La Manzana de Newton (Afi) – Descripción de proyecto

1. Agradecimientos

Desde La Manzana de Newton deseamos mostrar nuestro más sincero agradecimiento a Cajamar, por haber posibilitado un evento de tal magnitud y características que invita a tantos participantes a estudiar, investigar, aprender, en definitiva, reinventarse, para hacer frente a los diversos retos propuestos. A la comunidad educativa de Afi por habernos guiado con tan buen hacer y detalle en este camino que es el llegar a ser nuestra mejor versión como analistas de datos. Finalmente, quien suscribe estas líneas quiere agradecer a sus compañeros de equipo, que durante esta competición le han llevado sobre hombros para ser, aunque solo por unos instantes, un poco más alto.

2. Breve resumen del trabajo desarrollado

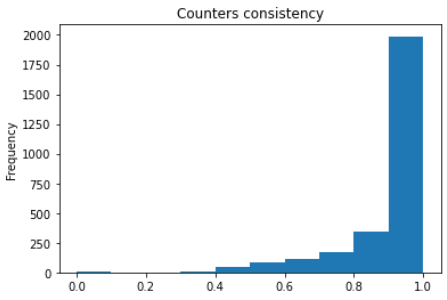
Gran parte de los esfuerzos se han centrado en el análisis exploratorio de los datos y más en particular de los *missing values* que existen en el *dataset* original (horas para las cuales un determinado contador no había tomado una medida). Se acometen varias estrategias para imputar *missing values* atendiendo a cuán malo o bueno es un contador, entendiendo por un contador bueno aquel con casi todas las medidas y uno malo aquel al que le faltan muchas. Se generan nuevas variables explicativas que se considera pueden ser relevantes para el modelo (dado que en el fondo el *dataset* solo tenía una variable ya que DELTA y READING aportan información extremadamente parecida). Más adelante en el presente informe se detalla qué variables se han generado y por qué. Finalmente, con un *dataset* limpio, sin *missing values* y con nuevas variables explicativas existen dos líneas de operatividad: modelos naive para contadores débiles y entrenamiento con modelos de *machine learning* y series temporales para el resto de los contadores. Finalmente se seleccionan los dos mejores modelos (entendiendo por mejores aquellos que obtenían un menor *RMSE* en el conjunto de *test* que se fabrica con la última semana de enero).

3. Resumen del análisis exploratorio llevado a cabo

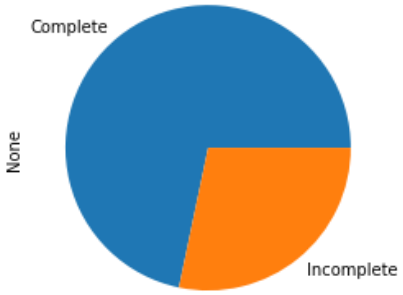
Tras la carga de datos se observa que se disponen las siguientes variables: ID del contador, SAMPLETIME, READINGINTEGER, READINGTHOUSANDTH, DELTAINTEGER, DELTATHOUSANTH. Se conoce que DELTA es el consumo de agua para un contador en un registro y READING es la medida acumulativa de dicho contador. Se asume que la diferencia entre dos medidas READING consecutivas de un mismo contador debe ser igual a DELTA. Se procede a la realización de un pequeño estudio de consistencia de contadores, asumiendo como consistente aquel contador que cumple con el siguiente axioma:

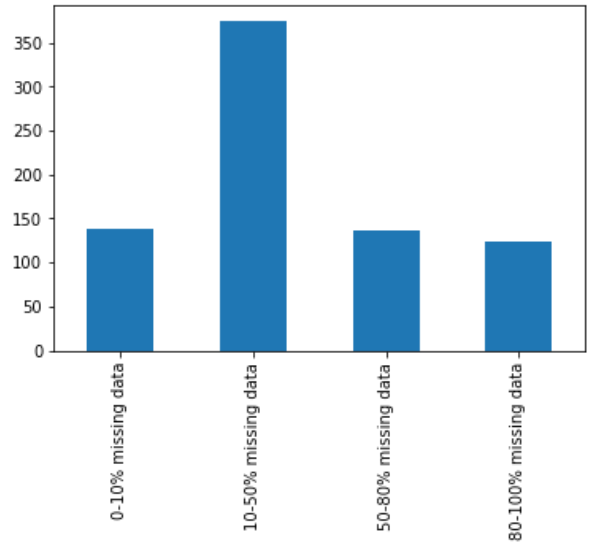
$$DELTA = READING_{t+1} - READING_t$$

Si la consistencia es igual a 1, el contador cumple siempre con dicho axioma y si es igual a 0, no cumple jamás. En la siguiente gráfica se puede analizar de forma visual la cantidad de contadores que se tienen consistentes (unos 2000 aproximadamente) y los que no lo son (unos 700).



Paralelamente se procede al estudio de la cantidad de contadores que poseen todas las medidas que se asume deben tener, que debe responder a lo siguiente, una medida por hora durante un año completo: $24\ horas * 265\ días - 1 = 8759$. Se tienen 1970 contadores completos y a 777 les faltan medidas en mayor o menor proporción.





Los valores de las variables READINGTHOUSANDTH y DELTATHOUSANDTH, que oscilan entre 0 y 99 se dividen entre 1000, entendiendo por la palabra *thousandth* de sus nombres, que hacen referencia a milésimas. Posteriormente se agrega el resultante a sus respectivas partes enteras, DELTA y READING. Cabe destacar que a pesar de la existencia una discrepancia entre nombre de variable (milésima) y su rango (0-99), se toma la anterior decisión con el fin de ser estrictos con el nombre de la variable. En caso de ser una hipótesis errónea, el efecto que tendría sería el de suavizar las lecturas redondeándolas a la baja. Tras el agregado a sus respectivas columnas, las variables milésimas son eliminadas del *dataset*.
La columna SAMPLETIME, que hace referencia al día y la hora en que las medidas READING y DELTA son tomadas, se formatea a tipo *datetime* con el fin de facilitar su manipulación. Con el fin de que la información esté agrupada de forma intuitiva, se ordenan los registros por ID y fecha. Debido a que las medidas de los contadores no son a la misma hora y ni siquiera el espacio entre medidas es de exactamente una hora se hace inviable trabajar a nivel horario y se opta por la solución de agrupar por días las medidas. De esta manera se disponen valores de DELTA y READING únicos para cada día y contador y se facilita la visualización.

4. Resumen de la manipulación de variables y su argumentación

Muchos de los esfuerzos durante el trabajo realizado se focalizan en el preprocesado de la información; limpieza de datos, alimentación del *dataset* con nuevas variables, imputación de *missing values* y tratamiento de *outliers*. La obtención de resultados satisfactorios tras la aplicación de un modelo depende, en crucial medida, del trabajo de preprocesado al que el set de datos se haya sometido previamente.

4. 1. Limpieza de datos

Las lecturas de datos de consumo cuyos valores sean negativos, se transforman a 0, en caso de existir previo a la aplicación de modelos predictivos, e incluso si se obtuviese una predicción con dichas características (postprocesado).

4. 2. Alimentación del *dataset* con nuevas variables

La variable DATE se extrae de la columna SAMPLETIME, de donde se aíslan fechas de mediciones de contadores sin horas.

Asimismo, se alimenta al set de datos con variables temporales tipo seno coseno, tanto del día de la semana como del día del año. El motivo de ello es el dotar de un carácter cíclico a nivel numérico a estas variables, evitar situaciones como: un 31 de diciembre, que es el día 365, sea interpretado por un algoritmo como 365 veces un 1 de enero, que es un día 1.
Se genera la variable IS_WEEKEND, binaria, que toma valores 1 (sábados o domingo) o 0 en caso contrario, entendiendo que los consumos de agua pueden verse afectado por este tipo de días.
Finalmente, se alimenta al set de datos con variables tipo climáticas (temperatura media, sol, precipitaciones), tomándose datos meteorológicos de las estaciones de AEMET en Valencia.

4. 3. Imputación de *missing values*

En el análisis de las variables originales del problema se descubren 140056 valores nulos, tanto en la variable READINGTHOUSANDTH y DELTATHOUSANDTH. Dichos valores son imputados a 0 por pertenecer a variables que miden la parte decimal de la medida registrada por el contador en litros.
Uno de los puntos críticos del preprocesado de datos es la toma de decisión sobre cómo tratar aquellas medidas inexistentes: creación de registro e imputación de valores. Se considera que un contador posee medidas inexistentes si y solo si no posee la totalidad de su serie temporal cubierta de forma continua, en definitiva, si faltan registros para determinadas fechas. La inexistencia de medidas para contadores se considera para el presente estudio muy problemática debido a la gran magnitud de valores que faltan. En consecuencia, se decide acometer el problema empleando los consumos DELTA y READING desde la siguiente perspectiva:

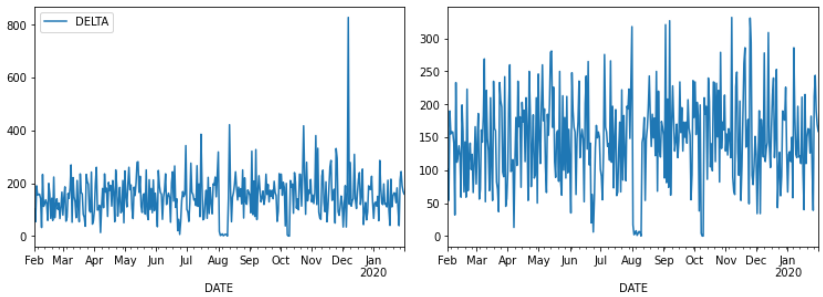
1. Si todos los datos existen y se asumen como correctos, se considera READING como valor fiable.
2. En caso de falta de datos, se recurre a DELTA con el fin de completar estos.

Asimismo, se procede a seguir el siguiente algoritmo:

1. Si existen 24 medidas, es decir, las medidas para un día están completas:
    i. Si existen 24 medidas para el día anterior se toma:
    $$max(READINGactual) - max(READINGanterior)$$
    ii. Si no existen 24 medidas para el día anterior, es decir, está incompleto, se toma:
    $$sum(DELTA_{actual})$$
2. Si no existen 24 medidas, es decir, el contador está incompleto para ese día:
    i. Se calcula el promedio de las medidas que existan y se multiplica por 24, como si virtualmente existiesen 24 medidas:
    $$24/N_{medidas} * sum(DELTA_{actual})$$
3. En caso de no existir una medida para un día se imputa como *None*.
    i. Para aquellos comprendidos entre dos días para los que sí se dispone de medidas, se toma la última medida de READING del día en el que empieza la secuencia de *None* y la primera del día que termina dicha secuencia. La resta de dichas medidas revela el consumo de agua total durante los días sin medidas, es decir, durante la secuencia *None*. Con el fin de interpolar entre estas dos medidas se divide este consumo total entre el número de días sin medidas.

4. 4. Tratamiento de *outliers*

Con el fin de aplicar de forma eficaz los modelos a los contadores se decide realizar una limpieza de *outliers*. La estrategia seguida consiste en la imputación a la media, contador a contador, de todos aquellos valores que exceden las fronteras de un *boxplot*, tanto superior (tercer cuartil más 1.5 veces el rango intercuartílico), como inferior (primer cuartil menos 1.5 veces el rango intercuartílico). A continuación, se exponen dos gráficas comparativas de las series de consumo, para un contador específico, previo y tras la limpieza de un *outlier* notablemente visible en diciembre de 2019.



4. 5. Set de datos final

Tras los procesos anteriormente descritos se logra disponer de cuatro tipos de contadores:

1. Contadores completos.
2. Contadores a los que les faltan medidas al principio.
3. Contadores a los que les faltan medidas al final.
4. Contadores a los que les faltan medidas al principio y al final.

Asimismo, se dividen los contadores en los tipos según su serie temporal de consumos. Esta división es de elevada importancia ya que dependiendo del tipo de contador se aplica una estrategia de predicción distinta.

Tipo 1. Contadores cuyas medidas son todas 0.
Tipo 2. Completos, como mínimo en enero de 2020 y no pertenecientes al Tipo 1.
Tipo 3. Contadores sin medidas en noviembre y diciembre de 2019 y enero de 2020 pero con medidas en febrero de 2019 y no pertenecientes al Tipo 1.
Tipo 0. Contadores no pertenecientes a ninguno de los anteriores tipos.

5. Justificación de la selección del modelo

Los modelos empleados se justifican en función del contador específico al que se apliquen. Se procede a desglosar, contador a contador, el procedimiento final.

5.1. Contadores de tipo 0 y tipo 1: el modelo más naive

Los contadores de Tipo 0 o bien no tienen medidas en enero de 2020 ni en febrero de 2019 (cabeza y cola), o bien tienen muy pocas. En cualquier caso, son contadores muy difíciles de predecir ya que no se dispone de la cantidad apropiada de datos y en el caso de disponerse, son datos poco relevantes (por ejemplo, consumos de junio). En estos casos se considera que la mejor opción es la realización de una estimación grosera, el modelo más naive existente: la media. Para el contador *i*, en la fecha *j*, se predice: $PREDICCIÓN_{i,j} = avg(DELTA_i)$
donde $avg(DELTA_i)$ es la media de consumos del contador *i*. A pesar de que la media puede no ser precisa ya que está fabricada a partir de pocos datos o datos lejanos a febrero de 2020, ante este tipo de contadores muy débiles se considera es la mejor estimación posible.
Los contadores de Tipo 1 son contadores cuyas medidas han sido 0. Se llega a la siguiente conclusión: quizás sean contadores ubicados en viviendas deshabitadas en las que no se consume agua por lo que se considera que la mejor opción es predecir consumo 0 para estos contadores, $PREDICCIÓN_{i,j} = 0$.

5.2. Contadores de tipo 3

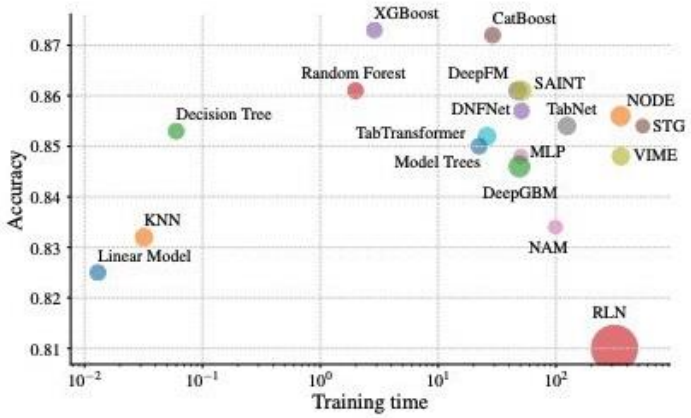
Los contadores de Tipo 3 son aquellos sin suficientes medidas en enero de 2020 (mes previo a la predicción) pero con medidas en febrero de 2019 (el mes de la predicción un año antes). Para estos contadores se considera que una mejor estimación que la media es la siguiente:

$$PREDICCIÓN_{i,j} = ½ * (DELTA_{j,2019}) + ½ * avg(\beta_i)$$

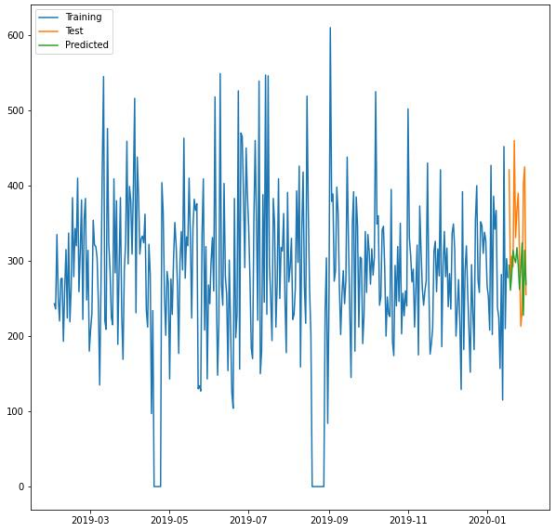
Donde $(DELTA_{j,2019})$ es el consumo el día *j* de 2019 y β_i son los consumos durante las dos primeras semanas de febrero del contador *i*.

5.2. Contadores de tipo 2

Las predicciones anteriores cubren el 7% de los contadores. No obstante, los principales esfuerzos de predicción y donde se emplean modelos de *machine learning* es en este tipo de contadores, que como mínimo cubren las medidas del mes de enero de 2020. Para estos contadores se han entrenado diversos modelos de *boosting*, XGBoost, Gradient Boosting, etc., así como modelos de series temporales, AutoARIMA y Prophet, un algoritmo de Facebook. De todos ellos, los contadores de Tipo 2 se entrenan con AutoARIMA y XGBoost, que han mostrado un rendimiento superior. En la siguiente tabla se visualiza en una tabla la precisión de diversos algoritmos frente al tiempo de entrenamiento que requieren, siendo XGBoost, el que obtiene mejores precisiones.



El primer modelo que se emplea AutoARIMA. Este tipo de modelo está enfocado a series temporales de carácter estacional (como el consumo de agua) y se basa en hacer *finetuning* de hiperparámetros (*p,d,q*) que habitualmente tiene un modelo ARIMA. Para la implementación se ha usado la biblioteca *pmdarima*. Las predicciones son guardadas individualmente. En la siguiente imagen se expone un ejemplo de visualización de predicción de AutoARIMA, en este caso, muy distante al conjunto de test, ya que se trata de unas predicciones que se hicieron en la fase de pruebas, sin tener el set de datos limpio.



XGBoost Regressor se aplica sobre set de datos enriquecido con variables tipo temporales y climáticas. Para cada contador *i*, se entrena un XGBoost donde *X_train* está compuesto por las variables temporales y climáticas generadas en las fechas para las que *i* tenga datos e *y_train* por los consumos del contador *i*. Una vez entrenado se emplea este modelo para predecir los consumos del contador *i* de las dos primeras semanas de febrero.

La predicción final que se añade al *dataframe results_df* es la media entre la predicción del modelo AutoARIMA y la predicción del modelo XGBoost.