz, Ray-Turi, Davies-Bouldin.

  Criterio Calinski-Harabasz: también se le conoce como criterio de relación de varianza, es el resultado de la suma de la dispersión entre conglomerados para todos los conglomerados y la dispersión entre conglomerados, cuanto mayor sea la puntuación, mejores serán los resultados.

  Criterio Ray-Turi: propone un índice que incorpora una función multiplicadora (para penalizar la selección de un pequeño número de clusters) a la relación entre distancias intra-clúster e inter-clúster.

  Criterio Davies-Bouldin: se trata de un criterio que se basa en la "similitud" promedio entre los conglomerados, la "similitud" se define como la medición que hace la comparativa del tamaño de los conglomerados en sí con la distancia entre los propios conglomerados. Cuanto menor sea el índice de Davies-Bouldin, mejor separación entre los conglomerados.

Modelo 1: RandomK.

k individuos se asignan aleatoriamente a un grupo, el resto de individuos no se asignan. Cada semilla es un solo individuo y no un promedio de varios individuos. Este método produce semillas iniciales que no están cerca unas de otras, de modo que este método puede producir semillas iniciales que son de diferentes grupos (posiblemente una semilla en cada grupo) lo que acelerará la convergencia. Nos devuelve que la mejor opción es tener 4 clústeres, como se puede ver en las figuras 5.2 y 5.3 explicadas a continuación.

Respecto a la figura 5.2, podemos observar que en el gráfico de la izquierda, las particiones con el mismo número de clúster se clasifican de acuerdo con el criterio de Calinski-Harabasz en orden decreciente, siendo mejor el primero. De todas las particiones, se selecciona una (punto negro).

Se selecciona el clúster óptimo, marcado con un punto negro. En nuestro caso, se determina cuatro clústeres.

Las trayectorias que define el criterio de Calinski-Harabasz se presentan en el lado derecho de las ventana, representando los cuatro clústeres trazando los datos longitudinales y resaltando la estructura del grupo de la partición seleccionada utilizando colores variados.

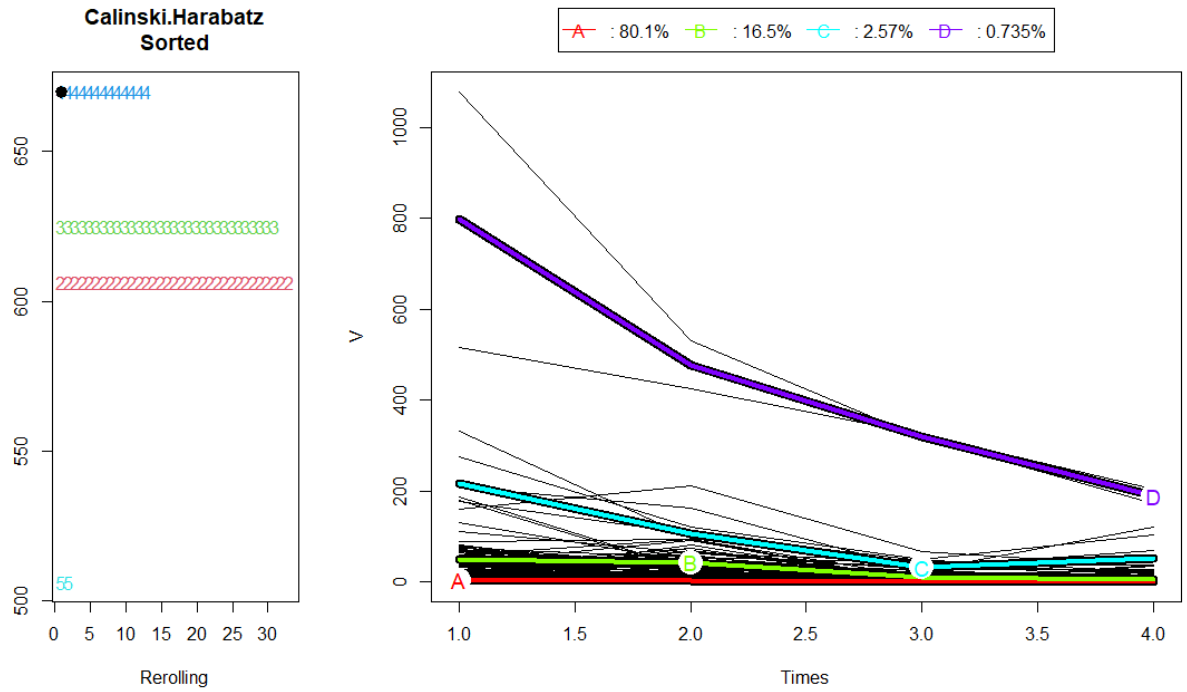

Figura 5.2: Modelo 1: RandomK.

En la figura 5.3, se muestra el criterio de calidad para seleccionar el número "correcto" de clústeres siguiendo los criterios estimados (Calinsky-Harabasz, Ray-Turi, Davies-Bouldin). El eje x nos indica el número de clústeres y el eje y, la puntuación del parámetro de evaluación.

Podemos decir que los criterios concuerdan con la elección de cuatro clústeres puesto que cuatro han sido los criterios que han coincidido.

Además, considerando los criterios como vectores en el espacio de las variables, un elevado valor de la distancia entre dos criterios nos indicará un alto grado de diferencia entre ellos.

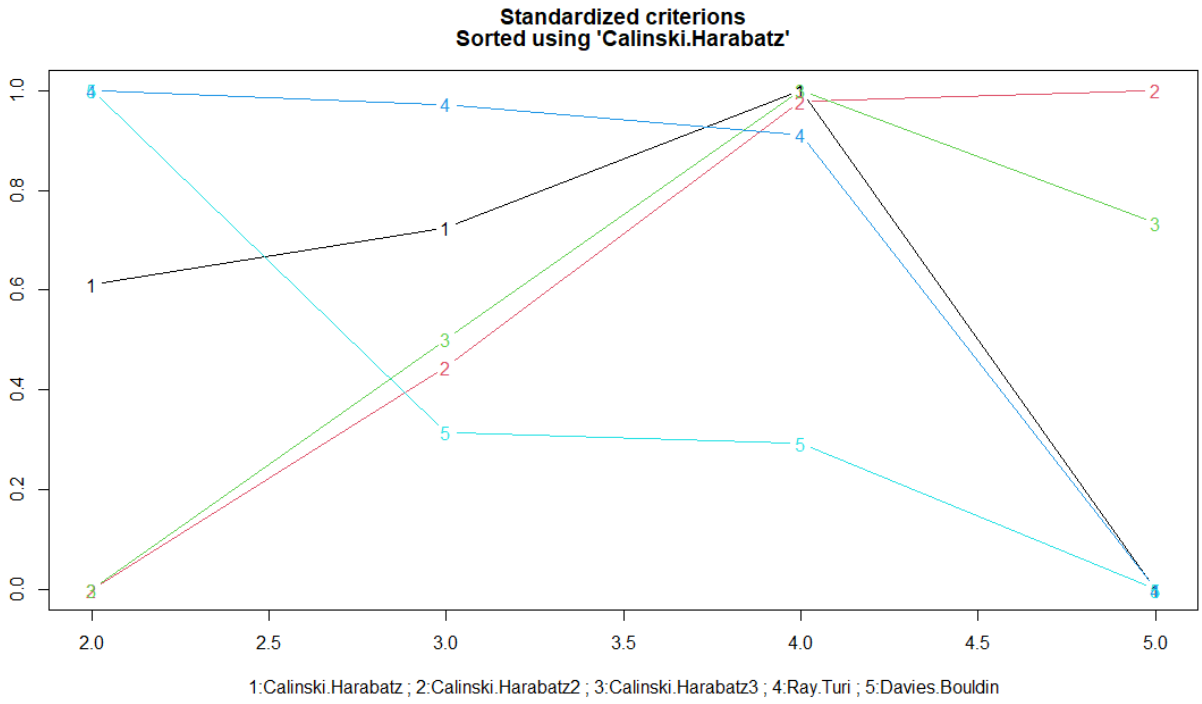

Figura 5.3: Criterio estimado para el modelo 1: RandomK.

Modelo 2: RandomAll.

Todos los individuos se asignan aleatoriamente, al azar, a un grupo con al menos un individuo en cada grupo. Este método produce semillas iniciales cercanas entre sí. Nos devuelve que la mejor opción es tener 4 clústeres, como se puede ver en las figuras 5.5 y 5.4, explicadas a continuación.

En la figura 5.4, el gráfico de la izquierda nos indica todas las particiones que han sido encontradas por la librería kml() siguiendo el criterio de Calinski-Harabasz.

En nuestro caso, el modelo 2 nos indica que el clúster óptimo es de cuatro, señalado con el punto negro. El gráfico de la derecha recoge las trayectorias de los cuatro clústeres, es decir, los casos de COVID-19 en los meses de noviembre, enero, marzo y junio.

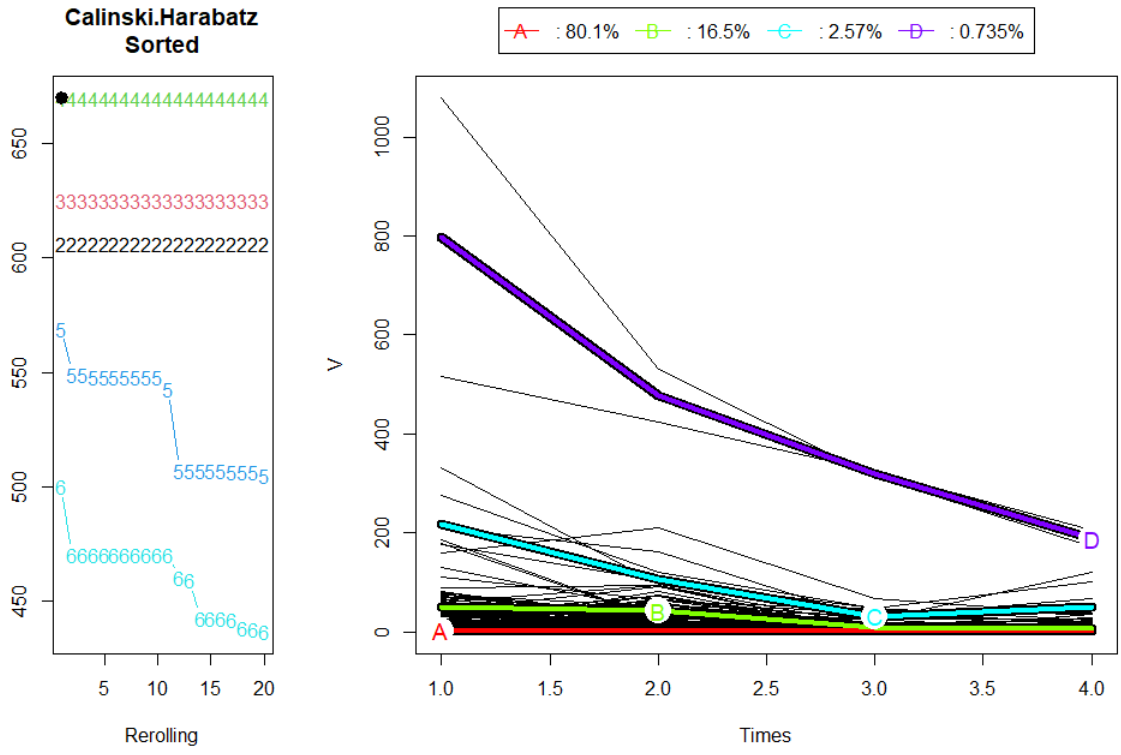

Figura 5.4: Modelo 2: RandomAll.

Para la figura 5.5, siguiendo los criterios estimados por Calinski-Harabasz, Ray-Turi, Davies-Bouldin, el punto de mayor coincidencia de los autores sería en el número cuatro. Por tanto, podríamos decir que, según el criterio de calidad, el número de clústeres es cuatro.

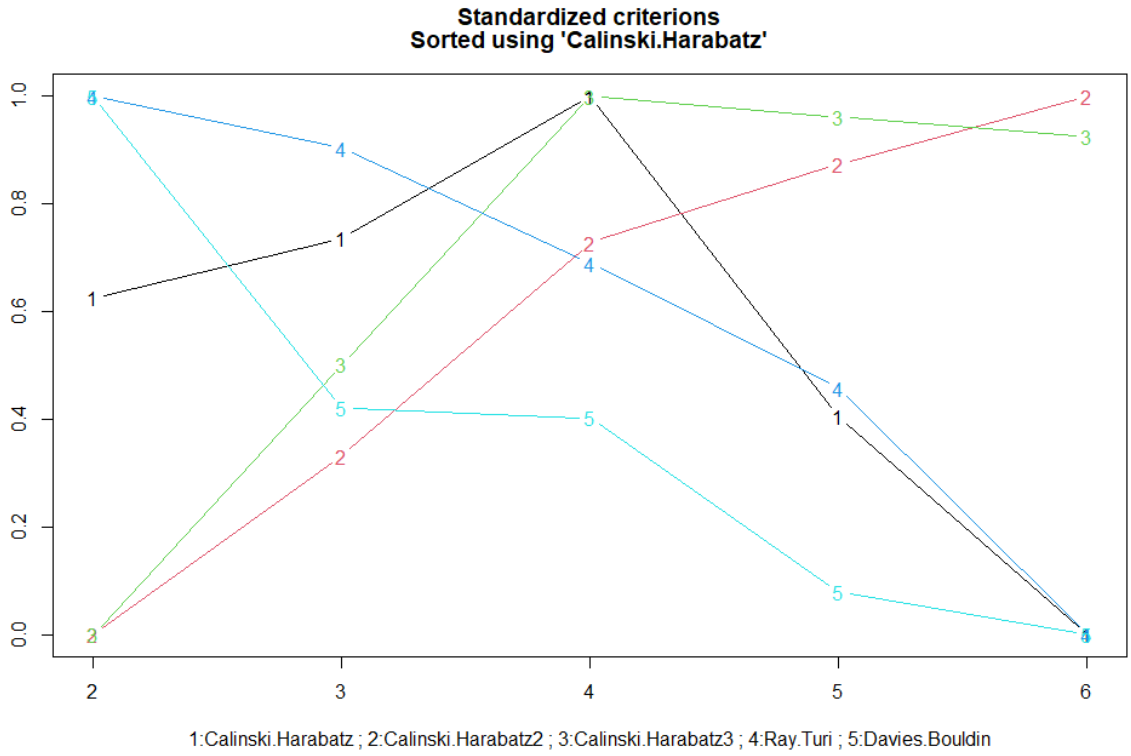

Figura 5.5: Criterio estimado para el modelo 2: RandomAll.

Modelo 3: MaxDist.

k individuos se eligen de forma incremental. Los dos primeros son los individuos que son los más alejados el uno del otro. Los siguientes individuos se agregan una a la vez y son los individuos más alejados de los que ya están seleccionados. El "más lejano" es el individuo con el que mayor distancia presenta de los individuos seleccionados. Si D es el conjunto de individuos ya seleccionados, entonces el individuo que se agregará es el individuo i para quien

$$s(\boldsymbol{i}) = minj.DDist(\boldsymbol{i}, j)$$

es máximo. Nos devuelve que la mejor opción es tener cuatro o cinco clústeres, como se puede ver en las figuras 5.6 y 5.7 explicadas a continuación.

En la figura 5.6, el gráfico de la izquierda recoge todas las particiones que han sido encontradas siguiendo el criterio de Calinski-Harabasz. Nos indica que el cinco son los clústeres óptimos.

El gráfico de la derecha recoge las trayectorias de los cuatro clústeres, es decir, los casos de COVID-19 en los meses de noviembre, enero, marzo y junio.

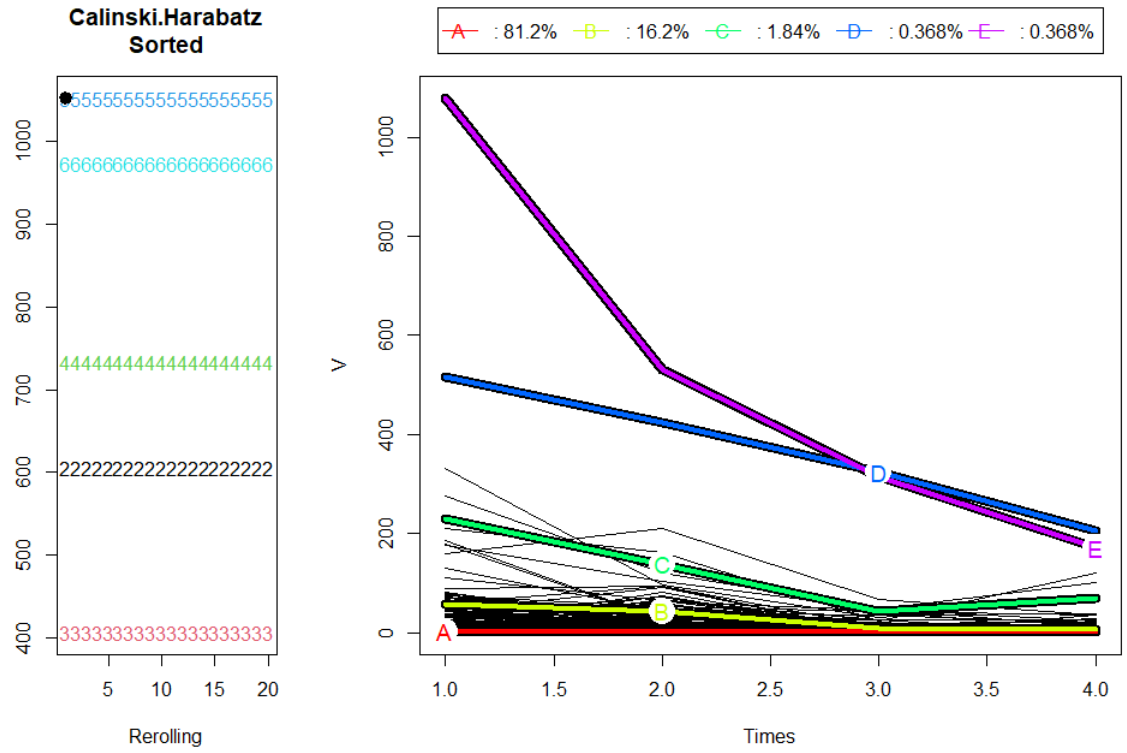

Figura 5.6: Modelo 3: MaxDist.

Por último, para la figura 5.7, el criterio estimado por Calinski-Harabasz, Ray-Turi, Davies-Bouldin, esto nos indica que podrían ser cuatro o cinco clústeres puesto que son los puntos donde mayor coincidencia de criterios hay.

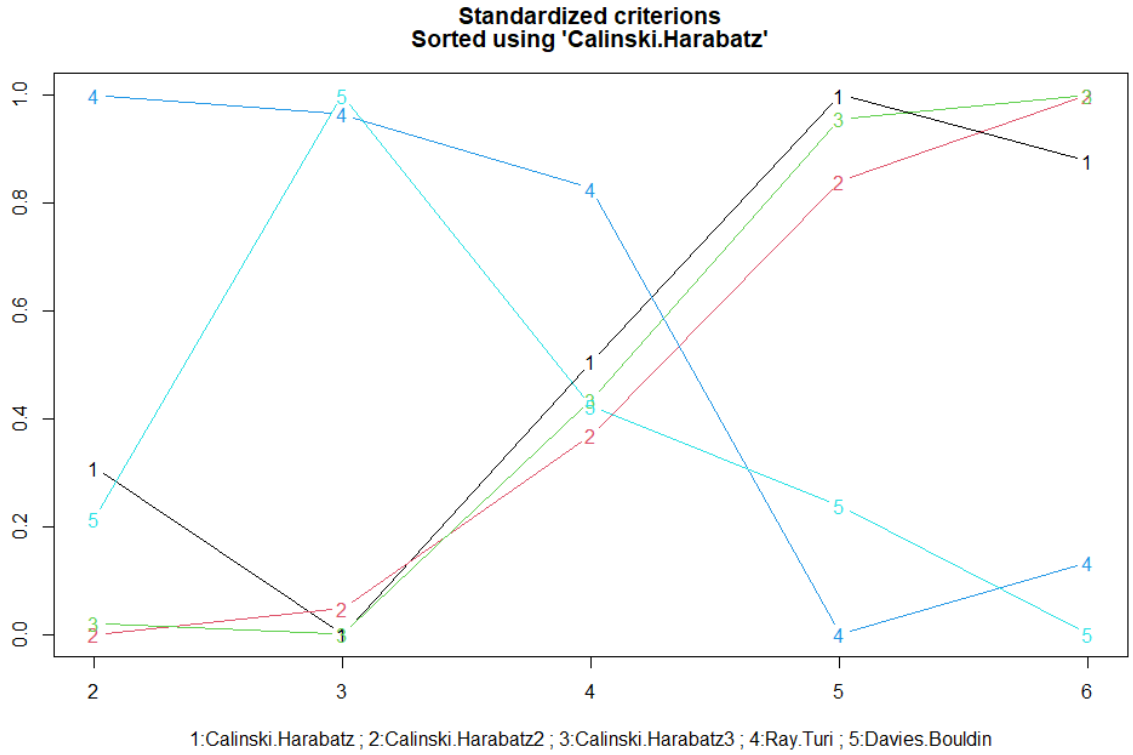

Figura 5.7: Criterio estimado para el modelo 3: MaxDist.

5.1.4 Resultados y conclusiones del clustering

Los resultados de los tres modelos nos arrojan que la partición con cuatro clústeres parece ser la más relevante. No obstante, el criterio del modelo 3 es algo dubitativo porque nos da la opción de tener cinco clústeres también.

En cuanto a las trayectorias de los tres modelos, el patrón que sigue es muy similar. Según sus particiones, se pueden encontrar cuatro tipos de casos:

- **A**: bajo, entonces trayectorias crecientes.
- **B** y **C**: algo más altas que **A**.
- **D**: alto, entonces trayectorias decrecientes.

Por tanto, esto se podría traducir que en trayectorias crecientes, mayor número casos de la COVID-19, mientras que en trayectorias decrecientes, menor número de casos.

En conclusión, podríamos decir que los tres modelos elaborados coinciden que el número óptimo de clústeres debe de ser cuatro, aunque el modelo 3 sugiere la opción de tener cinco clústeres.

Así mismo, los tres modelos arrojan que el mayor número de trayectorias medias de los clústeres se da en el grupo A, siendo al menos un 80% en los tres algoritmos coincidiendo con el primer tiempo que nuestro caso hace referencia al mes de noviembre de 2020.

En cuanto al modelo 1 y modelo 2, coinciden en los resultados obtenidos en el algoritmo (k = 4), aunque en el primer caso hemos utilizado randomK donde k individuos se asignan aleatoriamente a un grupo. Mientras que, en el segundo caso, hemos utilizado randomAll el cual todos los individuos se asignan aleatoriamente a un grupo con al menos un individuo en cada grupo.

Los resultados de las gráficas de los criterios estimados para los diferentes modelos 2 y 3, nos abordan unos resultados muy diferentes respecto al modelo 1, obteniendo los modelos 2 y 3 criterios discordantes en comparación al modelo 1.

- Número de coincidencias de criterios en el modelo 1: 9-10.
- Número de coincidencias de criterios en el modelo 2: 8.
- Número de coincidencias de criterios en el modelo 3: 4-6.

Respecto a los tres algoritmos creados, si tuviéramos que elegir uno de ellos nos quedaríamos con el **modelo 1** debido a que cumple con los principios de los criterios de calidad, los clústeres son los más compactos/concordantes, esto se puede ver en la figura 5.3 donde se aprecian los criterios concordantes y los clústeres están bien separados entre sí (de la misma manera que el modelo 2, pero este último posee un número menor de criterios concordantes).

Las diferentes agrupaciones han sido extraídas a razón de los llamados centroides. Al hablar del centroide de un cluster, hablamos del punto equidistante de los objetos pertenecientes a dicho cluster. Para ello, el método k-means se basa en las fases de inicialización, asignación y actualización. En la primera de ellas, se asignan los centroides para los k grupos de forma aleatoria. En la fase de asignación se agrupa cada dato al centroide más cercano que le corresponda y finalmente, en la etapa de actualización, se actualiza la posición del centroide a la media aritmética de las posiciones de los datos asignados a dicho grupo.

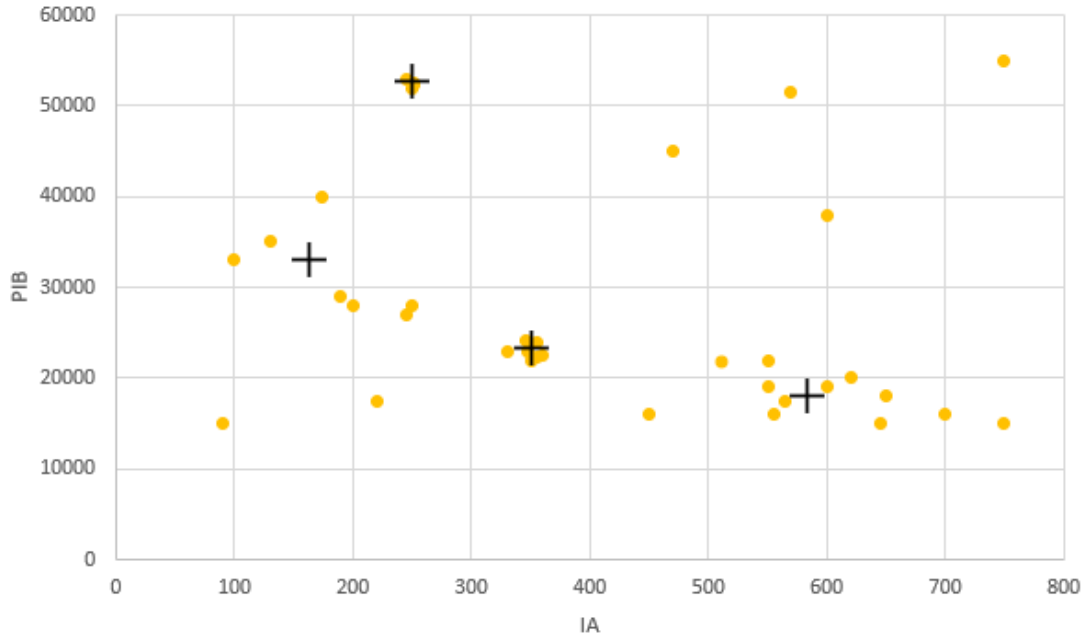

Figura 5.8: Representación de los centroides del clustering.

Siendo X el eje de las IAs y el eje Y los valores del PIB, los valores de los diferentes centroides se sitúan de la siguiente manera:

Centroides	Eje X	Eje Y
Agrupamiento 1	588	18365
Agrupamiento 2	350	23351
Agrupamiento 3	167	32512
Agrupamiento 4	252	52003

Cuadro 5.2: Cuadro de información de los valores de los centroides.

Como se ha explicado anteriormente, el clustering nos permite obtener grupos a partir del conjunto de los datos. Para ello, los datos son agrupados buscando maximizar la similitud entre los elementos y minimizar la similitud entre grupos. Debido a estas razones, los objetos de un mismo grupo son similares entre sí y distintos de los objetos de otro grupo.

El modelo escogido, el modelo 1, que emplea el método randomK, concluye que el número de clústeres óptimo para nuestro conjunto de datos es 4 (k = 4), y el agrupamiento que hace de ellos, agrupado por los valores de la IA en los últimos 14 días y el valor del PIB para los diferentes municipios ha sido:

- Agrupamiento 1: Conjunto de datos aleatoriamente asignados:

  Valores de IA => 509,587.

  Valores del PIB <= 21932.

- Agrupamiento 2: Conjunto de datos aleatoriamente asignados:

  Valores de IA <= 366,193.

  Valores del PIB que están entre 21932 y 26809.

- Agrupamiento 3: Conjunto de datos aleatoriamente asignados:

  Valores de IA <= 331,564.

  Valores del PIB que están entre 26809 y 51666.

- Agrupamiento 4: Conjunto de datos aleatoriamente asignados:

  Valores de IA <= 254,980.

  Valores del PIB => 51666.

Una vez se conocen dichos valores extremos (máximo o mínimo) para los atributos de las diferentes agrupaciones, se pueden reafirmar las conclusiones extraídas en la sección 4.4, donde se afirma que los municipios gallegos que se sitúan en el rango del PIB entre 10.000 euros y 20.000 euros por habitante, habían sufrido un impacto mayor a razón de la COVID-19 a lo largo de los meses.

En cambio, no se puede afirmar que los municipios gallegos con mayor PIB hayan tenido una IA tan baja como dicen los resultados del agrupamiento, debido a que son pocos los municipios gallegos (12) con un valor de PIB superior a 51666 y ha podido ser fruto de la aleatoriedad del método.

El resto de variables empleadas para el trabajo de clustering no tienen influencia en la formación de dichas agrupaciones, una situación que se entiende ya que los municipios costeros gallegos suman un total de 69, en cambio, son 246 los municipios gallegos de interior. Además, las variables habitantes y casos son dependientes entre sí (la IA depende de los casos y del número de habitantes).

Capítulo 6

Análisis predictivo

Ya finalizado todo el análisis exploratorio y exhaustivo de los datos que componen los diferentes conjuntos de los que se disponen para el estudio, y puesto en contexto la situación gallega en la que nos encontramos para el primer trimestre, se tratará de trabajar con modelos de predicción.

La analítica predictiva es crucial en la lucha para erradicar la COVID-19, ya que nos permite hacernos una idea del comportamiento que ha tenido la pandemia en cierta manera, pudiendo prever y anticipar situaciones de riesgo, así como estimar datos relevantes para la sociedad. Son estos los motivos por los cuales los gobiernos le han destinado tanta importancia.

Tal y como se define en [32], un modelo predictivo es un método de análisis de datos y estadísticas para definir hipótesis o deducir resultados o sucesos futuros. "El modelado genera predicciones con un grado de probabilidad según las variables analizadas. La mayoría de las veces el evento que se quiere predecir es en el futuro, pero el modelado predictivo puede aplicarse a cualquier tipo de evento desconocido, independientemente de cuando haya ocurrido".

Existen dos tipos de modelos predictivos: los modelos de clasificación y los modelos de regresión [33].
Los primeros, los de clasificación, permiten predecir la pertenencia a una clase, y los segundos nos permiten predecir un valor. Basándonos en nuestro objetivo final, el de predecir resultados que puedan ocurrir en un futuro, fijándonos en datos del pasado, para nuestro proyecto en concreto, el idóneo sería un modelo de regresión.

Existen técnicas, que pueden ser tanto específicas para cada tipo o que funcionen para ambos sin ningún tipo de problema, se mencionarán algunos de ellos:

- Árboles de decisión: es una técnica que permite el estudio de la toma de decisiones

secuenciales mediante el análisis del uso de diferentes resultados y probabilidades relacionadas con los mismos. Se utilizan para generar sistemas expertos, búsquedas binarias y árboles de juegos [34].

- Máquinas de vectores de soporte: permiten encontrar la mejor forma de clasificar entre varias clases. La mejor solución para esta técnica es maximizar el margen de separación entre clases tanto como sea posible. El vector que define este borde de separación es el vector de soporte [35].
- Regresión logística y regresión lineal: la regresión lineal es un algoritmo de regresión utilizado para predecir un valor numérico, mientras que la regresión logística es un algoritmo de clasificación utilizado para predecir entre dos opciones.
- K-Vecinos más cercanos: es uno de los algoritmos de clasificación más básicos y esenciales en el aprendizaje automático. Pertenece al dominio del aprendizaje supervisado y encuentra una aplicación intensa en el reconocimiento de patrones, la minería de datos y la detección de intrusos [36].
- Redes neuronales: se trata de una metodología computacional que ha sido construida a razón de diferentes aportaciones científicas que se han ido registrando a lo largo de la historia. Consiste en un conjunto de unidades, llamadas neuronas artificiales, que están conectadas entre sí para transmitirse señales [37].
- Análisis Bayesiano: la estadística bayesiana se basa en la probabilidad subjetiva, trabaja con la constante actualización de diferentes evidencias, teniendo en cuenta unos conocimientos que han sido adquiridos previamente, sumándoselos a la posterior investigación. El entender los resultados supone conocer la especificación de las diferentes hipótesis a contrastar, así como la probabilidad de que sucediera previamente al comienzo del estudio [38].

6.1 Trabajo práctico de análisis predictivo

Una vez comprendida y analizada la parte teórica del análisis predictivo, se ha procedido a la práctica.

Los datos empleados han sido:

- Casos de COVID-19 acumulados en Galicia: nos permite conocer el número de casos diarios.
- Altas acumuladas en Galicia: nos permite conocer el número de altas diarias.
- Fallecidos acumulados en Galicia: nos permite conocer el número de fallecidos diarios.
- Número de pruebas PCR diarias confirmadas.

El espacio temporal que se ha utilizado para este trabajo predictivo ha sido de 435 días, desde el 2020-04-30 hasta el 2021-07-08.

	Fecha	Galicia.casos.acum	Galicia.altas.acum	Galicia.fallecidos.acum	Galicia.confirmados.PCR.diarias	Galicia.casos.diarios	Galicia.altas.diarias	Galicia.fallecidos.diarios
0	2020-04-30	9579	5573	551	NaN	38	180	0
1	2020-05-01	9617	5816	558	NaN	35	243	7
2	2020-05-02	9652	5981	563	NaN	33	165	5
3	2020-05-03	9685	6075	569	NaN	22	94	6
4	2020-05-04	9707	6234	575	NaN	76	159	6
...	...	...	...	...	...	...	...	...
430	2021-07-04	130626	125929	2437	219.0	244	80	0
431	2021-07-05	130870	125992	2437	253.0	235	63	0
432	2021-07-06	131105	126055	2437	230.0	367	63	0
433	2021-07-07	131472	126142	2438	366.0	412	87	1
434	2021-07-08	131884	126239	2439	410.0	559	97	1

435 rows × 8 columns

Cuadro 6.1: Dataset empleado para el análisis predictivo.

Se ha empleado la técnica anteriormente explicada, las redes neuronales. En concreto se ha entrenado un MLP, se trata de una red neuronal que conecta varias capas en un gráfico dirigido, lo que supone que la ruta de la señal a través de los nodos solo va en una única dirección. Cada nodo, además de los nodos de entrada, tiene una función de activación no lineal. Un MLP utiliza la propagación hacia atrás como una técnica de aprendizaje supervisado. Dado que existen múltiples capas de neuronas, MLP es una técnica de aprendizaje profundo. MLP se usa para resolver problemas que requieren aprendizaje supervisado.

6.1.1 Código del análisis predictivo en Python

En el anexo C se detalla el código empleado y la ejecución de los cuadros resultantes que nos ayudan a realizar el análisis predictivo.

El código comienza importando las librerías básicas para el trabajo, explicadas en la sección 3.1. Una vez cargado el archivo .csv con los datos, se eliminan las variables acumuladas.

A continuación, se crea la variable día de la semana (numérica [0-6]) [39] y se crea una codificación cíclica basada en senos y cosenos de la variable día de la semana previamente creada, esto nos permite medir distancias entre los días de una manera numérica [40].

Una vez llegados a este punto, se calcula la matriz de correlación de Pearson. Esta matriz de correlación nos enseña los valores de correlación, que pueden estar entre -1 y +1. Estos valores de correlación, miden el grado de relación lineal entre cada par de variables. Si ambas tienen una tendencia de aumento o de descenso a la vez, el valor de correlación es positivo.

	Galicia.confirmados.PCR.diarias	Galicia.casos.diarios	Galicia.altas.diarias	Galicia.fallecidos.diarios
Galicia.confirmados.PCR.diarias	1.000000	0.956051	0.563923	0.653984
Galicia.casos.diarios	0.956051	1.000000	0.597274	0.717137
Galicia.altas.diarias	0.563923	0.597274	1.000000	0.778968
Galicia.fallecidos.diarios	0.653984	0.717137	0.778968	1.000000

Cuadro 6.2: Matriz de correlación de Pearson.

Se crean unas variables temporales con ventana deslizante, de esta manera se crea una realización por fila longitudinal en el tiempo. Así, tenemos una nueva forma de plantear la serie temporal. En este caso se hace con ventana de tamaño = 6 que equivale a todos los días anteriores de la semana.

Finalmente, se hace el estudio de los diferentes modelos (sección 6.2) y la obtención de unas medias finales, que nos ayudarán a sacar las posteriores conclusiones del análisis predictivo realizado.

6.2 Resultados del análisis predictivo

Como se ha dicho previamente, se estudian cuatro modelos. Los modelos realizados son equivalentes a un modelo de series temporales ARIMA [41], un modelo puramente estadístico que utiliza regresiones y variaciones de datos estadísticos con la finalidad de encontrar patrones para una predicción hacia el futuro, pero en esta ocasión utilizando un modelo de regresión, tratándose de una red neuronal con la ayuda de series temporales.

El proceso de validación empleado ha sido un "train test split", para ello tenemos una partición lineal del dataset en dos bloques, uno para datos de entrenamiento (80%) y otro referido a datos de test (20%), porcentajes habitualmente empleados a la hora de trabajar con este proceso de validación. El "train test split" suele ser empleado para datasets que presentan un orden temporal (como es nuestro caso), ya que no tendría sentido emplear datos futuros para intentar predecir valores pasados.

Modelo 1

Se realiza una serie temporal con una ventana de t-7 días únicamente.

Se entrena un MLP, un modelo no lineal explicado anteriormente con una gran capacidad de aprender comportamientos complejos. Son modelos con una gran tendencia al overfitting/sobreajuste (efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado, como es nuestro caso), por lo que hay que manejarlos con cuidado. Es importante que un modelo de aprendizaje automático tenga un buen ajuste en entrenamiento y una buena capacidad de generalización en datos de test o desconocidos. El equilibrio entre estas dos propiedades es la esencia de una buena predicción.

Se ha creado una red neuronal MLP de 50 neuronas con 50 épocas de entrenamiento. Estas redes evitan la multicolinealidad, un fenómeno que por la tipología de datos y las transformaciones era susceptible de aparecer (debido al momento en el que un regresor es combinación lineal de otro).

El resultado nos indica que este modelo tiene una esperanza de error de 3,9 fallecidos en test.

Modelo 2

Para este modelo, se realiza una serie temporal también, pero con una ventana de t-7 con senos y cosenos, representando la proximidad de los días de la semana en la que se toman los

	fallecidos_t-1	fallecidos_t-2	fallecidos_t-3	fallecidos_t-4	fallecidos_t-5	fallecidos_t-6	Galicia.fallecidos.diarios
fallecidos_t-1	1.000000	0.765415	0.715446	0.742438	0.736991	0.710438	0.765668
fallecidos_t-2	0.765415	1.000000	0.765273	0.715216	0.742282	0.735900	0.715761
fallecidos_t-3	0.715446	0.765273	1.000000	0.764998	0.714945	0.741163	0.742660
fallecidos_t-4	0.742438	0.715216	0.764998	1.000000	0.764664	0.714131	0.737400
fallecidos_t-5	0.736991	0.742282	0.714945	0.764664	1.000000	0.763201	0.710955
fallecidos_t-6	0.710438	0.735900	0.741163	0.714131	0.763201	1.000000	0.714310
Galicia.fallecidos.diarios	0.765668	0.715761	0.742660	0.737400	0.710955	0.714310	1.000000

Cuadro 6.3: Cuadro resultante del modelo 1 del análisis predictivo.

datos. Este modelo, en esencia es el mismo modelo que el modelo anterior (sección 6.2), pero con alguna parametrización distinta.

	DiaSemana_sin	DiaSemana_cos	fallecidos_t-1	fallecidos_t-2	fallecidos_t-3	fallecidos_t-4	fallecidos_t-5	fallecidos_t-6	Galicia.fallecidos.diarios
DiaSemana_sin	1.000000	0.001254	0.093560	0.030355	-0.055810	-0.100999	-0.070302	0.010100	0.086018
DiaSemana_cos	0.001254	1.000000	-0.037723	-0.097927	-0.086420	-0.012009	0.070487	0.095976	0.050375
fallecidos_t-1	0.093560	-0.037723	1.000000	0.765415	0.715446	0.742438	0.736991	0.710438	0.765668
fallecidos_t-2	0.030355	-0.097927	0.765415	1.000000	0.765273	0.715216	0.742282	0.735900	0.715761
fallecidos_t-3	-0.055810	-0.086420	0.715446	0.765273	1.000000	0.764998	0.714945	0.741163	0.742660
fallecidos_t-4	-0.100999	-0.012009	0.742438	0.715216	0.764998	1.000000	0.764664	0.714131	0.737400
fallecidos_t-5	-0.070302	0.070487	0.736991	0.742282	0.714945	0.764664	1.000000	0.763201	0.710955
fallecidos_t-6	0.010100	0.095976	0.710438	0.735900	0.741163	0.714131	0.763201	1.000000	0.714310
Galicia.fallecidos.diarios	0.086018	0.050375	0.765668	0.715761	0.742660	0.737400	0.710955	0.714310	1.000000

Cuadro 6.4: Cuadro resultante del modelo 2 del análisis predictivo.

El resultado nos indica que este modelo tiene una esperanza de error de 3,8 fallecidos en test, mejora un poco con respecto al anterior (error de 3,9 fallecidos).

Modelo 3

En esta ocasión, se realiza el mismo modelo que el modelo 2 (sección 6.2), pero añadiendo los datos de pruebas PCR. De la misma forma, implementando el mismo modelo pero con alguna parametrización distinta, con la idea principal de probar con distintos datasets para averiguar cuál es el que mejor funciona en términos de información resultante.

El resultado (cuadro 6.5) nos indica que este modelo tiene una esperanza de error de 3,85 fallecidos en test, empeorando muy poco con respecto al anterior, pero se está empezando a tener una complejidad alta en término de número de variables, al añadir la variable de pruebas PCR, por lo que es difícil saber el alcance que tiene este fenómeno.

Modelo 4

De la misma manera, se realiza una serie temporal con una ventana de t-7 con senos y cosenos, representando la proximidad de los días de la semana en la que se toman los datos, se añaden datos de PCR y los de altas. Se implementa el mismo modelo que antes con alguna parametrización distinta, pero en esencia es el mismo modelo.

La idea principal es probar con distintos datasets para analizar nuevamente cual es el que mejor funciona en términos de información, nos encontramos con un caso límite, teniendo demasiada información para el tamaño que tenemos de dataset.

El resultado (cuadro 6.6) nos indica que este modelo tiene una esperanza de error de 4,04 fallecidos en test, empeorando respecto a modelos anteriores, una situación razonable y de fácil explicación: el número de variables que se está introduciendo es muy elevado y el número de observaciones que tenemos es muy pequeño.

	DiaSemana_sin	DiaSemana_cos	fallecidos_t-1	fallecidos_t-2	fallecidos_t-3	fallecidos_t-4	fallecidos_t-5	fallecidos_t-6	PCR_t-1	PCR_t-2	PCR_t-3	PCR_t-4	PCR_t-5	PCR_t-6	Galicia.fallecidos.diarios
DiaSemana_sin	1.000000	0.001999	0.120621	0.035425	-0.081801	-0.141316	-0.098703	0.014342	-0.064927	-0.134219	-0.101816	0.006740	0.110443	0.128824	0.111681
DiaSemana_cos	0.001999	1.000000	-0.051624	-0.131409	-0.116670	-0.013637	0.096963	0.133579	-0.119536	-0.023752	0.088347	0.134122	0.076985	-0.041240	0.066398
fallecidos_t-1	0.120621	-0.051624	1.000000	0.731658	0.674195	0.708198	0.699213	0.667671	0.656554	0.672208	0.695120	0.744647	0.774341	0.819921	0.732104
fallecidos_t-2	0.035425	-0.131409	0.731658	1.000000	0.730903	0.673579	0.707668	0.698731	0.686095	0.657351	0.672007	0.694715	0.744062	0.774313	0.674413
fallecidos_t-3	-0.081801	-0.116670	0.674195	0.730903	1.000000	0.730003	0.672927	0.706899	0.648147	0.686678	0.656656	0.670513	0.692883	0.743653	0.708750
fallecidos_t-4	-0.141316	-0.013637	0.708198	0.673579	0.730003	1.000000	0.729363	0.672300	0.588615	0.649235	0.686572	0.656320	0.669900	0.692862	0.699614
fallecidos_t-5	-0.098703	0.096963	0.699213	0.707668	0.672927	0.729363	1.000000	0.728868	0.541103	0.589593	0.649011	0.686049	0.655451	0.669828	0.668187
fallecidos_t-6	0.014342	0.133579	0.667671	0.698731	0.706899	0.672300	0.728868	1.000000	0.507054	0.541823	0.589305	0.648558	0.685394	0.655309	0.677210
PCR_t-1	-0.064927	-0.119536	0.656554	0.686095	0.648147	0.588615	0.541103	0.507054	1.000000	0.961114	0.929292	0.899515	0.886152	0.886423	0.671569
PCR_t-2	-0.134219	-0.023752	0.672208	0.657351	0.686678	0.649235	0.589593	0.541823	0.961114	1.000000	0.961467	0.929355	0.899710	0.886477	0.695491
PCR_t-3	-0.101816	0.088347	0.695120	0.672007	0.656656	0.686572	0.649011	0.589305	0.929292	0.961467	1.000000	0.961298	0.929130	0.899649	0.745529
PCR_t-4	0.006740	0.134122	0.744647	0.694715	0.670513	0.656320	0.686049	0.648558	0.899515	0.929355	0.961298	1.000000	0.961306	0.928833	0.775454
PCR_t-5	0.110443	0.076985	0.774341	0.744062	0.692883	0.669900	0.655451	0.685394	0.886152	0.899710	0.929130	0.961306	1.000000	0.960973	0.820059
PCR_t-6	0.128824	-0.041240	0.819921	0.774313	0.743653	0.692862	0.669828	0.655309	0.886423	0.886477	0.899649	0.928833	0.960973	1.000000	0.848582
Galicia.fallecidos.diarios	0.111681	0.066398	0.732104	0.674413	0.708750	0.699614	0.668187	0.677210	0.671569	0.695491	0.745529	0.775454	0.820059	0.848582	1.000000

Cuadro 6.5: Cuadro resultante del modelo 3 del análisis predictivo.

	DiaSemana_sin	DiaSemana_cos	fallecidos_t-1	fallecidos_t-2	fallecidos_t-3	fallecidos_t-4	fallecidos_t-5	fallecidos_t-6	PCR_t-1	PCR_t-2	...	PCR_t-4	PCR_t-5	PCR_t-6	altas_t-1	altas_t-2	altas_t-3	altas_t-4	altas_t-5	altas_t-6	Galicia.fallecidos.diarios
DiaSemana_sin	1.000000	0.001999	0.120621	0.035425	-0.081801	-0.141316	-0.098703	0.014242	-0.064927	-0.134219	...	0.006740	0.110443	0.128824	-0.109472	-0.143692	-0.073282	0.049042	0.130143	0.109207	0.111681
DiaSemana_cos	0.001999	1.000000	-0.051624	-0.131409	-0.116670	-0.013637	0.096963	0.135579	-0.119536	-0.023752	...	0.134122	0.076985	-0.041240	-0.096543	0.019893	0.119905	0.127276	0.035180	-0.065784	0.066398
fallecidos_t-1	0.120621	-0.051624	1.000000	0.731658	0.674195	0.708198	0.699213	0.667671	0.656554	0.672208	...	0.744647	0.774341	0.819921	0.743725	0.709948	0.712032	0.686562	0.691918	0.680575	0.732104
fallecidos_t-2	0.035425	-0.131409	0.731658	1.000000	0.730903	0.673579	0.707668	0.698731	0.686095	0.657351	...	0.694715	0.744062	0.774313	0.815613	0.743242	0.709375	0.711423	0.695994	0.691340	0.674413
fallecidos_t-3	-0.081801	-0.116670	0.674195	0.730903	1.000000	0.730003	0.672927	0.706899	0.648147	0.666678	...	0.670513	0.692883	0.743653	0.838868	0.815222	0.742274	0.708132	0.710727	0.695153	0.708750
fallecidos_t-4	-0.141316	-0.013637	0.708198	0.673579	0.730003	1.000000	0.729363	0.672300	0.586615	0.649235	...	0.656320	0.669900	0.692862	0.835552	0.838438	0.814731	0.741544	0.707404	0.710014	0.699614
fallecidos_t-5	-0.098703	0.096963	0.699213	0.707668	0.672927	0.729363	1.000000	0.728868	0.541103	0.569593	...	0.656049	0.655451	0.669828	0.813410	0.835221	0.837999	0.814152	0.740971	0.706730	0.668187
fallecidos_t-6	0.014342	0.133579	0.667671	0.698731	0.706899	0.672300	0.728868	1.000000	0.507054	0.541823	...	0.648358	0.665394	0.655309	0.807363	0.813046	0.834930	0.837684	0.813815	0.740486	0.677210
PCR_t-1	-0.064927	-0.119536	0.656554	0.686095	0.646147	0.586615	0.541103	0.507054	1.000000	0.961114	...	0.899515	0.886152	0.886423	0.565970	0.520914	0.443970	0.401721	0.370631	0.353641	0.671569
PCR_t-2	-0.134219	-0.023752	0.672208	0.657351	0.686678	0.649235	0.589593	0.541823	0.961114	1.000000	...	0.929355	0.899710	0.886477	0.601922	0.586917	0.511010	0.444968	0.402584	0.371436	0.695491
PCR_t-3	-0.101816	0.088347	0.695120	0.672007	0.656656	0.686572	0.649011	0.589305	0.929292	0.961467	...	0.961298	0.929130	0.899549	0.617727	0.601598	0.566679	0.510678	0.444980	0.402056	0.745529
PCR_t-4	0.006740	0.134122	0.744647	0.694715	0.670513	0.656320	0.686049	0.648358	0.899515	0.929355	...	1.000000	0.961306	0.928833	0.633731	0.617010	0.601276	0.566407	0.510008	0.443812	0.775454
PCR_t-5	0.110443	0.076985	0.774341	0.744062	0.692383	0.669900	0.655451	0.665394	0.886152	0.899710	...	0.961306	1.000000	0.960973	0.658502	0.634707	0.616375	0.600730	0.565472	0.529065	0.820059
PCR_t-6	0.128824	-0.041240	0.819921	0.774313	0.743653	0.692862	0.669828	0.655309	0.886423	0.886477	...	0.928833	0.960973	1.000000	0.703221	0.658397	0.634667	0.616288	0.600602	0.565290	0.846582
altas_t-1	-0.109472	-0.096543	0.743725	0.815613	0.838868	0.835552	0.813410	0.807363	0.565970	0.601922	...	0.633731	0.658502	0.703221	1.000000	0.948057	0.910641	0.883390	0.877444	0.868648	0.710277
altas_t-2	-0.143692	0.019893	0.709948	0.743242	0.815222	0.838438	0.835221	0.813046	0.520914	0.586917	...	0.617010	0.634707	0.658397	0.948057	1.000000	0.947848	0.910246	0.883126	0.877131	0.712615
altas_t-3	-0.073282	0.119905	0.712032	0.709375	0.742274	0.814731	0.837999	0.834930	0.443970	0.511010	...	0.601276	0.616375	0.634667	0.910641	0.947848	1.000000	0.947670	0.909974	0.882802	0.697306
altas_t-4	0.049042	0.127276	0.686562	0.711423	0.708132	0.741544	0.814152	0.837684	0.401721	0.444968	...	0.566407	0.600730	0.616288	0.883390	0.910246	0.947670	1.000000	0.947440	0.909682	0.682369
altas_t-5	0.130143	0.035180	0.691918	0.695994	0.710727	0.707404	0.740971	0.813815	0.370631	0.402584	...	0.510008	0.565472	0.600602	0.877444	0.883126	0.909974	0.947440	1.000000	0.947304	0.681112
altas_t-6	0.109207	-0.065764	0.680575	0.691340	0.695153	0.710014	0.706730	0.740486	0.353641	0.371436	...	0.443812	0.529065	0.565290	0.868648	0.877131	0.882802	0.909682	0.947304	1.000000	0.639723
Galicia.fallecidos.diarios	0.111681	0.066398	0.732104	0.674413	0.708750	0.699614	0.668187	0.677210	0.671569	0.695491	...	0.775454	0.820059	0.846582	0.710277	0.712615	0.697306	0.682369	0.681112	0.639723	1.000000

Cuadro 6.6: Cuadro resultante del modelo 4 del análisis predictivo.

6.3 Conclusiones del análisis predictivo

El último modelo estudiado, el modelo 4 hace que nos demos cuenta de que, para predecir datos en el futuro haría falta más tiempo, el número de observaciones (filas que representan los días) y de columnas de datos, debería de ser mucho mayor.

La media de fallecidos es de 6,7. Lo que nos indica que, por lo general en Galicia para el rango temporal estudiado, han muerto alrededor de 6,7 personas cada día, existiendo obviamente una variabilidad dependiendo de la variante del virus.

Errores de alrededor de 4 fallecidos, implican un intervalo de [2.4 , 10.4] fallecidos, lo que nos indica que los modelos estudiados anteriormente con el sistema MLP obtienen mucho error, debido a la cantidad de datos que se tiene en el dataset. Lo ideal sería obtener un conjunto de datos mucho mayor, pero debido a que contemplamos series diarias desde que se tienen datos verídicos sobre la COVID-19 en Galicia, el espacio de tiempo, y por lo tanto el número de filas/observaciones, es "pequeño".

Partiendo de la problemática anterior y comprendiendo que son muchas las variables que influyen en el fallecimiento de una persona, es muy difícil predecir los fallecimientos, sabiendo, además, que desde que una persona es positiva en una prueba PCR, y por lo tanto contabilizada como un caso más de la enfermedad, el período medio del posible fallecimiento abarca semanas.

En conclusión, aún utilizando herramientas punteras como pueden ser una red neuronal o técnicas como MLP, si el error que se espera obtener es pequeño, no se puede llevar a cabo con las condiciones de dataset anteriores y, por lo tanto, el problema no se puede llegar a resolver.

Centrándonos en un trabajo futuro, se podrían incorporar otras variables que también resultan interesantes, como puede ser la tendencia (positiva o negativa) en el número de casos del día anterior, entre otras.

Capítulo 7

Conclusiones

Una vez concluido el estudio, me gustaría sacar unas conclusiones divididas en diferentes puntos: conclusiones del estudio de los datos, relación de lo estudiado en el proyecto con las competencias del grado y un posible trabajo futuro.

7.1 Sobre el estudio de los datos

Se puede concluir el proyecto analizando la llegada a los objetivos concretos principales que se habían fijado al comienzo del proyecto:

- La letalidad: el AS con mayor letalidad es Ourense, por delante de Ferrol. Con mucha diferencia, el área sanitaria en la que menos fallecidos ha habido con respecto a los casos existentes por COVID-19 ha sido Pontevedra, con una diferencia clara con tan solo 163 fallecidos.

- El comportamiento de las UCIs a nivel gallego: siguiendo la tendencia española pero, llegados al último tramo de enero del 2021, convirtiéndose en la comunidad española con más ingresos para pacientes con la COVID-19. Vigo ha sido el AS que más ha sufrido la presión de pacientes en las UCIs.

- La relación entre la positividad de las pruebas PCR y el tramo por el que atraviesa la variante del virus que se está expandiendo por el área en cuestión, va estrechamente ligada.

- Se han encontrado datos valiosos que se desconocían sobre el comportamiento del virus entre los diferentes municipios de todas las ASs gallegas: se destaca que la COVID-19 ha tenido un impacto mayor en las zonas/municipios que tienen un valor del PIB entre 10.000 y 20.000 euros.

Respecto al trabajo realizado de análisis predictivo, al conocer finalmente que con las variables consideradas no ha sido posible predecir los fallecidos relacionados con la COVID-19, en el futuro se podrán utilizar más variables y métodos alternativos.

Ha sido interesante conocer el funcionamiento de las metodologías y técnicas usadas, así como la incertidumbre a la que te expones cuando realizas un estudio de análisis predictivo.

En líneas generales, los análisis, tanto el descriptivo como el predictivo, nos han ayudado a comprender la importancia que pueden suponer este tipo de técnicas y tareas en situaciones de emergencia sanitaria, como ha sido el caso.

7.2 Relación con competencias del grado

Respecto a la relación con las competencias de la titulación en general y en la mención en particular, tecnologías de la información, he de decir que a pesar de tener una incidencia directa a la hora de escoger el tema de mi trabajo fin de grado, he podido afianzar los conocimientos y la materia aprendida en diferentes asignaturas del grado, en especial a las siguientes:

- **"Estadística"**
- **"Administración de Bases de Datos"**
- **"Explotación de Almacenes de Datos"**

7.3 Trabajo futuro

A pesar de haber llegado de manera satisfactoria a los objetivos que se han planteado desde el inicio del proyecto, a lo largo del mismo me han surgido diferentes ramas de estudio o de futura puesta en marcha en un posible trabajo futuro.

- Incluir fuentes de datos diferentes: añadir nuevos conjuntos de datos. Siempre es beneficioso a la hora de investigar o de informar.
- Emplear lo aprendido y la metodología seguida para otros tipos de virus o enfermedades de transmisión, ya que, teniendo una leve experiencia en el mundo de la investigación con explotación de datos a través de la COVID-19, puede suponer un avance muy grande de cara al futuro.
- Destinar investigaciones o cualquier tipo de trabajo que aporte y apoye una mejora de la calidad de los datos que se proporcionan desde diferentes webs, pudiendo hacer

un portal unificado para la recolección de este tipo de datos o trabajos. La cantidad de datos falsos o engañosos que se generan diariamente, y más en temas de actualidad y de emergencia social, son abrumadores. Ha sido uno de los temas a los que más importancia le he dado durante el trabajo debido a la cantidad de problemas que me ha dado a lo largo de los meses empleados en el proyecto, pudiendo un dato falso o no validado, complicar en gran medida un trabajo o proyecto.

- Migrar gran cantidad de datos relevantes a una nube común, siendo mucho más fácil su accesibilidad y manipulación.
- Realizar diferentes pruebas con otros tipos de técnicas de análisis predictivo.
- Realizar el análisis predictivo con un dataset mucho más amplio en el tiempo.
- Replicar los modelos del clustering en los meses antes, durante y después del confinamiento domiciliario sufrido en Galicia, para ver las diferencias respecto al resultado obtenido en el proyecto.

Apéndices

Apéndice A

Gráficas adicionales

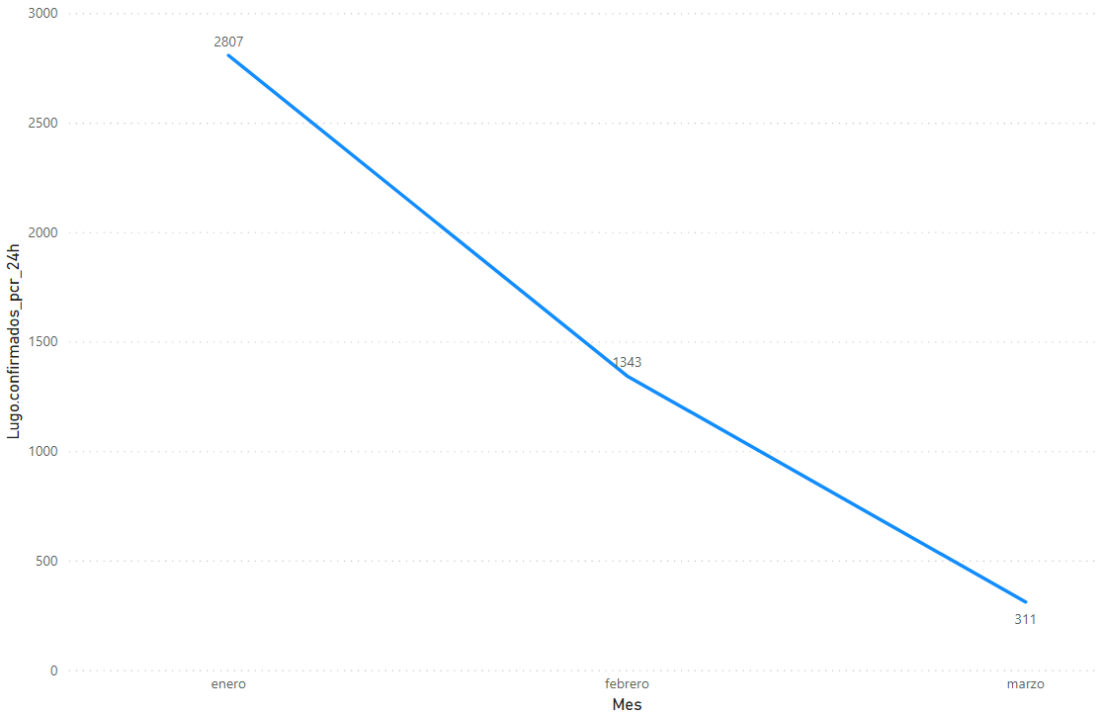

Figura A.1: Evolución de casos de COVID-19 por PCR positiva confirmada para el área de Lugo

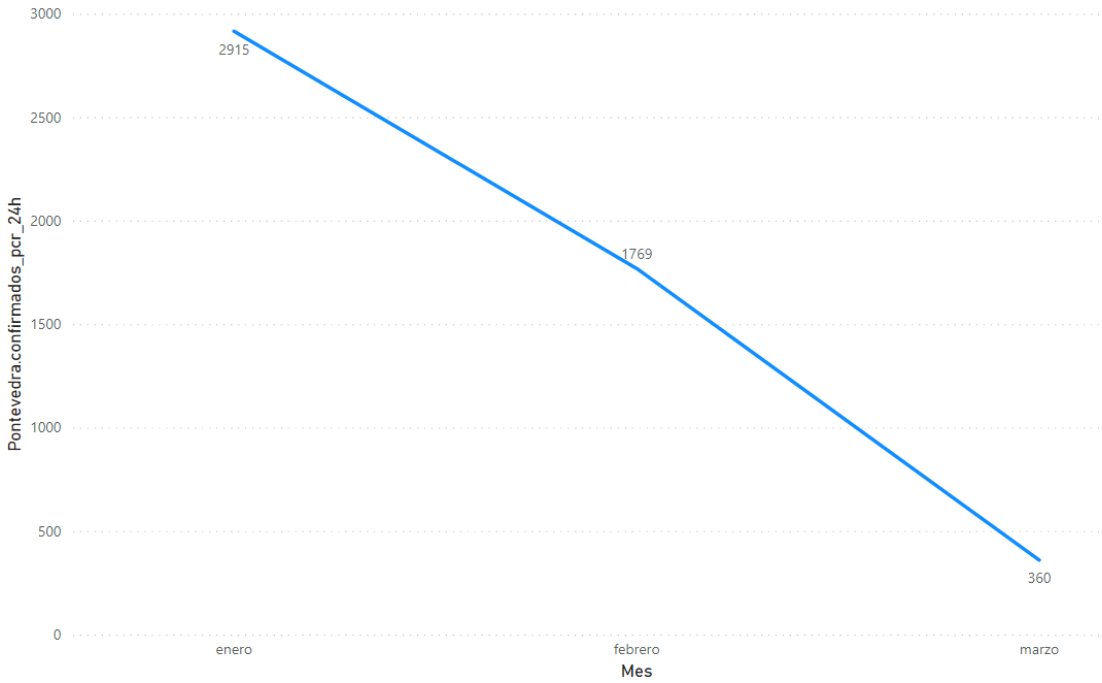

Figura A.2: Evolución de casos de COVID-19 por PCR positiva confirmada para el área de Pontevedra

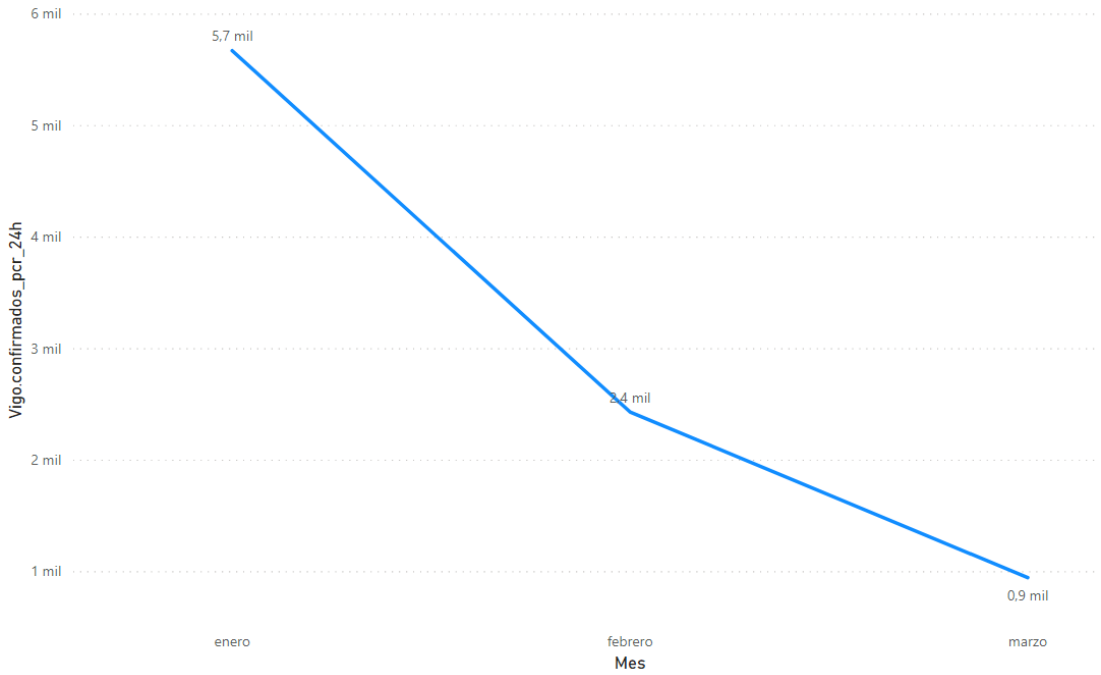

Figura A.3: Evolución de casos de COVID-19 por PCR positiva confirmada para el área de Vigo

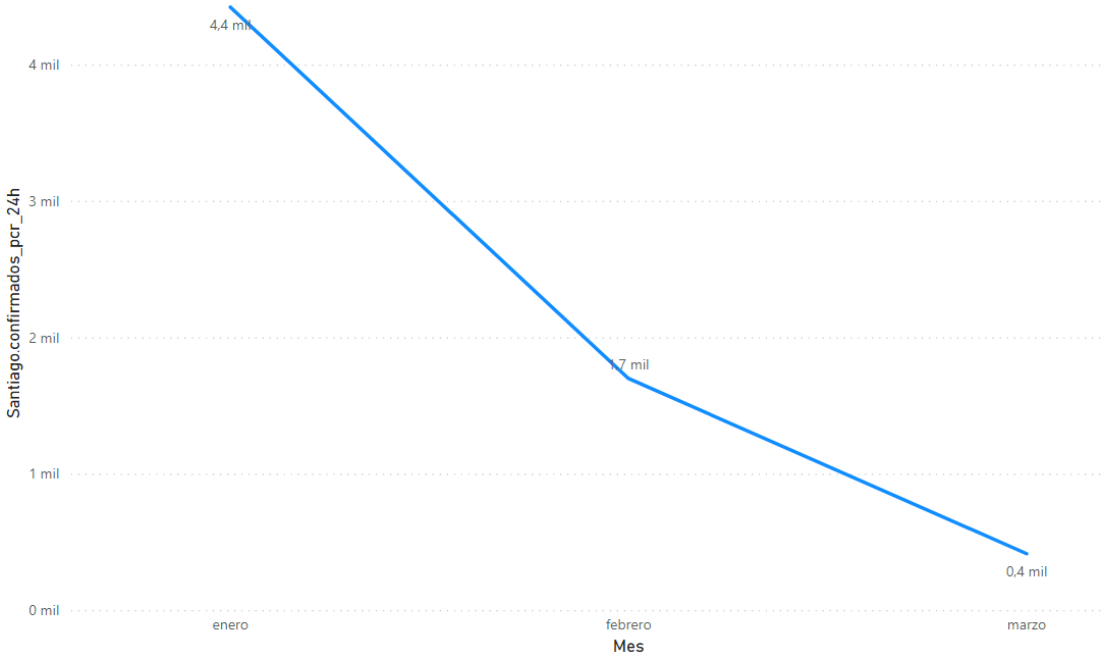

Figura A.4: Evolución de casos de COVID-19 por PCR positiva confirmada para el área de Santiago

Mes	Día	Positividad	A_Coruna.PCR.diaria
enero	6	0,13	1264
enero	7	0,14	1138
enero	8	0,14	856
enero	9	0,11	1494
enero	10	0,13	1395
enero	11	0,14	1104
enero	12	0,17	823
enero	13	0,13	1724
enero	14	0,11	2254
enero	15	0,11	2360
enero	16	0,12	2145
enero	17	0,16	2091
enero	18	0,15	1627
enero	19	0,16	1255
enero	20	0,15	2362
enero	21	0,12	2407
enero	22	0,13	2654
enero	23	0,14	3166
enero	24	0,14	2748
enero	25	0,12	2065
enero	26	0,16	1579
enero	27	0,14	2659
enero	28	0,08	3282
enero	29	0,08	3101
enero	30	0,08	3180
enero	31	0,11	2696
febrero	1	0,08	2415
febrero	2	0,12	1414
febrero	3	0,10	2122
febrero	4	0,10	2881
febrero	5	0,08	2545
febrero	6	0,06	3711

Figura A.5: Ejemplo del descenso de la positividad de las pruebas PCR para el AS de A Coruña

Apéndice B

Código clustering

```r
# Se importan las librerías
library(dplyr)
library(tidyverse)
library(kml)

# Carga de datos
nov <- read.csv("Nov2020.csv", encoding = "UTF-8")
ene <- read.csv("Enero2021.csv", encoding = "UTF-8")
mar <- read.csv("Marzo2021.csv", encoding = "UTF-8")
jun <- read.csv("Junio2021.csv", encoding = "UTF-8")

# Unificación de los cuatro dataframes
nov_ene <- merge(nov, ene, all = TRUE)
nov_ene_mar <- merge(nov_ene, mar_,  all = TRUE)
df <- merge(nov_ene_mar, jun_,  all = TRUE)
View(df)

# pivot_wider: "amplía" los datos, aumentando el número de columnas y disminuyendo el número de filas
df <- as.data.frame(df%>% arrange(fecha)%>%pivot_wider(id_cols = c(municipio, codigo_municipio, habitantes), names_from = fecha, values_from = IA14, PIB, zona))
is.na(df) %>%sum()

# Reemplazo de los valores NA por 0
df[is.na(df)] <- 0

########## MODELOS ##########
# Creación de un cld puesto que kml espera un objeto. clusterLongData necesita saber que columnas son de serie temporal (timeInData)
# Formato dataframe
```

```
df <- as.data.frame(df)
View(df)

------------------------------------------------------------

# Modelo 1: sin especificar cluster, randomK
# Creación de un clusterLongData objecto
model1 <- kml::cld(df, timeInData=4:7)

# Creación del objeto para usarlo en kml()
(option1<-parALGO(startingCond = c("randomK")))
class(model1)

# Creación del modelo sin indicar cuantos clúster queremos, ya que
    por defecto contiene un vector que contiene el número de
    conglomerados con los que kml debe de trabajar. De forma
    predeterminada, nbclústeres es 2: 6, lo que indica que kml debe
    buscar particiones con 2, luego 3, ... hasta 6 grupos,
    respectivamente.
kml::kml(model1, parAlgo = option1)

if (.Platform$OS.type != "windows") {
  X11(type = "Xlib")
}

# Se lanza el modelo
kml::choice(model1)
plotAllCriterion(model1)
print(model1)

#  Trayectos Municipios y tiempos
mun_tim1 <- data.frame(model1@traj)
mun_tim1

------------------------------------------------------------

# Model 2: sin especificar cluster, indicando randomAll
model2 <- kml::cld(df, timeInData=4:7)
(option2<-parALGO(startingCond = c("randomAll")))

# Creación del modelo
kml::kml(model2,  parAlgo = option2)
if (.Platform$OS.type != "windows") {
  X11(type = "Xlib")
}

```

```
# Se lanza el modelo
kml::choice(model2,typeGraph = "y/h")

# Muestra los tres criterios estimados por el algoritmo.
plotAllCriterion(model2)
print(model2)

# Trayectos municipio-tiempo
mun_tim2 <- data.frame(model2@traj)
mun_tim2

------------------------------------------------------------

# Modelo 3: sin especificar cluster, maxDist
# creamos clusterLongData objecto
model3 <- kml::cld(df, timeInData=4:7)
(option3<-parALGO(startingCond = c("maxDist")))

# Creación del modelo
kml::kml(model3, parAlgo = option3)
if (.Platform$OS.type != "windows") {
  X11(type = "Xlib")
}

# Se lanza el modelo
kml::choice(model3, typeGraph = "f")

# Muestra los tres criterios estimados por el algoritmo.
plotAllCriterion(model3)

# Trayectos municipio-tiempo
mun_tim3 <- data.frame(model3@traj)
mun_tim3
```

Apéndice C

Código análisis predictivo

```
#Se importan las librerias
import pandas as pd
import numpy as np

from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

#Se carga el .csv
GaliciaPredictivo = pd.read_csv("GaliciaPredictivo.csv")
GaliciaPredictivo

#Se eliminan las variables acumuladas que no tienen interes
df1 = GaliciaPredictivo.drop(columns=["Galicia.casos.acum",
    "Galicia.altas.acum", "Galicia.fallecidos.acum"])
df1

#Se crea la variable dia de la semana (numerica [0-6])
df1['Fecha'] = pd.to_datetime(df1['Fecha'])
df1['DiaSemana'] = df1['Fecha'].dt.dayofweek

#Creacion senos/cosenos
def time_encoding(data, features, items):

    for col in features:
        data[col+'_sin'] = np.sin((data[col]-1)*(2.*np.pi/items))
        data[col+'_cos'] = np.cos((data[col]-1)*(2.*np.pi/items))

    return data

df2 = time_encoding(df1, ['DiaSemana'], items=7)
```

```
df2.sort_values('Fecha')
df2

#Calcular matriz de correlacion de pearson
df2[["Galicia.confirmados.PCR.diarias", "Galicia.casos.diarios",
    "Galicia.altas.diarias",
    "Galicia.fallecidos.diarios"]].corr("pearson")

#Variables temporales
df2["fallecidos_t-1"] = df2["Galicia.fallecidos.diarios"].shift(1)
df2["fallecidos_t-2"] = df2["Galicia.fallecidos.diarios"].shift(2)
df2["fallecidos_t-3"] = df2["Galicia.fallecidos.diarios"].shift(3)
df2["fallecidos_t-4"] = df2["Galicia.fallecidos.diarios"].shift(4)
df2["fallecidos_t-5"] = df2["Galicia.fallecidos.diarios"].shift(5)
df2["fallecidos_t-6"] = df2["Galicia.fallecidos.diarios"].shift(6)
#df2["fallecidos_t-7"] = df2["Galicia.fallecidos.diarios"].shift(7)

df2["PCR_t-1"] = df2["Galicia.confirmados.PCR.diarias"].shift(1)
df2["PCR_t-2"] = df2["Galicia.confirmados.PCR.diarias"].shift(2)
df2["PCR_t-3"] = df2["Galicia.confirmados.PCR.diarias"].shift(3)
df2["PCR_t-4"] = df2["Galicia.confirmados.PCR.diarias"].shift(4)
df2["PCR_t-5"] = df2["Galicia.confirmados.PCR.diarias"].shift(5)
df2["PCR_t-6"] = df2["Galicia.confirmados.PCR.diarias"].shift(6)
#df2["PCR_t-7"] = df2["Galicia.confirmados.PCR.diarias"].shift(7)

df2["altas_t-1"] = df2["Galicia.altas.diarias"].shift(1)
df2["altas_t-2"] = df2["Galicia.altas.diarias"].shift(2)
df2["altas_t-3"] = df2["Galicia.altas.diarias"].shift(3)
df2["altas_t-4"] = df2["Galicia.altas.diarias"].shift(4)
df2["altas_t-5"] = df2["Galicia.altas.diarias"].shift(5)
df2["altas_t-6"] = df2["Galicia.altas.diarias"].shift(6)
#df2["altas_t-7"] = df2["Galicia.altas.diarias"].shift(7)

df2


#MODELO1
df3 =  df2[["fallecidos_t-{}".format(x) for x in range(1,7)] +
    ["Galicia.fallecidos.diarios"]].dropna()
X = df3.iloc[:,0:-1]
y = df3["Galicia.fallecidos.diarios"]

X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.25, random_state=42)

```

```
serie_temporal_univariante_k7 =  MLPRegressor(hidden_layer_sizes=(100),
                        random_state=1,
                        max_iter=10000,
                        activation = 'logistic',
                        early_stopping = True,
                        alpha = 0.001,
                        learning_rate='invscaling',
                        solver="lbfgs").fit(X_train, y_train)#LinearRegression().fit(X_train, y_train)

mse_serie_temporal_univariante_k7_test = mean_squared_error(y_test, serie_temporal_univariante_k7.predict(X_test), squared=False)
mse_serie_temporal_univariante_k7_train = mean_squared_error(y_train, serie_temporal_univariante_k7.predict(X_train), squared=False)

print("error de train", mse_serie_temporal_univariante_k7_train, "\nerror de test:", mse_serie_temporal_univariante_k7_test)


#MODELO2
df4 =  df2[["DiaSemana_sin", "DiaSemana_cos"]+["fallecidos_t-{}".format(x) for x in range(1,7)] + ["Galicia.fallecidos.diarios"]].dropna()
X = df4.iloc[:,0:-1]
y = df4["Galicia.fallecidos.diarios"]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

serie_temporal_univariante_k7_cossin = MLPRegressor(hidden_layer_sizes= (100),
                            random_state=1,
                            max_iter=10000,
                            activation = 'logistic',
                            early_stopping = True,
                            alpha = 0.0001,
                            learning_rate='invscaling',
                            solver="lbfgs").fit(X_train, y_train)#LinearRegression().fit(X_train, y_train)

mse_serie_temporal_univariante_k7_cossin_test = mean_squared_error(y_test, serie_temporal_univariante_k7_cossin.predict(X_test),
```

```
    squared=False)

mse_serie_temporal_univariante_k7_cossin_train =
    mean_squared_error(y_train,
    serie_temporal_univariante_k7_cossin.predict(X_train),
    squared=False)

print("error de train",
    mse_serie_temporal_univariante_k7_cossin_train, "\nerror de
    test:", mse_serie_temporal_univariante_k7_cossin_test)


#MODELO3
df5 =  df2[["DiaSemana_sin",
    "DiaSemana_cos"]+["fallecidos_t-{}".format(x) for x in
    range(1,7)]+ ["PCR_t-{}".format(x) for x in range(1,7)] +
    ["Galicia.fallecidos.diarios"]].dropna()
X = df5.iloc[:,0:-1]
y = df5["Galicia.fallecidos.diarios"]


X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.25, random_state=42)

serie_temporal_univariante_k7_cossin_PCR =
    MLPRegressor(hidden_layer_sizes= (50),
                              random_state=1,
                              max_iter=1000000,
                              activation = 'logistic',
                              early_stopping = True,
                              alpha = 0.0001,
                              learning_rate='invscaling',
                              solver="lbfgs").fit(X_train,
    y_train)#LinearRegression().fit(X_train, y_train)

mse_serie_temporal_univariante_k7_cossin_PCR_test =
    mean_squared_error(y_test,
    serie_temporal_univariante_k7_cossin_PCR.predict(X_test),
    squared=False)
mse_serie_temporal_univariante_k7_cossin_PCR_train =
    mean_squared_error(y_train,
    serie_temporal_univariante_k7_cossin_PCR.predict(X_train),
    squared=False)

print("error de train",
    mse_serie_temporal_univariante_k7_cossin_PCR_train, "\nerror de
```

```
    test:", mse_serie_temporal_univariante_k7_cossin_PCR_test)
133
134
135 #MODELO4
136 df6 =  df2[["DiaSemana_sin",
    "DiaSemana_cos"]+["fallecidos_t-{}".format(x) for x in
    range(1,7)] +
137            ["PCR_t-{}".format(x) for x in range(1,7)] +
    ["altas_t-{}".format(x) for x in range(1,7)] +
    ["Galicia.fallecidos.diarios"]].dropna()
138 X = df6.iloc[:,0:-1]
139 y = df6["Galicia.fallecidos.diarios"]
140
141
142 X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.25, random_state=42)
143
144 serie_temporal_univariante_k7_cossin_PCR =
    MLPRegressor(hidden_layer_sizes= (50),
145                              random_state=1,
146                              max_iter=10000,
147                              activation = 'logistic',
148                              early_stopping = True,
149                              alpha = 0.0001,
150                              learning_rate='invscaling',
151                              solver="lbfgs").fit(X_train,
    y_train)#LinearRegression().fit(X_train, y_train)
152
153 mse_serie_temporal_univariante_k7_cossin_PCR_test =
    mean_squared_error(y_test,
    serie_temporal_univariante_k7_cossin_PCR.predict(X_test),
    squared=False)
154 mse_serie_temporal_univariante_k7_cossin_PCR_train =
    mean_squared_error(y_train,
    serie_temporal_univariante_k7_cossin_PCR.predict(X_train),
    squared=False)
155
156 print("error de train",
    mse_serie_temporal_univariante_k7_cossin_PCR_train, "\nerror de
    test:", mse_serie_temporal_univariante_k7_cossin_PCR_test)
157
158
159 media_y_test = np.mean(y_test)
160 media_y_test
161 media_y_train = np.mean(y_train)
162 media_y_train
```

Lista de acrónimos

API Application Programming Interfaces. 16, 17

AS Área Sanitaria. iv, v, 20, 21, 27–29, 37, 67, 77

BBDD Bases de Datos. 18

CRISP-DM Cross Industry Standard Process for Data Mining. v, 5

IA Incidencia Acumulada. v, 33–35, 55, 56

MLP Multi Layer Perceptron. 59, 61, 66

PDIA Pruebas Diagnósticas de Infección Activa. 21

PIB Producto Interior Bruto. v, 23, 35, 42, 55, 56, 67

RRSS Redes Sociales. 21

UCI Unidad de Cuidados Intensivos. iii, 1, 2, 20, 21, 29–32, 67

Bibliografía

[1] "Clasificación de los diferentes clústeres." [En línea]. Disponible en: https://www.diegocalvo.es/cluster-jerarquicos-y-no-jerarquicos/

[2] "Crisp-dm 2." [En línea]. Disponible en: https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=dm-crisp-help-overview

[3] "Crisp-dm." [En línea]. Disponible en: https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/

[4] "Microsoft excel." [En línea]. Disponible en: https://www.microsoft.com/es-es/microsoft-365/excel

[5] "Power bi." [En línea]. Disponible en: https://powerbi.microsoft.com/es-es/

[6] "Power bi 2." [En línea]. Disponible en: https://www.quonext.com/software-gestion-business-intelligence-reporting/microsoft-power-bi

[7] "Anaconda navigator." [En línea]. Disponible en: https://docs.anaconda.com/anaconda/navigator/index.html

[8] "Web corporativa de python." [En línea]. Disponible en: https://www.python.org/

[9] "Libreria pandas." [En línea]. Disponible en: https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library-f1b678f34673

[10] "Libreria numpy." [En línea]. Disponible en: https://aprendeconalf.es/docencia/python/manual/numpy/

[11] "Libreria sklearn." [En línea]. Disponible en: http://datascience.recursos.uoc.edu/es/preprocesamiento-de-datos-con-sklearn/

[12] "Limpieza de datos en r." [En línea]. Disponible en: https://cran.r-project.org/doc/contrib/de_Jonge+van_der_Loo-Introduction_to_data_cleaning_with_R.pdf

[13] G. Williams, *Data mining with rattle and r. Use r!* Springer, 2011.

[14] "Limpieza de datos en r 2." [En línea]. Disponible en: https://www.dataquest.io/blog/load-clean-data-r-tidyverse/

[15] "Paquete tidyverse." [En línea]. Disponible en: https://rpubs.com/paraneda/tidyverse

[16] "Paquete dplyr." [En línea]. Disponible en: https://mauricioanderson.com/curso-r-dplyr/

[17] "Paquete kml." [En línea]. Disponible en: https://cran.r-project.org/web/packages/kml/kml.pdf

[18] "Pandemia covid-19 en españa." [En línea]. Disponible en: https://es.wikipedia.org/wiki/Pandemia_de_COVID-19_en_Espa%C3%B1a

[19] "Web del sergas." [En línea]. Disponible en: https://coronavirus.sergas.gal/datos/#/gl-ES/galicia

[20] "Twitter del chuo." [En línea]. Disponible en: https://twitter.com/Com_CHUO

[21] "Cnecovid." [En línea]. Disponible en: https://cnecovid.isciii.es/covid19/

[22] "Municipios y áreas sanitarias." [En línea]. Disponible en: https://github.com/lipido/galicia-covid19/blob/master/municipios-areas.csv

[23] "Portal do instituto galego de estatística." [En línea]. Disponible en: http://www.ige.eu/igebdt/selector.jsp?COD=9958

[24] "Diagnostico de la covid-19." [En línea]. Disponible en: https://www.semfyc.es/wp-content/uploads/2020/05/COVID19-semFYC-28-_05_2020-DIAGNOSTICO.pdf

[25] "Porcentaje de fallo en una prueba pcr." [En línea]. Disponible en: https://www.redaccionmedica.com/recursos-salud/faqs-covid19/cual-es-el-porcentaje-de-fallo-de-la-pcr

[26] "Medidas covid-19 ourense enero 2021." [En línea]. Disponible en: https://www.ceo.es/wp-content/uploads/2021/01/Medidas_enero-.pdf

[27] "Datos oficiales de covid-19 en españa." [En línea]. Disponible en: https://github.com/rubenfcasal/COVID-19

[28] "¿qué es el clustering?" [En línea]. Disponible en: https://www.grapheverywhere.com/que-es-el-clustering/

[29] "K-means." [En línea]. Disponible en: https://vimeo.com/109114518

[30] "K-means 2." [En línea]. Disponible en: https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html

[31] "Calinski-harabasz." [En línea]. Disponible en: https://www.geeksforgeeks.org/calinski-harabasz-index-cluster-validity-indices-set-3/

[32] "Análisis predictivo." [En línea]. Disponible en: https://neoattack.com/neowiki/modelo-predictivo/

[33] "Análisis predictivo 2." [En línea]. Disponible en: https://www.baoss.es/analisis-predictivo-que-es/

[34] "Árboles de decisión." [En línea]. Disponible en: https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n

[35] "Máquinas de vectores de soporte svm." [En línea]. Disponible en: https://www.iartificial.net/maquinas-de-vectores-de-soporte-svm/

[36] "K-vecinos." [En línea]. Disponible en: https://aprendeia.com/k-vecinos-mas-cercanos-teoria-machine-learning/#:~:text=K%20vecinos%20m%C3%A1s%20cercanos%20es,y%20la%20detecci%C3%B3n%20de%20intrusos

[37] "Redes neuronales." [En línea]. Disponible en: https://es.wikipedia.org/wiki/Red_neuronal_artificial

[38] "Analisis bayesiano." [En línea]. Disponible en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2448-91902018000300205

[39] "Codigo python." [En línea]. Disponible en: https://stackoverflow.com/questions/9216138/find-the-day-of-a-week

[40] "Libreria pandas." [En línea]. Disponible en: Encodingcyclicalcontinuousfeatures

[41] "Arima." [En línea]. Disponible en: https://es.wikipedia.org/wiki/Modelo_autorregresivo_integrado_de_media_m%C3%B3vil