'Covid-19 predictions and clustering in Castilla-La Mancha's provinces'

FELIX ANGEL MARTINEZ MUELA* and MIGUEL DE LA CAL BRAVO*, Universidad de Castilla La Mancha, Spain

Proyecto final para la asignatura del Máster en Ingeniería Informática llamada Desarrollo de Sistemas Inteligentes, en la cual, se empleará el lenguaje de programación *Python* y un conjunto de librerías para aplicar técnicas de predicción y de *clustering* de la evolución del coronavirus *Covid-19* en las diferentes provincias de la región de Castilla-La Mancha.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Machine learning algorithms**.

Additional Key Words and Phrases: Prediction, Clustering, Covid-19, Castilla-La Mancha, Spain

ACM Reference Format:
Felix Angel Martinez Muela and Miguel de la Cal Bravo. 2020. 'Covid-19 predictions and clustering in Castilla-La Mancha's provinces'. *ACM Trans. Graph.* 0, 0, Article 0 (June 2020), 6 pages.

1 INTRODUCCIÓN

En este trabajo se realiza un estudio aplicando técnicas de predicción y *clustering* sobre la evolución del nuevo coronavirus *Covid-19* en las distintas provincias de la región de Castilla-La Mancha.

Para ello, cogeremos el *dataset* creado manualmente por Miguel de la Cal Bravo (autor también de este trabajo), el cual recoge todos los datos de casos detectados, hospitalizados, altas epidemiológicas, fallecidos y casos activos por provincias, publicados en los informes diarios de Sanidad de Castilla-La Mancha en el portal de la Junta[1].

En la *Figura 1*, podemos ver un extracto del *dataset* mencionado, en la hoja donde se recogen los datos de hospitalizados por cada hospital. Este *dataset* se viene actualizando de forma frecuente desde el inicio de la pandemia y para descargarlo completo, nos dirigiremos al siguiente repositorio y descargaremos el fichero **CLM.xlsx**:

https://github.com/miguelcal97/dsi_prediccionCovid19

Nuestro objetivo consistirá en realizar una serie de análisis, aplicar técnicas de predicción y *clustering* sobre la evolución y el comportamiento del virus en las diferentes provincias, es decir, en Albacete, Ciudad Real, Cuenca, Guadalajara y Toledo. Principalmente, nos centraremos en la variable de **hospitalizados**, en cada uno de los hospitales de Castilla-La Mancha destinados al tratamiento del virus.

Fecha	24-mar	25-mar	26-mar	27-mar	28-mar	29-mar	30-mar	31-mar	01-abr	02-abr	03-abr	04-abr	05-abr	06-abr	07-abr	08-abr	09-abr	10-ab
	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V
Hospitalizados Totales																		
H. Tomelloso	109	117	111	125	134	133	148	142	141	144	144	143	127	114	120	117	116	99
H. Manzanares	30	38	34	41	51	56	63	65	68	64	65	61	53	53	53	53	48	48
H. U. CR	116	132	253	299	351	328	383	372	390	374	379	360	334	344	349	336	304	270
H. Mancha Centro	207	213	213	299	326	342	353	383	391	389	394	423	378	350	336	319	290	261
H. Puertollano	40	36	48	97	96	111	112	112	117	121	122	106	89	80	78	76	74	60
H. Valdepeñas	46	51	51	53	54	79	71	73	92	78	78	76	89	77	81	71	71	66
Ciudad Real	548	587	710	914	1012	1049	1130	1147	1199	1170	1182	1169	1070	1018	1017	972	903	804
C. H. Albacete	286	323	398	424	492	502	551	576	545	554	570	585	575	577	568	531	489	460
Almansa	7	5	5	45	53	59	58	60	56	58	57	56	51	47	39	30	30	30
H. Villarrobledo	70	79	85	110	101	110	119	101	121	129	118	117	108	103	100	92	84	66
H. Hellín	41	50	53	64	66	70	71	71	67	66	54	51	49	49	49	49	47	39
Albacete	404	457	541	643	712	741	799	808	789	807	799	809	783	776	756	702	650	595
H.Toledo	461	488	466	464	536	550	565	590	600	590	587	584	552	564	571	529	511	487
H.N. Parapléjicos	2	4	5	8	7	7	7	7	6	6	5	5	5	5	5	5	3	2
H. Talavera	108	121	66	179	178	161	137	145	140	135	132	129	125	129	137	127	122	123
Toledo	571	613	537	649	721	718	709	742	746	731	724	718	682	698	713	661	636	612
H. Guadalajara	179	149	133	320	341	326	315	312	310	295	283	279	260	253	270	248	246	248
Guadalajara	179	149	133	320	341	326	315	312	310	295	283	279	260	253	270	248	246	248
H. Cuenca	121	146	146	181	191	184	181	189	186	181	177	158	155	156	153	141	136	134
Cuenca	124	146	146	181	191	184	181	189	186	181	177	158	155	156	153	141	136	134
Total CLM	1826	1952	2067	2707	2977	3018	3134	3198	3230	3184	3165	3133	2950	2901	2909	2724	2571	2393
Diferencia día anterior																		
Ciudad Real	93	39	123	204	98	37	81	17	52	-29	12	-13	-99	-52	-1	-45	-69	-99
Albacete	47	53	84	102	69	29	58	9	-19	18	-8	10	-26	-7	-20	-54	-52	-55
Toledo	86	42	-76	112	72	-3	-9	33	4	-15	-7	-6	-36	16	15	-52	-25	-24
Guadalajara	20	-30	-16	187	21	-15	-11	-3	-2	-15	-12	-4	-19	-7	17	-22	-2	2
Cuenca	6	22	0	35	10	-7	-3	8	-3	-5	-4	-19	-3	1	-3	-12	-5	-2
CLM	252	126	115	640	270	41	116	64	32	-46	-19	-32	-183	-49	8	-185	-153	-178

Fig. 1. Extracto de la hoja "Hospitalizados", durante los días más críticos de la pandemia, del dataset realizado

En la *Figura 2*, podemos ver una gráfica con la evolución de hospitalizados, desde que comenzó la pandemia del *Covid-19* hasta la actualidad.

Es importante mencionar que, para el *dataset* realizado, se ha desarrollado una **macro** en *VisualBasic* para la exportación de los datos de cada hoja de *Excel* a diferentes ficheros *.csv*, que cargaremos en los cuadernos desarrollados en *Python* en lugar del fichero global original (explicado más en detalle en el *Anexo A* de la memoria).

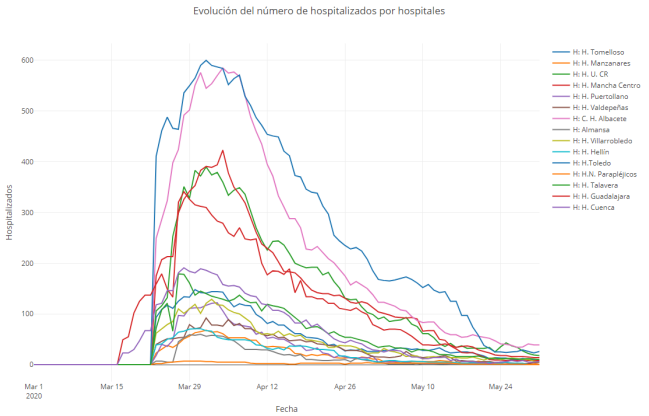

Fig. 2. Evolución de hospitalizados en el tiempo por hospitales

El repositorio del proyecto, se encuentra en el enlace de *Github*: *https://github.com/FelixAngelMartinez/dsi_covid19_final*

*Both authors contributed equally to this research.
[1]https://castillalamancha.es/actualidad/notasdeprensa/

Authors' address: Felix Angel Martinez Muela, FelixAngel.Martinez@alu.uclm.es; Miguel de la Cal Bravo, Miguella.Cal@alu.uclm.es, Universidad de Castilla La Mancha, Ciudad Real, Ciudad Real, Spain, 13005.

0730-0301/2020/6-ART0 $15.00
https://doi.org/

2 PREDICCIÓN DE SERIES TEMPORALES

Entrando en materia, comenzaremos aplicando una técnica de predicción llamada ***ARIMA***, la cual hemos visto que funciona muy bien para predecir con series temporales no estacionarias, en los cuales se puede aplicar un paso diferenciador que consiga eliminar la no estacionariedad en dichas series temporales.

2.1 Motivación y objetivos

En este apartado, tenemos el objetivo de realizar predicciones a corto plazo sobre la variable hospitalizados para los distintos hospitales con plantas destinadas al tratamiento de *Covid-19*.

Gracias a esto, seremos capaces de **predecir la evolución de hospitalizados en cada hospital** y, con ello, podríamos **anticiparnos a potenciales colapsos en hospitales a corto plazo**.

Para realizar las predicciones hemos optado por utilizar la técnica *ARIMA*, cuyos fundamentos se explican en el siguiente apartado. Gracias a esta técnica, podemos estimar el futuro comportamiento del virus y atender eficientemente a los pacientes en los hospitales.

En la *Figura 3*, se muestra una gráfica con la tendencia media y de desviación estándar (*Rolling Mean y Rolling Standard Deviation*).

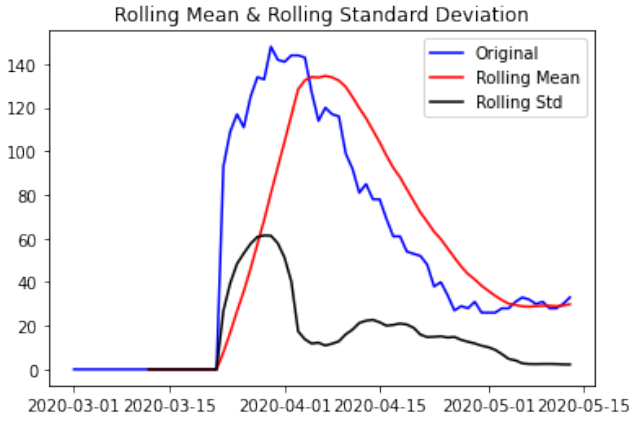

Fig. 3. Cálculo de rolling mean y rolling standard deviation

En la *Figura 4*, podemos ver un ejemplo de predicción realizado con la técnica de *ARIMA*, para el hospital de Tomelloso.

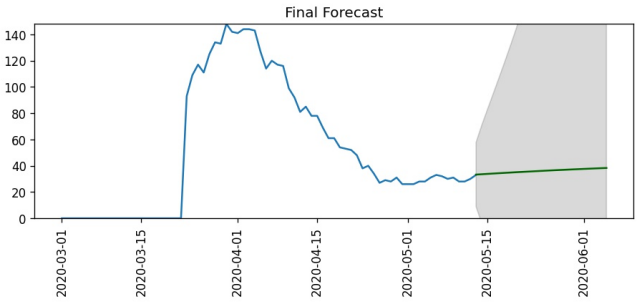

Fig. 4. Predicción realizada con la técnica Arima

2.2 Explicación de la técnica ARIMA

ARIMA proviene de las siglas en Inglés: *AutoRegressive Integrated Moving Average*, desarrollado a finales de los setenta del siglo XX, y sistematizándolo *Box* y *Jenkins* en 1976.

A continuación, se explica cómo utilizar la técnica *ARIMA* para realizar predicciones a corto plazo sobre la evolución del virus.

ARIMA es un modelo estadístico que utiliza variaciones y regresiones de datos estadísticos para encontrar patrones y así lograr una predicción a futuro. Se trata de un modelo dínamico, lo que implica que las predicciones son en base a **datos pasados** y no a variables independientes.

Matemáticamente ARIMA (p,d,q) se expresa como:

$$Y_t = -(\Delta^d Y_t - Y_t) + \phi_0 + \sum_{i=1}^{p} \phi_i \Delta^d Y_{t-i} - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

Modelo en palabras: Predicción Yt = constante + combinación lineal *lag* de y (hasta *p lags*) + combinación lineal de errores de pronósticos del *lag* (hasta *q lags*)

- **d**: se trata de la diferencia necesaria para convertir la serie temporal original en estacionaria.
- **p**: parámetros pertenecientes a la parte AutoRegresiva.
- **q**: parámetros pertenecientes a las medias móviles.

Debemos tener en cuenta que puede generalizarse más el efecto de la estacionalidad empleando el modelo *SARIMA (seasonal autoregressive integrated moving average)*

2.3 prediction_arima.ipynb

Se trata del *notebook* en *Jupyter* ideado para ser ejecutado en *Google Colab*, el cual tiene un carácter didáctico de cómo funciona *ARIMA* de manera manual para en caso de que se quiera ajustar algún hospital para realizar una predicción más personalizada, se pueda realizar, ya que se explica cómo se obtienen los hiperparámetros del modelo (p, d, q), todo ello mostrando las gráficas correspondientes y detalles del modelo.

A la hora de guiarnos para la obtención del mejor modelo se detallan un conjunto de valores de error, los cuales son: *eMAPE*, *ME*, *MAE*, *MPE*, *RMSE*, *ACF1*, *corr* y *minmax*.

2.4 prediction_auto_arima.ipynb

Se trata, al igual que en el apartado 2.3, de un *notebook* en *Jupyter* ideado para ser ejecutado en *Google Colab*. En dicho *notebook* se ha descrito en texto y código la aplicación de *ARIMA* automáticamente sobre un *dataset*, en este caso de hospitalizados, aunque funcionando de igual manera sobre los otros *dataset* presentes en la carpeta *data/* del repositorio. Encontraremos el tratamiento del *dataset* previo a la ejecución, la ejecución de *ARIMA* sobre todos los hospitales.

- Adaptación del modelo *ARIMA* a la serie temporal de dicho hospital y por tanto, ajustando los hiperparámetros del modelo (p, d, q).
- Generación de *plots* con intervalos de confianza sobre las predicciones y posterior almacenamiento de dichos *plots*.
- Almacenamiento de los modelos *ARIMA* empleados y sus correspondientes gráficas.
- Finalmente el almacenamiento de las predicciones, tanto en un archivo común, como en un archivo individualizado por hospital, todo ello en *.csv* con la versatilidad que nos otorga dicho formato.

Después de aplicar *ARIMA* sobre los hospitales en este caso, conseguiremos un resultado de hospitalizados[2] como el que aparece en la tabla 1.

Table 1. Extracto de ejemplo de la predicción del nuevo dataframe

Fecha	H. Tomelloso	...	H. Cuenca
30-05-2020	8.967	...	5.746
31-05-2020	7.913	...	4.989
...	...	...	...
28-06-2020	0	...	0

Las predicciones se han decidido realizar a treinta días vista a partir del último dato del *dataset* original, aunque debemos tener en cuenta que las predicciones más alejadas en el tiempo al día final, serán predicciones con muy poca precisión, siendo las más cercanas aquellas que más se acercarán a la futura realidad.

Tras ejecutar el cuaderno desarrollado, las predicciones de cada uno de los serán almacenadas en *Google Colab* dentro de un fichero *.zip*, pudiendo ser fácilmente descargadas para su posterior análisis.

3 CLUSTERING CON SERIES TEMPORALES

Tras aplicar la técnica de *ARIMA* para realizar predicciones de hospitalizados en nuestra región, pasaremos a realizar un *clustering* con series temporales.

Para ello, hemos desarrollado un cuaderno de *Jupyter* en lenguaje *Python*, haciendo uso de diferentes librerías como *Pandas* para el tratamiento de *dataframes*, *Plotly* para representar las gráficas con los resultados obtenidos, y otras librerías necesarias para realizar algunos cálculos matemáticos, etc.

De nuevo, hemos vuelto a cargar los *datasets* que generamos, poniendo el foco en el correspondiente a los hospitalizados.

3.1 Motivación y objetivos

En este apartado, tenemos el objetivo de realizar un *clustering* con los datos de los diferentes hospitales de nuestra comunidad autónoma, con el fin de estudiar y analizar comportamientos similares entre los hospitales.

Esto podría ser útil para hacer frente a nuevas oleadas del virus en un futuro, mediante la **detección de patrones de hospitalizados** gracias a la aplicación de **técnicas de *clustering* de series temporales**. Esto nos permitirá analizar diferentes evoluciones de la pandemia y que los hospitales sean capaces de estar preparados, para de esta manera conseguir evitar o minimizar colapsos.

Al ser capaces de detectar estos **patrones de comportamiento**, también podríamos **anticiparnos y distribuir aquellos pacientes más leves que se encuentren en hospitales que casi llenen su capacidad, a otros hospitales menos sobrecargados**. Así, podríamos atender a todos los pacientes con la mayor eficacia posible, **adelantándonos a potenciales saturaciones** en los distintos hospitales de Castilla-La Mancha.

[2]A la hora de realizar las predicciones no se tienen en cuenta hospitalizados negativos, ya que en este caso no sería posible, por lo que se sustituye por un 0.

En la *Figura 5*, podemos ver una gráfica con la media de hospitalizados en cada uno de los hospitales, desde que comenzó la pandemia del *Covid-19* hasta la actualidad.

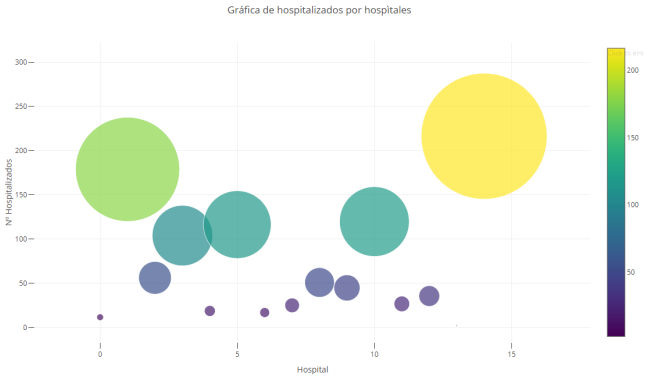

Fig. 5. Número de hospitalizados de media por hospitales

En base a los tamaños y colores de los círculos dibujados, podemos distinguir aquellos hospitales con mayor número de pacientes hospitalizados de aquellos con menor carga.

Los colores también nos podrían servir de referencia y darnos indicaciones para el *clustering* posteriormente, ya que nos muestran aquellos hospitales con una carga de pacientes similar.

3.2 Explicación de la técnica de clustering utilizada para series temporales

Una vez hemos visto la gran utilidad de emplear técnicas de *clustering* con series temporales, es momento de explicar la técnica empleada en nuestros *scripts*.

Tras investigar diferentes técnicas de *clustering* con series temporales, finalmente nos decantamos por utilizar ***K-means*** para la detección de patrones de comportamiento en el tiempo.

La técnica de *K-medias* (o *K-means*), se trata de una de los algoritmos de *clustering* más conocidos en la actualidad para *BigData*.

El algoritmo de las *K-medias* tiene un inicio aleatorio, ya que en primer lugar selecciona unos **centroides** de los datos del *dataset* de forma aleatoria que serían los representantes iniciales de los *K clusters* o grupos diferentes. En cada iteración, dichos centroides se van actualizando para dar lugar a la generación de los *clusters* con aquellos elementos con características lo suficientemente similares entre sí, y lo suficientemente diferenciados del resto, hasta que no se produzcan nuevas actualizaciones.

Esto se calcula con alguna de las técnicas de medición de distancias, como por ejemplo la distancia euclídea entre dos puntos (en nuestro caso, dichos puntos se corresponden con los valores de las series temporales entre los hospitales)

Generalmente, es utilizado para agrupar elementos con características muy parecidas (por ejemplo, aquellos productos más vendidos en una tienda). Sin embargo, nosotros le daremos un enfoque distinto, adaptando su uso a la aplicación de series temporales. Gracias a ello, podremos detectar patrones evolutivos con comportamientos similares en el tiempo de los diferentes hospitales de la región.

3.3 clustering_timeseries.ipynb

Se trata de un *notebook* en *Jupyter* que utilizaremos nuevamente en *Google Colab*, el cual nos permitirá aprender el funcionamiento de un algoritmo de *clustering* y su aplicación a series temporales, con la citada técnica de las *K-medias*[3].

En primer lugar, es necesario tener cargado el *dataset* de los hospitalizados en Castilla-La Mancha, el cual podemos encontrar en el repositorio de *Github* bajo el nombre de *hospitalizados.csv*.

Hecho esto, y tras cargar las librerías necesarias, procedemos con el tratamiento del *dataset* de los hospitalizados, tratándolo como *dataframe* con *Pandas*. En este procesamiento se realizan algunas modificaciones, interpolaciones, etc, de la misma manera que se hizo en los ejemplos de predicción con *ARIMA*.

Adicionalmente, también han sido necesarios otros cambios para un correcto tratamiento del *dataframe*, para así aplicar bien las técnicas de *clustering*. Un ejemplo de este *dataframe* se muestra en la *Tabla 2*:

Table 2. Extracto de ejemplo del nuevo dataframe resultante

Fecha	Hospital	Hospitalizados
2020-03-01	H. Tomelloso	0
2020-03-02	H. Tomelloso	0
...	...	...
2020-05-30	H. Cuenca	5
2020-03-31	H. Cuenca	6

Con nuestro *dataframe* listo para aplicar la técnica de *clustering* de *K-medias* para series temporales, comenzaremos a aplicar la técnica y generar *clústeres* con grupos de hospitales con un **comportamiento similar en la evolución de la curva de hospitalizados**.

Para facilitar la aplicación del algoritmo, se han definido un conjunto de funciones que se explican a continuación:

- `euclid_dist`: realiza el cálculo de la distancia euclídea, dados dos valores *t1* y *t2*.
- `init_centroids`: dado un conjunto de datos y un número de clústeres determinado, inicializa tantos centroides como clústeres se desean formar.
- `calc_centroids`: dado un conjunto de datos y centroides, hace un cálculo de los nuevos centroides.
- `closest_centroids`: dado un conjunto de datos y centroides, devuelve los centroides más cercanos en función de la distancia.
- `move_centroids`: se encarga de la actualización de los centroides.
- `k_means`: dado un conjunto de datos, número de clústeres y número de iteraciones, aplica el algoritmo de K-means llamando a las distintas funciones definidas.
- `cosine_similarity`: realiza el cálculo de la similaridad del coseno, dados dos valores *t1* y *t2*.

[3]Para el desarrollo de este cuaderno, hemos tomado como referencia el siguiente ejemplo de time series clustering: https://www.kaggle.com/egregori/clustering-time-series

Una vez definidos los *K* clústeres necesitaremos definir un diccionario para traducir los identificadores numéricos de los hospitales, a su nombre en formato *string* para su posterior representación. Para ello, llamaremos a la función *plot_hospital* pasándole la lista de hospitales agrupados en cada clúster a representar.

En las *Figuras 6 y 7*, podemos ver dos ejemplos gráficas generadas de la aplicación de nuestra técnica de *clustering* para series temporales.

En ellas, se puede observar dos agrupaciones de diferentes hospitales de Castilla-La Mancha, los cuales detectamos que han seguido un patrón similar en su evolución.

Estos patrones nos indican que el número de pacientes hospitalizados ha crecido y decrecido de forma similar en el tiempo en los hospitales que representan. Esto, nos puede resultar de gran utilidad, como se explicó anteriormente, de cara al futuro para anticiparnos a potenciales colapsos de hospitales.

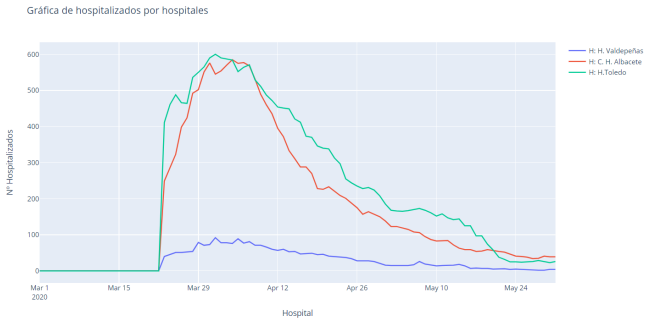

Fig. 6. Ejemplo de patrón de hospitalizados en el tiempo de distintos hospitales, detectado con la técnica de clustering de K-means

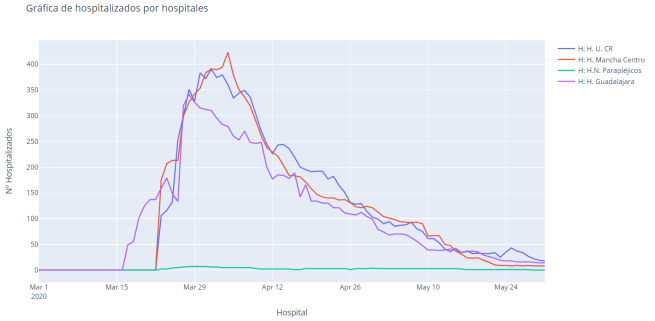

Fig. 7. Ejemplo de otro patrón de hospitalizados detectado con K-means

Si nos fijamos detenidamente en ambas gráficas, observamos una tendencia creciente de forma exponencial en los primeros días, después se producen unos días de estabilización y finalmente una tendencia decreciente más suave en el en número de hospitalizados.

Los clústeres o patrones formados se asemejarán lo suficiente las tres etapas de la serie temporal, es decir, en la escalada, contención/estabilización y desescalada de la curva.

De esta manera, concluimos el apartado correspondiente a *clustering*, habiendo cubierto los dos aspectos clave de la asignatura en este trabajo final, es decir, tanto la parte de predicción como *clustering*.

4 CORRELACIÓN DE SERIES TEMPORALES POR PROVINCIAS EN CASTILLA-LA MANCHA

Para poner el broche final a este trabajo, y después de haber aplicado técnicas tanto de predicción como de *clustering* a la variable de hospitalizados, concluiremos con un análisis aplicando algunas **técnicas de correlación** de las diferentes variables para estudiar el comportamiento del virus en las provincias de nuestra región.

4.1 Motivación y objetivos

A todo lo demás realizado en el trabajo, si le añadimo el incentivo de la desescalada por fases que está teniendo lugar en nuestro país, tendremos un gran interés por conocer cómo se está comportando el virus en las provincias de Castilla-La Mancha.

Al aplicar técnicas de correlación, también podemos extraer comparaciones entre nuestras provincias, y concluir si la desescalada está avanzando de manera similar entre todas ellas.

4.2 correlation_timeseries.ipynb

Para este apartado final, hemos creado un cuarto y último *notebook* de *Jupyter* para *Google Colab*, en el cual cargaremos aquel *dataset* (de los ficheros *.csv* generados) que deseemos para seguidamente extraer estas correlaciones.

Una vez importado, podremos hacer uso de las algunas técnicas de correlación, las cuales se encuentran definidas en diferentes funciones del cuaderno desarrollado:

- `Pearson`: realiza el cálculo del coeficiente de *Pearson*, aplicando la librería incluida en *Pandas*. También podría calcularse con la función *pearsonr* de la librería *scipy.stats*.
- `DTW`: también conocida como *Dynamic Time Warping*, nos servirá para calcular deformaciones dinámicas en el tiempo entre dos secuencias temporales importando la librería *dtw*.

Al aplicar el coeficiente de *Pearson*, hemos sido capaces de extraer los comportamientos epidemiológicos más similares (y diferentes) de las provincias en las distintas variables.

Recordemos que el coeficiente de *Pearson* no es una operación conmutativa, por lo que no obtendremos los mismos resultadosal comparar Toledo con Ciudad Real, por ejemplo, que a la inversa.

Algunas de estas conclusiones se adjuntan en la *Tabla 3*:

Table 3. Conclusiones obtenidas al aplicar la correlación de Pearson

Variable	Más similares	Coeficiente
Activos	Ciudad Real-Albacete	0.9988
Altas	Albacete-Guadalajara	0.9998
Casos	Ciudad Real-Toledo	0.9971
Fallecidos	Albacete-Toledo	0.9974
Hospitalizados	H. Mancha Centro-H. Albacete	0.9896

Como vemos, los mejores coeficientes al comparar las provincias para cada variable se encuentran próximos a 1. Esto significa que se sigue una correlación prácticamente idéntica en estos casos, aunque en el resto de comparaciones también ocurre algo similar.

La tendencia epidemiológica que se sigue es muy parecida en todas las variables para todas las provincias, aunque estas correlaciones nos podrían servir en un futuro para detectar tendencias epidemiológicas anómalas en cualquiera de sus variables.

En cuanto a la técnica de *DTW*, podemos decir que se trata de otra técnica alternativa a *Pearson* para realizar correlaciones en base a secuencias temporales.

Nosotros hemos aplicado ambas técnicas para toda la serie temporal (desde el inicio de la pandemia hasta la actualidad), aunque también podría tener sentido aplicarlo sobre un periodo de tiempo concreto. Esto nos facilitaría la detección de brotes de la enfermedad en una provincia frente a las otras.

Por otra parte, también tendría sentido comparar diferentes variables entre sí dentro de una provincia, por ejemplo, casos detectados y fallecidos. Así, podríamos detectar anomalías dentro de una provincia en cuanto a alguna de las variables, en este ejemplo, ver si se producen nuevos brotes o un cambio en el índice de mortalidad ocasionado por el *Covid-19* a lo largo del tiempo.

5 CONCLUSIONES Y PROPUESTAS

Para terminar, daremos nuestras conclusiones y opinión personal sobre las técnicas de predicción y *clustering* utilizadas.

Para realizar este trabajo comenzamos confeccionando los *datasets* de las diferentes variables por provincias de la región, algo que ha llevado mucho tiempo al trasladar los datos publicados en los informes diarios por Sanidad de Castilla-La Mancha.

Con este conjunto de datos "exclusivo", hemos sido capaces de extraer información de valor al aplicar tanto técnicas de predicción como de *clustering* a las series temporales, en especial, en lo referente a los pacientes hospitalizados.

Hemos utilizado el lenguaje *Python*, ideal para aplicar las técnicas mencionadas, y cuadernos de *Jupyter* en *Google Colab*, lo cual nos ha permitido comentar de forma muy completa los pasos desarrollados. Todo ello, junto con un fácil tratamiento de los datos al generar *dataframes* con la librería *Pandas* y demás librerías para representar gráficas, cálculos matemáticos, etc.

También, nos gustaría añadir algunas propuestas o sugerencias para el futuro, que también podrían resultar de utilidad. Por ejemplo, podríamos extender el uso de las técnicas de predicción y *clustering* a otras variables, como la de casos positivos para determinar brotes dentro de una misma provincia.

Además, también podríamos coger los datos de nuestro país en otras comunidades autónomas y estudiar cómo siguen los patrones epidemiológicos en comparación con las provincias de Castilla-La Mancha.

Como conclusión final, en este trabajo hemos aprovechado los conocimientos adquiridos en la asignatura para aplicar técnicas tanto de predicción, como de *clustering* a los datos de nuestra región, realizando un estudio con datos muy concretos dentro de cada provincia y extrayendo información de valor que consideramos podría resultar de gran utilidad en un futuro.

6 REFERENCIAS

A continuación, se muestran las referencias empleadas para realizar tanto el desarrollo del proyecto, como la presente memoria:

- https://castillalamancha.es/actualidad/notasdeprensa/
- https://machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
- https://pypi.org/project/pmdarima/
- http://alkaline-ml.com/pmdarima/1.6.1/index.html
- https://kaggle.com/egregori/clustering-time-series
- https://towardsdatascience.com/time-series-hierarchical-clustering-using-dynamic-time-warping-in-python-c8c9edf2fda5
- https://statsmodels.org/stable/generated/statsmodels.tsa.arima_model.ARIMA.html
- https://towardsdatascience.com/four-ways-to-quantify-synchrony-between-time-series-data-b99136c4a9c9

A EJECUCIÓN

El proyecto ha sido realizado en lenguaje *Python*, para el cual se han desarrollado los siguientes cuadernos en formato *.ipynb* para *Google Colab*:

(1) **prediction_arima.ipynb**
(2) **prediction_auto_arima.ipynb**
(3) **clustering_timeseries.ipynb**
(4) **correlation_timeseries.ipynb**

Todos los cuadernos están comentados aprovechando la versatilidad que nos ofrecen los cuadernos o *notebooks* de *Jupyter*, por lo que el usuario debería entender los diferentes pasos del cuaderno a medida que se vayan ejecutando.

Recomendamos su ejecución en el orden indicado anteriormente, ya que se corresponde con el orden en el que dichos cuadernos vienen explicados en la presente memoria.

Para su ejecución, también es necesario importar en *Google Colab* los *datasets* en formato *.csv* que se encuentran en la carpeta */data* del repositorio final. En el código de cada cuaderno, y al principio de la ejecución, se pedirá importar un archivo para posteriormente formar nuestro *dataframe* con *Pandas*.

Por otro lado, y para facilitar la exportación de los diferentes ficheros *.csv* correspondientes a cada variable, hemos desarrollado una macro de *Excel* en *Visual Basic* para el *dataset* global (*CLM.xlsx*, o *CLM.xlsm* con la macro ya importada).

Dicha macro tiene el nombre de *Exportar_CSV.bas* y al ejecutarla nos guardará en la ruta indicada los siguientes cinco ficheros, para poder fácilmente importarlos y tratarlos en nuestros cuadernos:

(1) **activos.csv**: contiene el número de casos activos por provincias en Castilla-La Mancha. Representa el número de casos totales menos el número de altas y fallecidos.
(2) **altas.csv**: contiene el número de altas totales en cada provincia de la región.
(3) **casos.csv**: contiene el número de casos totales por provincias. Cabe destacar que se produce un importante descenso en el número de casos contabilizados, debido al cambio de criterio en el conteo definido por Sanidad.
(4) **fallecidos.csv**: contiene el número de fallecidos en total de cada provincia.
(5) **hospitalizados.csv**: contiene el número de hospitalizados en total de cada uno de los quince hospitales regionales para el tratamiento del *Covid-19*.

NOTA: Para ejecutar la macro, es necesario tener habilitado en Excel la pestaña de Desarrollador. Después, buscaremos la macro llamada Exportar_CSV y le daremos a ejecutar.

De esta sencilla manera, podremos tratar por separado las distintas variables a predecir o hacer *clustering* en base a series temporales, de cada una de las provincias de Castilla-La Mancha.