21-2-2022

Trabajo fin de Máster

Predicción del consumo eléctrico mediante Machine Learning

Carlos García del Pozo
KSCHOOL
CALLE JOSÉ PICÓN, 31, MADRID

Máster en Data Science

Trabajo de fin de Máster

Predicción del consumo eléctrico mediante Machine Learning

Autor: Carlos García del Pozo

Febrero de 2022

Contenido

1. Introducción

1.1 Motivación

El constante desarrollo de las nuevas tecnologías con el fin de mejorar la calidad de vida de las personas tiene como consecuencia un aumento de la demanda de energía eléctrica y una búsqueda por encontrar la mejor fuente de obtención, por medio de energías baratas, limpias y renovables.

El primer paso para conseguir estos objetivos es intentar predecir las necesidades eléctricas para poder hacerlas frente en el futuro. Actualmente existe distintos debates entre estas fuentes de generación, tanto en la nuclear, como en las renovables. Es bien sabido, que todas fuentes tienen sus pros y sus contras, por lo que es necesario hacer un estudio previo bastante profundo para decantarnos por una de ella.

España ha optado por apostar fuertemente por las renovables, concretamente por la Solar, Eólica e hidráulica; pero en momentos de grandes picos de demandas o periodos meteorológicos desfavorables la red se alimenta con fuentes no renovables como ciclo combinado, cogeneración o el carbón. También es importante tener en cuenta que las renovables están fuertemente respaldadas por la energía nuclear, que produce de manera constante durante todo el año.

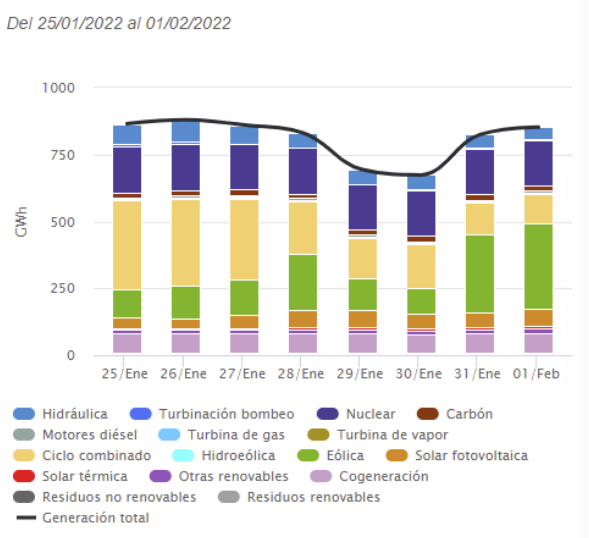


Ilustración 1.1. *Generación eléctrica de enero 2022. Fuente REE.*

1.1 Objetivo TFM

El objetivo de este TFM es averiguar cuáles son los factores de los que dependen el consumo eléctrico para poder predecir el consumo anual per cápita en KWh, aplicando distintas técnicas de Machine Learning.

El primer paso, será obtener información de cómo funciona el mercado eléctrico español y como se gestiona la demanda y la producción de energía. Con esta información seremos capaces de saber qué factores afectan, en mayor o menos medida, al consumo eléctrico.

Una vez identificados, tendremos que obtener datos de calidad para nutrir el modelo. Para ello, harán falta distintos procesos de recolección y tratamientos de datos, además de un análisis y estudio estadístico para terminar con las técnicas de Machine Learning.

Utilizaremos distintos modelos y compararemos los resultados obtenidos para seleccionar el que mejor se adapte a nuestras necesidades. Una vez decidido el modelo, lo optimizaremos para que nos proporcione la mejor predicción posible y guardaremos esos resultados.

Para terminar, expondremos los resultados obtenidos en un informe, comparándolos con los datos reales.

2. Consumo eléctrico en España

El consumo de energía eléctrica en España siguió en 2018 la tendencia de crecimiento del 2015. En concreto, España incrementó su consumo de energía eléctrica un 1,8% respecto al año anterior. Creció el consumo de petróleo, un 2,6%, y creció el consumo de energías renovables, un 1,7%. El consumo de carbón descendió en 2018 más de 17 puntos (lo que BP también asocia a las lluvias, que dejaron más agua en España el año pasado). Se redujo también la quema de gas un 0,8% y descendió el consumo de energía nuclear un 4,3%.

La generación eléctrica volvió a descender ligeramente en 2018 en un 0,2%. España produjo el año pasado con fuentes renovables el 38,5% de los kilovatios hora del mix eléctrico nacional. Tras ellas, casi empatadas, dos fuentes sucias de electricidad: el gas (20,8%) y la nuclear (20,2). Del carbón salió el 14% de la electricidad de España.

Con estos datos, debemos tener claro que tenemos que reducir la emisión de contaminantes producidos por las fuentes de generación de energía más contaminantes. Para ello, debemos hacer una predicción más o menos exacta y sustituir esas fuentes contaminantes por energías limpias.

2.1 Por industria y sector

El sector industrial ha sido el mayor consumidor de energía eléctrica en cualquier país del mundo.

Con el paso del tiempo, este consumo se ha reducido en el sector, debido a la mejora y modernización de los procesos industriales, y ha sido superado por otros sectores como el del transporte y servicios. Este cambio ha sido posible gracias al cambio de necesidades de la sociedad moderna, siendo el ejemplo más representativo el uso diario del tren (metros y cercanías).

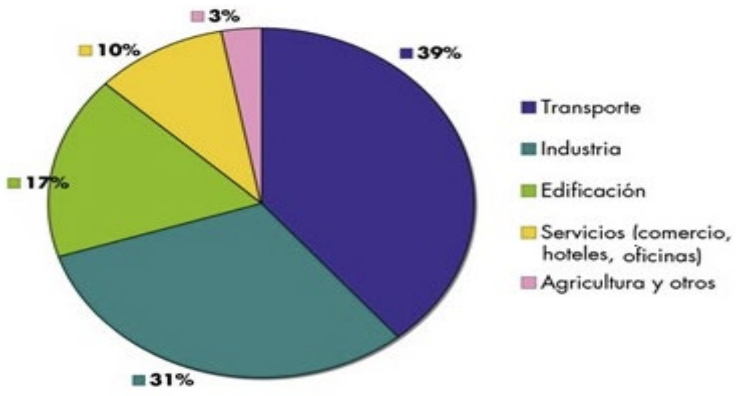


Ilustración 2. *Consumo final por sectores en España. Fuente Ministerio de Fomento.*

2.2 Por vivienda

Como hemos visto en el apartado anterior, el sector de la vivienda consume el 17% del total de la energía eléctrica, siendo las instalaciones térmicas las de mayor peso dentro de esta.

Podemos hacernos una idea de la importancia que tiene el aislamiento térmico dentro de este apartado. Por lo tanto, el consumo energético se dispara en los meses con temperaturas más extremas como son los de verano e invierno, y son más bajos en primera y otoño. El consumo energético también está relacionado con la zona geográfica a la que pertenezca la vivienda.

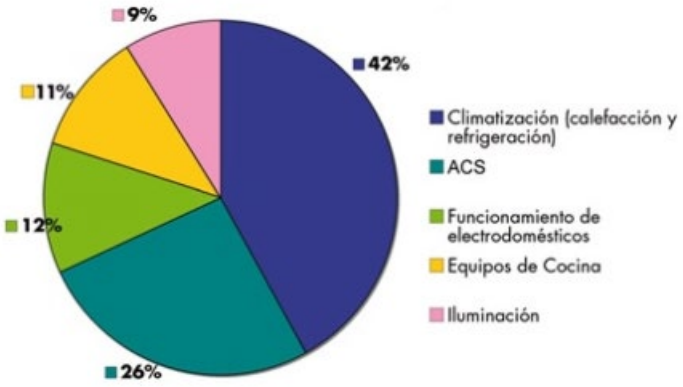


***Ilustración 3**: Distribución del consumo eléctrico en la vivienda. Fuente IDEA.*

3. Estado del Arte

Este trabajo se desarrolla dentro del ámbito de la ciencia de datos por lo que engloba los tres puntos de los que se sustenta esta rama. En primer lugar, deberemos tener conocimiento del ámbito en el que se desenvuelve el problema, el consumo energético. Por otro lado, deberemos tener una base de matemáticas y estadística para poder interpretar los datos y resultados, elaborados mediante modelos matemáticos. Por último, conocimientos de programación en Python y de Tableau, utilizado tanto para la representación de datos como de los resultados obtenidos.

3.1 Trabajo Previo

El primer paso para empezar cualquier proyecto es definir una metodología de trabajo. Para ello, lo primero que hice fue definir los pasos que iba a seguir. Para tener una visión global en todo momento decidí apostar el método Kanban utilizando el software de Atlassian Jira. Con el creé una hoja de ruta a seguir, de manera aproximada, para marcarme unos tiempos estimados reales e ir viendo como de rápido avanzada. Invertí algo de tiempo en adquirir los conocimientos necesarios para manejar este software. Me resultó especialmente útil para saber el tiempo que invertía en cada paso, y los problemas que me iban surgiendo, ya que al ser un proyecto que se iba a desarrollar en un periodo largo de tiempo, me parecía innecesario marcarme fechas concretas.

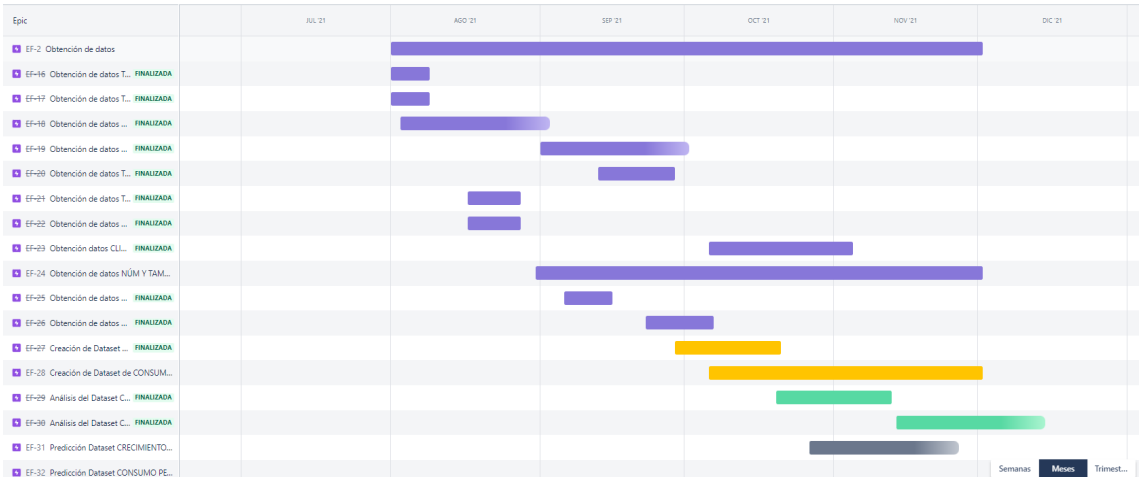


Ilustración 4: *Hoja de ruta con Jira. Fuente: Propia.*

Tras definir la metodología de trabajo, tuve que obtener todos los conocimientos necesarios para poder resolver el problema en cuestión. En un primer lugar, planteé otro proyecto, pero al darme cuenta de que con los datos que había en internet no iba a ser capaz de resolverlo opté por este otro proyecto. En este caso había mucha más información al respecto y pude obtener los datos necesarios para su desarrollo.

El primer planteamiento fue bastante complejo y enrevesado, ya que planteé la posibilidad de predecir dos variables distintas, la primera el consumo eléctrico per cápita y la segunda variable fue la población de cada provincia. Empecé por la segunda opción, teniendo terminado el dataset final que iba a utilizar, pero cambié de opinión al respecto, en gran medida al encontrar predicciones bastantes realistas del crecimiento de la población que podía utilizar. Como he citado anteriormente, finalmente no elegí ese proyecto.

Me gustó la idea de una parte de ese proyecto, la del consumo que tenemos cada habitante del país y de cómo varía dependiendo de sus necesidades. Encontré gran cantidad de dataset que podía utilizar, pero fue un auténtico quebradero de cabeza elegir cuáles podía utilizar y cuáles no. Tras analizar a fondo la cuestión, e informarme de los pasos a seguir pase al siguiente nivel.

El tipo de problema se resuelve con un modelo de regresión donde obtenemos como resultado la predicción del consumo per cápita de una provincia de España. Para ello utilizaremos distintos modelos y evaluaremos cuál es el que mejor se adapta a nuestras necesidades.

3.2 Modelos de predicción

En este apartado explicaremos, de manera breve, el funcionamiento de los algoritmos utilizados:

- **Regresión Lineal (LR):** Esta forma de análisis estima los coeficientes de la ecuación lineal, involucrando una o a más variables independientes que mejor predicen el valor de la variable dependiente. La regresión lineal se ajusta a una línea recta o a una superficie que minimiza las discrepancias entre los valores de salida previstos y reales.

- **K Nearest Neighbours (KNN):** El K-NN es un algoritmo de aprendizaje supervisado, es decir, que a partir de un juego de datos inicial su objetivo será el de clasificar correctamente todas las instancias nuevas. El juego de datos típico de este tipo de algoritmos está formado por varios atributos descriptivos y un solo atributo objetivo (también llamado clase).

- **Random Forest (RF):** El algoritmo de Random Forest (también conocido como Bosques Aleatorios) es ampliamente utilizado para la creación de modelos supervisados. Basado en una idea simple: combinar diferentes árboles de decisión. Permite obtener modelos con menor propensión al sobreajuste que un árbol de decisión.

- **Gradient Boosting Tree (GBT):** El GBT es una familia de algoritmos usados tanto en clasificación como en regresión basados en la combinación de modelos predictivos débiles (*weak learners*) -normalmente árboles de decisión- para crear un modelo predictivo fuerte. La generación de los árboles de decisión débiles se realiza de forma secuencial, creándose cada árbol de forma que corrija los errores del árbol anterior. Los aprendices suelen ser árboles "poco profundos" (*shallow trees*), de apenas uno, dos o tres niveles de profundidad, típicamente.

- **Stacking:** En stacking combinamos las predicciones de diferentes clasificadores o regresores, y las pasamos como features de un estimador final, el cual es el que genera la predicción de todo el conjunto, concretamente, entrenaremos una primera capa de modelos directamente sobre los datos.

Posteriormente, entrenaremos un segundo modelo sobre las predicciones de los de la primera capa. Es así como logramos reducir el “bias” o parcialidad de todo el conjunto.

3.3 Criterio de evaluación para aprendizaje automático

En este apartado hablaremos sobre la métrica utilizada para medir el rendimiento del modelo creado. Dependiendo del tipo de problema que queramos resolver, clasificación o regresión, existen unas métricas distintas. En mi caso, he decidido elegir la Raíz del Error Cuadrático Medio o RMSE por sus siglas en inglés. Existen otras muchas como el Error Medio Absoluto (EAM), Error Cuadrático Medio (MSE), etc.

Me he decantado por este criterio, por ser el más fácil de interpretar, en mi opinión. Puedes ver de manera rápida y clara que error está teniendo tu modelo con respecto el valor real.

La **Raíz del Error Cuadrático Medio** o **RMSE** (**Root Mean Squared Error**) es una medida de desempeño cuantitativa utilizada comúnmente para evaluar métodos de pronóstico de demanda. En este contexto RMSE consiste en la raíz cuadrada de la sumatoria de los errores cuadráticos. En comparación con la Error Medio Absoluto o EMA, RMSE amplifica y penaliza con mayor fuerza aquellos errores de mayor magnitud.

$$RMSE = \sqrt{1/N \sum_{i=1}^{N} (Y_{J-} \overline{Y_J})}$$

4. Desarrollo

En este capítulo recorreremos el proceso de búsqueda de las variables responsables de encontrar la solución al problema. Explicaremos cómo y dónde las hemos obtenido, el trabajo que hemos realizado para normalizar y limpiar los datos y construir el dataframe final del que se alimentará el modelo. Continuaremos con la visualización y explicación del dataframe final, preprocesado, y finalizaremos comentando los resultados de los distintos modelos.

4.1 Fuentes de datos

Quiero empezar este apartado diciendo que, sin duda, esta ha sido la parte más laboriosa del proyecto.

En la mayoría de los dataset he realizado los mismos pasos para preparar los datos. He cogido los valores de las provincias, normalizando sus nombres, mayúsculas y minúsculas, he cambiado puntos por comas, tipos de datos y eliminado algunos valores que no me servían. He seleccionado los años comunes a todos ellos, otro problema que he tenido, ha sido que no he tenido la misma cantidad de años en todos los datasets por lo que el número de filas se ha visto disminuido de manera drástica, teniendo datos completos de 19 años.

Estos datasets han sido preparados por separados y luego se han ido juntando hasta formar el dataframe final utilizado para las siguientes fases del proyecto.

Al principio, pensé que tener tan pocos datos podía ser un problema, pero decidí continuar con ellos y hacer una primera predicción rápida para ver el resultado que tendría.

4.1.1 Climatología

La información meteorológica proviene de 52 excel obtenidos del AEMET. No me ha hecho falta descargarme estos archivos de la página, ya que los tenía de un proyecto El conjunto de archivos se extiende a más de 400, ya que hay varias estaciones meteorológicas en cada provincia. Al principio, la selección de estas me resultó difícil, pero terminé por seleccionar aquellas que coincidían en tiempo con el resto de los datos recolectados. He unido los 52 excel creando un dataframe de los años 2000 hasta 2018 agrupados por años. Para ello, he calculado los datos medio de las variables temperatura en ºC y horas de sol, y he sumado los valores de precipitación en mm.

Las variables seleccionadas han sido la fecha, las provincias, las precipitaciones, horas de sol, temperatura mínima, media y máxima. He de decir, que cuando entreno el modelo, decido quitar la temperatura media por su correlación con las otras dos.

4.1.2 Población

Este dataset se ha obtenido de la página del INE, y posiblemente sea el más simple de todos. Hemos seleccionado todos los años posibles con su correspondiente valor de población total y provincia.

4.1.3 PIB per cápita

Este dato sale de un simple cálculo, dividir el PIB anual de cada provincia entre el número de habitantes de ese año. Para ello, descargamos de la página del INE el PIB a precios de mercado de cada provincia. Ese valor, será el que dividamos entre la población correspondiente a la provincia y al año. Así obtenemos el PIB por habitante en euros.

4.1.4 Número de viviendas

El contenido de este archivo es bastante simple, el número de viviendas por provincia y años. La dificultad residió en darle la forma deseada, ya que los ejes estaban cambiados.

4.1.5 Consumo de energía eléctrica per cápita

Este dato no pude encontrarlo, por lo que fue elaborado por mí. Para ello, encontré los valores de consumos totales de las provincias del año 2014 Y calculé la proporción de cada provincia, con respecto a la suma total. He supuesto que esta proporción no ha variado en el tiempo, cosa que no es real, aunque la proporción sea pequeña. Al comparar datos totales no hay tantas diferencias.

Los datos totales de energía eléctrica, en KWh, fueron cogidos a mano del periódico especializado en economía Expansión, en los datos macro. Con estos, y la proporción de cada provincia calculada anteriormente obtenemos los datos a nivel de provincia, que luego dividimos por el número de habitantes, como hicimos anteriormente con el PIB per cápita.

4.1.6 Precio del KWh

Estos datos, en euros, proceden de los datos macro del diario Expansión.

4.2 Análisis exploratorio

En el siguiente punto del proyecto procederemos a analizar la variable a predecir y su interacción con el resto.

La variable a predecir tiene una distribución beta con cola positiva, un tipo de distribución continua. Esta identificación ha sido posible gracias a la librería 'fitter', la cual permite trazar los resultados para verificar cuál es la distribución más probable y los mejores parámetros.

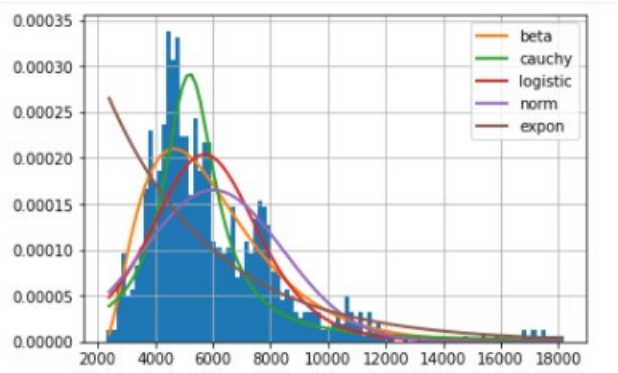


Ilustración 5: *Tipo de distribución de la variable a predecir. Fuente: Propia*

En las gráficas de abajo podemos comprobar la baja correlación existente entre la variable a predecir y el resto. Las más correlacionadas con ella son el 'PIB per cápita' y las variables meteorológicas 'TMIN', 'TMEDIA', 'TMAX' y 'SOL'. Las variables temperaturas estas muy correlacionadas entre ellas, pero la 'TMAX' y 'TMIN' me parecen importantes como para prescindir de una de ellas. La que sí eliminaremos es la 'TMEDIA', innecesaria teniendo las otras dos.

Me ha sorprendido que la variable 'Num de viviendas' esté tan poco correlacionada.

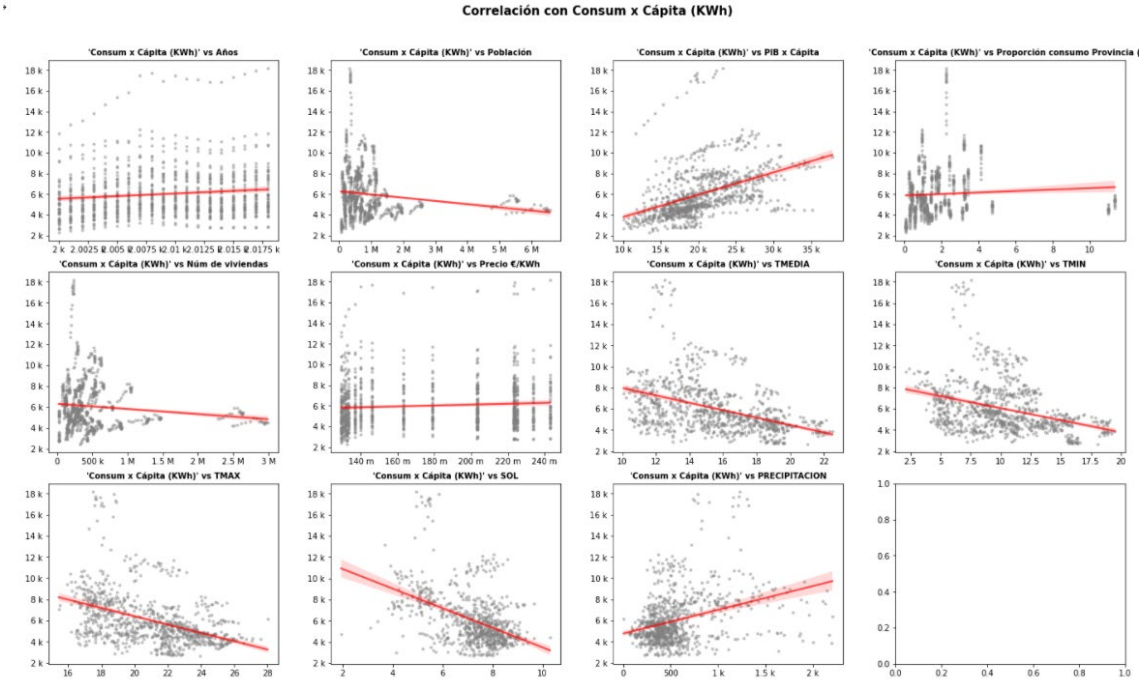


Ilustración 6: *Correlación con variable a predecir. Fuente: Propia*

Ilustración 7: *Correlación entre variables. Fuente: Propia*

El análisis de datos anormales lo he hecho con el software Tableau. He hecho un dashboard con toda la información relevante y he ido examinando variable por variable. Lo he hecho de esta manera debido a la diferencia de valores existentes entre las diferentes provincias. Al hacerlo en Python, me salían 'outlayers' en todas las variables y era mucho más trabajoso el ir comprobando dato por dato. Al final, este análisis exhaustivo me ha dado bastante buen resultado y al analizar los datos, descubrí que estos 'outlayers' eran valores totalmente normales.

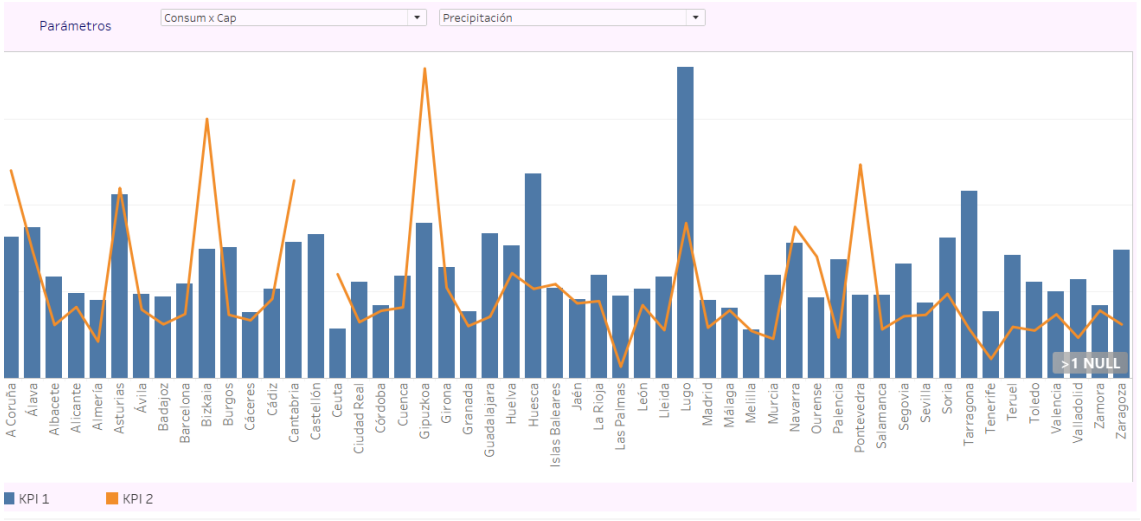


Ilustración 8: *Representación de variables. Fuente: Propia.*

Por comentar un dato curioso de esta ilustración, quizá el consumo per cápita más llamativo sea el de la provincia de Lugo, con un gran núcleo industrial en la capital y un bajo número de habitantes en toda la provincia. Esto hace que el consumo per cápita se dispare, hasta ser la más alta de España.

En el informe de Tableau, en la pestaña de datos se podrá visualizar todos los datos que contiene el dataframe creado, y en la última pestaña, 'My Repor', se podrá construir un informe a medida para comparar valores.

4.3 Machine Learning

En este apartado trataremos de buscar el modelo que mejor resuelva el problema propuesto.

En el punto 3.2 he explicado los algoritmos utilizadosahora explicaremos los pasos previos que hemos dado para preparar los datos. En este punto he probado diferentes métodos, y tras varias pruebas, hemos decidido dejar lo que mejor resultado ha dado.

4.3.1 Preprocesamiento de datos

Voy a empezar hablando de como he dividido el dataframe. Al principio decidí dividir el conjunto con el habitual 80/20 para el Train/Test, pero más adelante me di cuenta de que al evaluar el modelo final, no tenía un conjunto de datos que no hubiese visto el modelo, por lo que decidí guardar los datos del año 2018 para evaluar, de manera real, el algoritmo y para representar estos resultados en el informe final de Tableau. Con el problema de los datos resuelto, procedí a dividir el conjunto de datos como he citado anteriormente.

El siguiente paso era saber que hacer con los datos nulos que tenía. Estos estaban ubicados en las variables meteorológicas debido al mantenimiento de las estaciones meteorológicas. En un primer momento utilicé 'Iterative Imputer', un método de la librería 'Sklearn' que lo que hace es rellenar los valores faltantes utilizando distintas estrategias, como la media, el valor más frecuente, mediana… Un método muy interesante, pero que en mi caso no valía, ya que mis datos pertenecen a distintas provincias, con valores que tienen poca relación las unas con las otras. Por ejemplo, las precipitaciones anuales entre provincias distan mucho unas de otras. Si me falta la variable precipitación de Madrid del año 2012, no puedo poner la media de precipitaciones de las demás provincias, porque estaría falseando los datos. En todo caso podría hacerlo con los valores de otros años de esa misma comunidad, pero en una primera prueba, al eliminar esos valores faltantes, los resultados obtenidos fueron bastante buenos, por lo que decidí dejarlo así.

La siguiente transformación de la que hablaremos, es 'Standar Scaler', también de la biblioteca 'Sklearn', cuya función es normalizar los valores de las distintas variables. Muchos de los modelos no hacen buenas predicciones si los valores no están estandarizados. Esta estandarización de las variables numéricas las metí dentro de una 'pipe', ya que como he dicho anteriormente, en un primer momento lo combiné con 'Iterative Imputer'. En las pruebas que hice, me salieron mejores resultados si introducía 'PolynomialFeatures' de grado 2 que 'passthrough'. Las características polinómicas son aquellas características creadas elevando las características existentes a un exponente. El '*grado*' del polinomio se utiliza para controlar el número de características añadidas, por ejemplo, un grado de 2 añadirá una nueva variable para cada variable de entrada.

La población es la única variable categórica en el dataframe final. Opté por eliminar esta columna porque no me parecía correcto trabajar con 52 etiquetas. Tanto 'One hot Encoder' como el 'Label Encoder' iba a aumentar de manera innecesaria el número de variables.

En la búsqueda de los hiperparámetros me he decantado por un 'Random Grid Search', donde evaluamos los valores de manera aleatoria, y siempre dentro de un límite establecido por nosotros.

La estrategia de validación utilizada en la 'RepeatedKFold', la cual evalúa el modelo, utilizando distintos subconjuntos de datos.

4.3.2 Comparación de modelos

4.3.2.1 Regresión Lineal (LR)

Para este modelo he utilizado una regularización 'Ridge', que es la que mejor me ha funcionado, creo que debido a la correlación existente entre las variables meteorológicas. Tengo que reconocer que la diferencia con respecto a las oras dos, Lasso y Elasticnet, era pequeña. En este modelo, los mejores resultados se dan con una 'alpha' muy pequeña.

```
El error (rmse) de test es: 1628.7614092733302
```

Ilustración 9: *Error de la Regresión Lineal. Fuente: Propia*

4.3.2.2 K-Nearest Neighbor (KNN)

Para este modelo, a medida que aumentamos los 'neighbors' los resultados son peores. Por eso, el GridSearch nos recomienda 'n_neighbors' = 2.

En la siguiente gráfica podremos ver la comparación entre distintos valores.

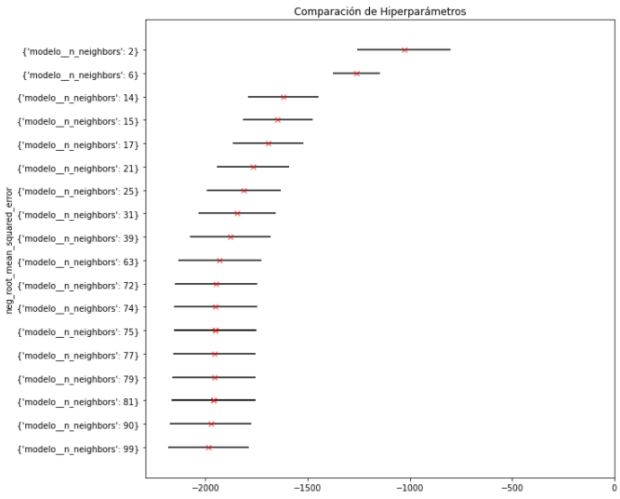


Ilustración 10: *Comparación de hiperparámetros. Fuente: Propia*

Como podemos observar, el resultado de este modelo es mejor que el anterior.

```
El error (rmse) de test es: 1628.7614092733302
```

Ilustración 11: *Resultado KNN. Fuente : Propia*

4.3.2.3 Random Forest (RF)

Para este modelo, tendremos más hiperparámetros que configurar. En este caso tenemos 'n_stimators', 'max_features' y 'max_depth'. Como siempre, utilizaremos el GridSearch definido anteriormente. Aquí los resultados son más ajustados, siendo los mejores 'n_stimators' = 1000, 'max_features' = 5 y 'max_depth' = None.

	param_modelo__n_estimators	param_modelo__max_features	param_modelo__max_depth	mean_test_score	std_test_score	mean_train_score	std_train_score
6	1000	5	None	-759.448139	160.232753	-295.960752	9.964705
16	100	5	20	-770.749338	147.085956	-306.995756	14.795048
7	1000	3	20	-771.442531	152.161637	-296.929808	7.388823

Ilustración 12: *Resultados GridSearch. Fuente: Propia*

La mejoría de este modelo con respecto los anteriores es bastante notable, reduciendo su error en más de la mitad del primer modelo.

```
El error (rmse) de test es: 621.9161722203767
```

Ilustración 13: *Resultado de RF. Fuente: Propia*

4.3.2.4 Gradient Boosting Trees (GBT)

Es el modelo más complejo utilizado y, junto con en random forest, el que más tiempo de ejecución conlleva. El nivel de personalización de los hiperparámetros es más elevado que los anteriores. Como en el caso anterior, tenemos 'n_stimators', 'max_features' y 'max_depth', pero en este caso añadimos uno más, 'subsample'. Los mejores valores serían 'n_stimators' = 2000, 'max_features' = 7, 'max_depth' = 3 y 'subsample' = 0,5.

En este caso, sí habría una diferencia considerable entre los parámetros que están en primer lugal y los siguientes, como vemos en la imagen.

	param_modelo__subsample	param_modelo__n_estimators	param_modelo__max_features	param_modelo__max_depth	mean_test_score	std_test_score	mean_train_score	std_train_score
2	0.5	2000	7	3	-474.504171	123.436434	-1.688143	0.167800
12	0.5	1000	3	5	-606.880993	239.066121	-0.217226	0.019578
14	1	100	3	5	-611.346674	136.988567	-71.058866	6.185370

***Ilustración 14**: Grid Search GBT. Fuente: Propia.*

El resultado de este modelo es increíble, es verdad que podemos apreciar mucha diferencia entre el conjunto de train y el de test, pero aún así, es un resultado muy muy bueno.

```
El error (rmse) de test es: 395.7162606712849
```

Ilustración 15: Resultado GBT. Fuente: Propia

4.3.2.5 Stacking

En este último punto, evaluaremos de manera conjunta distintos regresores para ver si existe alguna mejora. En este caso he hecho dos, combinando en primer lugar, una Regresión Lineal con un Random Forest; y en segundo lugar un Random forest con Gradient Boosting Trees.

Los parámetros utilizados son los resultados del GridSearch de los apartados anteriores.

El resultado del primer modelo es:

```
El error (rmse) de test es: 819.2495783203962
```

***Ilustración 16**: Resultado Stacking 1. Fuente: Propia*

El resultado es peor de lo esperado, ya que no mejora el Random Forest.

El resultado del segundo modelo es:

```
El error (rmse) de test es: 474.6101953494662
```

***Ilustración 17:** Resultado Stacking 2. Fuente: Propia.*

En este caso el bastante mejor, pero sigue sin superar el del Gradient Boosting Trees.

Por lo tanto, la comparación entre los modelos creados sería el siguiente:

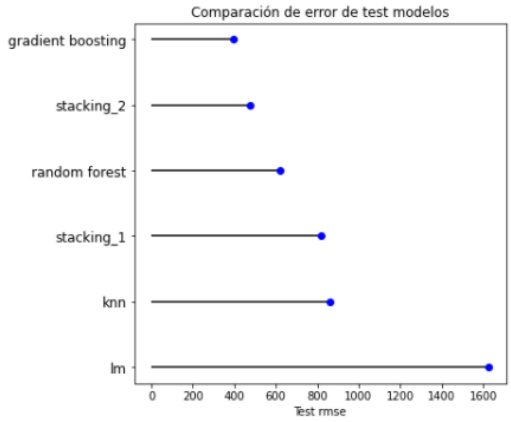


Ilustración 18: *Comparación de modelos. Fuente: Propia.*

El Gradient Boosting Trees sería el modelo ganador.

4.3 Optimización del modelo seleccionado

En este punto, queremos buscar la configuración óptima para que prediga el mejor valor posible, sin sufrir 'overfiting'. Los modelos tipos 'boosting' trabajan de manera secuencial con varios modelos más sencillos, aprendiendo de los errores del modelo anterior.

Al ser un proyecto que no es computacionalmente muy costoso, esta búsqueda la haremos por medio de una validación cruzada. El número de arboles será elevado y por medio de la parada temprana evitaremos que el modelo haga 'overfiting' y, además que se pueda detener antes.

Tras el GridSearch, los valores óptimos y el error final son:

```
Mejores hiperparámetros encontrados (cv)
########################################
{'learning_rate': 0.1, 'max_depth': 3, 'max_features': 3, 'subsample': 1}
```

Ilustración 19*: Mejores hiperparámetros. Fuente: Propia.*

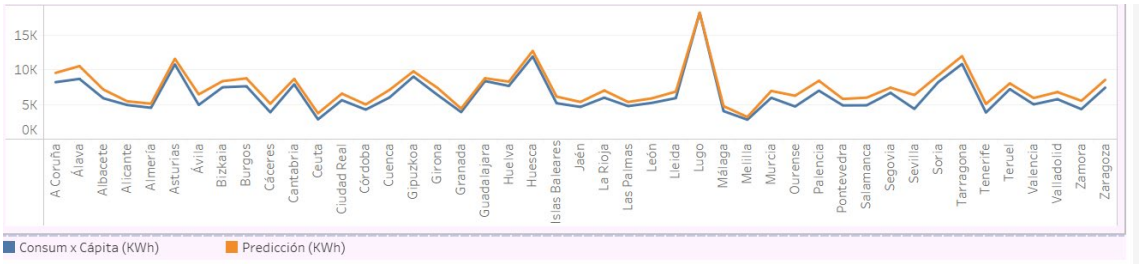
```
El error (rmse) de test del modelo final es: 462.10342575895936
```

Ilustración 20: *Error del modelo final. Fuente: Propia*

5. Resultados

Tras guardar el modelo final, lo probamos con los datos del 2018 que guardamos al principio. Estos datos no han sido visto vistos por el modelo en ningún momento, por lo tanto, podremos ver de una manera real, el comportamiento de nuestro modelo.

Vamos a representar una comparativa entre los datos reales y los predichos:



6. Conclusiones y trabajo futuro

6.1 Conclusiones

Con este proyecto, hemos podido desarrollar con claridad los distintos puntos de los que se compone la ciencia de datos.

Hemos empezado con una introducción al ámbito del proyecto, recabando información para tener una idea de como abordar el problema. Una vez que teníamos el conocimiento necesario para poder idear una solución hemos tenido que recabar los datos que íbamos a necesitar de las distintas bases de datos públicas que hay en internet.

Esos datos lo hemos limpiado y procesado, y los hemos unido en un dataframe.

Tras su exploración y análisis, procedemos a preprocesar los datos y separarlos en conjuntos de train/test. Haremos otra división más cuya función será la evaluación del modelo final.

Compararemos los distintos modelos de manera simple y cogeremos el que más se adapte a nuestras necesidades.

El modelo seleccionado, Gradient Boosting Trees, es optimizado y se le pasa el conjunto de datos virgen.

El RMSE del modelo con los datos que hemos separado al principio, los del año 2018, es de 506.50 KWh, un buen resultado para un modelo que, en un principio, tiene dificultades para extrapolar fuera del rango de entrenamiento.

6.2 Trabajo futuro

En un futuro, me gustaría aumentar la cantidad de datos, añadiendo los posteriores a la pandemia e introduciendo las regiones de otros países.

Por otro lado, me gustaría enfocarme más en las tecnologías limpias de generación de energía eléctrica. Tras lograr predecir con tanta exactitud el consumo per cápita, y conocer la tendencia de crecimiento de la población en España, me gustaría analizar la transición ecológica que se plantea Europa en un futuro, y ver como de viable es que suceda en un futuro próximo.

7. Referencias

1. “REE (Red Eléctrica de España).” https://www.ree.es/es [Online; acceso 01-septiembre-2021].

2. “Energía y Sociedad.” https://www.energiaysociedad.es/manual-de-la-energia/5-3-contribucion-del-sector-electrico-y-gasista-a-la-sociedad/ [Online; acceso 01-septiembre-2021].

3. “Fundación BBVA” https://www.fbbva.es/wp-content/uploads/2017/05/dat/DE_2006_estadisticas_historicas.pdf [Online; acceso 05-septiembre-2021].

4. “Ministerio de transportes, Movilidad y Agenda Urbana (mitma).” https://www.mitma.gob.es/recursos_mfom/paginabasica/recursos/201804_estudio_distribucion_consumo_energetico_res.pdf [Online; acceso 10-septiembre-2021].

5. “Ministerio de transportes, Movilidad y Agenda Urbana (mitma).” https://energia.gob.es/balances/Balances/LibrosEnergia/Libro-Energia-2018.pdf

6. “Organismo Internacional de la Energía Atómica” https://www.iaea.org/es/publications/7794/modelo-para-el-analisis-de-la-demanda-de-energia-maed-2 [Online; acceso 11-septiembre-2021].

7. “Kanbanize.” https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban [Online; acceso 12-julio-2021].

8. “Atlassian.” https://www.atlassian.com/es/software/jira/guides/getting-started/basics#step-3-set-up-your-columns [Online; acceso 12-julio-2021].

9. “Uniwebsidad.” https://uniwebsidad.com/libros/pro-git [Online; acceso 12-julio-2021].

10. https://sinchi.org.co/files/Base%20de%20Datos%20Inirida/PDF/17_Consumo%20de%20energia%20electrica%20por%20habitante.pdf [Online; acceso 20-septiembre-2021].

11. “Comisión Nacional de la Energía.” https://www.cne.cl/wp-content/uploads/2015/07/Informe-Final-TOMO-I.pdf [Online; acceso 20-septiembre-2021].

12. “REE (Red Eléctrica de España).” https://www.ree.es/es/datos/publicaciones/informe-anual-sistema/informe-del-sistema-electrico-espanol-2020 [Online; acceso 20-septiembre-2021].

13. “Instituto para la Diversificación y Ahorro de la Energía (IDAE).”https://www.idae.es/uploads/documentos/documentos_Documentacion_Basica_Residencial_Unido_c93da537.pdf [Online; acceso 20-septiembre-2021].

14. Caballero Roldán, Rafael, Martín Martín, Enrique, Riesco Rodríguez, Adrián, “BIG DATA con PYTHON. Recolección, almacenamiento y proceso”. RC libros, 2018

15. Geron, Aurelien, “Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc, USA; 2nd edition

16. Hebert, Jones, “Analítica de datos”, 2019.

17. “IBM.” https://www.ibm.com/mx-es/analytics/learn/linear-regression [Online; acceso 25-septiembre-2021].

18. “Merkleinc.” https://www.merkleinc.com/es/es/blog/algoritmo-knn-modelado-datos [Online; acceso 30-septiembre-2021].

19. “Analyticslane.” https://www.analyticslane.com/2019/05/20/random-forest/ [Online; acceso 30-septiembre-2021].

20. “Interactivechaos.” https://interactivechaos.com/es/manual/tutorial-de-machine-learning/gradient-boosting [Online; acceso 30-septiembre-2021].

21. “Datasmarts.” https://datasmarts.net/es/que-es-stacking/ [Online; acceso 30-septiembre-2021].

22. “Cienciadedatos.” https://www.cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn.html [Online; acceso 30-septiembre-2021].

    https://www.cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn.html

23. “Instituto Nacional y Estadística (INE).” https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736177012&menu=resultados&idp=1254734710990 [Online; acceso 29-septiembre-2021].

24. “Instituto Nacional y Estadística (INE).” https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736167628&menu=resultados&idp=1254735576581 [Online; acceso 22-octubre-2021].

25. “Instituto Nacional y Estadística (INE).” https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176992&menu=ultiDatos&idp=1254735572981 [Online; acceso 17-octubre-2021].

26. “Expansión.” https://datosmacro.expansion.com/energia-y-medio-ambiente/electricidad-consumo/espana [Online; acceso 22-octubre-2021].

27. “Expansión.” https://datosmacro.expansion.com/energia-y-medio-ambiente/electricidad-consumo [Online; acceso 22-octubre-2021].

28. “Tableau” Tutorial: empezar a usar Tableau Desktop - Tableau [Online; acceso 30-diciembre-2021].