



1. RESUMEN.

Nuestro equipo enfrentó el reto de crear un modelo de predicción de consumos de agua enfrentando dos vertientes: Sk Forecast Vs Tensorflow | **MACHINE LEARNING Vs DEEP LEARNING**. Ambas poderosas herramientas de predicción. Las cuales mostraron funcionar efectivamente, y por detalles de ejecución, ajenos al potencial de las herramientas enunciadas, nos decantamos por Sk Forecast.

La ingeniería de datos, es en nuestro concepto, el factor decisorio en la implementación del modelo presentado. El entender la naturaleza de los datos y clasificarla, es lo que hicimos en la fase de Exploración, tenemos claro de que se puede hacer varias acciones que podrían mejorar el modelo, sin embargo presentamos lo que consideramos un acercamiento aceptable a la solución óptima.

2. METODOLOGÍA.

- a. Exploración: En general se pretende recalculer los datos de consumo DELTA's, teniendo en consideración que no deben existir lecturas ni consumos negativos. Los outliers se trataron con un patrón estadístico escogido discrecionalmente. Para los datos faltantes se decidió por usar data sintética creada a partir de interpolaciones bidireccionales. Nuestro modelo evidencia la necesidad de generar clases (clusters) afines para ser tratados en posibles procesos de entrenamiento y validación propios de su grupo.
 - i. Lectura de datos:
Se leen los datos cargados en un hosting propio.
 - ii. Ordenado de los datos:
Se ordenan los datos de forma ascendente.
 - iii. Comprobación de datos faltantes:
Los datos faltantes son se atribuyen a los decimales de algunos readings y deltas, se opta por colocarlos en cero.
 - iv. Comprobación de datos repetidos:
No hay datos repetidos.
 - v. Se comprueba que hay lecturas y consumos de valores negativos que se van a "corregir" poniéndolos en positivos.
 - vi. Consolidación de lecturas y consumos en columnas READING Y DELTA:
Se generan las lecturas y consumos provistos por el concurso.
 - vii. Recálculo del DELTA:
Se detectaron errores en el delta entregado (la suma de los consumos no es la diferencia entre la última lectura y la primera), usamos las lecturas para calcular los consumos. Se hace una primera interpolación a nivel de horas usando el método 'from_derivatives' con el objetivo de eliminar valores faltantes. Luego ya a nivel de días se realiza una interpolación lineal.
 - viii. Manejo de Outliers:
Se reemplazan los outliers que se alejan más de 3 veces la varianza con respecto a la media, con una ventana de 7 posiciones, se reemplaza por la media. Usamos HAMPEL.

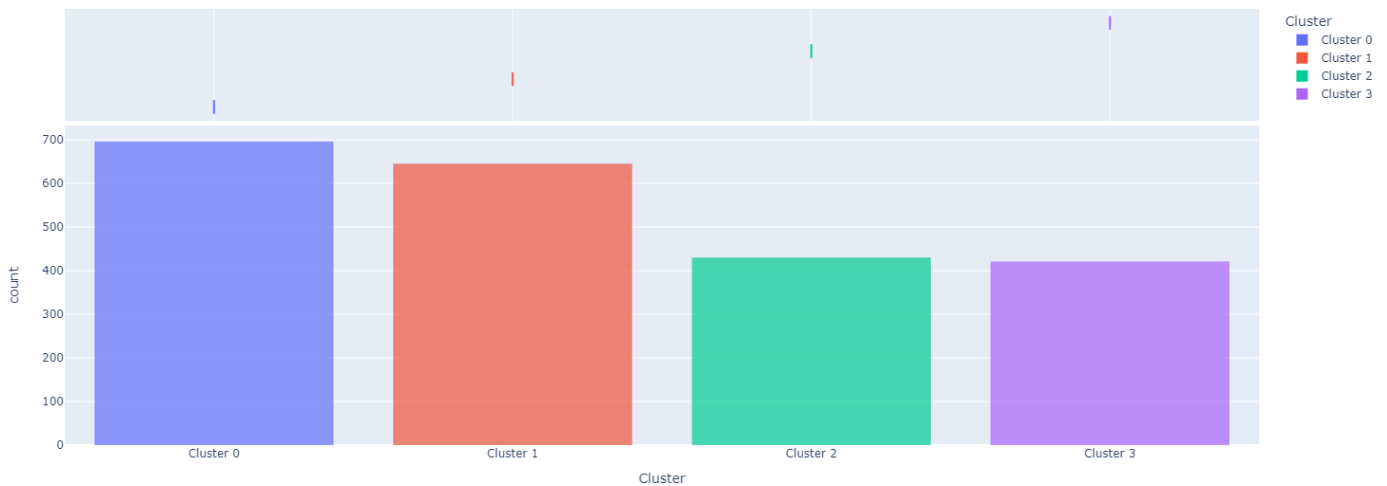
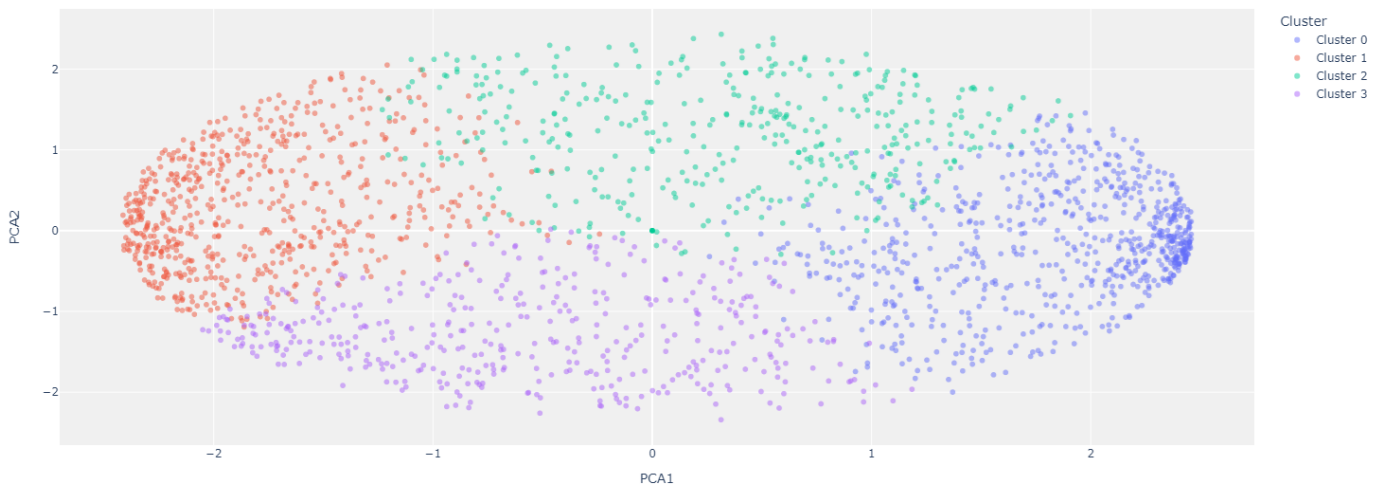


ix. Clustering de Entrenamiento:

Decidimos dividir el dataset por clusters, con características parecidas, para ayudar a que los modelos se ajusten mejor y evitar el underfitting. Resulta 4 clusters como el número óptimo para dividir el dataset. Usamos PYCARET.

x. Clasificación de todos los ID's (Contadores).

2D Cluster PCA Plot



Clasificamos todos los datos según el clúster asignado.

xi. Exportación de datos por clasificación.

Exportamos y alojamos datos en nube.

b. Desarrollo y Predicción:

Se usaron los datos clasificados de forma genérica, usando el mismo modelo para cada clúster.

Evaluamos 2 modelos: uno con Tensor Flow cuya base es dividir para cada clase lo que consideramos comportamiento de señal (oscilación de cada contador), y la media móvil, haciendo entrenamiento, y validaciones independientes, y al final conjugarlos para formar la predicción. El segundo modelo



utiliza Sk Forecast con optimización de parámetros por medio de autoregresión utilizando random forest.

El proceso que se describe a continuación se realiza para cada clúster, a excepción del cálculo del calibrador y de la consolidación (y exportación de los datos).

- i. Importación de datos clasificados:
Se leen datos del clúster cargados previamente en la nube.
- ii. Preparación de los datos:
 1. Se añade como datos de validación los 14 días del mes de febrero del año anterior.
 2. Se inicializa el calibrador el cuál es una “parámetro” que ajusta la serie a entrenar, con respecto a su predicción:
El calibrador se calcula como la mediana del conjunto de la media del factor de los datos de test entre la predicción.
Se decide utilizar la mediana como indicador preferente ante la media, para eliminar en última instancia, la influencia en los outliers del calibrador.
 3. Escogemos una serie característica, la cual se utilizará en el proceso de selección de hiper parámetros óptimos (número de estimadores y profundidad), utilizados para el autoregresor (random forest), del predictor (forecaster): Esta serie cumple con la caracterización de completitud y varianza.
Se decide no entrenar el valor óptimo para cada contador y de esta forma mermar el tiempo general de entrenamiento.
 4. Se definen como variables exógenas:
 - a. La media transversal: Se calcula la media por día de todo el clúster.
 - b. La serie calibrada: Multiplicada por el calibrador.
 - c. El día de la semana.
 - d. El mes.
- iii. Se realiza la optimización de hiper parámetros:



1. Se utiliza un grid con de 2X3 para hallar el número óptimo de estimadores y profundidad.

```

Number of models compared: 12
loop lags_grid: 0%| 0/2 [00:00:?, ?it/s]
loop param_grid: 0%| 0/6 [00:00:?, ?it/s]
loop param_grid: 17%| 1/6 [00:02<00:10, 2.02s/it]
loop param_grid: 33%| 2/6 [00:07<00:16, 4.12s/it]
loop param_grid: 50%| 3/6 [00:11<00:12, 4.10s/it]
loop param_grid: 67%| 4/6 [00:20<00:12, 6.06s/it]
loop param_grid: 83%| 5/6 [00:24<00:05, 5.37s/it]
loop param_grid: 100%| 6/6 [00:28<00:00, 4.90s/it]
loop lags_grid: 50%| 1/2 [00:28<00:28, 28.92s/it]
loop param_grid: 0%| 0/6 [00:00:?, ?it/s]
loop param_grid: 17%| 1/6 [00:01<00:09, 1.83s/it]
loop param_grid: 33%| 2/6 [00:05<00:11, 2.79s/it]
loop param_grid: 50%| 3/6 [00:07<00:07, 2.44s/it]
loop param_grid: 67%| 4/6 [00:11<00:05, 2.99s/it]
loop param_grid: 83%| 5/6 [00:13<00:02, 2.67s/it]
loop param_grid: 100%| 6/6 [00:17<00:00, 3.16s/it]
loop lags_grid: 100%| 2/2 [00:46<00:00, 23.15s/it]
`Forecaster` refitted using the best-found lags and parameters, and the whole data set:
Lags: [1 2 3]
Parameters: {'max_depth': 7, 'n_estimators': 100}
Backtesting metric: 11.938391980106982

```

2. Se genera el predictor usando como regresor random forest, con los hiper parámetros hallados y con lags (número de datos anteriores para generar predicción siguiente) 14.
3. Se realiza un ciclo que recorre todo el clúster, el cual ejecuta:
 - a. Entrenamiento.
 - b. Predicción.
 - c. Acumulación de error (RMSE).
 - d. Acumulación de calibradores.
 - e. Cálculo de tiempo.

iv. Se calcula la media del RMSE del clúster.

v. Se genera en un data frame la preconsolidación de las predicciones del clúster.

- c. Análisis del calibrador:

Definimos como calibrador a la mediana de la lista de las medias de los cocientes de los datos test (validación) y las predicciones.

Decidimos usar la mediana en lugar de la media, para eliminar, de alguna forma, los outliers en última instancia.

El calibrado se realiza ejecutando primero la predicción con el calibrador (calibrate) inicializado en 1, a medida que se vaya ejecutando el modelo el calibrador se irá ajustando. En el caso particular enviado, fueron necesarias sólo 2 ejecuciones, para llegar a un calibrador redondeado de 0.9.

- d. Consolidación del modelo:

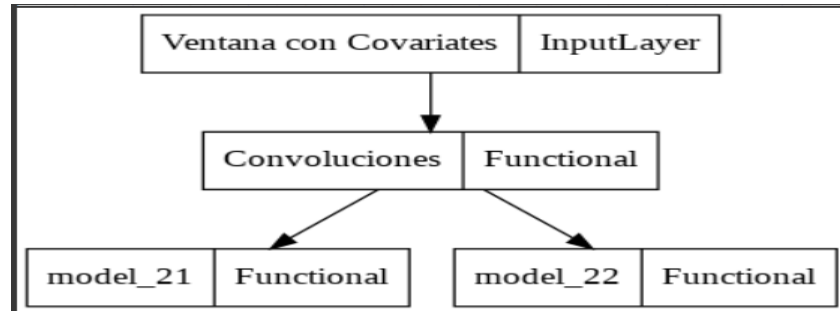
Finalmente concatenamos en un dataframe la pre consolidaciones y hacemos la exportación.

3. EVALUACIÓN DE MODELOS:

- a. Descripción modelo Tensor Flow con el cual se compara SK Forecaster:



Por cada cluster se entrena modelo que intenta predecir la serie estacionaria normalizada y la media móvil del contador. Al combinar estas dos predicciones se obtiene la predicción de los consumos.



El modelo consta de dos capas convolucionales que alimenta a dos modelos de 3 capas densas (completamente conectadas) que trabajan en paralelo cada una para predecir la serie normalizada y la media móvil respectivamente.

Para entrenar se eligieron al azar, manualmente, varios pares de contadores del cluster. Uno sirvió para entrenar y otro para validar el entrenamiento.

- b. Los hiper parámetros seleccionados en los modelos cumplen con los siguientes criterios:
 - i. Las predicciones se hacen usando 14 días (lags y steps).
 - ii. Los parámetros del regresor fueron calculados escogiendo los mejores del grid search propuesto.
 - iii. Los parámetros de configuración del grid search obedecen principalmente a tiempo de ejecución y fitting aceptable.
- c. Selección del modelo SK Forest (SKF), frente al modelo Tenserflow (TF):
 - i. Aunque ambos modelos utilizan series representativas de cada clúster para entrenar, el modelo SKF sólo usa la serie representativa para hallar los mejores hiper parámetros, no para el entrenamiento del modelo.
 - ii. El modelo SKF entrena y predice cada una de las series (contadores).
 - iii. El uso de un calibrador hace del modelo SKF más flexible e implementable: En desarrollos distintos su definición puede variar para ajustarse al dominio en el que se emplee.
 - iv. Al enfrentar los modelos en la entrega intermedia de la fase II, el modelo SKF arrojó el mejor resultado.