

Cajamar Water Footprint Datathon 2022

1. Introducción

La estimación correcta de la demanda de agua potable representa una condición indispensable para la planificación, diseño y operación eficiente y sostenible de todos los elementos que conforman los sistemas de captación, transporte y suministro de agua potable. Esta demanda está sujeta a variaciones interanuales, estacionales, semanales, diarias e incluso horarias, muy significativas y que dependen de múltiples factores como son los ciclos de actividad económica, la meteorología, las situaciones de crisis sanitaria, los cambios en los bloques tarifarios, etc.

Por tanto se establece como objetivo el crear un modelo de predicción de consumo de agua para realizar estimaciones a futuro, a partir de un conjunto de datos histórico.

2. Minería y comprensión de los datos

Como se ha comentado se parte de un conjunto de datos inicial histórico de consumos en el que se identificaron 6 campos de datos. Un ID, como identificador del contador que registra la medida, SAMPLETIME como variable independiente, la fecha de registro de cada medida. Y después hasta cuatro medidas de registro de datos, de las cuales decidimos centrarnos en el análisis de DELTAINTEGER, por ser esta el consumo realizado en cada uno de los registros.

Durante el proceso de exploración de los datos se determinó que se tenían 2747 contadores de agua diferentes siendo la fecha mínima el 2019-02-01 y la fecha máxima el 2020-01-31. La gran mayoría de los contadores registraron medidas durante el periodo completo, en este caso más de 2000 IDs registraron los 365 días, por lo que se puede comprobar que la serie de datos está bastante completa, pero es destacable también algunos grupos de IDs que registrar incluso menos de 100 días de datos.

Con el objetivo de estudiar el comportamiento de la serie de la manera más sencilla, en primer lugar se decidió trabajar con un grupo de 100 IDs, reduciendo así el tamaño del conjunto. Puesto que el conjunto de datos no tiene muchas características, debemos centrarnos en los aspectos principales de una serie temporal univariante, tales como los ciclos, la tendencia y la estacionalidad (Brockwell &

Davis, 2016). Además, discretizamos la serie temporal para tener un eje temporal con valores diarios, obteniendo la suma de consumo de cada día. Para ello realizamos la descomposición de la serie (Figura 1).

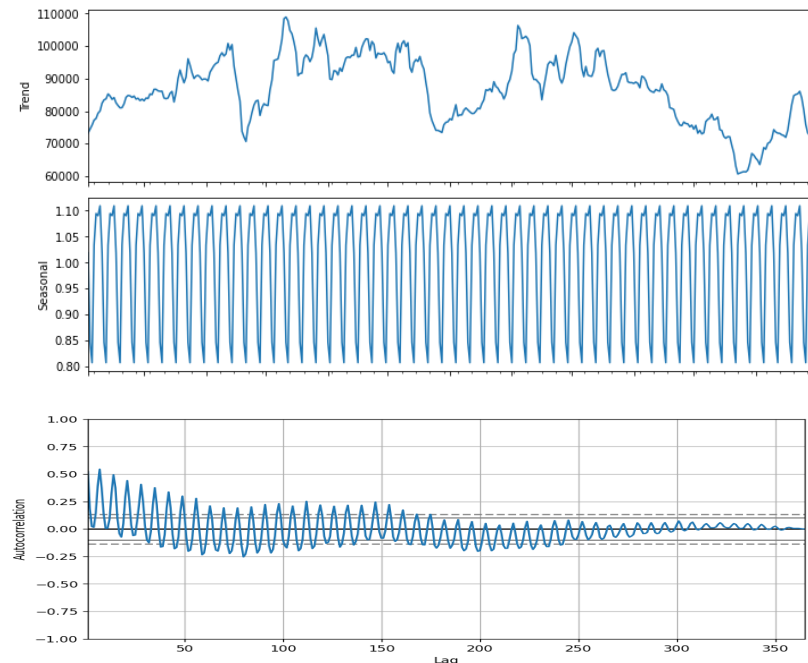


Figura 1. Ejemplo de descomposición de la serie para 100 IDs, junto a gráfico de autocorrelación.

Tras esta toma de contacto con los datos se puede apreciar muy claramente una gran estacionalidad, ya que el patrón se repite constantemente de manera periódica, por otro lado, en cuanto a la tendencia, que quizá es el aspecto más importante a tener en cuenta (Chatfield, 2000), no parece existir una clara a lo largo de la serie, ni que se de un ciclo anual representativo, pero si puede observarse algunas épocas de mayor consumo sostenido en el tiempo y un descenso hacia el final del año 2019 con un remonte en enero. Además la función de correlación empieza con un alto pico y después va oscilando hasta reducirse, lo que puede indicar que existe correlación en el primer desfase, seguida de correlaciones no significativas, en este caso se estaría hablando de una función autoregresiva.

3. Preparación de los datos

Durante el primer análisis hemos actuado sobre un subconjunto de datos de 100 IDs, sin embargo durante el inicio de la exploración ya se pudo observar que existían 2747 contadores, además un porcentaje de ellos no existían durante la serie completa.

En principio, es esperable que entre una gran cantidad de contadores estos no se comporten todos igual. Esto es debido a que algunos pueden ser industriales, otros de hogares particulares, otros de casas de veraneo, etc..

De este modo, el principal objetivo para la preparación de los datos es agrupar estos IDs por semejanza, y para ello utilizamos un modelo de aprendizaje no supervisado.

Para elegir la cantidad correcta de clusters, primero se realizó una normalización de los datos mediante la función MinMaxScaler de la librería Scikit learn, entonces se realizó el Elbow method, y finalmente se determinaron 10 clusters de IDs para aplicar nuestro modelo (figura 2).



Figura 2. Gráfica del codo, y selección de los clústeres. En el eje y, el número de contadores por cada clúster, así como una representación de estos.

Una vez obtenidos los clusters, y tras la exploración de los datos, ya está todo preparado para aplicar el modelo de predicción.

4. Modelado y discusión

Tras la clusterización, demostramos que existían diferentes grupos de contadores. Y tras la exploración se vió que teníamos un modelo altamente estacional en una serie univariante. En ese sentido tras probar múltiples ejecuciones, obtuvimos resultados satisfactorios con la función de predicción GradientBoostingRegressor (GBR).

Sin embargo, y puesto que tenemos diferentes grupos de contadores se decidió aplicar al grupo principal (Cluster 2, 420 lds) , con resultados satisfactorios, un modelo diferente.

Para ello recurrimos a uno de los modelos más populares en series temporales univariadas, los modelos ARMA (Box et al., 1970), y en concreto para prevenir la posibilidad de que el modelo no sea puramente estacionario, un modelo ARMA integrado, es decir un ARIMA. Sin embargo, y puesto que en la exploración de datos vimos una gran estacionalidad, se decidió recurrir al modelo SARIMA (Seasonal-ARIMA), (Box et al., 2016).

El modelo SARIMA requiere la parametrización de sus ordenes (p,d,q) , siendo p el orden de autorregresión, q el orden de la media móvil, (ambos del modelo ARMA) y d la diferenciación del modelo integrado (para el caso ARIMA). Estos órdenes se calcularon en función de criterios de la información según el índice de Akaike (Akaike, 1974), el cual indica que el menor índice conduce al modelo más parsimonioso, la mejor parametrización se obtuvo aplicando una función de tipo gridsearch.

Finalmente, se aplicaron los modelos a los clusteres, prediciendo los 7 primeros días de febrero 2020, para el caso de la predicción semanal, se predijo la media de consumo diaria y se multiplicó por el número de días. El resultado de esta estrategia resultó altamente ajustado, especialmente comparándolo con la estrategia en la que aplicamos a todos los clusteres el algoritmo GBR. Debido a esto, para la segunda fase, nos decidimos a retornar a la fase de modelado y aplicar SARIMA a todo el conjunto de datos. Para ello, primero rediseñamos la función gridsearch de parametrización, para que la aplicase en bucle para cada uno de los clusteres y así obtener los mejores parámetros en cada grupo. Tras esto aplicamos el modelo a los diferentes clusteres, además de en el cluster 2. A pesar de obtener los mismos resultados en este cluster, el error cometido en el conjunto general fue mucho mayor, por lo que en el resto de clusteres este algoritmo no pareció funcionar correctamente. Como alternativa, se decidió utilizar la librería prophet, una librería de predicción de series temporales ampliamente utilizado en este tipo de problemas, aplicando a cada uno de los clusteres. Sin embargo, los resultados arrojados fueron incluso más alejados de la estrategia de SARIMA + GBR.

Como conclusión ante este comportamiento, cabe destacar que el cluster 2, siendo el más grande, concentra aquellas series de bajo consumo (figura 2), algoritmos como el GBR trabajan muy bien en ambientes de poco pre-procesado de los datos y ausencia de estandarización de estos, en general no se ven muy influenciados por los outliers (Amat Rodrigo, 2020), por ello, SARIMA funciona realmente bien en el cluster más estandarizado, el cluster 2, mientras que produce alta cantidad de outliers en los demás, y por ello un empeoramiento en la capacidad predictiva. Esto, además, explicaría la razón por la cual el método SARIMA + GBR funcionaría mejor que si solo aplicáramos el boosting gradient.

Por esta razón, se decidió volver a la fase de preparación de datos y estandarizar las series como paso previo tanto para la clusterización como para la predicción, obteniendo así una distribución diferente de los clusters (Figura 3), pero más orientada al patrón de comportamiento que a la cantidad de consumo de los contadores. Tras ello, se procedió al modelado con la misma estrategia anterior, entrenando con SARIMA las nuevas series estandarizadas, mientras que las series altamente incompletas decidimos seguir tratándolas con GBR. El resultado se comparó con el mejor resultado obtenido hasta la fecha calculando su error cuadrático medio y obteniendo un resultado muy parejo, pero que entendemos mejor.

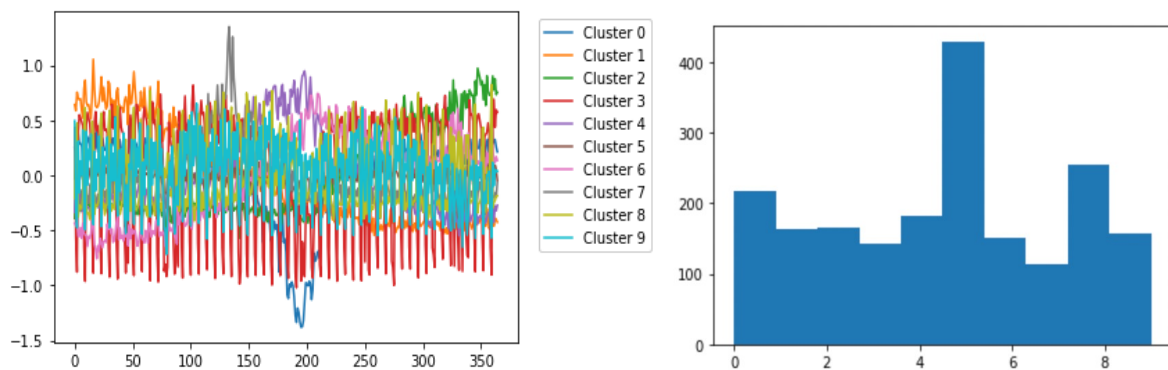


Figura 3. Nueva distribución de clusters de las series estandarizadas, junto con los patrones de los centroides de cada cluster.

5. Referencias

1. Akaike, H. (1973), Information theory and an extension of the maximum likelihood principle, in: BN Petrov and F. Csaki, eds., 2nd Internat. Syrup.
2. Box, G. E. P., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control. San Francisco: Holden Day.
3. Box, G. E., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. (2016), Time Series Analysis: Forecasting and Control, 5th ed., Wiley
4. Brockwell, P. J., Brockwell, P. J., Davis, R. A., & Davis, R. A. (2016). Introduction to time series and forecasting. Springer.
5. Chatfield, C. (2000). Time-series forecasting. CRC press.
6. Gradient Boosting con Python by Joaquín Amat Rodrigo, available under a Attribution 4.0 International (CC BY 4.0)
https://www.cienciadedatos.net/documentos/py09_gradient_boosting_python.html
 consultada en Abril 2022.