

Big Data.

Informe Parte 2

José Angel Lozano Pineda

Contenido

[1) Introducción 3](#_Toc59645774)

[2) Neural Network 4](#_Toc59645775)

[3) Random Forest 5](#_Toc59645776)

[4) Clustering - Knn 6](#_Toc59645777)

# 1) Introducción

Una vez realizada la parte 1 del proyecto, tenemos el fichero de datos de las medidas de los diferentes sensores a lo largo del mes, sobre el cual hemos trabajado en esta segunda parte, realizándole un procesado, donde obteníamos en el documento las medidas a lo largo de un día de un sensor, junto a la media de las medidas del día siguiente, para a continuación realizar un análisis con técnicas de Machine Learning y finalmente obteniendo diferentes graficas con técnicas de visualización.

En este documento se muestran los resultados que hemos obtenido en la fase de análisis con las diferentes técnicas.

Dado que no es un documento técnico de implementación, en el Anexo 1 se explican a grandes rasgos detalles de la implementación, estructura del entregable y ficheros que se generan, entre otros datos.

# 2) Neural Network

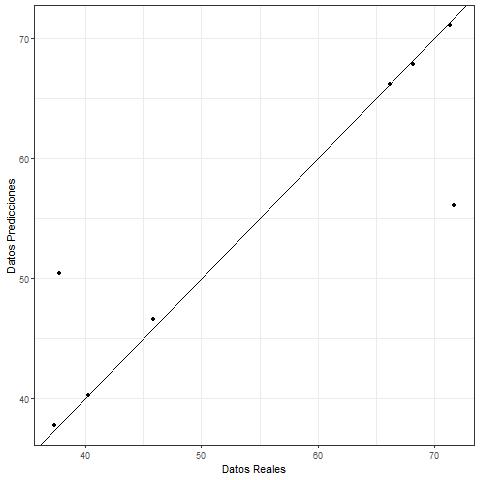
En este apartado se ha realizado un análisis de la línea de distribución “DG1000420”, utilizando un 70% de los datos como datos de entrenamientos y un 30% para pruebas. Para el entrenamiento del modelo, se ha definido que se prueben redes con un tamaño de capas ocultas de 1 a 6, y con un decay de 0.1 a 0.5, obteniendo los siguientes errores para cada caso:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **size** | **decay** | **RMSE** | **Rsquared** | **MAE** |
| 1 | 0.1 | 0.11707034 | 0.9611486 | 0.09939212 |
| 1 | 0.2 | 0.11594365 | 0.9669573 | 0.09860696 |
| 1 | 0.3 | 0.11808985 | 0.9522740 | 0.10164811 |
| 1 | 0.4 | 0.11686991 | 0.9567815 | 0.10286259 |
| 1 | 0.5 | 0.12183826 | 0.9500015 | 0.10671013 |
| 2 | 0.1 | 0.10735571 | 0.9373411 | 0.09391544 |
| 2 | 0.2 | 0.10751394 | 0.9503889 | 0.09512787 |
| 2 | 0.3 | 0.11036806 | 0.9525175 | 0.09849298 |
| 2 | 0.4 | 0.10592852 | 0.9528928 | 0.09490977 |
| 2 | 0.5 | 0.11276620 | 0.9605786 | 0.10153819 |
| 3 | 0.1 | 0.09791158 | 0.9569347 | 0.08866366 |
| 3 | 0.2 | 0.10144686 | 0.9552297 | 0.09252336 |
| 3 | 0.3 | 0.10800394 | 0.9511920 | 0.09817047 |
| 3 | 0.4 | 0.10587147 | 0.9514953 | 0.09688105 |
| 3 | 0.5 | 0.10440221 | 0.9600793 | 0.09477556 |
| 4 | 0.1 | 0.10096767 | 0.9561006 | 0.09190983 |
| 4 | 0.2 | 0.09771535 | 0.9552612 | 0.08894456 |
| 4 | 0.3 | 0.09833170 | 0.9564592 | 0.08962744 |
| 4 | 0.4 | 0.10084744 | 0.9553384 | 0.09136528 |
| 4 | 0.5 | 0.10085932 | 0.9525184 | 0.09116391 |
| 5 | 0.1 | 0.10072681 | 0.9582018 | 0.09207429 |
| 5 | 0.2 | 0.10094098 | 0.9564165 | 0.09330480 |
| 5 | 0.3 | 0.10279524 | 0.9586179 | 0.09472699 |
| 5 | 0.4 | 0.10258321 | 0.9555564 | 0.09512720 |
| 5 | 0.5 | 0.10407558 | 0.9572441 | 0.09655982 |
| 6 | 0.1 | 0.09740348 | 0.9586079 | 0.09038276 |
| 6 | 0.2 | 0.10023180 | 0.9590777 | 0.09317681 |
| 6 | 0.3 | 0.10214447 | 0.9581905 | 0.09452307 |
| 6 | 0.4 | 0.10295145 | 0.9577842 | 0.09550217 |
| 6 | 0.5 | 0.10413368 | 0.9568574 | 0.09656069 |

Con los valores calculados, se ha establecido como criterio de elección la que nos da un menor RMSE, que conseguimos con los valores size=6 y decay=0.1.

Con esta red neuronal establecida, se ha realizado la predicción con el conjunto de datos de prueba, obteniendo los siguientes resultados para las predicciones, en comparación a los reales.

|  |  |
| --- | --- |
| **Real** | **Predicción** |
| 66.28153 | 66.16260 |
| 45.82699 | 46.52666 |
| 40.20797 | 40.24702 |
| 68.18978 | 67.87985 |
| 71.75705 | 56.09095 |
| 71.41090 | 71.10339 |
| 37.26335 | 37.68678 |
| 37.73034 | 50.36196 |



Con estos resultados, podemos determinar que el modelo tiene buen funcionamiento, pero cuando realiza una predicción errónea, el error es de un tamaño considerable, por lo que no podríamos fiarnos de ese resultado. Esto podría ser debido a que el modelo es únicamente entrenado para una línea de producción, con los datos de un mes, y siendo pocos datos con los que se ha entrenado la red, obtiene en general buenos resultados. El modelo sin duda podría obtener más fiabilidad y menor error con una base de datos mayor para realizar el entrenamiento del modelo.

# 3) Random Forest

En este apartado se ha realizado una predicción basada en el método Random Forest para el que hemos utilizado los mismos datos de prueba y test que en el apartado anterior, para así poder realizar comparaciones de resultados. Para esto, se han definido las variables del árbol profundidad máxima como 5, y el número máximo de árboles como 100, que tras una investigación para determinar el número óptimo de árboles, e intentando realizar una aplicación del método lo más real posibles se determinó así *“... An optimal number of trees is obtained by reviewing the model performance, looking for a threshold from which increases in the number of trees would bring no significant performance gain, and would only increase the computational cost. Previous literature suggests that a random forest should have several trees between 64 and 128. An optimal number of predictors is obtained after running the optimal number of trees and testing how OOB error changes according to the number of random attribute candidates in each tree and selecting which reduces it. ...”* [1]

Con estos datos entrenamos el modelo, el cual obtenemos los siguientes errores:

|  |  |  |
| --- | --- | --- |
| **MAE** | **MSE** | **RMSE** |
| 10.610264547336175 | 173.9739821298828 | 13.18991971658216 |

Tras el entrenamiento, tenemos el conjunto de arboles generados para las predicciones, dado resultado a un modelo de árbol, como vemos para el árbol 0.

Tree 0:

If (feature 128 <= 52.829998)

If (feature 140 <= 37.079998)

If (feature 1 <= 33.929998499999996)

If (feature 84 <= 40.8599985)

If (feature 0 <= 32.039998)

Predict: 37.9686697272727

Else (feature 0 > 32.039998)

Predict: 37.918320090909106

Else (feature 84 > 40.8599985)

Predict: 37.82894951048951

Else (feature 1 > 33.929998499999996)

Predict: 38.76419404195801

Else (feature 140 > 37.079998)

If (feature 133 <= 44.279998500000005)

Predict: 40.618320055944

Else (feature 133 > 44.279998500000005)

If (feature 93 <= 36.539999)

Predict: 39.7724456153846

Else (feature 93 > 36.539999)

Predict: 39.3079702097902

Else (feature 128 > 52.829998)

If (feature 49 <= 85.3199995)

If (feature 89 <= 41.9399985)

Predict: 67.2293675734266

Else (feature 89 > 41.9399985)

Predict: 68.222094965035

Else (feature 49 > 85.3199995)

If (feature 49 <= 88.379997)

If (feature 80 <= 65.33999650000001)

Predict: 71.5053115734265

Else (feature 80 > 65.33999650000001)

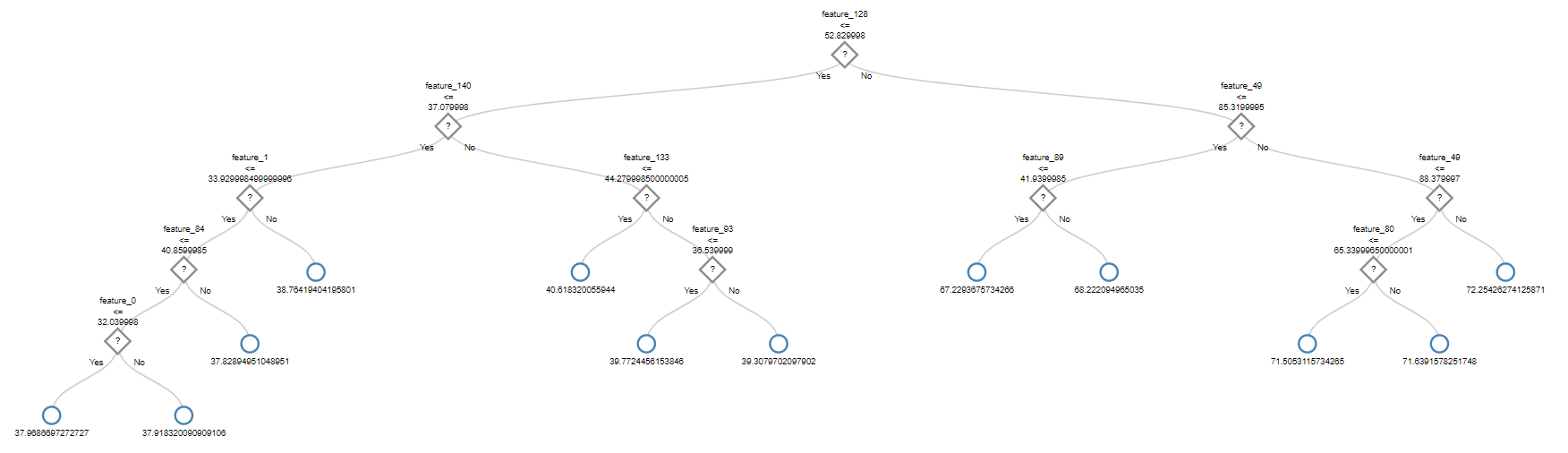
Predict: 71.6391578251748

Else (feature 49 > 88.379997)

Predict: 72.25426274125871

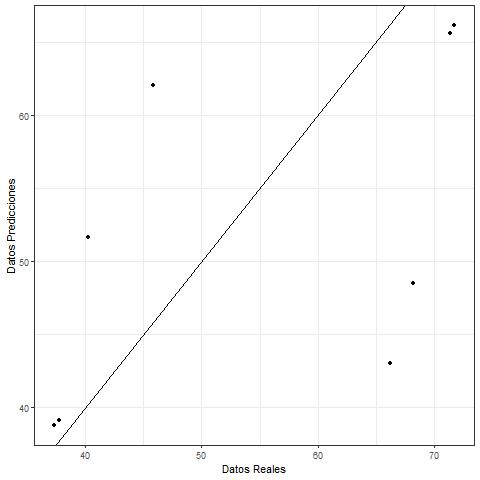
El modelo completo con todos los arboles se encuentra dentro de la carpeta ficheros, dentro del fichero de datos “2b-random\_forest.txt”

Claramente esta forma de ver el modelo es bastante abstracta, ya que la anidación de condiciones dificulta la visión del árbol, por lo que a partir de estos modelos se han creado los siguientes gráficos de árbol:



Los gráficos de árbol generados para cada uno de ellos se encuentran dentro de la carpeta “./Parte2/ejercicio3/Graficas/random\_forest”. Una vez entrenado el modelo, se procede a realizar las predicciones:

|  |  |
| --- | --- |
| **Real** | **Predicción** |
| 66.28153 | 43.00638 |
| 45.82699 | 62.04210 |
| 40.20797 | 51.67901 |
| 68.18978 | 48.52943 |
| 71.75705 | 66.16805 |
| 71.41090 | 65.60970 |
| 37.26335 | 38.74718 |
| 37.73034 | 39.11675 |



Viendo los resultados de los datos y los errores obtenidos, podemos determinar que en este modelo no podríamos confiar en absoluto, ya que los resultados son totalmente erróneos. Quizás, al igual que el modelo anterior, una mayor cantidad de datos de entrenamiento mejoraría los resultados.

# 4) Comparativa NNET – Random Forest

Una vez realizadas las predicciones con los dos modelos anteriores, procedemos a comprar los resultados.

|  |  |  |
| --- | --- | --- |
| **Real** | **NNET** | **Random Forest** |
| 66.28153 | 66.16260 | 43.00638 |
| 45.82699 | 46.52666 | 62.04210 |
| 40.20797 | 40.24702 | 51.67901 |
| 68.18978 | 67.87985 | 48.52943 |
| 71.75705 | 56.09095 | 66.16805 |
| 71.41090 | 71.10339 | 65.60970 |
| 37.26335 | 37.68678 | 38.74718 |
| 37.73034 | 50.36196 | 39.11675 |

Viendo los resultados y errores en el análisis de cada modelo, y los resultados finales en las predicciones, podemos asegurar que en este caso con estos datos el modelo de red neuronal funciona con una fiabilidad y precisión ampliamente mejor que el modelo random forest.

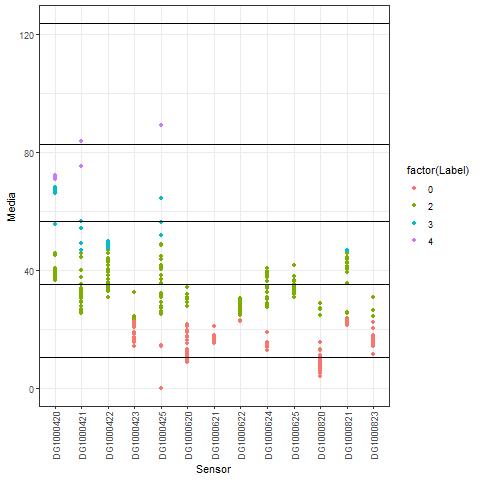
# 5) Clustering – Knn

Tras los análisis de predicción de datos, pasamos a la clasificación en clusteres de los sensores. En este caso, utilizamos todo el set completo de datos, sin distinción de sensor, ni necesidad de datos de entrenamiento y prueba, agrupándolos en 5 clusters.

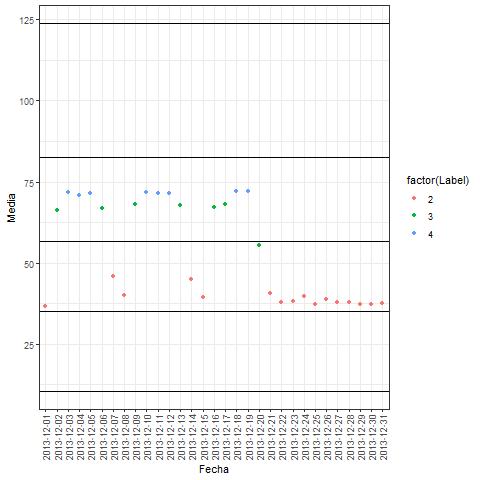
Ya que esta clasificación se ha realizado con el dataset de consumo completo, tenemos una amplia cantidad de dimensiones para el agrupamiento de las medidas en clusters, por lo que a la hora de representarse gráficamente se ha decidido hacerlo mediante dos dimensiones, que muestren de la forma mas clara posible como ha hecho la agrupación de todas las dimensiones.

El documento completo en el que aparecen los grupos para cada sensor y fecha se puede encontrar en la carpeta ficheros, dentro del fichero “2c-clusters.csv”.

Para poder representar las dimensiones múltiples se ha decidido agruparlas todas en la media de las medidas del día, al igual que para el valor de los centros. En esta primera grafica se muestran los grupos que se forman para los diferentes sensores según la media, para las que cada sensor tiene 31 puntos en diferentes valores para las medias, donde se puede aprecia que la agrupación se ha realizado según este valor de la media. Esta grafica ha sido montada con un extracto de 400 líneas

.

Se puede observar que aun realizandose las agrupaciones con multiples dimensiones, se sigue un patron de agrupación de las dimensiones relacionado con el valor de la media de todas las dimensiones. De esta misma forma, si aislamos un solo sensor a lo largo del tiempo obtenemos la siguiente representación.



# 6) Bibliografica

[1]<https://www.researchgate.net/publication/230766603_How_Many_Trees_in_a_Random_Forest>