MÁSTER UNIVERSITARIO EN INGENIERÍA DE SISTEMAS Y DE CONTROL

Trabajo de Fin de Máster

SEGMENTACIÓN DE CLIENTES MEDIANTE TÉCNICAS DE CLASIFICACIÓN PARA LA MEJORA DE PRECISIÓN COMO PARTE DEL DESARROLLO DE NUEVOS PRODUCTOS

Universidad Nacional de Educación a Distancia (UNED) y REPSOL

Curso 2021/2022

Alumno: José Manuel Palomar Andrea

Director UNED: Dr. D. Sebastián Dormido Canto

Director Repsol: Dr. D. Ricardo Enríquez Miranda

Agradecimientos

Quiero agradecer a Juan Diego Bernal, a Javier Juárez y a Clara Rey, miembros del equipo de Repsol Technology Lab, la oportunidad y las facilidades que me dieron para la realización de este trabajo en el marco de los proyectos de desarrollo que se llevaron a cabo en Repsol Technology Lab durante 2021-2022.

Agradezco también a Ricardo Enríquez, a Vasyl Gnyedykh, a David Casasnovas y a David Aller, así como a todo el equipo de matemáticas avanzadas, miembros de Repsol Technology Lab, por fomentar un ambiente de intercambio continuo de conocimiento que me ha permitido aprender mucho más en los temas aquí tratados.

También quiero agradecer muy especialmente al Dr. D. Sebastián Dormido y al Dr. D. José Sanchez Moreno, por el soporte proporcionado todo este tiempo y, ante todo, por la flexibilidad que se me ha concedido para poder terminar en tiempo y forma este trabajo, dadas las dificultades que han surgido, primero en el contexto de la pandemia de la COVID-19, y después, debido a circunstancias personales.

Resumen

Este trabajo pretende desarrollar una aplicación que permita obtener una clasificación de clientes a partir de datos de consumo de sus hogares utilizando métodos de aprendizaje automático, con el objetivo de mejorar la precisión del desempeño de algunos de los productos de software que desarrolla Repsol, en particular de uno de ellos, al que se denomina en este trabajo Desagregador y cuya función principal es proveer a los clientes de una desagregación de la energía total consumida en sus hogares en los principales tipos de consumo, tomando como únicos datos de entrada el consumo horario total del hogar del cliente.

This work aims to develop an application that allows to obtain a classification of customers from their household consumption data by using machine learning methods, with the objective of improving the accuracy in the performance of some of the Repsol's software product developments, in particular one of them, which is called in this work Disaggregator and whose main function is to provide customers with a disaggregation of their total household consumption into the main types of consumption, taking the total hourly household consumption as the only input data.

Nota: todos los términos se intentaron traducir, pero en aquellos casos en los que la traducción pudiera dar lugar a confusión se ha preferido mantener el término anglosajón.

Índice

Índice 4
1. Introducción 9
1.1. Motivación del trabajo 9
1.2. Objetivo 10
1.3. Definición del problema 11
1.4. Datos utilizados 13
1.5. Organización de la memoria 15
2. Estado del arte y fundamento teórico 16
2.1. Consideraciones acerca de los datos, la precisión y el procedimiento 17
2.2. Definición y selección de características 18
2.3. Métodos de aprendizaje automático 21
2.4. Medidas de precisión 30
3. Metodología de desarrollo del trabajo 31
3.1. Entorno de desarrollo utilizado 32
3.2. Preprocesado de datos 33
3.3. Metodología de selección de características 36
3.4. Exploración visual de los datos 37
3.5. Definición de características 39
3.6. Filtrado de características 40
3.7. Diseño de experimentos para la configuración definitiva del *dataframe* 41
3.8. Diseño de experimentos para selección del algoritmo de clasificación 44
4. Resultados 45
4.1. Experimentos para la configuración definitiva del *dataframe* 45
4.2. Análisis de importancia y redundancia de características. 62
4.3. Experimentos para selección del algoritmo de clasificación 65
5. Conclusiones y líneas futuras 80
6. Bibliografía 83
Anexo: Estructura del código 86

Índice de Figuras

Figura 1. Ejemplo de resultados del Desagregador de Repsol obtenido en su fase de desarrollo. 10
Figura 2. Consumos promedio de hogares según informe del IDAE. 12
Figura 3. Curva de carga asociada a Lavadoras y lavavajillas según informes del IDAE. 12
Figura 4. Consumos de una lavadora durante los 20 primeros minutos. Fuente: Jacobs, Maun, 2019. 13
Figura 5. Esquema de las respuestas tipo para un usuario. 15
Figura 6. Ecuaciones del filtro mRMR. 21
Figura 7. Taxonomía de métodos de aprendizaje automático. Fuente: Muhamedyev, 2015. 21
Figura 8. Ejemplo de clasificación lineal (a) y no lineal (b) en 2 dimensiones. Fuente: Muhamedyev, 2015. 22
Figura 9. Algunas distancias empleadas en los métodos instanciados, como el KNN. Fuente: Kotsiantis, 2007. 23
Figura 10. Esquema de un algoritmo de árbol de decisión. Fuente: Han, 2012 24
Figura 11. Ejemplo de cálculo de método combinado, con votaciones a cada uno de los resultados de los árboles. Fuente: Chen, 2016. 25
Figura 12. Estructura de procesamiento de una neurona biológica y su analogía con el Perceptrón de una red neuronal artificial. Fuente: Basheer, 2000. 27
Figura 13. Estructura de una red neuronal, con la capa de entrada (input), una capa oculta (hidden) y la capa de salida (output). Fuente: Han, 2012. 28
Figura 14. Dendograma de un algoritmo jerárquico aglomerativo. Fuente: UNED. 30
Figura 15. Diagrama esquemático del trabajo realizado. 31
Figura 16. Repositorio de notebooks utilizado durante el análisis. 33
Figura 17. Carpetas de datos y métodos utilizadas. 33
Figura 18. Frecuencia de respuestas a la franja de uso de cada electrodoméstico en el cuestionario. 34
Figura 19. Frecuencia de respuestas al número de encendidos semanales de cada electrodoméstico en el cuestionario. 34
Figura 20. Ilustración de la validación cruzada estratificada. 36
Figura 21. Ejemplo de consumo de 6 usuarios durante las 7 franjas que marca el cuestionario, y recuadrado, horario preferente de consumo de lavadora. 37
Figura 22. Franjas de consumo para las primeras 5 semanas del mes 5 (mayo), para el Usuario Objetivo, que representan 1)Suma, 2)mean, 3)STD y 4)max. 43
Figura 23. Matriz de confusión para predicción de franjas horarias al reducir el número de franjas a 4 (experimento 8). 57
Figura 24. Franjas tarifarias generales que marcan el coste de la energía según la hora del día. 58
Figura 25. Resumen de características utilizadas en los experimentos para el mes de mayo. 61
Figura 26. Importancia relativa de las características seleccionadas para predicción de la franja horaria de uso de electrodomésticos. 62

Figura 27. Importancia relativa de las características seleccionadas para predicción del número de encendidos semanales de electrodomésticos. 63
Figura 28. Correlación de Pearson para algunas de las características del análisis. 64
Figura 29. Análisis de multicolinearidad a partir del valor VIF. 64
Figura 30. Esquema de la estructura del código. 86
Figura 31. Fragmento del módulo de importación de métodos. 86
Figura 32. Fragmento del módulo methods.py donde se programaron los algoritmos. 87
Figura 33. Fragmento de código correspondiente a la lectura de datos de consumo y del cuestionario. 87
Figura 34. Fragmento de código de la función preprocessor() que sirvió para adecuar los datos a la forma de trabajo del método de clasificación. 88
Figura 35. Fragmento de código del método de clasificación 89

Índice de Tablas

Tabla 1. Extracto de la tabla de consumos horarios por día. 14
Tabla 2. Extracto de las respuestas del cuestionario para algunos de los clientes. 15
Tabla 3. Renta promedio obtenida a partir del código postal, según datos disponibles en el INE. 38
Tabla 4. Nombre de características asignado a cada franja. 39
Tabla 5. Ejemplo de características asociadas a picos de consumo. 40
Tabla 6. Característica asociada a renta. 40
Tabla 7. Ejemplo de características antes y después del filtrado con mRMR. 41
Tabla 8. Precisión en la predicción de la franja de uso según el número de características clave seleccionadas por el filtro en lavadora y lavavajillas (experimento1). 46
Tabla 9. Precisión en la predicción del número de encendidos según el número de características clave seleccionadas por el filtro en lavadora y lavavajillas (experimento1). 47
Tabla 10. Precisión en la predicción de la franja de uso de lavadora según análisis en una base semanal o mensual (experimento 2). 48
Tabla 11. Precisión en la predicción del número de encendidos de lavadora según análisis en una base semanal o mensual (experimento 2). 49
Tabla 12. Ejemplo de característica adicional de tipo indicador para el Usuario Objetivo. 50
Tabla 13. Precisión en la predicción de la franja de uso para 30 y 37 características clave incluyendo una característica adicional de tipo indicador (experimento 3). 50
Tabla 14. Precisión en la predicción del número de encendidos para 30 y 37 características clave incluyendo una característica adicional de tipo indicador (experimento 3). 51
Tabla 15. Precisión en la predicción de la franja de uso con 2 características adicionales de tipo indicador de presencia en hogar (experimento 4). 52
Tabla 16. Precisión en la precisión del número de encendidos con 2 características adicionales de tipo indicador de presencia en hogar (experimento 4). 52
Tabla 17. Precisión en la predicción de la franja de uso con una característica adicional de indicador de consumo en fin de semana (experimento 5). 53
Tabla 18. Precisión en la predicción del número de encendidos semanales con una característica adicional de indicador de consumo en fin de semana (experimento 5). 53
Tabla 19. Precisión en la predicción de la franja de uso sin considerar los horarios de la franja 0 o 'no tengo un horario fijo' (experimento 6). 54
Tabla 20. Precisión en la predicción de la franja del número de encendidos sin considerar los horarios de la franja 0 o 'no tengo un horario fijo' (experimento 6). 54
Tabla 21. Precisión en la predicción de la franja de uso considerando una variación en la interpretación de las franjas (experimento 7). 55
Tabla 22. Precisión en la predicción del número de encendidos considerando una variación en la interpretación de las franjas (experimento 7). 55
Tabla 23. Precisión en la predicción de la franja de uso al reducir el número de franjas a 4 (experimento 8). 56

Tabla 24. Precisión en la predicción de la franja de uso al reducir el número de franjas a 3 (experimento 8). 57
Tabla 25. Precisión en la predicción del número de encendidos al reducir el número de franjas a 4 (experimento 8). 58
Tabla 26. Costes de energía para cada hora de 2021 obtenidos de OMIE. 59
Tabla 27. Precisión de la predicción de la franja de uso al considerar el coste horario (experimento 9). 59
Tabla 28. Precisión de la predicción del número de encendidos al considerar el coste horario (experimento 9). 59
Tabla 29. Resumen de resultados de los experimentos para la predicción de franja horaria. 60
Tabla 30. Resumen de resultados de los experimentos para la predicción del número de encendidos semanales. 60
Tabla 31. Resultados de predicción de franja horaria en base temporal anual para la lavadora. 61
Tabla 32. Características eliminadas y características finales para el dataframe. 65
Tabla 33. Precisión obtenida con XGBoost para distintas funciones de coste. 66
Tabla 34. Precisión obtenida con XGBoost según la cantidad de estimadores (árboles). 67
Tabla 35. Resultados de precisión con XGBoost en función de la función objetivo. 68
Tabla 36. Precisión obtenida con el clasificador Gradient Boosting para dos criterios de error en la partición. 69
Tabla 37. Precisión obtenida con el clasificador Gradient Boosting según el número de árboles empleados. 70
Tabla 38. Precisión obtenida con Random Forest según la medida de bondad en la partición. 71
Tabla 39. Resultados de precisión con Random Forest según el número de árboles utilizados. 72
Tabla 40. Precisión obtenida con Random Forest según aleatoriedad indicada. 73
Tabla 41. Precisión obtenida con Support Vector Classification según variación de la función kernel. 74
Tabla 42. Precisión obtenida con Support Vector Classification según variación del método de cálculo multiclase. 75
Tabla 43. Precisión obtenida con K- Nearest Neighbor según variación del algoritmo. 76
Tabla 44. Precisión obtenida con K- Nearest Neighbor según variación del número de vecinos originales. 76
Tabla 45. Precisión obtenida con una Red Neuronal Artificial según variación del número de capas y número de épocas. 77
Tabla 46. Resumen de los resultados de precisión en la predicción de franjas para todos los algoritmos empleados en la segunda tanda de experimentos. 78
Tabla 47. Resumen de los resultados de precisión en la predicción de encendidos para todos los algoritmos empleados en la segunda tanda de experimentos. 79

1. Introducción

En esta introducción se detalla la motivación del trabajo, los objetivos de su realización, y se describe el problema concreto a resolver.

También se explica el conjunto de datos del que se partió, así como la organización de la memoria.

1.1. Motivación del trabajo

Nos encontramos dentro de la segunda gran transición energética, también denominada *Long-term Energy Transition* o LTE [Doh, 2021], que se caracteriza por un uso cada vez mayor de las fuentes de energía renovables y por la búsqueda de la reducción del consumo obtenido de los combustibles fósiles. Un ejemplo representativo se puede encontrar en que el porcentaje de energía renovable en España se sitúa en la actualidad en aproximadamente un 20% del total, y algunos estudios prevén que aumente hasta más del 27% en 2030 [Utkucan, 2021]. A pesar de las incertidumbres acerca de cuánto durará y de cómo se llevará a cabo dicha transición, las cifras indican que el cambio se está produciendo, afectando a la generación y también al almacenamiento y al consumo de energía.

Para alcanzar los porcentajes mencionados, la implicación del conjunto de los actores económicos resulta crucial, en especial la de las empresas multinacionales [Bass, Grøgaard, 2021], llamadas a desempeñar un papel determinante en este cambio contribuyendo a la mejora de las comunidades y la vida en los países, al tiempo que se adaptan para mantener un alto nivel de competitividad.

Por ejemplo, algunas compañías que trabajan en el desarrollo de sistemas automatizados para la gestión de los edificios y de las Smart Home han puesto foco en la eficiencia de los consumos del hogar. La complejidad de los sistemas instalados [Reinisch, 2011] ha llevado a las empresas a desarrollar nuevos productos aplicados a la gestión del consumo eléctrico inteligente, los llamados sistemas de gestión energética, que pueden constituir un elemento importante dentro de esta transición y ser de gran utilidad para el usuario.

Las técnicas de análisis de datos (aprendizaje automático, minería de datos) resultan determinantes en estos ámbitos, así como en otros sectores, como el de las *utilities*[1] [Sharma, 2014, Zhou, 2016] y constituyen herramientas fundamentales a la hora de crear productos y servicios innovadores de este tipo.

Repsol también ha manifestado su intención de contribuir a este cambio a través de unos compromisos en consonancia con los objetivos de la Cumbre de París y los de Desarrollo Sostenible de Naciones Unidas según un comunicado de prensa [Ibarra,

[1] Denominación genérica de aquellas compañías que ofrecen servicios públicos como electricidad, gas, agua o autopistas (Fuente: CNMV)

03/12/2019]. Para ello, contempla en su hoja de ruta, entre otras acciones, el desarrollo de tecnologías y productos que consigan mejoras significativas en esta dirección, también haciendo uso de las aplicaciones mencionadas.

En concreto, uno de los productos que se han desarrollado es el que denominaremos *Desagregador* en este trabajo por cuestiones de confidencialidad. En este contexto, conviene hacer notar que Repsol y la UNED han firmado un acuerdo de confidencialidad sobre los datos, desarrollos y resultados obtenidos por Repsol que se incluyen en este trabajo de fin de máster, tanto a modo de contexto como de contenido.

El Desagregador es un software diseñado por Repsol para proporcionar al cliente un servicio de desagregación de su factura eléctrica. A partir del consumo total de un hogar, ofrece una estimación del porcentaje destinado a cada tipo de consumo de dicho hogar, separando entre los consumos de lavadora, lavavajillas, nevera, cocina o consumos fantasma (o de *stand-by)*, entre otros. En la figura 1, se presenta un ejemplo de resultados de esta herramienta para un usuario durante una semana concreta para una de las primeras versiones del producto.

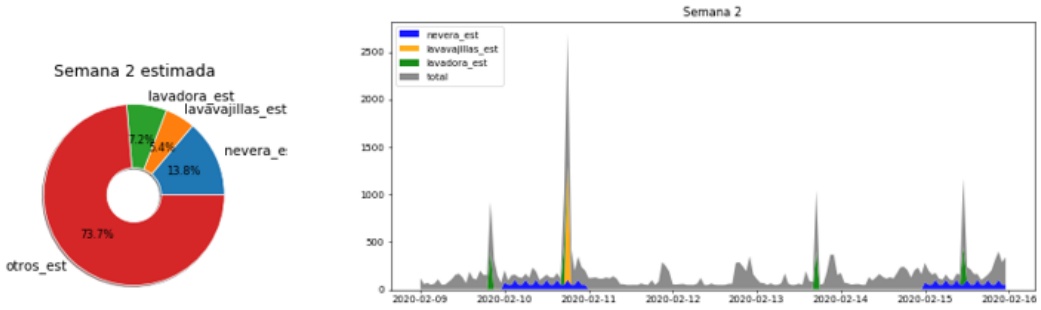

Figura 1. Ejemplo de resultados del Desagregador de Repsol obtenido en su fase de desarrollo.

En la primera versión del Desagregador se emplearon técnicas de análisis de datos basadas en optimización y métodos combinatorios. Para la mejora del comportamiento de la aplicación se pensó en desarrollos adicionales que facilitasen esa mejora de desempeño y este trabajo de fin de máster describe algunos de ellos.

1.2. Objetivo

El análisis que realiza el Desgagregador incorpora como datos entrada, entre otros, distribuciones de probabilidad de uso de cada electrodoméstico, así como señales que describen el comportamiento habitual de consumo de los electrodomésticos.

Para conseguir una mejora de los resultados de este producto, se pensó en algún tipo de clasificación de los clientes en cuanto a sus hábitos de consumo, o también en indicadores de uso de los electrodomésticos, es decir, información que permitiera separar grupos diferenciales o asignar probabilidades de uso específicas a por grupos. A

continuación, se expone un ejemplo de cómo este enfoque podría perfeccionar los resultados del Desagregador:

> El resultado de la clasificación que persigue este trabajo fin de máster podría ser del tipo: **el usuario U (objetivo del análisis) pone la lavadora en la franja de 'Entre las 10 y las 13' con una probabilidad del X%.** Este resultado alimentará, como una entrada más, al problema de desagregación. Dado que la desagregación utiliza un análisis combinatorio basado en señales tipo de consumo de electrodomésticos, esta nueva información podría contribuir a la mejora de la precisión en forma de restricción. Por tanto, para un usuario de este tipo, la información de que el cliente utilizará una lavadora con un porcentaje de probabilidad concreto en una franja horaria determinada puede ser clave para inferir si realmente se puso una lavadora en esa franja y no otro electrodoméstico.

El objetivo del trabajo es, por tanto, desarrollar un método de clasificación de clientes que permita mejorar la precisión del Desagregador.

Derivados del objetivo general se establecieron los siguientes objetivos específicos:

1. Obtener una muestra representativa de datos para el análisis.
2. Garantizar la confidencialidad de los datos correspondientes a usuarios reales.
3. Preprocesar y tratar los datos para facilitar un análisis efectivo.
4. Seleccionar las características más adecuadas.
5. Seleccionar y aplicar las técnicas necesarias para conseguir una clasificación de los usuarios lo más precisa posible.
6. Analizar y explicar los resultados.
7. Garantizar la confidencialidad de los desarrollos de Repsol en el ámbito de trabajo.

En la próxima sección se enuncia de un modo más conciso la definición del problema.

1.3. Definición del problema

Este trabajo tiene por objetivo obtener un método que sea capaz de clasificar a los clientes con métodos de aprendizaje automático para mejorar la precisión del resultado de los productos, en concreto del resultado del Desagregador.

Debido a que los clientes pueden clasificarse en base a muchos criterios, en primer lugar, se trató de definir el tipo de clasificación que podría ser más útil para mejorar dicho producto.

Como el Desagregador toma como datos de entrada la curva total de consumo de un hogar y la información genérica de las curvas tipo de consumo de cada electrodoméstico, estos datos fueron los primeros que se investigaron para definir el problema de un modo más concreto.

Según el Instituto para la diversificación y el ahorro de energía (IDAE), la lavadora y el lavavajillas representan en España, en promedio, consumos del 11,8% y del 6% del total de un hogar. Ambos electrodomésticos constituyen las principales cargas eléctricas de uso ocasional (después de otras permanentes de mayor consumo porcentual, como el frigorífico) de un hogar.

En la figura 2 se muestra información de referencia del IDAE acerca del consumo de los electrodomésticos de un hogar y en la figura 3, se presentan curvas de carga genéricas asociadas a la lavadora y al lavavajillas, también obtenidas del IDAE.

7.-% Consumo de Electrodomésticos según Tipo de Equipamiento

Frigorífico:	30,6%
Congeladores:	6,1%
Lavadora:	11,8%
Lavavajillas:	6,1%
Secadoras:	3,3%
Horno:	8,3%
TV:	12,2%
Ordenadores :	7,4%
Standby:	10,7%
Otros:	3,5%

8.-Consumo Medio de Iluminación y Electrodomésticos

Iluminación:	410 kWh/hogar
Frigorífico:	655 kWh/eq.
Lavadora:	254 kWh/eq.
Lavavajillas:	245 kWh/eq.
TV:	119 kWh/eq.
Ordenador:	145 kWh/eq.

Figura 2. Consumos promedio de hogares según informe del IDAE.

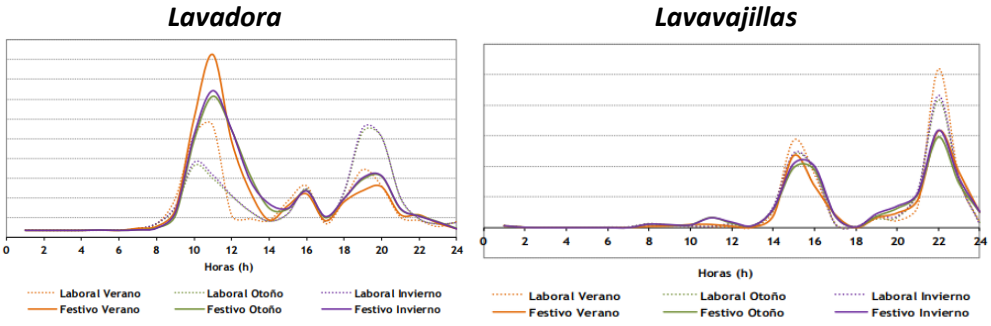

Figura 3. Curva de carga asociada a Lavadoras y lavavajillas según informes del IDAE.

A la vista de esta información, la lavadora, por ejemplo, parece un electrodoméstico más fácil de relacionar con la curva general, ya que sus picos de consumo son elevados en comparación con otros electrodomésticos del hogar, tal y como se muestra en la figura 4 [Jacobs, Maun, 2019]. Patrones de consumo también con picos asociados a incrementos de temperatura se producen en el lavavajillas [Michel, 2017].

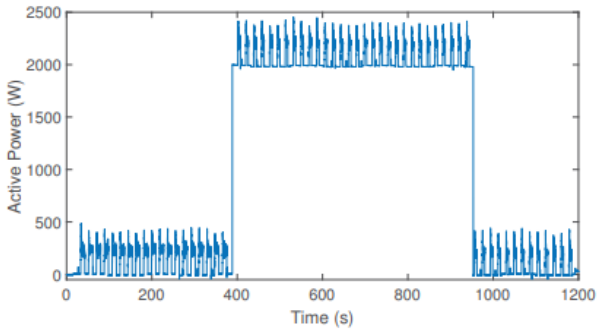

Figura 4. Consumos de una lavadora durante los 20 primeros minutos. Fuente: Jacobs, Maun, 2019.

Se pensó, por tanto, que los usos de lavadora y de lavavajillas podrían reflejarse en la curva de consumo general en forma de picos, y comportarse como indicadores de uso diferencial de cada usuario, siendo por tanto importantes en la clasificación deseada.

Los datos de los que se disponía, como se describirá a continuación, eran los de consumo horario total de un hogar [Kwac, 2014], por lo que el problema en concreto se definió como: **desarrollar un método de clasificación para, a partir de los datos de la curva total horaria de consumo de un usuario, inferir la hora de encendido y el número de encendidos semanales de lavadora y lavavajillas**.

Aunque se eligieron estos electrodomésticos por su tipo de uso ocasional, es preciso señalar que la resolución del problema es compleja, al tratar de clasificar hábitos de uso de equipos individuales a partir de una curva general agregada de consumo del hogar, que engloba, a la vez, consumos de otros electrodomésticos.

La razón de partir de la curva agregada de consumo de un usuario es importante, ya que se pretende que el Desagregador sea autónomo y que, a ser posible, sólo tome datos de consumo general del hogar como datos de entrada.

1.4. Datos utilizados

Para resolver el problema, se utilizaron dos fuentes de datos.

En primer lugar, datos de clientes de Repsol Electricidad y Gas. Según se ha mencionado anteriormente, se firmó un acuerdo de confidencialidad entre Repsol y la UNED y por tanto se mostrarán los resultados con pseudónimos o con variaciones para garantizar la confidencialidad. El acceso a esta base de datos se llevó a cabo cumpliendo la normativa de protección de datos de Repsol y garantizando en todo momento la confidencialidad de los usuarios para lograr un análisis ético, seguro y confidencial.

Se seleccionó una muestra de 747 usuarios para los que se dispuso de datos de consumo horario total del hogar durante el período comprendido entre el 01 de mayo y el 22 de noviembre del 2021 y también se dispuso del código postal de estos usuarios.

En la tabla 1 se muestra un extracto de los consumos horarios, para cada día, con las horas de la 1 a la 23 (Media_magnitud_activa01, … Media_magnitud_activa23), situadas en cada columna.

	user	fecha	Medida_magnitud_activaH01	Medida_magnitud_activaH02	Medida_magnitud_activaH03	Medida_magı
0	user1	01/05/2021	95.0	132.0	86.0	
1	user1	02/05/2021	53.0	51.0	55.0	
2	user1	03/05/2021	59.0	54.0	54.0	
3	user1	04/05/2021	54.0	70.0	60.0	
4	user1	05/05/2021	72.0	55.0	58.0	
...	...	...	...	...	...	
150827	user747	18/11/2021	169.0	175.0	126.0	
150828	user747	19/11/2021	173.0	175.0	176.0	
150829	user747	20/11/2021	189.0	183.0	153.0	
150830	user747	21/11/2021	193.0	132.0	86.0	
150831	user747	22/11/2021	66.0	68.0	94.0	

150832 rows × 26 columns

Tabla 1. Extracto de la tabla de consumos horarios por día.

En segundo lugar, se solicitó a este mismo grupo de usuarios que respondieran a un cuestionario sobre hábitos de uso de los electrodomésticos, y las respuestas se utilizaron para realizar un etiquetado de uso de lavadora y lavavajillas.

En el cuestionario se preguntó por la hora más frecuente de encendido de lavadora y la hora más frecuente de encendido de lavavajillas, proporcionando como posibles respuestas para ambos una lista cerrada de horarios:

- 'No tengo un horario fijo'
- 'Entre las 2 y las 7'
- 'Entre las 7 y las 10'
- 'Entre las 10 y las 13'
- 'Entre las 13 y las 16’
- 'Entre las 16 y las 20'
- 'Entre las 20 y las 23'
- 'Entre las 23 y las 2’

También se preguntó por el número de encendidos semanales para ambos electrodomésticos, proporcionando como posibles respuestas la siguiente lista cerrada de opciones:

- ‘Menos de 1 vez por semana’
- ‘1 vez por semana’
- ‘2 o 3 veces por semana’
- ‘4 o 5 veces por semana’
- ‘Más de 5 veces por semana’

Por otra parte, se preguntó por la fecha de encendido de alguno de estos electrodomésticos, la eficiencia, o si se utiliza o no un programa ECO, con la intención de ser empleados en este u otro método dentro del desarrollo del Desagregador o en otros productos, aunque estos datos no llegaron a usarse en el presente trabajo.

En la figura 5 se muestra un esquema de las respuestas tipo para un usuario.

	Lavadora		Lavavajillas	
	Día	**Hora**	**Día**	**Hora**
¿Recuerdas cuándo fue la última vez que pusiste este electrodoméstico? Día y hora aproximada a ser posible.	29/05/2021	9:00:00	30/05/2021	2:00:00

	Lavadora	Lavavajillas
¿Cuántos días a la semana lo sueles poner?	2-3 veces por semana	2-3 veces por semana
Horario habitual en el que lo pones	07:00-10:00	02:00-07:00

	Lavadora	Lavavajillas
¿Conoces la eficiencia de alguno de tus equipos?	lo desconozco	lo desconozco

Figura 5. Esquema de las respuestas tipo para un usuario.

En la tabla 2, se recoge un extracto de las respuestas al cuestionario para algunos clientes.

USUARIO	N_HORNO_SEMANA	N_LAVADORA_SEMANA	HORA_LAVADORA	N_LAVAVAJILLAS_SEMANA	HORA_LAVAVAJILLAS
1	2 o 3 veces por semana	2 o 3 veces por semana	No tengo un horario fijo	2 o 3 veces por semana	Entre las 23 y las 2
2	Menos de 1 vez por semana	2 o 3 veces por semana	No tengo un horario fijo	Más de 5 veces por semana	No tengo un horario fijo
3	Menos de 1 vez por semana	2 o 3 veces por semana	Entre las 7 y las 10		
4	1 vez por semana	Más de 5 veces por semana	Entre las 10 y las 13	2 o 3 veces por semana	Entre las 13 y las 16
5	Menos de 1 vez por semana	2 o 3 veces por semana	No tengo un horario fijo	2 o 3 veces por semana	No tengo un horario fijo
6	4 o 5 veces por semana	Más de 5 veces por semana	Entre las 10 y las 13	4 o 5 veces por semana	No tengo un horario fijo
7	2 o 3 veces por semana	Más de 5 veces por semana	Entre las 7 y las 10	Más de 5 veces por semana	Entre las 23 y las 2
8	2 o 3 veces por semana	4 o 5 veces por semana	No tengo un horario fijo		
9	4 o 5 veces por semana	4 o 5 veces por semana	No tengo un horario fijo	4 o 5 veces por semana	No tengo un horario fijo
10	Menos de 1 vez por semana	1 vez por semana	Entre las 16 y las 20	1 vez por semana	Entre las 16 y las 20
11	1 vez por semana	1 vez por semana	Entre las 10 y las 13	1 vez por semana	Entre las 23 y las 2
12	2 o 3 veces por semana	Más de 5 veces por semana	Entre las 10 y las 13	4 o 5 veces por semana	Entre las 20 y las 23
13	Menos de 1 vez por semana	4 o 5 veces por semana	Entre las 10 y las 13	4 o 5 veces por semana	Entre las 13 y las 16
14	2 o 3 veces por semana	2 o 3 veces por semana	Entre las 10 y las 13	4 o 5 veces por semana	No tengo un horario fijo
15	2 o 3 veces por semana	1 vez por semana	Entre las 23 y las 2	Menos de 1 vez a la semana	Entre las 23 y las 2
16	Menos de 1 vez por semana	4 o 5 veces por semana	Entre las 16 y las 20	Menos de 1 vez a la semana	No tengo un horario fijo
17	1 vez por semana	1 vez por semana	Entre las 16 y las 20	Menos de 1 vez a la semana	No tengo un horario fijo

Tabla 2. Extracto de las respuestas del cuestionario para algunos de los clientes.

Como se puede observar no todos los campos del cuestionario fueron respondidos por todos los usuarios por lo que no se dispuso de un número completo de respuestas.

Adicionalmente, como se ha mencionado, la dificultad de estimar consumos individuales a partir de los datos de consumo total conlleva cierta complejidad para el problema.

1.5. Organización de la memoria

El resto de la memoria se organiza de la siguiente forma:

En la sección 2 se detalla el estado del arte referente al procesado de datos, a la precisión, a la selección de características y a los métodos de clasificación.

En la sección 3 se describe en detalle el trabajo realizado, que incluyó el análisis y el tratamiento de los datos, la selección y filtrado de características y el diseño de los experimentos para comprobar la importancia de características y seleccionar el método de clasificación para obtener la precisión máxima.

En la sección 4 se presentan los resultados obtenidos a partir de dichos experimentos.

En la sección 5 se exponen las conclusiones y líneas futuras del trabajo.

Por último, en la sección 6 se recogen las referencias bibliográficas y finalmente se añade un Anexo para detallar la estructura general del código empleado en el trabajo.

2. Estado del arte y fundamento teórico

En esta sección se describe el fundamento teórico de los métodos que se emplearán durante el análisis.

El caso que nos ocupa, al ser novedoso y particular, no tiene precedentes claros en estudios publicados, por lo que se recurrió a trabajos que podrían considerarse similares, aunque no fueran realizados con los mismos objetivos, como los de Beckel [2013], Sodenkamp [2014] o Yu[2020].

El estudio de Beckel se centra en utilizar algoritmos no intrusivos[2] para, a partir de la curva de carga de hogares, conseguir una clasificación de los tipos de vivienda.

El estudio de Sondenkamp parte del estudio de Beckel, y utiliza, además de la curva de carga de los hogares, datos de los usuarios (por ejemplo, tipo de calefacción, edad de la vivienda, número de personas que lo habitan), con la intención de aportar luz en el campo de la optimización de las redes inteligentes[3], uno de los pilares de la futura gestión energética sostenible y que podría aumentar la eficiencia energética bilateralmente, tanto desde el punto de vista del consumidor como desde el del proveedor de electricidad. En su análisis abordan, entre otros puntos, la selección de características, así como el filtrado de las más relevantes para obtener un análisis preciso.

El trabajo de Yu intenta averiguar patrones de comportamiento de los consumidores a partir de su consumo con el objetivo de predecir la demanda. Trata con especial detalle los métodos de selección de características y se centra en aquellas basadas en el consumo medido, a partir de dispositivos instalados en las líneas de carga de los hogares. También aplica filtros de características y utiliza un método de clasificación no supervisada para obtener las clases o grupos de consumidores.

[2] Non intrusive load monitoring (NILM).
[3] *Smart Grids*, en inglés.

2.1. Consideraciones acerca de los datos, la precisión y el procedimiento

Para el propósito de la clasificación, una muestra de datos de consumo medidos con un espaciamiento de 60 minutos puede ser considerado suficiente según Sodenkamp [2014] aunque siempre que sea posible sería preferible tener medidas con una frecuencia mayor.

Algunos estudios, como el de Beckel [2013], han demostrado que es posible inferir propiedades de un hogar utilizando métodos de clasificación sobre datos de consumo eléctrico total. En concreto, su estudio empleó una muestra de 3488 hogares (con granularidad de 30 minutos y 1,5 años de datos) para estimar algunas propiedades como habitantes del hogar o metros cuadrados y se alcanzaron precisiones superiores al 70%.

El trabajo de Sondenkamp [2014] partió de 4200 hogares y de datos de 72 semanas con cierta homogeneidad (sin días festivos y con climas similares), para averiguar características del hogar y en él se lograron precisiones variables según el tipo de características analizadas. Para características relacionadas con los dispositivos del hogar las precisiones estuvieron alrededor del 53%, para características relacionadas con empleo estuvieron en torno a 56% y para características relacionadas con la edad de la vivienda se alcanzaron precisiones de hasta el 79%.

El trabajo de Yu [2020] utilizó datos de 6277 viviendas, con 48 puntos de muestreo en cada hogar a frecuencias de 30 minutos. Con un análisis de *clustering* logró clasificar hábitos de comportamiento con precisiones de entre 79% y 98%.

Como se ha mencionado, para el presente trabajo se contó con datos de consumo de tipo horario (cada 60 minutos) recogidos en los meses de mayo a noviembre de 2021 correspondientes a 29 semanas, para 747 usuarios.

Tomando como referencia los trabajos anteriores, la precisión objetivo del presente análisis podría suponerse, a priori, en el entorno de valores entre el 50 y el 98%. Como se utiliza una muestra de datos menor que las de los estudios mencionados, y dada la complejidad añadida del problema mencionado, la precisión esperada podría estar en el entorno bajo de la horquilla definida y se podría dar por válida. Más en concreto, el análisis se asemejaría en complejidad al estudio de Sondenkamp, que alcanzó precisiones de entre 53% y 79%.

En cualquier caso y siguiendo los trabajos señalados, el hecho de complementar los datos de consumo con información proveniente de la interpretación de sus usuarios parece la mejor aproximación a este tipo de problemas para abordar una clasificación supervisada, evitando la inferencia de patrones más teóricos a partir de métodos de clasificación no supervisada.

2.2. Definición y selección de características

Una adecuada definición de características[4] resulta fundamental para realizar un buen análisis. Según Sodenkamp [2014], una buena selección de características puede mejorar la precisión, en concreto la utilización de técnicas de filtrado y por tanto de elección de las características más adecuadas, puede mejorar la precisión en órdenes de entre el 3% y el 12%. [Sondenkamp, 2014]. No obstante, resulta necesario encontrar un equilibrio para no seleccionar un número excesivamente alto de características (que en algunos casos puede comprometer el resultado), ni tampoco muy bajo.

En el estudio de la clasificación de usuarios por su comportamiento de consumo, se suele emplear el consumo global para definir características. Yu [2020] señala dos posibles tipos: por un lado, las intuitivas, asociadas a diferentes tipos de consumo, y por otro, las basadas en ratios. Esta división se muestra continuación:

Intuitivas:

- Consumo diario
- Consumo máximo y mínimo diario
- Promedio diario
- Diferencia pico-valle

Relacionados con tarifa:

- Factor de potencia en valle
- Ratio pico/valle diario
- % de consumo en horas habituales
- % de consumo en horas pico

Para este tipo de análisis, Sodenkamp [2014] partió de características del tipo consumo diario, como indicadores de consumo a horas concretas del día o ratios de consumo, pero también definió otras adicionales basadas en parámetros estadísticos, así como en relaciones estacionales o patrones semanales y de fin de semana. Una vez definidas, realizó una normalización de características para optimizar el cálculo.

Algunas de las métricas en las que se podría pensar inicialmente como características para el problema que nos ocupa se muestran a continuación.

Para consumos diarios, por ejemplo, se puede asumir agrupaciones de 3 franjas horarias para cada día, mañana, tarde y noche:

- Promedio total
- Promedio mañana
- Promedio tarde
- Promedio noche
- Suma total

[4] *Features,* en inglés

- Suma mañana
- Suma tarde
- Suma noche
- Mínimo total
- Mínimo mañana
- Mínimo tarde
- Mínimo noche
- Máximo total
- Máximo mañana
- Máximo tarde
- Máximo noche

En un análisis a nivel semanal:

- Promedio semanal
- Total semanal
- Máximo semanal
- Mínimo semanal
- Máximo entre semana
- Máximo fin de semana
- Mínimo entre semana
- Mínimo fin de semana
- Total entre semana
- Total fin de semana
- Porcentaje de consumo de entre semana respecto a total de semana
- Porcentaje de consumo de fin de semana respecto a total de semana

También se podrían utilizar indicadores horarios:

- Promedio por hora para cada mes
- Total por hora para cada mes
- Promedio por hora para cada día de la semana
- Total por hora para cada día de la semana

Si se abordasen variables estadísticas:

- Varianza por franja horaria
- Varianza entre semana
- Varianza el fin de semana

O variables no asociadas a consumo:

- Temperatura exterior por zona
- Renta media a partir de código postal

O asociadas a picos:

- Media picos diarios entre la semana
- Media picos diarios el fin de semana
- Máximo picos diarios
- Mínimo picos diarios

Tal como describe Cai [2018], las estrategias para la selección de características han proliferado y mejorado en los últimos años para conjuntos de datos de alta dimensión. Pueden variar según el tipo de datos (supervisadas o no supervisadas), de acuerdo con el método empleado (filtrados o embebidos), en base al criterio de evaluación (basados en la distancia euclídea o en otras distancias), o según las técnicas de búsqueda (hacia adelante o hacia atrás[5]).

Esta selección se puede realizar usando algoritmos, como *Random Forest* o *Extreme Gradient Boosting* [Ayub et al., 2020], pero también a partir de métodos de filtrado.

Sondenkamp [2014] prefiere los métodos de filtrado a los envolventes[6] o a los estadísticos porque resultan independientes del tipo de clasificador usado. Por eso en este trabajo se decidió aplicar una de esas técnicas, que suelen tener bajo coste computacional y potencia de cálculo aceptable. En concreto se empleó un filtro como el descrito por Yu [2020], denominado de máxima relevancia y mínima redundancia (mRMR), llamado así porque, en cada iteración, selecciona la característica que tiene la máxima relevancia con respecto a la variable de destino y la mínima redundancia con respecto al resto de características seleccionadas.

En la figura 6 se muestran las ecuaciones en las que está basado el filtro mRMR, donde $D(S,d)$ es el índice de máxima correlación y $R(S)$ es el índice de mínima redundancia. S representa el conjunto de características y N_S el número de características contenidas en él. $I(t_d; d)$ es un término complejo que se obtiene de cálculo de la entropía de sus parámetros y representa la información mutua entre la característica t_i y una variable objetivo d. De manera que el conjunto de características solución satisface el criterio óptimo.

$$D(S,d) = \frac{1}{N_S} \sum_{i=1}^{N_S} I(t_i; d)$$

$$R(S) = \frac{1}{N_S{}^2} \sum_{t_i, t_j \in S} \left| \rho_{t_i t_j} \right|$$

[5] *Forward increase, backward deletion*, en inglés.
[6] *Wrapper*, en inglés.

$$mRMR = \max_{S}[D(S,d) - R(S)]$$

Figura 6. Ecuaciones del filtro mRMR.

2.3. Métodos de aprendizaje automático

En esta sección se van a comentar los conceptos teóricos de los métodos que se han utilizado en el desarrollo del trabajo, prestando especial atención a los procedimientos de aprendizaje automático[7] empleados en las series de experimentos.

Se describirán métodos de clasificación supervisada y no supervisada. Gran parte de la información para este apartado se obtuvo de Muhamedyev [2015] y del libro *Data Mining Concepts and Techniques* [Han, 2012].

En la figura 7 se muestra una taxonomía de los métodos de aprendizaje automático según distingue Muhamedyev [2015].

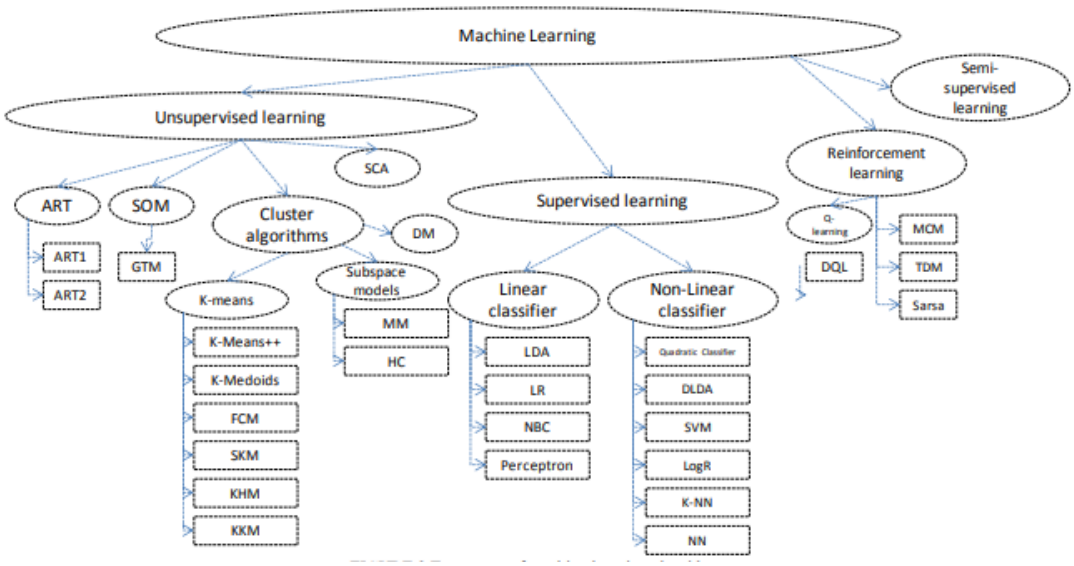

Figura 7. Taxonomía de métodos de aprendizaje automático. Fuente: Muhamedyev, 2015.

MÉTODOS DE CLASIFICACIÓN SUPERVISADA

La clasificación es un proceso de búsqueda de un modelo (o función) para diferenciar clases o conceptos dentro de un conjunto de individuos.

En la clasificación supervisada el modelo se obtiene del análisis de un conjunto de datos[8], donde las etiquetas que identifican a cada clase son conocidas para cada individuo. Una vez definido, este modelo se puede usar para predecir la etiqueta de clase de nuevos individuos en los que esa etiqueta sea desconocida. En la figura 8 se muestra un gráfico

[7] En inglés, *Machine Learning*.
[8] En inglés, *dataset.*

con el resultado de clasificación lineal (*a*) y no lineal (*b*) para individuos a partir de dos características (una en cada eje).

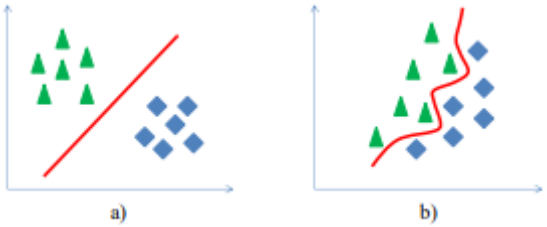

Figura 8. Ejemplo de clasificación lineal (a) y no lineal (b) en 2 dimensiones. Fuente: Muhamedyev, 2015.

La clasificación o aprendizaje supervisado es, por tanto, un proceso de dos pasos.

Es necesario un primer paso de *entrenamiento* (o fase de aprendizaje), donde un algoritmo de clasificación construye el modelo o clasificador, analizando o "aprendiendo de" un conjunto de datos de entrenamiento. Se puede imaginar este conjunto como una matriz X_{ent} de dimensión n x k y sus etiquetas de clase asociadas como un vector y_{ent}. Dentro de esta matriz, cada elemento $x_{ij,ent}$ toma un valor, donde la fila i corresponde a cada individuo (con $i = 1,\ldots,n$) y la columna j, a cada característica seleccionada de la base de datos (con $j = 1,\ldots,k$). Cada individuo pertenece a una clase predefinida, que se indica con la denominada etiqueta de clase, contenida en el vector $y_{ent} = (y_1, y_2,\ldots, y_n)$. Nótese que para los problemas de regresión esta etiqueta es un número real, mientras que para clasificación es una etiqueta de clase que puede ser o no numérica.

Así el objetivo a través de este entrenamiento es encontrar un modelo o función $y_{ent} = f(X_{ent})$, que describa la relación entre los grupos de individuos y sus etiquetas.

El segundo paso se denomina *test* y en él se comprueba la capacidad de la función $f(X)$, extraída del paso anterior, para predecir una etiqueta de clase de individuos no etiquetados. Esto se realiza con un nuevo conjunto, datos de test, $X_{test} = (x_{test,ij})$, para los que también se dispone de etiquetas asociadas a clase, y_{test}. La comparativa entre las etiquetas que predice la función $y_{estimada} = f(X_{test})$ y las reales y_{test}, proporciona una idea de la bondad de acierto o de la precisión de la función.

Para llevar a cabo estos dos pasos es necesario, por tanto, realizar una división en los datos de entrada disponibles utilizando parte como muestra de *entrenamiento* y parte como muestra de *test*.

Estos métodos tienen aplicación en diversos campos y tipos de problema, desde el reconocimiento de documentos al procesado de imágenes, pero en este proyecto se aplicarán al reconocimiento de patrones de comportamiento de consumidores de energía.

A continuación, se describen algunos de los métodos más utilizados:

Regresión logística. El modelo más sencillo de clasificación es la regresión. La logística es un caso particular de regresión que se usa para predecir resultados de variables categóricas de 2 clases. Mide la relación entre una variable dependiente categórica y una o varias variables independientes continuas, asignándoles una probabilidad a cada una de ellas. Se utiliza, por tanto, preferentemente para clasificaciones binarias y no multiclase.

Es un método sencillo, aunque plantea problemas frente a datos no lineales o complejos y las precisiones no son tan altas como en otros.

K-Nearest-Neighbour*[9] *(KNN). Es un método basado en instancias[10] de los más utilizados. Dado un conjunto de elementos (o individuos) etiquetados en clases, la predicción de la clase de un nuevo elemento se asocia a la clase que dista menos de los elementos etiquetados. Clasifica las nuevas muestras, por tanto, a partir de la asignación de clase de las muestras de entrenamiento más próximas.

En este tipo de algoritmos entra en juego una medida de la distancia entre los elementos de cada clase, cuya selección condiciona el resultado de la clasificación. En la figura 9 se recogen algunas de las distancias empleadas entre dos elementos *x* e *y* que señala Kotsiantis [2007].

Minkowsky: $D(x,y)=\left(\sum_{i=1}^{m}\lvert x_i - y_i\rvert^r\right)^{1/r}$
Manhattan: $D(x,y)=\sum_{i=1}^{m}\lvert x_i - y_i\rvert$
Chebychev: $D(x,y)=\max_{i=1}^{m}\lvert x_i - y_i\rvert$
Euclidean: $D(x,y)=\left(\sum_{i=1}^{m}\lvert x_i - y_i\rvert^2\right)^{1/2}$
Camberra: $D(x,y)=\sum_{i=1}^{m}\frac{\lvert x_i - y_i\rvert}{\lvert x_i + y_i\rvert}$

Figura 9. Algunas distancias empleadas en los métodos instanciados, como el KNN. Fuente: Kotsiantis, 2007.

El *KNN* es fácil de implementar y de comprender, sin embargo, puede dar resultados erróneos ante instancias con ruido. Este método acarrea menor tiempo de computación durante la fase de *entrenamiento*, pero mayor durante la de *test* o la de predicción.

Support Vector Machines (SVM o SVC) o máquinas de vectores soporte. Son métodos basados en la técnica del plano solución que se usan para clasificar datos lineales o no lineales. Pueden utilizar operaciones para realizar un mapeo no lineal transformando los datos de entrenamiento a un espacio de dimensión superior. En ese nuevo espacio busca el hiperplano óptimo de separación de los individuos.

[9] O vecino más cercano.
[10] *Instance based learning (IBL)*, en inglés.

Son métodos potentes en precisión y para tratamiento de datos no lineales y menos proclives a sobreajuste[11] que otros.

Son capaces de operar bien en grandes dimensiones de datos. Por el contrario, requieren de un mayor tiempo de ejecución y un ajuste preciso de sus hiperparámetros, en concreto de la función *kernel*, que es una característica clave de este tipo de algoritmo [Cristianini, 2002].

Árboles de decisión o CART[12]. Un árbol de decisión puede ser de clasificación o de regresión. Los de clasificación se emplean para agrupar datos en clases y predecir una categoría y los de regresión para predecir valores numéricos reales.

Poseen una estructura de árbol similar a un diagrama de flujo, donde cada paso interno (nodo) denota una prueba en un atributo, cada rama representa un resultado de la prueba, y cada nodo hoja (o nodo terminal) tiene una etiqueta de clase o un valor. El nodo superior en un árbol se denomina nodo raíz. Por ejemplo, para un árbol de clasificación, la evaluación tiene lugar en cada rama, que equivale a una partición de valores, hasta llegar al resultado de clase, tal como se muestra en la figura 10, obtenida de Han [2012].

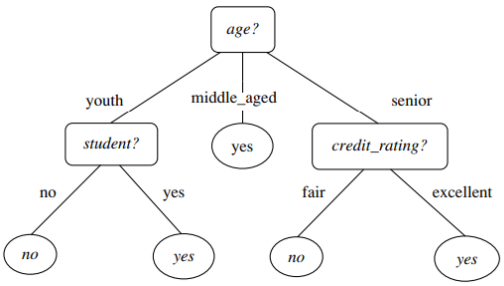

Figura 10. Esquema de un algoritmo de árbol de decisión. Fuente: Han, 2012

En general, los árboles de decisión son métodos sencillos y rápidos, pero en algunos casos tienen riesgo de producir sobreajuste.

Métodos combinados o de conjunto[13]. Son métodos para la clasificación basados en un modelo compuesto, formado por una combinación de clasificadores, que generalmente son árboles de decisión.

Los clasificadores individuales votan y la etiqueta que se predice se devuelve como resultado en función del número de votos recolectados para cada una. Un ejemplo se muestra en la figura 11, en la que se votan los resultados de dos árboles de decisión, penalizando los resultados equivocados [Chen, 2016].

[11] *Overfitting,* en inglés, indicador de que un modelo de aprendizaje automático sea poco generalizable.
[12] *Classification and Regression Trees.*
[13] *Ensemble methods*, en inglés.

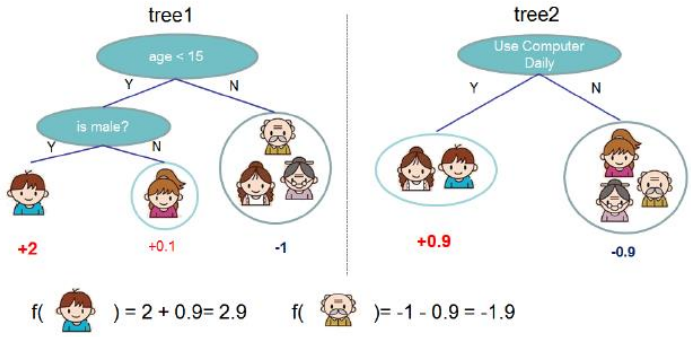

Figura 11. Ejemplo de cálculo de método combinado, con votaciones a cada uno de los resultados de los árboles. Fuente: Chen, 2016.

Estos métodos tienden a lograr mayor precisión que los clasificadores individuales que los forman y por eso son muy utilizados. Según Dietterich [2000] existen varias razones por las que estos métodos consiguen mayores precisiones: estadística (mediante votaciones se puede reducir el riesgo de elegir un clasificador erróneo), computacional (iniciar la búsqueda de mínimos desde varios puntos para evitar mínimos locales) y representacional (la asignación de pesos puede extender el espacio de funciones representadas más allá de las hipótesis iniciales).

Estos métodos pueden ser capaces de evitar el sobreajuste gracias al uso de funciones de regularización como parte de las funciones de coste, que logran suavizar los pesos finales del algoritmo. Aunque esta capacidad puede verse afectada por muestras que contengan ruido elevado.

Dentro de estos métodos se distinguen, según la forma en que se manipule el conjunto de entrenamiento, dos tipos de procedimiento: el *bagging* y el *boosting* [Bauer, 1999].

El *bagging*[14] consigue combinaciones de modelos a partir de una familia inicial, con la intención de minimizar la varianza y evitar el sobreajuste, paralelizando para ello los cálculos de varios de los modelos. A partir de un conjunto de entrenamiento inicial, se generan *m* nuevos conjuntos de entrenamiento tomando elementos al azar del conjunto original (y admitiendo repetición). Para cada conjunto se construye un modelo y después se utiliza una técnica de promedio o de votación para dar la predicción definitiva a partir de los *m* modelos. Estos modelos son especialmente buenos para combinar algoritmos individuales como los árboles de decisión, pero quizá no tanto con modelos como el antes mencionado *KNN*.

Uno de los algoritmos más extendidos de este tipo es el ***Random Forest.*** Cada clasificador en este método es un árbol de decisión. Los árboles individuales se generan usando una selección aleatoria de atributos en cada nodo. Cada árbol depende de los valores de un vector aleatorio muestreado de manera independiente y con la misma

[14] También conocido como *Bootstrap Aggregation*

distribución para todos los árboles empleados. La particularidad del método reside en que el resultado es la media de los resultados obtenidos con los árboles individuales.

El *boosting*, en cambio, utiliza una metodología secuencial [Natekin, 2013]. Parte del mismo conjunto de entrenamiento para todos los árboles y les asigna inicialmente un peso. En cada etapa siguiente se evalúa la predicción del clasificador y a los predictores incorrectos se le asigna un peso inversamente proporcional al error de medida, de forma que sean ponderados (en inglés *boosted*) con un peso mayor, al contrario que a los que predicen de forma correcta, que reducen su peso. En cada una de las rondas posteriores, los clasificadores más pesados se tienen menos en cuenta. Estos algoritmos suelen obtener muy buenos resultados, sobre todo para grandes volúmenes de datos.

Algunos casos particulares de *boosting* son *Adaboost*, *Gradient Boosting* o *Extreme Gradient Boosting* (*XGBoost*). A continuación, se describen los dos últimos:

Gradient Boosting. Es un caso particular de *boosting* en el que se emplea el método de optimización *Gradient Descent* para encontrar el mínimo de la función de coste. Este método iterativo parte de un punto aleatorio y encuentra el mínimo de una función recorriendo su pendiente, lo que lo convierte en un método más eficiente, rápido y preciso. Adicionalmente, los métodos de *Gradient Boosting* pueden incluir métodos de regresión o clasificación.

Un caso particular de este tipo de métodos es el ***Extreme Gradient Boosting***, *o XGBoost*, que ha tenido un impacto relevante en diferentes ámbitos, desde el reconocimiento de textos a la predicción de comportamiento de consumidores, pasando por la predicción de riesgos o la detección de movimientos [Chen, 2016]. La razón de dicha potencia se atribuye principalmente a su escalabilidad, amparada en métodos de optimización novedosos. Permiten manejar grandes cantidades de datos, así como muestras con dispersión o datos ausentes. Emplean métodos de ponderado de árboles mejorados, así como procedimientos de cálculo para incrementar la velocidad. Por el contrario, este tipo de algoritmos requieren de mayor uso de la memoria RAM, así como de un ajuste preciso de los parámetros para minimizar el error.

Redes Neuronales Artificiales[15]**.** Son otro de los algoritmos más populares a la hora de resolver problemas de aprendizaje automático [Chen, 2016]. Aunque son abstracciones simplificadas de los sistemas biológicos en los que se basan, utilizan los fundamentos de lo que se conoce acerca de las redes neuronales biológicas para resolver problemas.

Son estructuras compuestas de una serie de unidades de procesamiento, las neuronas, capaces de realizar operaciones masivas en paralelo. Las neuronas biológicas reciben información en forma de impulsos eléctricos a través de unas conexiones llamadas dendritas y envían información a través de una rama, el axón, que les permite conectarse a otras neuronas. Esta estructura de procesamiento es la que hereda la unidad de

[15] *Artificial Neural Networks (ANN), en inglés.*

procesamiento individual de una red neuronal artificial: el Perceptrón. Dicha equivalencia se puede ver en la figura 12 [Basheer, 2000].

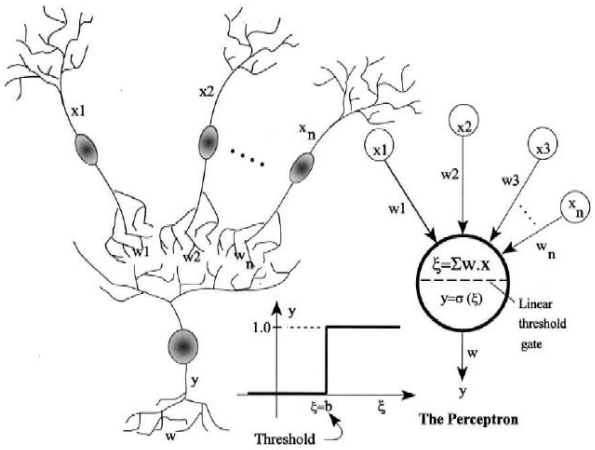

Figura 12. Estructura de procesamiento de una neurona biológica y su analogía con el Perceptrón de una red neuronal artificial. Fuente: Basheer, 2000.

De la misma forma que la neurona biológica, la neurona artificial (o Perceptrón) recibe datos de entrada de cada rama x_1 a x_n, (que representan las dendritas), los pondera (esta ponderación es análoga a la de las sinapsis de los axones) con los pesos w_1 a w_n respectivamente, y construye una señal ponderada ξ que, si supera el umbral marcado por el núcleo de la neurona, produce su activación y transmite la señal a la siguiente neurona.

La red neuronal se construye mediante la conexión de varias capas (figura 13) a partir del modelo expuesto de la neurona individual [Han, 2012]. Consta de una capa de entrada[16], que ha de tener un número de neuronas igual a la dimensión de las características que se desean incluir en el análisis. Después se añaden varias capas intermedias[17], en las que se establecen las conexiones y por último existe una capa de salida[18], cuya dimensión debe ser igual al número de salidas esperadas; por ejemplo, para una red de análisis binario, la capa de salida tendrá dos unidades, pero para una red de análisis multiclase, la capa de salida tendrá el mismo número de unidades que de clases etiquetadas.

[16] *Input layer*
[17] *Hidden layers* o capas ocultas
[18] *Output layer*

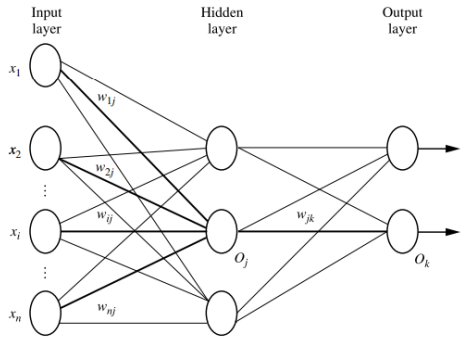

Figura 13. Estructura de una red neuronal, con la capa de entrada (input), una capa oculta (hidden) y la capa de salida (output). Fuente: Han, 2012.

Estos algoritmos suelen alcanzar buenos resultados, debido a varios puntos fuertes, como la tolerancia a fallos y a no linealidades, la habilidad de manejar información difusa y la robustez. Por contra, suelen ser más exigentes en procesamiento que otros métodos.

MÉTODOS DE CLASIFICACIÓN NO SUPERVISADA

Estos métodos, a diferencia de los de clasificación supervisada, permiten agrupar datos no etiquetados y asignarles etiquetas de clase. En particular, se detallarán a continuación algunos métodos de *clustering*, que son un caso particular de este tipo ya que uno de ellos se empleó en el trabajo.

Las agrupaciones de clase se forman utilizando procedimientos que maximicen la similitud entre los elementos de una clase y minimicen la similitud entre los elementos de clases diferentes y, por lo general, esto se traduce en el uso de distintas medidas de la distancia entre los elementos.

El *clustering* conlleva la partición de un conjunto de elementos o individuos, definidos por una matriz *X* análoga a la descrita para clasificación supervisada, pero en este caso el problema no considera el vector de etiquetas de clase *y*. Las características de los individuos de *X* se agrupan con estos métodos en subconjuntos de individuos similares y pueden aplicarse en multitud de ámbitos, desde el reconocimiento de objetos en imágenes, búsqueda web, aplicaciones de seguridad o de *business intelligence*.

Algunas ventajas de aplicar un método de *clustering* son la escalabilidad, la capacidad de procesar datos ruidosos, la capacidad de tratar con diferentes tipos de atributos, etc.

La mayoría de los métodos de *clustering* utilizan técnicas jerárquicas o particionales de división, como se describe a continuación.

Métodos particionales:

***K-means*.** El método comienza asignando centroides aleatorios para cada *cluster* (estos pueden ser o no parte de la muestra) y asigna cada elemento a uno de esos

centroides por distancia más próxima (esta distancia puede ser la euclídea u otra). Una vez asignada se recalcula el centroide de cada *cluster* usando la media de las distancias de todos los nuevos elementos del *cluster* y se vuelve a realizar la asignación de todos los puntos por proximidad a los nuevos centroides. La iteración se repite hasta que no se produce variación sustancial en los centros (definida en base a una función de error).

Las principales ventajas son su simplicidad de cálculo y la linealidad, y el principal inconveniente es la menor repetibilidad de los resultados, ya que los centros originales se seleccionan de forma aleatoria, así como el valor del número de *clusters* que ha de especificarse desde el principio.

Existen variaciones de este método, como por ejemplo el *K-Medians*, que computa la mediana en lugar de la media.

Agrupamiento secuencial. Se trata de un caso particular del *K-means* en el que el número de agrupamientos a encontrar, *K*, se trata como un valor máximo. Es un algoritmo incremental que procesa lotes individuales. Incorpora, además, una fase de evaluación que permite reducir el número de agrupamientos según tamaño y distancia al centroide.

Los parámetros principales son *K*, número de agrupamientos, *R*, umbral para crear agrupamientos, *C*, umbral para mezclar agrupamientos, *M*, longitud de un lote, *T*, umbral para eliminar agrupamientos (% de *M*).

En cada paso se procesa un punto nuevo y se evalúa su pertenencia o no al agrupamiento más cercano (si está a menos distancia del centro que la umbral, *R*). Así se van formando agrupamientos hasta que se detecta un elemento que está más alejado del umbral, en cuyo caso se crea un nuevo agrupamiento. En cada paso se recalcula el centroide del agrupamiento.

Esta iteración se lleva a cabo para cada uno de los miembros del lote, hasta que se alcanza el número máximo del lote, *M*. En ese momento se reevalúan los agrupamientos creados para esos puntos. Si el número es mayor que *K*, número de *cluster* máximo, se mezclan (en base a 3 criterios, cercanía, tamaño o forzada) y según el umbral definido, *C*, para obtener los *K* agrupamientos.

Al igual que el *K-means* este algoritmo es rápido, eficiente y flexible, al poder ajustarse muchos parámetros, pero tiene menor repetibilidad por el sesgo de orden de los patrones y existen algunos parámetros difíciles de establecer a priori.

Métodos jerárquicos:

Aglomerativo. Comienza situando cada objeto en su propio *cluster* y funciona de abajo hacia arriba. Realiza mezclas de dichos *clusters* atómicos en *clusters* cada vez mayores hasta que los objetos están en un único *cluster*. La mayoría de los métodos jerárquicos pertenecen a esta categoría y se diferencian en la medida de distancia o similitud que emplean.

Este tipo de *clustering* jerárquico se puede visualizar con un dendograma, como se muestra en la figura 14, donde cada nodo representa un subconjunto del conjunto de datos inicial y la altura de cada rama representa la distancia entre los objetos.

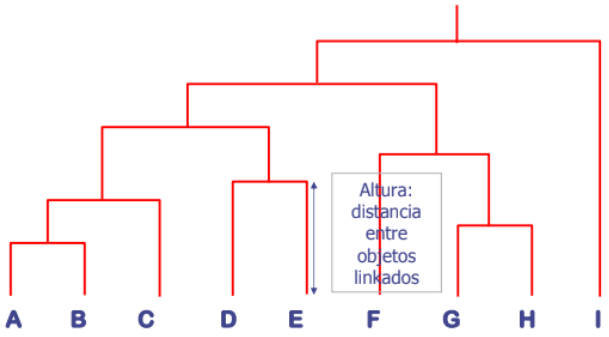

Figura 14. Dendograma de un algoritmo jerárquico aglomerativo. Fuente: UNED.

Divisivo. Se diferencia del aglomerativo en que comienza agrupando todos los objetos en el mismo *cluster* y los divide iterativamente hasta llegar a un número de particiones óptimo.

También existen otro tipo de métodos, como los basados en densidad o lo probabilísticos, que no se describen en esta sección.

En los métodos de *clustering* no se puede saber con antelación el número de clases que se acabarán obteniendo, por eso se utilizan métodos para indicar al algoritmo un número aproximado de clases objetivo en el que detenerse. Los más utilizados son los siguientes:

- El método del codo o *Elbow method*
- ISODATA o *Iterative Self-Organizing Data Analysis Technique*)
- DYNOC o *Dynamic Optimal Cluster-seek*
- MML o *Minimum Message Length*
- MLBG o *Modified Linde-Buzo-Gray*
- *Upper Tail method*
- SYNERACT

2.4. Medidas de precisión

A continuación, se describen algunas de las medidas de precisión más utilizadas para evaluar la bondad de los métodos descritos:

Exactitud [19], se define como el cociente entre el número de observaciones que el modelo predice correctamente (verdaderos positivos y negativos) y el número total de observaciones (verdaderos y falsos):

[19] *Accuracy* en inglés.

$$Exactitud = (VP + VN) / (VP + FP + FN + VN)$$

Donde,

VP = Verdaderos Positivos
VN = Verdaderos Negativos
FP = Falsos Positivos
FN = Falsos Negativos

Precisión es la relación entre las observaciones positivas que el modelo predice correctamente frente al total de falsos positivos más verdaderos positivos.

$$Precisión = VP / (VP + FP)$$

Exhaustividad[20], predice correctamente la cantidad de observaciones que es capaz de identificar el modelo. Viene dada por el siguiente cociente:

$$Exhaustividad = VP/(VP + FN)$$

F1 se define como la media armónica entre los indicadores *Exhaustividad* y *Precisión*

$$F1 = 2 * (Exhaustividad * Precisión) / (Exhaustividad + Precisión)$$

En los experimentos de este trabajo se empleó sobre todo la medida de *Exactitud*.

3. Metodología de desarrollo del trabajo

El trabajo se ha planteado en tres partes claramente diferenciadas (véase figura 15): preprocesado de los datos, selección de características y elección del algoritmo de clasificación.

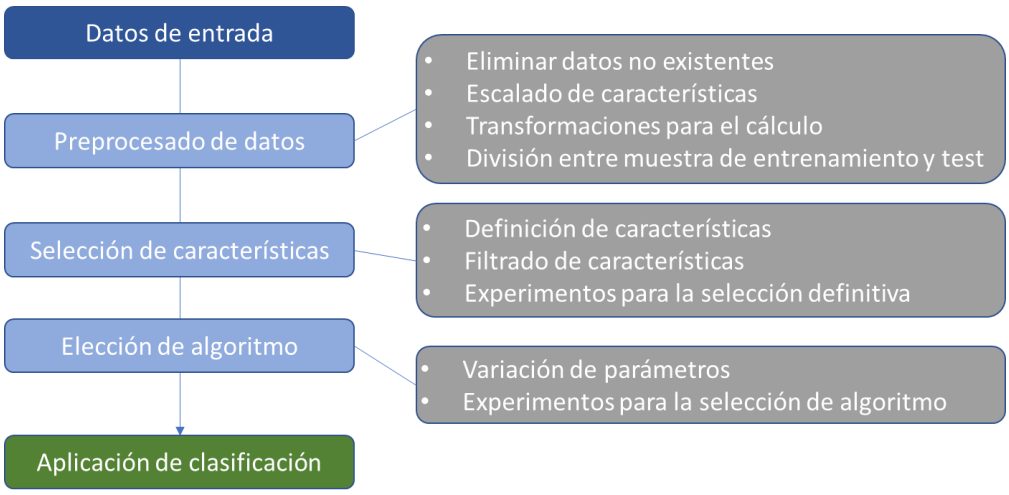

Figura 15. Diagrama esquemático del trabajo realizado.

[20] *Recall* en inglés.

Las aplicaciones basadas en aprendizaje automático requieren de un análisis previo para entender los datos y de un preprocesado de los mismos para garantizar un buen funcionamiento de los algoritmos.

El preprocesado de datos incluye la eliminación de valores no disponibles, el escalado de características y transformaciones del conjunto de datos para poder procesarlos de forma adecuada, así como una división apropiada en muestra de *entrenamiento* y de *test*, tal y como se describió en el estado del arte.

La segunda parte del análisis es la selección de características. A partir de los estudios descritos en el estado del arte, se definieron una serie de características y después se realizó una selección de las más importantes a través del método de filtrado descrito por Yu [2020]. Por último, se realizó una tanda de experimentos para comparar el impacto de cada una. Dada la relevancia de esta parte del trabajo, después de la selección se incluyó también un análisis de importancia y redundancia de las características.

En tercer lugar, se realizó un análisis del desempeño de varios algoritmos de clasificación supervisada para elegir el método más preciso a aplicar al conjunto de características finalmente seleccionadas. Para ello se realizó una segunda tanda de experimentos en la que se probaron distintas configuraciones de los parámetros y se compararon los resultados.

Una vez definida la configuración del algoritmo se dio por terminada la aplicación de clasificación.

3.1. Entorno de desarrollo utilizado

Para este proyecto se emplearon varios *notebooks* en un entorno web Jupyter sobre Python 3, tal como se detalla en la figura 16.

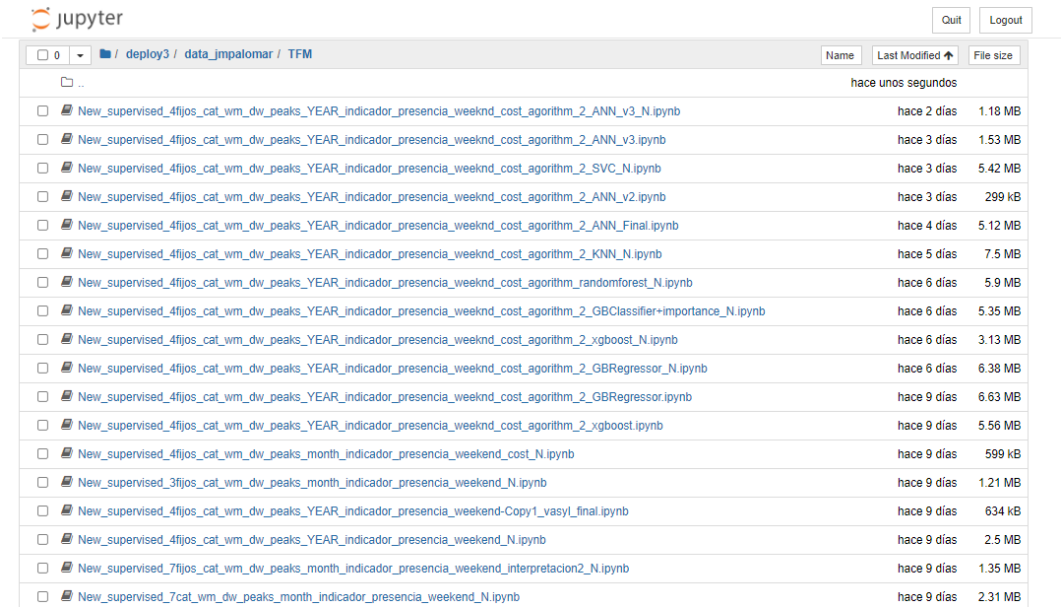

Figura 16. Repositorio de notebooks utilizado durante el análisis.

Se organizaron los datos en carpetas y se codificaron *notebooks* auxiliares con diferentes métodos, como se muestra en la Figura 17.

Figura 17. Carpetas de datos y métodos utilizadas.

En el Anexo se incluye una descripción más detallada de los principales *notebooks* y fragmentos de código empleados.

3.2. Preprocesado de datos

El resultado del análisis dependerá de la calidad de los datos, algo que, como señala Han [2012], sucede si se satisfacen ciertos requerimientos, como consistencia o interpretabilidad.

Han [2012] especifica que hay varias técnicas de preprocesado, como la limpieza, que permite eliminar datos extraños o ruidosos, repoblar datos ausentes o resolver inconsistencias, la transformación, que comprende procedimientos como la discretización o normalización, o la reducción de dimensionalidad, que permite obtener una representación comprimida o más específica de los datos originales sin perder información relevante.

Al estudiar las respuestas del cuestionario (véase figura 18), el uso de lavadora y lavavajillas más frecuente aparece en la franja 0, es decir, sin un horario fijo. Para la lavadora va seguida de la franja 3, entre las 10 y las 13 horas, y para el lavavajillas, va seguida de la 7, entre las 23 y las 2 horas, contando que muchas de las respuestas no estaban disponibles, ND, debido a que muchos usuarios no disponen de este último electrodoméstico.

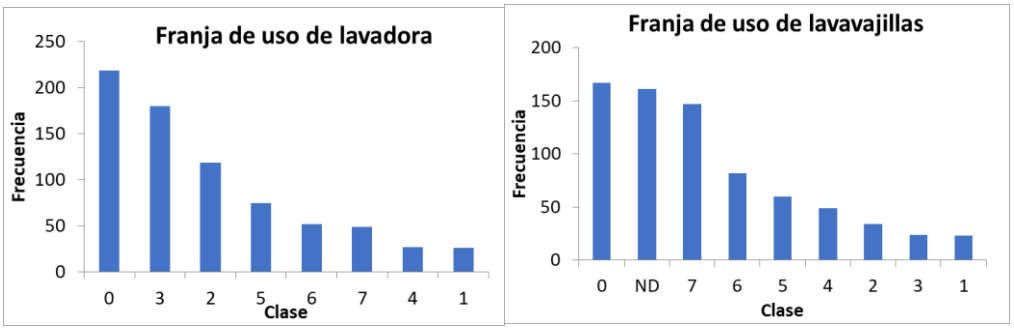

Figura 18. Frecuencia de respuestas a la franja de uso de cada electrodoméstico en el cuestionario.

Por otra parte, la figura 19 recoge las respuestas más frecuentes para el número de encendidos, en la lavadora, de 2 a 3 veces por semana y en el lavavajillas, la mayoría alega no tener ese equipo o no responde.

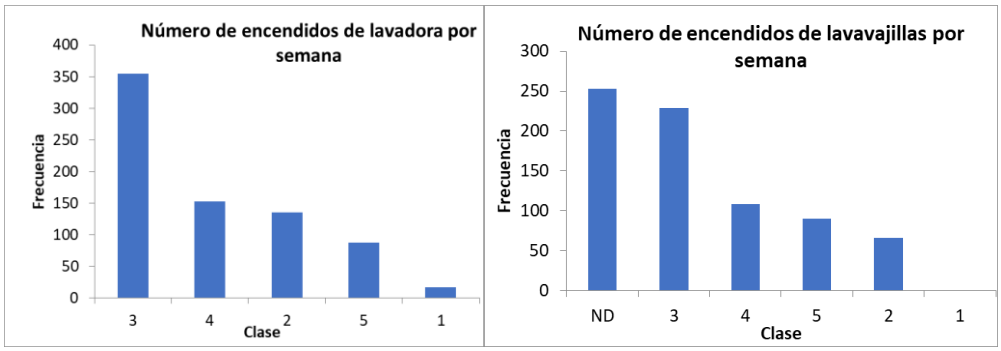

Figura 19. Frecuencia de respuestas al número de encendidos semanales de cada electrodoméstico en el cuestionario.

En el preprocesado se eliminaron, por tanto, datos no disponibles o extraños y se ordenaron las características deseadas en forma de un *dataframe* de calidad y manejable para la aplicación de los métodos y para garantizar una solución fiable. A continuación, se describen los pasos seguidos en el preprocesado:

1. **Transformaciones de los datos originales.** Los datos originales se encontraban en un formato diferente, separados en dos fuentes, base de datos de consumo y cuestionario. En la base de datos de consumo, cada línea correspondía a un día concreto para cada usuario y cada columna a la hora concreta del día. Por eso se agruparon los datos de consumo en franjas

horarias y se unificaron con los datos del cuestionario, en el que se encontraban las etiquetas. En el Anexo se describe esta transformación en más detalle.

2. **Limpieza de datos ausentes.** Se eliminaron algunos usuarios que no habían respondido a las preguntas solicitadas. En algunos casos, sobre todo cuando se trata de datos cuantitativos, es posible repoblar los datos ausentes en una característica a partir de datos del resto de los individuos para esa misma característica, mediante ratios o promedios, sin embargo, en este caso, al ser cualitativos esto resultaba más complicado, y se decidió eliminar los individuos que habían dejado vacío el campo. También se eliminaron datos de consumo que no habían sido recogidos, aunque estos fueron muchos menos.

3. **Transformaciones para el cálculo.** Se realizaron varias transformaciones para conseguir una matriz de datos por columnas de las características elegidas, en la que cada fila corresponda a todas las características asociadas a un mismo usuario. En el Anexo se describe esta transformación en más detalle.

4. **Escalado de características[21].** Es recomendable que las características tengan la misma escala y sean así comparables, antes de aplicar el algoritmo de clasificación elegido. Para ello se utilizó el escalado de características mediante la clase *StandardScaler* incluida en el paquete sklearn.preprocessing:

   class `sklearn.preprocessing.`**`StandardScaler`**(*, *copy=True*, *with_mean=True*, *with_std=True*)

5. **División entre muestra de *entrenamiento* y muestra de *test*.** Como se indicó anteriormente, esta división es necesaria para la ejecución del algoritmo. Hay que tener en cuenta que es bueno tratar de mantener la proporción de cada clase en dichos conjuntos de *entrenamiento* y de *test*.

   Para ello se ha empleado la función de validación cruzada estratificada, *StatifiedKFold* de Python:

   class `sklearn.model_selection.`**`StratifiedKFold`**(*n_splits=5*, *, *shuffle=False*, *random_state=None*)

   Es una variación de la validación cruzada *k-fold*, que utiliza grupos estratificados, de forma que cada conjunto contiene aproximadamente el

[21] *Feature Scaling* en inglés.

mismo porcentaje de muestras de cada clase que el conjunto completo, como se muestra en la figura 20.

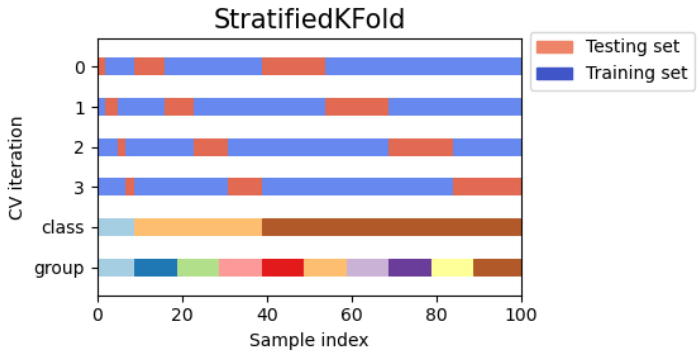

Figura 20. Ilustración de la validación cruzada estratificada.

Para implementarlo se ha aplicado el argumento *stratify = y* dentro del método *train_test_split*

`sklearn.model_selection.`**`train_test_split`**`(`*arrays, test_size=None, train_size=None, random_state=None, shuffle=True, stratify=None*`)`

Este procedimiento permite mantener las proporciones de cada clase tanto en la muestra de *entrenamiento* como en la de *test* de forma que sean similares a las de la muestra original.

3.3. Metodología de selección de características

La selección de características no es un paso trivial, sino que requiere del estudio de los datos y de análisis con anticipación al resultado del problema.

En este paso se llevaron a cabo las siguientes acciones:

1. Exploración visual de los datos.
2. Definición de características.
3. Filtrado de características.
4. Selección de un algoritmo de clasificación fijo para comparar precisión de las características elegidas.
5. Análisis comparativo de la precisión de características a partir de una serie de experimentos.
6. Análisis de importancia y redundancia de características.

3.4. Exploración visual de los datos

Para definir las características más apropiadas, se realizó una exploración visual de los datos.

Dado que uno de los objetivos era predecir la franja horaria de consumo habitual, se estudió el perfil de 6 usuarios que indicaron respuestas diferentes en el campo “hora de encendido habitual de lavadora”.

En la figura 21 se presenta un gráfico con el consumo para cada uno de estos 6 usuarios agrupado por las 8 franjas horarias indicadas en el cuestionario para varias semanas (cada semana está representada por una línea), indicando la franja que el usuario respondió en el cuestionario mediante un recuadro. Las franjas corresponden a estas 8 categorías posibles para la respuesta (7 franjas horarias y un caso adicional sin un horario fijo):

- 'No tengo un horario fijo' (0)
- 'Entre las 2 y las 7' (1)
- 'Entre las 7 y las 10' (2)
- 'Entre las 10 y las 13' (3)
- 'Entre las 13 y las 16’ (4)
- 'Entre las 16 y las 20' (5)
- 'Entre las 20 y las 23' (6)
- 'Entre las 23 y las 2’ (7)

Para una exploración más completa se calcularon y graficaron en cada franja las métricas Suma, desviación típica (STD), media (mean) y máximo (max).

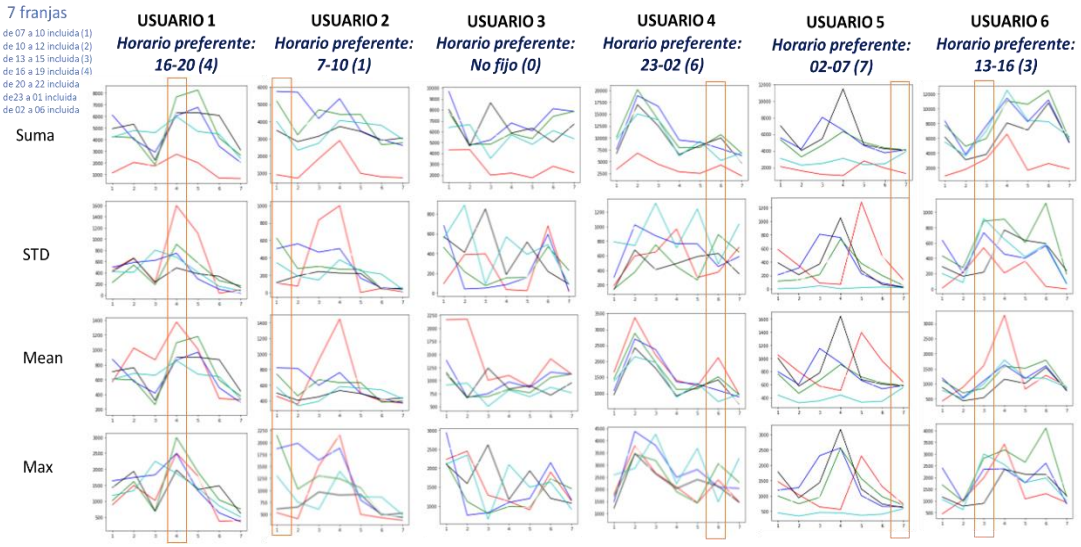

Figura 21. Ejemplo de consumo de 6 usuarios durante las 7 franjas que marca el cuestionario, y recuadrado, horario preferente de consumo de lavadora.

La exploración visual de estos usuarios parece indicar que podría haber cierta relación, aunque no siempre directa, entre los picos de consumo y el horario indicado, por

ejemplo, para la lavadora, lo que llevó a considerar la utilización de características relacionadas con los picos de consumo, tal y como se verá en la siguiente sección.

Teniendo en cuenta los trabajos de Yu [2020] y Sondenkampf [2014], y a la vista de los datos graficados, se definieron además de las basadas en picos, características asociadas al consumo total y variable, así como a estadísticos calculados sobre el consumo.

También se calculó, a partir del código postal, disponible como dato, una estimación de la renta promedio de cada región, a partir de datos disponibles en el Instituto Nacional de Estadística (INE), como se muestra en la tabla 3.

Comunidad Autónoma	CP	Provincia	CCAA	RENTA MEDIA
Andalucía	4	Almería	Andalucía	23826
Andalucía	11	Cádiz	Andalucía	23826
Andalucía	14	Córdoba	Andalucía	23826
Andalucía	18	Granada	Andalucía	23826
Andalucía	21	Huelva	Andalucía	23826
Andalucía	23	Jaén	Andalucía	23826
Andalucía	29	Málaga	Andalucía	23826
Andalucía	41	Sevilla	Andalucía	23826
Aragón	22	Huesca	Aragón	29672
Aragón	44	Teruel	Aragón	29672
Aragón	50	Zaragoza	Aragón	29672
Asturias, Principado de	33	Asturias	Asturias, Principado de	27897
Balears, Illes	7	Balears, Illes	Balears, Illes	32179
Canarias	35	Palmas, Las	Canarias	24565
Canarias	38	Santa Cruz de Tenerife	Canarias	24565
Cantabria	39	Cantabria	Cantabria	29144
Castilla y León	5	Ávila	Castilla y León	27679
Castilla y León	9	Burgos	Castilla y León	27679
Castilla y León	24	León	Castilla y León	27679
Castilla y León	34	Palencia	Castilla y León	27679
Castilla y León	37	Salamanca	Castilla y León	27679
Castilla y León	40	Segovia	Castilla y León	27679
Castilla y León	42	Soria	Castilla y León	27679
Castilla y León	47	Valladolid	Castilla y León	27679
Castilla y León	49	Zamora	Castilla y León	27679
Castilla-La Mancha	2	Albacete	Castilla-La Mancha	24853
Castilla-La Mancha	13	Ciudad Real	Castilla-La Mancha	24853
Castilla-La Mancha	16	Cuenca	Castilla-La Mancha	24853
Castilla-La Mancha	19	Guadalajara	Castilla-La Mancha	24853
Castilla-La Mancha	45	Toledo	Castilla-La Mancha	24853
Cataluña	8	Barcelona	Cataluña	33321
Cataluña	17	Girona	Cataluña	33321
Cataluña	25	Lleida	Cataluña	33321
Cataluña	43	Tarragona	Cataluña	33321
Comunitat Valenciana	3	Alicante/Alacant	Comunitat Valenciana	26176
Comunitat Valenciana	12	Castellón/Castelló	Comunitat Valenciana	26176
Comunitat Valenciana	46	Valencia/València	Comunitat Valenciana	26176
Extremadura	6	Badajoz	Extremadura	21611
Extremadura	10	Cáceres	Extremadura	21611
Galicia	15	Coruña, A	Galicia	27541
Galicia	27	Lugo	Galicia	27541
Galicia	32	Ourense	Galicia	27541
Galicia	36	Pontevedra	Galicia	27541
Madrid, Comunidad de	28	Madrid	Madrid, Comunidad de	35587
Murcia, Región de	30	Murcia	Murcia, Región de	24448
Navarra, Comunidad Foral de	31	Navarra	Navarra, Comunidad Foral de	34612
País Vasco	1	Araba/Álava	País Vasco	36399
País Vasco	48	Bizkaia	País Vasco	36399
País Vasco	20	Gipuzkoa	País Vasco	36399
Rioja, La	26	Rioja, La	Rioja, La	30162
Ceuta	51	Ceuta	Ceuta	30432
Melilla	52	Melilla	Melilla	37552

Tabla 3. Renta promedio obtenida a partir del código postal, según datos disponibles en el INE.

3.5. Definición de características

Según lo especificado en la sección anterior se definieron las siguientes características:

Características asociadas a consumo. Para cada día se calculó el consumo de cada franja como la suma de los consumos en las horas de esa franja. Pueden abarcar un período de tiempo variable (semana, mes o año completo). En la tabla 4 se muestra el nombre asignado a cada franja para identificar las características.

Franjas	Nombre de característica
'Entre las 2 y las 7' (1)	madrugada
'Entre las 7 y las 10' (2)	amanecer
'Entre las 10 y las 13' (3)	mañana
'Entre las 13 y las 16' (4)	mediodía
'Entre las 16 y las 20' (5)	tarde
'Entre las 20 y las 23' (6)	anochecer
'Entre las 23 y las 2' (7)	noche

Tabla 4. Nombre de características asignado a cada franja.

Después se estimaron las características: media, desviación típica, suma, máximo, mínimo y total, en total 6 para cada franja de consumo. Al inicio se consideraron 7 franjas más la franja sin horario fijo (0), lo que se tradujo en 48 características. De esta forma las características asociadas a consumo, asumiendo un análisis mensual, quedarían por ejemplo para el mes 5 de la siguiente forma:

'mean_amanecer_m5', 'mean_mañana_m5', 'mean_mediodía_m5', 'mean_tarde_m5', 'mean_anochecer_m5', 'mean_noche_m5', 'mean_madrugada_m5', 'mean_total_m5', 'sd_amanecer_m5', 'sd_mañana_m5', 'sd_mediodía_m5', 'sd_tarde_m5', 'sd_anochecer_m5', 'sd_noche_m5', 'sd_madrugada_m5', 'sd_total_m5', 'max_amanecer_m5', 'max_mañana_m5', 'max_mediodía_m5', 'max_tarde_m5', 'max_anochecer_m5', 'max_noche_m5', 'max_madrugada_m5', 'max_total_m5', 'min_amanecer_m5', 'min_mañana_m5', 'min_mediodía_m5', 'min_tarde_m5', 'min_anochecer_m5', 'min_noche_m5', 'min_madrugada_m5', 'min_total_m5', 'sum_amanecer_m5', 'sum_mañana_m5', 'sum_mediodía_m5', 'sum_tarde_m5', 'sum_anochecer_m5', 'sum_noche_m5', 'sum_madrugada_m5', 'sum_total_m5'

- **Características asociadas a franja pico de consumo o franjas adyacentes**, en este caso, el número de la franja pico, y el número de las dos franjas adyacentes al pico, en total 3 características más por usuario. Por ejemplo, en la tabla 5 se muestran estas 3 características para el mes 5 en un análisis mensual.

Nombre de característica
'peak_m5',
'adj_peak1_m5',
'adj_peak2_m5'

Tabla 5. Ejemplo de características asociadas a picos de consumo.

- **Características asociadas a los datos enriquecidos por la encuesta**, como el nivel de renta, (tabla 6), que es 1 característica más por usuario.

Nombre de característica
'RENTA'

Tabla 6. Característica asociada a renta.

Esto resultó en un número inicial de 52 características.

En cuanto a la frecuencia del análisis, el algoritmo se aplicó inicialmente en base semanal, es decir, estimando un modelo para cada semana. Pese a no ser un requisito obligado, se consideró que facilitaría la versatilidad de la aplicación a los clientes, ya que se podría aplicar desde la primera semana de la que se dispongan datos para nuevos clientes. Posteriormente, como se explicará, también se utilizó un análisis mensual y otro anual de los datos.

3.6. Filtrado de características

Una vez elegido un conjunto de características, una buena práctica, tal como se indicó en la sección 2, es determinar aquellas de mayor peso y por tanto mayor importancia en el desempeño de la clasificación.

Como se indicó en la sección 2.2 en este caso se usó un filtro mRMR. Se empleó el método de Python *mrmr_regression* incluido dentro del paquete *mrmr*.

Al aplicar este filtro se obtuvieron las características más relevantes para el análisis, que se denominaron características clave y en los desarrollos del código *key_features*[22]. El número de características filtradas también fue una variable a considerar en los experimentos posteriores.

Este filtro se implementa de forma sencilla en Python:

```
def feature_filter_mi(X,y,numero):
    from mrmr import mrmr_regression
    selected_features = mrmr_regression(X, y, K = numero)
    return selected_features
```

En la tabla 7 se muestra un ejemplo de características iniciales y características filtradas en un análisis mensual, para el mes 5, en una de las rondas de experimentos.

[22] Características clave.

Características iniciales	Características filtradas
features_wm = ['RENTA', 'mean_amanecer_m5', 'mean_mañana_m5', 'mean_mediodía_m5', 'mean_tarde_m5', 'mean_anochecer_m5', 'mean_noche_m5', 'mean_madrugada_m5', 'mean_total_m5', 'sd_amanecer_m5', 'sd_mañana_m5', 'sd_mediodía_m5', 'sd_tarde_m5', 'sd_anochecer_m5', 'sd_noche_m5', 'sd_madrugada_m5', 'sd_total_m5', 'max_amanecer_m5', 'max_mañana_m5', 'max_mediodía_m5', 'max_tarde_m5', 'max_anochecer_m5', 'max_noche_m5', 'max_madrugada_m5', 'max_total_m5', 'min_amanecer_m5', 'min_mañana_m5', 'min_mediodía_m5', 'min_tarde_m5', 'min_anochecer_m5', 'min_noche_m5', 'min_madrugada_m5', 'min_total_m5', 'sum_amanecer_m5', 'sum_mañana_m5', 'sum_mediodía_m5', 'sum_tarde_m5', 'sum_anochecer_m5', 'sum_noche_m5', 'sum_madrugada_m5', 'sum_total_m5', 'peak_m5', 'adj_peak1_m5', 'adj_peak2_m5', 'p_amanecer_m5', 'p_mañana_m5', 'p_mediodía_m5', 'p_tarde_m5', 'p_anochecer_m5', 'p_noche_m5', 'p_madrugada_m5', 'ratio_jornada_m5', 'ratio_weekend_m5', 'presencia', 'HORA_FIJA_WM', 'HORA_LAVADORA']	*selected_features_wm ['sd_mañana_m5', 'p_tarde_m5', 'RENTA', 'peak_m5', 'min_mañana_m5', 'sd_mediodía_m5', 'min_amanecer_m5', 'sd_amanecer_m5', 'adj_peak1_m5', 'max_mediodía_m5', 'min_anochecer_m5', 'adj_peak2_m5', 'max_madrugada_m5', 'p_mediodía_m5', 'max_mañana_m5', 'mean_mediodía_m5', 'sum_mediodía_m5', 'max_noche_m5', 'min_noche_m5', 'p_mañana_m5', 'min_tarde_m5', 'max_amanecer_m5', 'sd_madrugada_m5', 'sd_noche_m5', 'mean_noche_m5', 'sum_noche_m5', 'sd_total_m5', 'sd_anochecer_m5', 'sum_amanecer_m5', 'mean_amanecer_m5']*

Tabla 7. Ejemplo de características antes y después del filtrado con mRMR.

3.7. Diseño de experimentos para la configuración definitiva del *dataframe*

Después de la selección y el filtrado de las características y tomando como punto de partida las definidas en el epígrafe 3.5, se realizaron experimentos para estudiar el impacto de estas y de otras características, así como de otros factores, como el número de características a filtrar o las agrupaciones temporales de los datos, con la intención de configurar un *dataframe* lo más adecuado posible antes de abordar la selección del método de clasificación.

El procedimiento consistió en variar las características existentes, incluir otras nuevas o aplicar transformaciones a los datos para un mejor procesado y comprobar las variaciones que producían en la precisión.

Se siguió un proceso iterativo acumulativo respecto a cada caso previo. Si bien no se han realizado con un orden particular, se considera que puede ser una forma válida y práctica. Se entendió que más tandas de experimentos en secuencias de órdenes diferentes podría alterar los resultados, pero no sustancialmente para el objetivo del problema.

A continuación, se describe las hipótesis para esta primera tanda de experimentos:

Se consideraron 7 franjas de consumo diario.

Se partió de un análisis semanal considerando las métricas para los consumos de cada franja sólo de los días contenidos a cada semana para cada usuario. Así, el algoritmo se ejecuta y genera un clasificador para cada semana.

Se empleó el filtro de máxima relevancia y mínima redundancia y se varió en algún experimento el número de características de salida del filtro.

Se empleó un único algoritmo como punto de partida para la comparativa de los experimentos, *XGBoost*, debido a su versatilidad y precisión y al uso habitual en el sector, aunque esta decisión pudiera verse alterada en la segunda parte del análisis. Los hiperparámetros seleccionados para el algoritmo se muestran a continuación (la mayoría tomados por defecto del algoritmo y otros seleccionados):

- Tanto por uno de datos para test= 0.15,
- Función objetivo = MSE[23],
- Métrica de evaluación = regresión logística multiobjetivo (por defecto para clasificación).
- Número de estimadores = 200,
- Profundidad de árbol = 10.

Se eligió un 15% como porcentaje de datos de *test*, un número de estimadores de 200 y profundidad de árbol de 10 al considerarlas cifras razonables en este tipo de estudios, y dado que en unas primeras pruebas parecían arrojar resultados aceptables.

Se emplearon las técnicas descritas de validación cruzada estratificada utilizando *StratifiedKFold*, como se indicó en la sección 3.2.

Se analizaron las cifras de la lavadora y el lavavajillas, utilizando dos indicadores de precisión:

1. Por una parte, la medida de precisión general de algoritmo, basada en la medida de *Exactitud*, presentada en la sección 2.4 y estimada como como el porcentaje del número de aciertos exactos frente al número total de estimaciones, que se implementó mediante el método.

   `sklearn.metrics.accuracy_score(y_true, y_pred, *, normalize=True, sample_weight=None)`

[23] *MSE: Mean Squared Error.*

- En este cálculo se utilizaron técnicas de validación cruzada utilizando el método *cross_val_score*:

  `sklearn.model_selection.cross_val_score(estimator, X, y=None, *, groups=None, scoring=None, cv=None, n_jobs=None, verbose=0, fit_params=None, pre_dispatch='2*n_jobs', error_score=nan)`

- Se tuvo en cuenta un parámetro de 10 validaciones cruzadas para recoger una posible variabilidad en los resultados.

2. Por otra parte, se tomó como ejemplo de referencia un usuario concreto con consumos semanales bien delimitados referido en la sección 3.4, al que se denominó Usuario Objetivo, y para el que se conocían las respuestas del cuestionario. La franja señalada como de consumo habitual de lavadora y de lavavajillas, ambas entre las 16 horas y las 20horas, así como el número de encendidos semanales, en ambos casos, 1 por semana. En la figura 22 se muestran los consumos totales de este usuario para el mes 5, y se observa que la franja de uso coincide con el pico de consumo general.

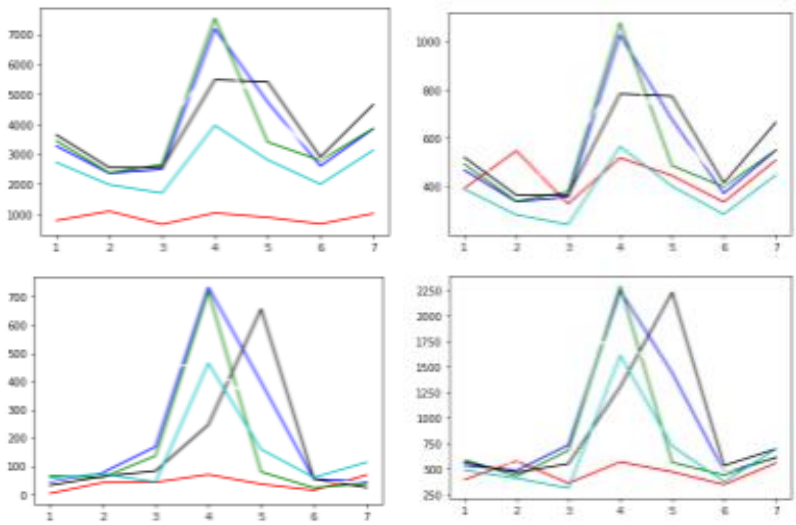

Figura 22. Franjas de consumo para las primeras 5 semanas del mes 5 (mayo), para el Usuario Objetivo, que representan 1)Suma, 2)mean, 3)STD y 4)max.

Según la categorización realizada, la franja preferente del Usuario Objetivo se corresponde con la franja a la que se asignó el número 5, debido a la inclusión como franja 0 de la opción 'No tengo un horario fijo':

- 'No tengo un horario fijo' (0)
- 'Entre las 2 y las 7' (1)
- 'Entre las 7 y las 10' (2)
- 'Entre las 10 y las 13' (3)
- 'Entre las 13 y las 16' (4)

- **'Entre las 16 y las 20' (5)**
- 'Entre las 20 y las 23' (6)
- 'Entre las 23 y las 2' (7)

Respecto al número de encendidos, el número de categoría sería la 2 para ambos electrodomésticos:

- 'Menos de 1 vez por semana' (1)
- **'1 vez por semana' (2)**
- '2 o 3 veces por semana' (3)
- '4 o 5 veces por semana' (4)
- 'Más de 5 veces por semana' (5)

Con todo ello, se realizaron los experimentos que se enumeran en la siguiente lista y cuyos resultados se recogen en el epígrafe 4.1:

1. Número de características clave seleccionadas
2. Análisis semanal frente a análisis mensual
3. Características adicionales incluidas para recoger picos de consumo
4. Características adicionales incluidas para recoger presencia en hogar
5. Característica adicional incluida para recoger consumo del fin de semana
6. Eliminación de la franja 0, 'No tengo un horario fijo'
7. Identificación de una posible mala interpretación de las franjas por parte del usuario
8. Reducción del número de franjas horarias
9. Característica adicional incluida para recoger impacto de coste de la energía

3.8. Diseño de experimentos para selección del algoritmo de clasificación

Han [2012] señala que la calidad del resultado depende también del algoritmo empleado y de la configuración de sus parámetros (que en ocasiones son muchos, como el umbral de distancia, el tipo de modelo o el número de árboles).

Una vez configuradas las características del *dataframe*, se realizó un análisis para elegir el algoritmo de clasificación más eficaz, incluyendo la configuración de sus parámetros, para así alcanzar la mayor precisión de clasificación.

Según se describió en la sección 2.3, algunos algoritmos usados por su alta precisión para la clasificación de variables asociadas a consumo son, por ejemplo, *Random Forest* o métodos basados en *Support Vector Machines* [Hedén, 2016], aunque existen otro tipo de modelos.

El presente trabajó probó otros algoritmos además de los mencionados debido a su uso extendido en la industria, como el *K-Nearest Neighbors*, el *Gradient Boosting* o las redes neuronales.

Cada algoritmo consta de una serie de hiperparámetros, que pueden configurarse de forma manual o mediante métodos de selección de óptimos. Para una primera tanda de análisis en la predicción de horario de uso de electrodomésticos se configuraron de forma manual. Posteriormente, para la predicción del número de encendidos semanales, se empleó un método automatizado como el *cross-validated grid-search* [Hedén, 2016].

En coherencia con el estado del arte expuesto en la sección 2.3, para esta tanda de experimentos se emplearon los siguientes algoritmos:

1. *Extreme Gradient Boosting Classification (XGBoost)*
2. *Gradient Boosting Classification*
3. *Random Forest*
4. *Support Vector Machines*
5. *K Nearest Neighbor (KNN)*
6. *Artificial Neural Network (ANN)*

Como se puede observar, el primero de ellos corresponde al algoritmo de *XGBoost*. La comparativa de todas las cifras de precisión de esta tanda de experimentos se refiere a un Caso Base, que consideraba *XGBoost* como algoritmo de clasificación, tal y como se usó en la tanda anterior de experimentos.

Estos casos se compararon tomando como indicador de precisión el promedio de cinco ejecuciones en cada uno, con el objetivo de recoger la posible variabilidad entre distintas ejecuciones sin comprometer los tiempos de cálculo.

4. Resultados

En esta sección se exponen los resultados de los experimentos explicados en la sección 3.7 para la configuración definitiva del *dataframe* y también se muestran los resultados obtenidos en los experimentos presentados en la sección 3.8 para la selección del algoritmo de clasificación.

Adicionalmente, antes de los resultados de la segunda tanda de experimentos, se incluye una revisión de la importancia de las características definitivas utilizando varios indicadores de heterogeneidad y redundancia.

4.1. Experimentos para la configuración definitiva del *dataframe*.

A continuación, se describen los resultados obtenidos para cada uno de los experimentos de la primera tanda.

1. Número de características filtradas

En este primer experimento se evaluó el número de características clave que establecer como salida del filtro.

Se especificaron inicialmente 10 características clave o *key features* y después se aumentó el número a 20, 30 y 40, realizando una evaluación en base semanal para las primeras 5 semanas.

Como se muestra en la tabla 8, la precisión mejora ligeramente con el incremento de características clave al subir a 20, o a 30 pero siempre dentro de un orden de magnitud similar, además el aumento progresivo de características requiere mayor tiempo de cálculo.

LAVADORA			LAVAVAJILLAS		
10 key features	Predicción Franja horaria		*10 key features*	Predicción Franja horaria	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	23%	3	SEMANA1	25%	0
SEMANA2	27%	3	SEMANA2	26%	5
SEMANA3	28%	0	SEMANA3	23%	6
SEMANA4	20%	2	SEMANA4	26%	0
SEMANA5	17%	5	SEMANA5	21%	6
PROMEDIO	**22,9%**		**PROMEDIO**	**24,2%**	
20 key features	Predicción Franja horaria		*20 key features*	Predicción Franja horaria	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	22%	0	SEMANA1	24%	0
SEMANA2	27%	0	SEMANA2	30%	6
SEMANA3	31%	0	SEMANA3	24%	0
SEMANA4	19%	3	SEMANA4	24%	0
SEMANA5	20%	6	SEMANA5	24%	0
PROMEDIO	**23,9%**		**PROMEDIO**	**25,1%**	
30 key features	Predicción Franja horaria		*30 key features*	Predicción Franja horaria	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	19%	7	SEMANA1	26%	0
SEMANA2	28%	0	SEMANA2	24%	6
SEMANA3	25%	2	SEMANA3	26%	0
SEMANA4	29%	2	SEMANA4	25%	0
SEMANA5	25%	2	SEMANA5	32%	0
PROMEDIO	**25,0%**		**PROMEDIO**	**26,7%**	
40 key features	Predicción Franja horaria		*40 key features*	Predicción Franja horaria	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	21%	5	SEMANA1	26%	0
SEMANA2	17%	5	SEMANA2	20%	6
SEMANA3	27%	5	SEMANA3	25%	0
SEMANA4	24%	3	SEMANA4	28%	0
SEMANA5	25%	0	SEMANA5	25%	0
PROMEDIO	**22,8%**		**PROMEDIO**	**24,9%**	

Tabla 8. Precisión en la predicción de la franja de uso según el número de características clave seleccionadas por el filtro en lavadora y lavavajillas (experimento1).

En la tabla 9, se muestra el efecto de estas variaciones para la clasificación en el número de encendidos, que arroja en torno al 40%, que incrementan ligeramente también al aumentar el número de características.

LAVADORA			LAVAVAJILLAS		
10 key features	Predicción encendidos/semana		*10 key features*	Predicción encendidos/semana	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	40%	3	SEMANA1	43%	3
SEMANA2	44%	3	SEMANA2	34%	3
SEMANA3	45%	2	SEMANA3	35%	4
SEMANA4	38%	3	SEMANA4	42%	3
SEMANA5	32%	4	SEMANA5	41%	3
PROMEDIO	**39,6%**		**PROMEDIO**	**38,9%**	
20 key features	Predicción encendidos/semana		*20 key features*	Predicción encendidos/semana	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	42%	4	SEMANA1	43%	3
SEMANA2	42%	3	SEMANA2	41%	3
SEMANA3	41%	3	SEMANA3	37%	3
SEMANA4	44%	2	SEMANA4	47%	2
SEMANA5	41%	4	SEMANA5	38%	3
PROMEDIO	**42,0%**		**PROMEDIO**	**41,1%**	
30 key features	Predicción encendidos/semana		*30 key features*	Predicción encendidos/semana	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	38%	2	SEMANA1	36%	3
SEMANA2	39%	3	SEMANA2	35%	2
SEMANA3	41%	2	SEMANA3	39%	2
SEMANA4	48%	2	SEMANA4	43%	3
SEMANA5	36%	3	SEMANA5	37%	2
PROMEDIO	**40,4%**		**PROMEDIO**	**38,1%**	
40 key features	Predicción encendidos/semana		*40 key features*	Predicción encendidos/semana	
SEMANA	Precisión General	Precisión Usuario objetivo	SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	43%	2	SEMANA1	45%	3
SEMANA2	42%	3	SEMANA2	41%	2
SEMANA3	40%	2	SEMANA3	50%	2
SEMANA4	39%	3	SEMANA4	46%	3
SEMANA5	39%	3	SEMANA5	42%	2
PROMEDIO	**40,6%**		**PROMEDIO**	**44,6%**	

Tabla 9. Precisión en la predicción del número de encendidos según el número de características clave seleccionadas por el filtro en lavadora y lavavajillas (experimento1).

Después de este experimento, se optó por mantener 30 características como compromiso entre precisión y velocidad de cómputo.

2. Análisis semanal frente a análisis mensual

En este experimento se evaluó el modelo utilizando dos bases temporales del clasificador, inicialmente establecida por semana, se probó también con una base mensual.

En la tabla 10 se muestran los resultados de predicción de uso de lavadora, para un análisis semana a semana, desde mayo a noviembre. En promedio, las precisiones están en torno a 23-29%. Sin embargo, en algunos casos esa precisión cae por debajo del 20%. Para el usuario objetivo, aunque es un experimento aislado, se predice una franja 0 con cierta frecuencia. Esto puede ser debido a que la cantidad de respuestas del cuestionario correspondientes a esa franja es elevada.

Analizando los resultados con una ejecución mes a mes los promedios incrementan la precisión de predicción, superan en algún caso el 30%, pero en términos generales el promedio total coincide para los dos escenarios analizados, en torno al 25%.

Lavadora, base semanal

30 key features

Predicción Franja horaria

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	26%	2
SEMANA2	20%	0
SEMANA3	31%	2
SEMANA4	25%	0
SEMANA5	26%	3
PROMEDIO	**25,7%**	

Predicción Franja horaria

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA15	32%	3
SEMANA16	26%	0
SEMANA17	29%	5
SEMANA18	30%	6
SEMANA19	26%	0
SEMANA20	26%	0
PROMEDIO	**28,3%**	

Predicción Franja horaria

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA6	25%	0
SEMANA7	18%	0
SEMANA8	26%	0
SEMANA9	28%	0
SEMANA10	22%	0
PROMEDIO	**23,9%**	

Predicción Franja horaria

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA21	23%	0
SEMANA22	17%	6
SEMANA23	25%	3
SEMANA24	29%	3
SEMANA25	25%	3
PROMEDIO	**23,7%**	

Predicción Franja horaria

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA11	22%	0
SEMANA12	25%	0
SEMANA13	25%	0
SEMANA14	22%	0
PROMEDIO	**23,7%**	

Lavadora, base mensual

30 key features

Mes	Precisión General	Precisión Usuario objetivo
MAYO	20%	5
JUNIO	28%	3
JULIO	30%	0
AGOSTO	21%	3
SEPTIEMBRE	29%	3
OCTUBRE	20%	5
NOVIEMBRE	33%	0
PROMEDIO	**25,6%**	

Tabla 10. Precisión en la predicción de la franja de uso de lavadora según análisis en una base semanal o mensual (experimento 2).

El comportamiento es similar en el caso de la predicción del número de encendidos, como se muestra en la tabla 11, donde el promedio para ambos escenarios es de alrededor de 42%.

Lavadora, base semanal

40 key features — Predicción encendidos/semana

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA1	44%	2
SEMANA2	42%	3
SEMANA3	38%	2
SEMANA4	40%	2
SEMANA5	46%	4
PROMEDIO	**42,0%**	

Predicción encendidos/semana

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA15	36%	3
SEMANA16	46%	3
SEMANA17	37%	2
SEMANA18	45%	2
SEMANA19	40%	3
SEMANA20	46%	2
PROMEDIO	**40,8%**	

Predicción encendidos/semana

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA6	41%	3
SEMANA7	46%	3
SEMANA8	37%	2
SEMANA9	42%	3
SEMANA10	39%	2
PROMEDIO	**41,0%**	

Predicción encendidos/semana

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA21	42%	3
SEMANA22	42%	3
SEMANA23	42%	3
SEMANA24	50%	3
SEMANA25	45%	3
PROMEDIO	**44,2%**	

Predicción encendidos/semana

SEMANA	Precisión General	Precisión Usuario objetivo
SEMANA11	33%	3
SEMANA12	42%	2
SEMANA13	48%	2
SEMANA14	38%	3
PROMEDIO	**40,4%**	

Lavadora, base mensual

40 key features

Mes	Precisión General	Precisión Usuario objetivo
MAYO	44%	3
JUNIO	44%	2
JULIO	39%	2
AGOSTO	38%	3
SEPTIEMBRE	45%	2
OCTUBRE	46%	3
NOVIEMBRE	39%	3
PROMEDIO	**42,2%**	

Tabla 11. Precisión en la predicción del número de encendidos de lavadora según análisis en una base semanal o mensual (experimento 2).

Debido a esta equivalencia, se optó por utilizar una agrupación mensual, por una cuestión de robustez y eficiencia, y para absorber también la ausencia de datos puntuales para algunas semanas.

3. Características adicionales incluidas para recoger picos de consumo

En este experimento se añadieron características de tipo indicador, que diesen información de la franja pico de consumo.

Como se trabajó con 7 franjas horarias, se incluyeron 7 características nuevas para cada mes, denominadas franjas picos (en abreviatura *p_* seguido del nombre de la franja específica), que tomasen valor 0 en caso de no ser una franja pico y valor 1 en caso de serlo. Por ejemplo, para el usuario de referencia, estas características tomarían los valores que se muestran en la tabla 12.

p_madrugada	p_amanecer	p_mañana	p_mediodia	**p_tarde**	p_anochecer	p_noche
0	0	0	0	**1**	0	0

Tabla 12. Ejemplo de característica adicional de tipo indicador para el Usuario Objetivo.

Estas características adicionales se mantuvieron fijas en el análisis (sin aplicarles el filtro mRMR) y se decidió mantener el número de características clave en 30 aunque también se probó incrementarlo hasta 37 para recoger las nuevas incorporaciones.

Los resultados no condujeron a una significativa precisión adicional, tal y como se puede ver en las tablas 13 y 14, aunque se mantuvieron las nuevas características para posteriores análisis, ya que, en el filtrado, el algoritmo desecharía las menos relevantes.

Lavadora

30 key features	Predicción Franja horaria	
Mes	**Precisión General**	**Precisión Usuario objetivo**
MAYO	29%	2
JUNIO	25%	2
JULIO	28%	0
AGOSTO	22%	3
SEPTIEMBRE	26%	5
OCTUBRE	27%	4
NOVIEMBRE	22%	0
PROMEDIO	**25,6%**	

37 key features	Predicción Franja horaria	
Mes	**Precisión General**	**Precisión Usuario objetivo**
MAYO	29%	5
JUNIO	34%	5
JULIO	25%	0
AGOSTO	22%	3
SEPTIEMBRE	30%	5
OCTUBRE	31%	4
NOVIEMBRE	29%	0
PROMEDIO	**28,6%**	

Lavavajillas

30 key features	Predicción Franja horaria	
Mes	**Precisión General**	**Precisión Usuario objetivo**
MAYO	26%	6
JUNIO	26%	5
JULIO	27%	0
AGOSTO	25%	6
SEPTIEMBRE	29%	6
OCTUBRE	31%	7
NOVIEMBRE	19%	0
PROMEDIO	**26,2%**	

37 key features	Predicción Franja horaria	
Mes	**Precisión General**	**Precisión Usuario objetivo**
MAYO	27%	6
JUNIO	16%	5
JULIO	25%	6
AGOSTO	25%	6
SEPTIEMBRE	29%	6
OCTUBRE	23%	5
NOVIEMBRE	22%	0
PROMEDIO	**23,9%**	

Tabla 13. Precisión en la predicción de la franja de uso para 30 y 37 características clave incluyendo una característica adicional de tipo indicador (experimento 3).

Lavadora		
37 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	44%	3
JUNIO	39%	2
JULIO	42%	2
AGOSTO	42%	3
SEPTIEMBRE	39%	3
OCTUBRE	47%	3
NOVIEMBRE	41%	3
PROMEDIO	**41,9%**	

Lavavajillas		
37 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	41%	2
JUNIO	38%	2
JULIO	42%	2
AGOSTO	41%	3
SEPTIEMBRE	45%	2
OCTUBRE	38%	4
NOVIEMBRE	32%	3
PROMEDIO	**39,5%**	

40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	50%	3
JUNIO	38%	2
JULIO	44%	2
AGOSTO	38%	3
SEPTIEMBRE	41%	2
OCTUBRE	38%	3
NOVIEMBRE	37%	3
PROMEDIO	**40,9%**	

40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	46%	3
JUNIO	32%	2
JULIO	38%	2
AGOSTO	39%	3
SEPTIEMBRE	44%	2
OCTUBRE	36%	2
NOVIEMBRE	40%	2
PROMEDIO	**39,3%**	

Tabla 14. Precisión en la predicción del número de encendidos para 30 y 37 características clave incluyendo una característica adicional de tipo indicador (experimento 3).

4. Características adicionales incluidas para recoger la presencia en hogar

En este experimento se planteó incluir características adicionales relacionadas con una posible presencia en el hogar con la intención de recoger el tiempo que el usuario pasa en casa; para ello se utilizaron ratios de consumo de la semana laboral.

Se añadió la característica '*ratio_jornada'* como posible indicador de teletrabajo. Se calculó como el consumo de la franja de 8 a 18 horas, correspondientes a la jornada laboral, frente al consumo de la franja 6 a 23 horas, durante los días laborables, de lunes a jueves (para evitar el fin de semana y el viernes, que en ocasiones incorpora la jornada intensiva).

Adicionalmente, se realizó un análisis de *clustering* para estimar una característica adicional como indicador binario de presencia, a partir de los datos de la característica *ratio_jornada*. Para ello se utilizó un algoritmo *K-means*, uno de los métodos más empleados [Nordahl, 2019]. La característica, llamada *presencia*, clasificó dos grupos de usuario: 1, presente en casa y 0, no presente en casa.

A continuación, se especifican los parámetros del algoritmo de *clustering* utilizado:

```
K-means (n_clusters = 2, init = 'k-means++',
random_state = 42)
```

Partiendo del experimento previo, se mantuvieron el número de características filtradas o calve, en 30 y 40, respectivamente para los dos análisis, franja horaria y encendidos.

Los resultados se presentan en las tablas 15 y 16, aunque la utilización de estas nuevas características no conlleva una mejora sustancial de la precisión.

Lavadora			Lavavajillas		
30 key features	Predicción Franja horaria		*30 key features*	Predicción Franja horaria	
Mes	**Precisión General**	**Precisión Usuario objetivo**	**Mes**	**Precisión General**	**Precisión Usuario objetivo**
MAYO	21%	5	MAYO	28%	6
JUNIO	24%	5	JUNIO	21%	0
JULIO	25%	0	JULIO	27%	0
AGOSTO	21%	3	AGOSTO	19%	6
SEPTIEMBRE	32%	0	SEPTIEMBRE	23%	6
OCTUBRE	19%	4	OCTUBRE	27%	7
NOVIEMBRE	25%	0	NOVIEMBRE	28%	0
PROMEDIO	**23,8%**		**PROMEDIO**	**24,8%**	

Tabla 15. Precisión en la predicción de la franja de uso con 2 características adicionales de tipo indicador de presencia en hogar (experimento 4).

Lavadora			Lavavajillas		
40 key features	Predicción encendidos/semana		*40 key features*	Predicción encendidos/semana	
Mes	**Precisión General**	**Precisión Usuario objetivo**	**Mes**	**Precisión General**	**Precisión Usuario objetivo**
MAYO	37%	3	MAYO	46%	2
JUNIO	42%	2	JUNIO	43%	2
JULIO	47%	1	JULIO	35%	2
AGOSTO	32%	3	AGOSTO	32%	3
SEPTIEMBRE	40%	2	SEPTIEMBRE	49%	2
OCTUBRE	34%	3	OCTUBRE	38%	2
NOVIEMBRE	45%	3	NOVIEMBRE	29%	2
PROMEDIO	**39,5%**		**PROMEDIO**	**39,0%**	

Tabla 16. Precisión en la precisión del número de encendidos con 2 características adicionales de tipo indicador de presencia en hogar (experimento 4).

5. Características adicionales incluidas para recoger consumo del fin de semana

Se incluyó una nueva característica, *ratio_weekend*, calculada como el porcentaje de consumo durante el fin de semana frente al total de la semana. En este cálculo se incluyó a modo de fin de semana el consumo de los viernes, sábados y domingos.

En las tablas 17 y 18 se muestra que los resultados al incluir esta característica, aun sin grandes incrementos, mejoran ligeramente para algunos de los meses en los dos análisis realizados, sobre todo para la lavadora.

Lavadora		
30 key features	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	27%	2
JUNIO	24%	5
JULIO	29%	0
AGOSTO	25%	3
SEPTIEMBRE	23%	0
OCTUBRE	27%	1
NOVIEMBRE	27%	0
PROMEDIO	**25,9%**	

Lavavajillas		
30 key features	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	21%	6
JUNIO	20%	0
JULIO	18%	0
AGOSTO	10%	6
SEPTIEMBRE	33%	6
OCTUBRE	22%	0
NOVIEMBRE	21%	0
PROMEDIO	**20,7%**	

Tabla 17. Precisión en la predicción de la franja de uso con una característica adicional de indicador de consumo en fin de semana (experimento 5).

Lavadora		
40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	44%	2
JUNIO	43%	3
JULIO	45%	2
AGOSTO	42%	3
SEPTIEMBRE	42%	1
OCTUBRE	38%	3
NOVIEMBRE	44%	3
PROMEDIO	**42,6%**	

Lavavajillas		
40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	41%	3
JUNIO	42%	2
JULIO	39%	2
AGOSTO	41%	3
SEPTIEMBRE	41%	2
OCTUBRE	29%	2
NOVIEMBRE	36%	2
PROMEDIO	**38,3%**	

Tabla 18. Precisión en la predicción del número de encendidos semanales con una característica adicional de indicador de consumo en fin de semana (experimento 5).

6. Eliminación de la franja 0, 'No tengo un horario fijo'

Como se ha comentado en esta sección, para el tipo de usuario escogido como referencia, se predecían demasiadas franjas de tipo 0 (franja "no tengo un horario fijo"), cuando la franja predominante debería ser de tipo 5, como se muestra en el siguiente extracto de predicción de uno de los experimentos anteriores:

```
y_pred [3 0 0 3 3 0 5 0 2 7 3 0 2 6 1 2 2 0 0 2 3 0 0 3 2
0 2 0 3 6 6 0 3 5 3 3 0 2 2 5 3 0 3 0 0 0 7 3 0 4 3 3 2 0
2 7 2 0 7 0 0 2 0 0 5 1 3 3 0 5 0 0 3 0 0 5 3 0 0 5 3 0 2
0 0 6 2 3 0 3 3 0 7 0 0 0 0 2 3 6 0 0 0 7 2 1 7 5 3 2 3 5]
```

Se asoció la causa a un alto número de respuestas de valor de franja 0. Para intentar paliar este problema se eliminaron los usuarios correspondientes al valor de franja 0.

Tras introducir este cambio se produce un ligero incremento de la precisión, como se ve en la tabla 19 y 20, más en el caso del lavavajillas que en el de la lavadora, donde casi no hay variación, y sobre todo en la predicción de la franja horaria. Aunque también se prescinde de parte de los datos, se reducen los usuarios a 478, que representa un 64% de la muestra original.

Lavadora		
30 key features	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	28%	2
JUNIO	28%	3
JULIO	24%	5
AGOSTO	19%	3
SEPTIEMBRE	32%	3
OCTUBRE	21%	5
NOVIEMBRE	28%	5
PROMEDIO	**25,5%**	

Lavavajillas		
30 key features	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	29%	6
JUNIO	21%	7
JULIO	35%	6
AGOSTO	35%	6
SEPTIEMBRE	23%	6
OCTUBRE	36%	2
NOVIEMBRE	31%	5
PROMEDIO	**29,8%**	

Tabla 19. Precisión en la predicción de la franja de uso sin considerar los horarios de la franja 0 o 'no tengo un horario fijo' (experimento 6).

Lavadora		
40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	40%	2
JUNIO	39%	2
JULIO	45%	2
AGOSTO	35%	3
SEPTIEMBRE	42%	3
OCTUBRE	38%	3
NOVIEMBRE	40%	3
PROMEDIO	**39,9%**	

Lavavajillas		
40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	32%	3
JUNIO	47%	2
JULIO	50%	3
AGOSTO	46%	3
SEPTIEMBRE	38%	3
OCTUBRE	42%	2
NOVIEMBRE	50%	2
PROMEDIO	**43,6%**	

Tabla 20. Precisión en la predicción de la franja del número de encendidos sin considerar los horarios de la franja 0 o 'no tengo un horario fijo' (experimento 6).

7. Identificación de una posible mala interpretación de las franjas por parte del usuario

En este experimento se cambió la hora límite de las franjas especificadas en el cuestionario, con la intención de capturar el efecto de una posible interpretación errónea de la pregunta del cuestionario: el usuario podría interpretar la última hora como incluida o como no incluida en la franja.

- 'Entre las 2 y las 7' (1)
- 'Entre las 7 y las 10' (2)
- 'Entre las 10 y las 13' (3)
- 'Entre las 13 y las 16' (4)
- 'Entre las 16 y las 20' (5)
- 'Entre las 20 y las 23' (6)
- 'Entre las 23 y las 2' (7)

Hasta ahora las franjas eran definidas de la siguiente forma:

```
dataset['amanecer_s'] = dataset.iloc[:,7:10].sum(axis=1) # de 07 a 09 incluida
dataset['mañana_s'] = dataset.iloc[:,10:13].sum(axis=1) # de 10 a 12 incluida
dataset['mediodía_s'] = dataset.iloc[:,13:16].sum(axis=1) # de 13 a 15 incluida
dataset['tarde_s'] = dataset.iloc[:,16:20].sum(axis=1) # de 16 a 19 incluida
dataset['anochecer_s'] = dataset.iloc[:,20:23].sum(axis=1) # de 20 a 22 incluida
```

```
dataset['noche_s'] = dataset.iloc[:,[23,24,1]].sum(axis=1) # de 23 a 01 incluida
dataset['madrugada_s'] = dataset.iloc[:,2:7].sum(axis=1) # de 02 a 06 incluida
```

En este caso se varió la definición para incluir la última hora de cada franja en el análisis:

```
dataset['amanecer_s'] = dataset.iloc[:,8:11].sum(axis=1) # de 08 a 10 incluida
dataset['mañana_s'] = dataset.iloc[:,11:12].sum(axis=1) # de 11 a 13 incluida
dataset['mediodía_s'] = dataset.iloc[:,14:17].sum(axis=1) # de 14 a 16 incluida
dataset['tarde_s'] = dataset.iloc[:,17:21].sum(axis=1) # de 17 a 20 incluida
dataset['anochecer_s'] = dataset.iloc[:,21:24].sum(axis=1) # de 21 a 23 incluida
dataset['noche_s'] = dataset.iloc[:,[24,1,2]].sum(axis=1) # de 24 a 02 incluida
dataset['madrugada_s'] = dataset.iloc[:,3:8].sum(axis=1) # de 03 a 07 incluida
```

Con esta variación, los resultados de predicción de franja experimentaron una mejoría de algunos puntos porcentuales, como se muestra en la tabla 21, pero no tanto los de predicción de número de encendidos, como se muestra en la tabla 22.

Lavadora		
30 key features	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	21%	5
JUNIO	29%	3
JULIO	29%	3
AGOSTO	24%	3
SEPTIEMBRE	31%	3
OCTUBRE	29%	5
NOVIEMBRE	32%	1
PROMEDIO	**27,8%**	

Lavavajillas		
30 key features	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	25%	5
JUNIO	29%	5
JULIO	25%	7
AGOSTO	18%	6
SEPTIEMBRE	27%	6
OCTUBRE	34%	7
NOVIEMBRE	34%	5
PROMEDIO	**27,5%**	

Tabla 21. Precisión en la predicción de la franja de uso considerando una variación en la interpretación de las franjas (experimento 7).

Lavadora		
40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	40%	3
JUNIO	43%	2
JULIO	39%	2
AGOSTO	41%	2
SEPTIEMBRE	55%	2
OCTUBRE	40%	2
NOVIEMBRE	41%	4
PROMEDIO	**42,6%**	

Lavavajillas		
40 key features	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo
MAYO	41%	2
JUNIO	45%	2
JULIO	36%	3
AGOSTO	41%	3
SEPTIEMBRE	45%	2
OCTUBRE	40%	2
NOVIEMBRE	39%	2
PROMEDIO	**41,0%**	

Tabla 22. Precisión en la predicción del número de encendidos considerando una variación en la interpretación de las franjas (experimento 7).

8. Reducción del número de franjas horarias

Con los experimentos anteriores se comprobó la dificultad de aumentar la precisión, asociada, como se ha mencionado, a la naturaleza del problema.

Por eso, se diseñó un nuevo experimento para reducir el número de franjas horarias de 7 a 4, considerando que 7 franjas podían ser demasiada exigencia en un ejercicio de clasificación de este tipo para los datos disponibles.

Con la nueva categorización, la franja indicada por el usuario objetivo en el cuestionario se correspondería con la número 3:

- 'Entre las 2 y las 7' (1)
- 'Entre las 7 y las 10' (2)
- 'Entre las 10 y las 13' (2)
- **'Entre las 13 y las 16' (3)**
- **'Entre las 16 y las 20' (3)**
- 'Entre las 20 y las 23' (4)
- 'Entre las 23 y las 2' (4)

Con esta reducción de franjas se observa una mejora notable de la precisión en la predicción de la franja horaria, por encima del 50%, aunque para el usuario objetivo acierta en un 40% (véase tabla 23).

Lavadora			Lavavajillas		
30 key features	Predicción Franja horaria		*30 key features*	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo	Mes	Precisión General	Precisión Usuario objetivo
MAYO	54%	3	MAYO	51%	4
JUNIO	52%	3	JUNIO	43%	3
JULIO	43%	2	JULIO	54%	3
AGOSTO	57%	2	AGOSTO	41%	3
SEPTIEMBRE	53%	4	SEPTIEMBRE	37%	4
OCTUBRE	55%	2	OCTUBRE	43%	4
NOVIEMBRE	47%	1	NOVIEMBRE	56%	4
PROMEDIO	**51,3%**		**PROMEDIO**	**46,4%**	

Tabla 23. Precisión en la predicción de la franja de uso al reducir el número de franjas a 4 (experimento 8).

En varios de los casos de 4 franjas, la selección lleva a un menor peso de la franja 1, dado que esa franja (que corresponde a las horas entre las 2 y las 7) no suele ser la habitual para encender estos electrodomésticos, según se ve en las matrices de confusión de algunos meses (véase figura 23).

LAVADORA mes 5

	1	2	3	4
1	0	4	0	0
2	0	34	8	4
3	0	10	2	3
4	0	6	2	7

LAVADORA mes 6

	1	2	3	4
1	0	0	1	2
2	1	1	1	6
3	0	1	2	13
4	0	1	5	29

LAVADORA mes 7

	1	2	3	4
1	0	1	0	3
2	0	37	2	6
3	0	15	0	0
4	1	9	1	4

LAVADORA mes 8

	1	2	3	4
1	0	0	0	4
2	0	1	1	7
3	0	1	2	13
4	0	3	7	24

LAVADORA mes 9

	1	2	3	4
1	0	2	1	1
2	1	29	10	6
3	0	13	2	0
4	0	11	1	3

LAVADORA mes 10

	1	2	3	4
1	0	0	0	3
2	1	0	1	7
3	1	2	6	7
4	1	0	6	28

Figura 23. Matriz de confusión para predicción de franjas horarias al reducir el número de franjas a 4 (experimento 8).

Por eso la variación no es muy alta en el caso de utilizar 3 franjas, como se muestra en la tabla 24. La precisión se sitúa también por encima del 50%.

Lavadora		
30 key features	**Predicción Franja horaria**	
Mes	**Precisión General**	**Precisión Usuario objetivo**
MAYO	61%	3
JUNIO	51%	3
JULIO	49%	3
AGOSTO	48%	2
SEPTIEMBRE	47%	2
OCTUBRE	48%	1
NOVIEMBRE	54%	3
PROMEDIO	**51,2%**	

Lavavajillas		
30 key features	**Predicción Franja horaria**	
Mes	**Precisión General**	**Precisión Usuario objetivo**
MAYO	57%	3
JUNIO	59%	3
JULIO	40%	3
AGOSTO	54%	2
SEPTIEMBRE	53%	2
OCTUBRE	63%	1
NOVIEMBRE	51%	3
PROMEDIO	**53,8%**	

Tabla 24. Precisión en la predicción de la franja de uso al reducir el número de franjas a 3 (experimento 8).

Aunque los resultados de precisión aislados son bajos en comparación con otro tipo de problemas, en el contexto del problema actual, según se comentó en secciones anteriores, podrían llegar a ser aceptables.

En el caso de la predicción del número de encendidos, la mejoría no es tan alta, aunque también se aprecia, como se ve en la tabla 25.

Lavadora			Lavavajillas		
40 key features	Predicción encendidos/semana		*40 key features*	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo	Mes	Precisión General	Precisión Usuario objetivo
MAYO	44%	3	MAYO	50%	2
JUNIO	56%	2	JUNIO	52%	2
JULIO	54%	2	JULIO	45%	3
AGOSTO	47%	3	AGOSTO	44%	3
SEPTIEMBRE	41%	3	SEPTIEMBRE	47%	2
OCTUBRE	52%	2	OCTUBRE	53%	3
NOVIEMBRE	46%	3	NOVIEMBRE	41%	2
PROMEDIO	**48,4%**		**PROMEDIO**	**47,3%**	

Tabla 25. Precisión en la predicción del número de encendidos al reducir el número de franjas a 4 (experimento 8).

9. Característica adicional incluida para recoger el impacto de coste de la energía

En este experimento se planteó incluir el efecto del coste de la energía en el análisis utilizando una característica adicional: *'coste'*. El objetivo era recoger los efectos asociados al consumo en una franja horaria u otra, dado que el coste de la energía varía con las horas de un día, como se observa en la figura 24, donde se exponen las franjas tarifarias generales.

HORA	LUNES A VIERNES	SABADOS-DOMINGOS
0	VALLE	VALLE
1	VALLE	VALLE
2	VALLE	VALLE
3	VALLE	VALLE
4	VALLE	VALLE
5	VALLE	VALLE
6	VALLE	VALLE
7	VALLE	VALLE
8	LLANA	VALLE
9	LLANA	VALLE
10	PICO	VALLE
11	PICO	VALLE
12	PICO	VALLE
13	PICO	VALLE
14	LLANA	VALLE
15	LLANA	VALLE
16	LLANA	VALLE
17	PICO	VALLE
18	PICO	VALLE
19	PICO	VALLE
20	PICO	VALLE
21	PICO	VALLE
22	LLANA	VALLE
23	LLANA	VALLE

Figura 24. Franjas tarifarias generales que marcan el coste de la energía según la hora del día.

Para ello se estimó el coste promedio de franja a un precio para cada período de tiempo elegido (en este caso mensual) a partir de los precios de energía de OMIE de 2021.

Se asumió que todos los usuarios tuvieron una potencia contratada de menos de 15kW (que determina un uso doméstico) y una tarifa DHA con discriminación horaria. A partir del 1 de julio de 2021 se implantó un nuevo régimen tarifario en España. Se

consideraron por eso dos períodos de tarificación separados: hasta 1 de junio de 2021, se aplicó la tarifa 2.0 y a partir de entonces, la 2.0TD.

Se definió un método para extraer los precios de la base de datos de OMIE (Operador del Mercado Eléctrico Español) y se calcularon los precios promedios para cada hora de 2021, como se recoge en la tabla 26.

	month	day	hour	price
0	1.0	1.0	2.0	48.19
1	1.0	1.0	3.0	44.68
2	1.0	1.0	4.0	38.50
3	1.0	1.0	5.0	36.80
4	1.0	1.0	6.0	36.55
...	...	...	...	...
8391	12.0	31.0	20.0	187.95
8392	12.0	31.0	21.0	182.30
8393	12.0	31.0	22.0	175.12
8394	12.0	31.0	23.0	140.38
8395	12.0	31.0	24.0	127.81

8396 rows × 4 columns

Tabla 26. Costes de energía para cada hora de 2021 obtenidos de OMIE.

Los resultados para la predicción de franja horaria se muestran en la tabla 27 y para la predicción del número de encendidos, en la tabla 28. Ambos son del mismo orden que el caso anterior y por tanto no suponen una mejora sustancial.

Lavadora			Lavavajillas		
30 key features	Predicción Franja horaria		*30 key features*	Predicción Franja horaria	
Mes	Precisión General	Precisión Usuario objetivo	Mes	Precisión General	Precisión Usuario objetivo
MAYO	48%	3	MAYO	46%	3
JUNIO	57%	3	JUNIO	51%	3
JULIO	50%	2	JULIO	46%	3
AGOSTO	56%	2	AGOSTO	49%	3
SEPTIEMBRE	51%	2	SEPTIEMBRE	31%	4
OCTUBRE	51%	2	OCTUBRE	40%	4
NOVIEMBRE	47%	1	NOVIEMBRE	49%	3
PROMEDIO	**51,4%**		**PROMEDIO**	**44,6%**	

Tabla 27. Precisión de la predicción de la franja de uso al considerar el coste horario (experimento 9).

Lavadora			Lavavajillas		
40 key features	Predicción encendidos/semana		*40 key features*	Predicción encendidos/semana	
Mes	Precisión General	Precisión Usuario objetivo	Mes	Precisión General	Precisión Usuario objetivo
MAYO	43%	2	MAYO	50%	3
JUNIO	43%	2	JUNIO	32%	2
JULIO	44%	3	JULIO	46%	3
AGOSTO	42%	3	AGOSTO	47%	3
SEPTIEMBRE	41%	3	SEPTIEMBRE	47%	3
OCTUBRE	43%	2	OCTUBRE	33%	2
NOVIEMBRE	41%	2	NOVIEMBRE	33%	2
PROMEDIO	**42,2%**		**PROMEDIO**	**41,3%**	

Tabla 28. Precisión de la predicción del número de encendidos al considerar el coste horario (experimento 9).

Resumen de resultados

En la tabla 29 se presenta un resumen de los resultados de los experimentos relativos a la predicción de la franja horaria. El conjunto de experimentos conlleva una mejora de la precisión, llegando a valores superiores al 50%, donde los más determinantes fueron la diferente interpretación de las franjas (experimento 7) o la reducción de franja (experimento 8).

	Precisión predicción franja horaria													
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6	Exp.7	Exp.8	Exp.9	Lavadora	Lavavajillas	Incremento respecto a caso base		
EXP 1	10 key features									23%	24%	LAVAD	LAVAV	**EXP 1**
EXP 1	20 key features									24%	25%	1%	1%	**EXP 1**
EXP 1	30 key features									25%	27%	2%	3%	**EXP 1**
EXP 1	40 key features									23%	25%	0%	1%	**EXP 1**
EXP 2	30/40 key features	Semanal								25%		2%		**EXP 2**
EXP 2	30/40 key features	Mensual								26%	24%	3%	0%	**EXP 2**
EXP 3	30/40 key features	Mensual	Picos							26%	26%	3%	2%	**EXP 3**
EXP 3	30/40 key features	Mensual	Picos							29%	24%	6%	0%	**EXP 3**
EXP 4	30/40 key features	Mensual	Picos	Presencia						24%	25%	1%	1%	**EXP 4**
EXP 5	30/40 key features	Mensual	Picos	Presencia	Weekend					26%	21%	3%	-3%	**EXP 5**
EXP 6	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo				25%	30%	3%	6%	**EXP 6**
EXP 7	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo	Interpretación			28%	28%	5%	3%	**EXP 7**
EXP 8	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo	Interpretación	4 franjas		51%	46%	28%	22%	**EXP 8**
EXP 9	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo	Interpretación	4 franjas	Coste	51%	45%	28%	20%	**EXP 9**

Tabla 29. Resumen de resultados de los experimentos para la predicción de franja horaria.

Para los experimentos relativos a la predicción del número de encendidos, los resultados se recogen en la tabla 30. Como se ve, la principal mejoría también se produjo para los experimentos, 7 y 8. En total se obtuvieron valores algo menores que en el análisis anterior, de hasta 47-48%.

	Precisión predicción encendidos													
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6	Exp.7	Exp.8	Exp.9	Lavadora	Lavavajillas	Incremento respecto a caso base		
EXP 1	10 key features									40%	39%	LAVAD	LAVAV	**EXP 1**
EXP 1	20 key features									42%	41%	2%	2%	**EXP 1**
EXP 1	30 key features									40%	38%	1%	-1%	**EXP 1**
EXP 1	40 key features									41%	45%	1%	6%	**EXP 1**
EXP 2	30/40 key features	Semanal								42%		2%		**EXP 2**
EXP 2	30/40 key features	Mensual								42%	42%	3%	3%	**EXP 2**
EXP 3	30/40 key features	Mensual	Picos							40%	39%	1%	0%	**EXP 3**
EXP 3	30/40 key features	Mensual	Picos							42%	39%	2%	1%	**EXP 3**
EXP 4	30/40 key features	Mensual	Picos	Presencia						39%	39%	0%	0%	**EXP 4**
EXP 5	30/40 key features	Mensual	Picos	Presencia	Weekend					43%	38%	3%	-1%	**EXP 5**
EXP 6	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo				40%	44%	0%	5%	**EXP 6**
EXP 7	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo	Interpretación			43%	41%	3%	2%	**EXP 7**
EXP 8	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo	Interpretación	4 franjas		48%	47%	9%	8%	**EXP 8**
EXP 9	30/40 key features	Mensual	Picos	Presencia	Weekend	Horario fijo	Interpretación	4 franjas	Coste	42%	41%	3%	2%	**EXP 9**

Tabla 30. Resumen de resultados de los experimentos para la predicción del número de encendidos semanales.

Utilización de una base temporal anual

Por último, se añadió un experimento aislado, que consideró la variación de la base temporal de nuevo, con una única agrupación anual. Los resultados para la predicción de franja de uso se muestran en la tabla 31 y son mejores que para otras bases temporales

anteriores, en algunos casos es cerca del 60% de precisión general, lo que indica que el método podría llegar a usarse en base anual siempre que existieran datos para un número aceptable de semanas.

	37 key features Lavadora		*37 key features Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
Run 1	56%	3	52%	3
Run 2	64%	3	56%	3

Tabla 31. Resultados de predicción de franja horaria en base temporal anual para la lavadora.

Existen otros experimentos que podrían añadirse en este apartado, como por ejemplo incluir características que penalizasen la predicción de una franja horaria muy alejada de la franja real y no adyacente, pero dado que las franjas finales elegidas fueron 4 no se incluyó en este trabajo.

En la figura 25 se recogieron las características que se utilizaron en los experimentos y que se han preseleccionado para el *dataframe*, como ejemplo para el primer mes de datos (el número 5).

```
[promedio extraído del INE a partir del CP del usuario
'RENTA',

Características basadas en consumo (1 por franja horaria para las 4 franjas + total).
'mean_madrugada_m5', 'mean_mañana_m5', 'mean_tarde_m5', 'mean_noche_m5', 'mean_total_m5',
'sd_madrugada_m5', 'sd_mañana_m5', 'sd_tarde_m5', 'sd_noche_m5', 'sd_total_m5',
'max_madrugada_m5', 'max_mañana_m5', 'max_tarde_m5', 'max_noche_m5', 'max_total_m5',
'min_madrugada_m5', 'min_mañana_m5', 'min_tarde_m5', 'min_noche_m5', 'min_total_m5',
'sum_madrugada_m5', 'sum_mañana_m5', 'sum_tarde_m5', 'sum_noche_m5', 'sum_total_m5',

Número de franja pico de consumo. Número de franjas adyacentes al pico por delante y detrás.
'peak_m5', 'adj_peak1_m5', 'adj_peak2_m5',

Característica indicador para cada franja, con 1 si es pico, 0 si no es pico
'p_madrugada_m5', 'p_mañana_m5', 'p_tarde_m5', 'p_noche_m5',

Ratio de consumo durante la jornada laboral (% de consumo durante la jornada laboral 8-18 vs 6-
23)
'ratio_jornada_m5',

Característica de presencia (resultado del clustering no supervisado K-means a partir del ratio
de jornada.
'presencia',

Ratio de consumo durante el fin de semana (V-D)/(Semana completa).
'ratio_weekend_m5',

Característica de hora fija (si en la encuesta se responde "No consumo en una hora fija" se
elimina ese dato de ese usuario, quedándonos sólo con Hora fija = 1
'HORA_FIJA_WM',

Coste de cada franja horaria, estimado como el consumo* precio medio OMIE de la franja.
'c_madrugada_m5', 'c_mañana_m5', 'c_tarde_m5', 'c_noche_m5',

Característica objetivo: franja horaria de consumo respondida en el cuestionario.
'HORA_LAVADORA']

Característica objetivo: franja horaria de consumo respondida en el cuestionario.
'HORA_LAVAVAJILLAS']
```

Figura 25. Resumen de características utilizadas en los experimentos para el mes de mayo.

4.2. Análisis de importancia y redundancia de características.

Como ya se ha mencionado al inicio de la sección, antes de pasar a la segunda tanda de experimentos se evaluó la importancia de las características seleccionadas utilizando la función *feature_importance*, que permite graficar las características en orden de peso o importancia para el análisis.

En la figura 26 se observa que las características más importantes para la predicción de franjas horarias fueron las definidas para recoger la presencialidad en la vivienda, *ratio_jornada*, y el consumo de fin de semana, *ratio_weekend*, pero también algunos valores de desviación típica y de máximos y mínimos.

Las de menor impacto, por el contrario, fueron las que recogen la franja pico; esto puede ser debido a varias razones, como la existencia de más de un pico en cada día, a que no siempre la franja pico coincida con la indicada por los usuarios en el cuestionario, o a que los picos pueden verse camuflados por la agrupación mensual o anual de los datos.

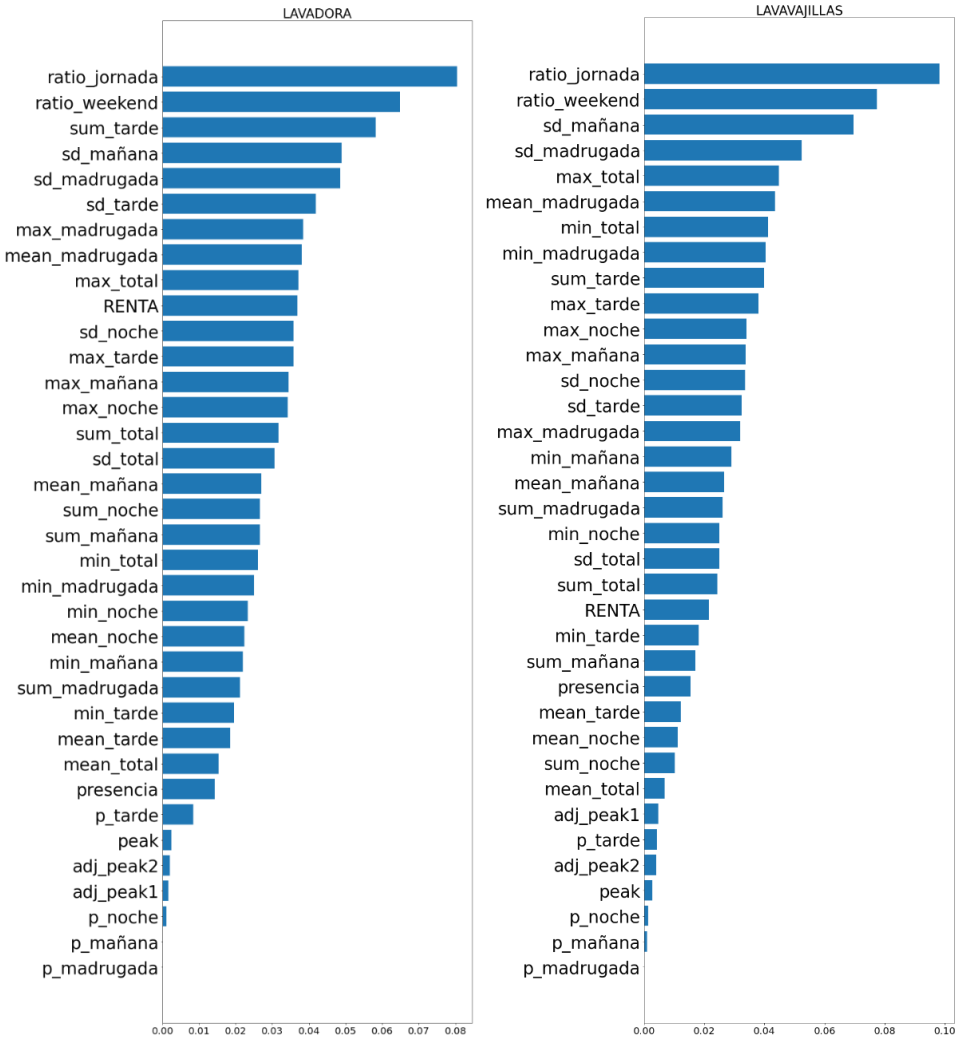

Figura 26. Importancia relativa de las características seleccionadas para predicción de la franja horaria de uso de electrodomésticos.

Para la predicción del número de encendidos, el consumo de fin de semana, *ratio_weekend*, es la característica más importante, también seguida de las desviaciones típicas y, en este caso, de algunas sumas, como se muestra en la figura 27.

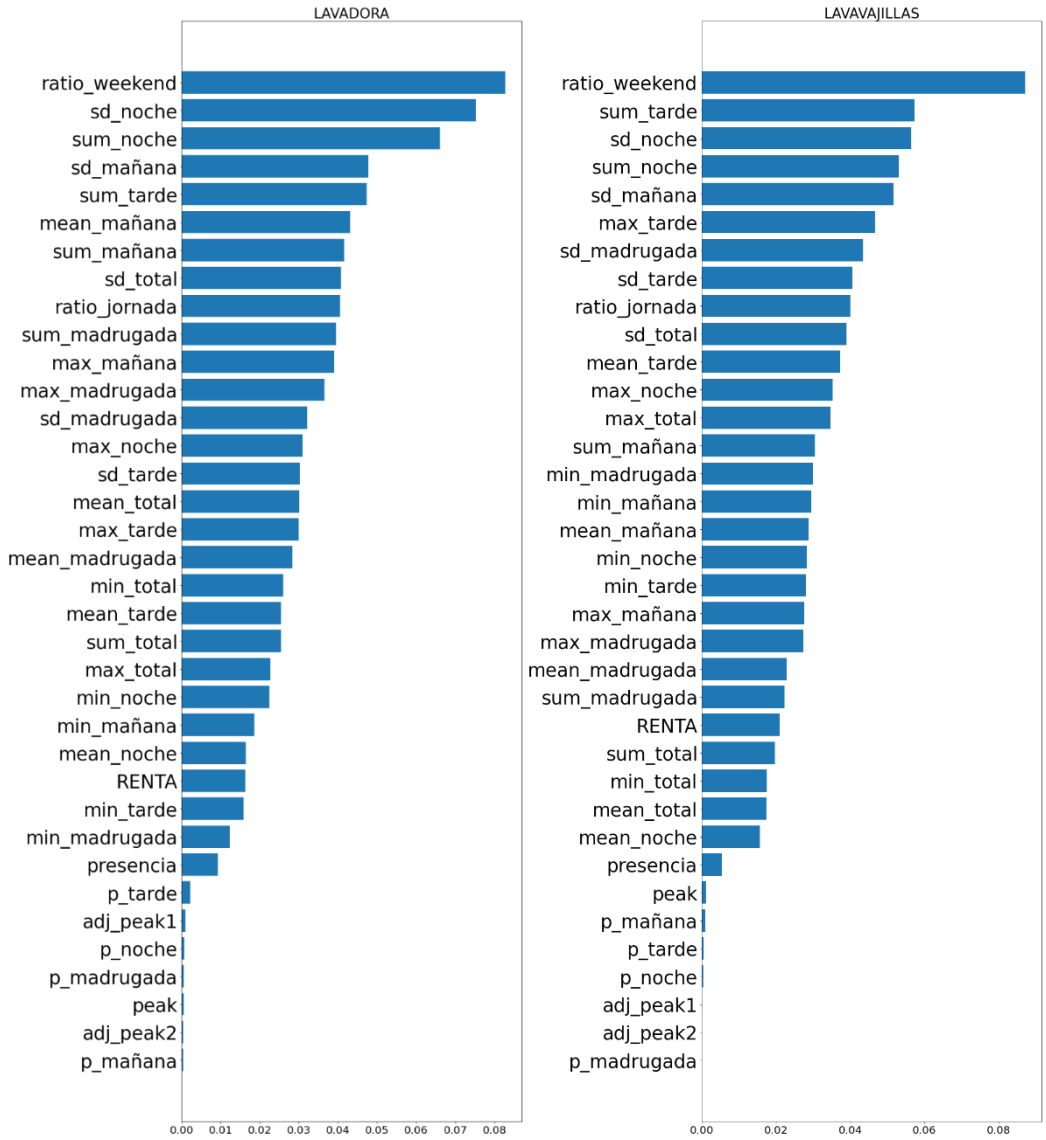

Figura 27. Importancia relativa de las características seleccionadas para predicción del número de encendidos semanales de electrodomésticos.

También se realizó un análisis de correlación y de multicolinearidad, es decir, la correlación simultanea entre 3 o más variables, para comprobar la posible redundancia de características [Kesornsit, 2022].

Para detectar la correlación, se aplicó el método *DataFrame.corr()* de pandas, disponible en Python, directamente sobre el *dataframe* objetivo:

```
DataFrame.corr(method='pearson', min_periods=1)
```

En la figura 28 se muestra el resultado de este análisis, aplicando la correlación de Pearson. Se aprecia correlación entre algunas variables, como las medias, ya que sus valores se acercan lo suficiente a la unidad.

	RENTA	mean_madrugada	mean_mañana	mean_tarde	mean_noche	mean_total	sd_madrugada	sd_mañana	sd_tarde	sd_noche	...
RENTA	1.000000	-0.090184	-0.077077	-0.123055	-0.096341	-0.110627	-0.056987	-0.011813	-0.150116	-0.088988	...
mean_madrugada	-0.090184	1.000000	0.661817	0.534500	0.682172	0.766293	0.758883	0.592749	0.397190	0.551901	...
mean_mañana	-0.077077	0.661817	1.000000	0.725834	0.724724	0.892654	0.448715	0.749554	0.485429	0.519883	...
mean_tarde	-0.123055	0.534500	0.725834	1.000000	0.816855	0.906942	0.357751	0.501841	0.802861	0.594396	...
mean_noche	-0.096341	0.682172	0.724724	0.816855	1.000000	0.925680	0.537882	0.566258	0.642426	0.783655	...
mean_total	-0.110627	0.766293	0.892654	0.906942	0.925680	1.000000	0.554799	0.678184	0.688283	0.697835	...
sd_madrugada	-0.056987	0.758883	0.448715	0.357751	0.537882	0.554799	1.000000	0.660937	0.359516	0.659812	...
sd_mañana	-0.011813	0.592749	0.749554	0.501841	0.566258	0.678184	0.660937	1.000000	0.479070	0.634004	...
sd_tarde	-0.150116	0.397190	0.485429	0.802861	0.642426	0.688283	0.359516	0.479070	1.000000	0.672979	...
sd_noche	-0.088988	0.551901	0.519883	0.594396	0.783655	0.697835	0.659812	0.634004	0.672979	1.000000	...
sd_total	-0.086557	0.656264	0.613386	0.663760	0.725129	0.750181	0.787812	0.812545	0.767269	0.885776	...
max_madrugada	-0.061080	0.726825	0.488249	0.423666	0.579428	0.596845	0.910331	0.660799	0.431198	0.738378	...
max_mañana	-0.040908	0.623951	0.752381	0.552724	0.616767	0.717118	0.653954	0.929476	0.522875	0.692647	...
max_tarde	-0.134693	0.389250	0.468567	0.706049	0.590383	0.632850	0.342127	0.449520	0.898238	0.637154	...
max_noche	-0.095811	0.579288	0.539788	0.600540	0.767144	0.705684	0.666763	0.635981	0.649568	0.936770	...

Figura 28. Correlación de Pearson para algunas de las características del análisis.

Para detectar la multicolinearidad se utilizó el método *variance_inflation_factor* (*VIF*), dentro del paquete *statsmodels.stats.outliers_influence.*

En la figura 29 se muestra el vector de valores de VIF. Este índice indica correlación cuando los valores son especialmente altos.

```
multicolinearity_vector RENTA            1.077301e+00
mean_madrugada     1.239379e+06
mean_mañana        4.209352e+06
mean_tarde         4.981033e+06
mean_noche         4.092683e+06
mean_total         4.328495e+07
sd_madrugada       1.497470e+01
sd_mañana          2.380254e+01
sd_tarde           2.020880e+01
sd_noche           2.667411e+01
sd_total           5.728756e+01
max_madrugada      1.095830e+01
max_mañana         1.830975e+01
max_tarde          1.229013e+01
max_noche          2.110277e+01
max_total          3.341810e+01
min_madrugada      5.374288e+00
min_mañana         1.477248e+01
min_tarde          1.102773e+01
min_noche          1.210141e+01
min_total          1.788479e+01
sum_madrugada               inf
sum_mañana                  inf
sum_tarde                   inf
sum_noche                   inf
sum_total                   inf
peak                        inf
adj_peak1                   inf
adj_peak2                   inf
```

Figura 29. Análisis de multicolinearidad a partir del valor VIF.

A la vista de estos análisis, algunas características parecen redundantes, como los valores de las medias, para las que bastaría incluir una de ellas, por ejemplo, el total, o también en las sumas.

Tampoco la Renta o la presencia, parecen ser características críticas, aunque la Renta se mantuvo porque no presenta multicolinearidad con el resto y porque en algunos casos la importancia fue relevante.

Todo ello confirma que la selección que realiza el filtro mRMR funciona ya que prescinde de algunas de estas características, en concreto, varias de las sumas, medias, o las variables de tipo indicador definidas, tal como se presenta en la tabla 32.

Características a eliminar por redundancia	Características finales
mean_madrugada','mean_mañana', 'mean_tarde', 'mean_noche', 'mean_total', 'sum_madrugada', 'sum_mañana', 'sum_tarde', 'sum_noche', 'mean_tarde', 'sum_total', 'peak', 'adj_peak1', 'adj_peak2'	*['RENTA', 'sd_madrugada', 'sd_mañana', 'sd_tarde', 'sd_noche', 'sd_total', 'max_madrugada', 'max_mañana', 'max_tarde', 'max_noche', 'max_total', 'min_madrugada', 'min_mañana', 'min_tarde', 'min_noche', 'min_total', 'p_madrugada', 'p_mañana', 'p_tarde', 'p_noche', 'ratio_jornada', 'ratio_weekend', 'presencia']*

Tabla 32. Características eliminadas y características finales para el dataframe.

4.3. Experimentos para selección del algoritmo de clasificación

En esta sección se recogen los resultados obtenidos en la segunda tanda de experimentos, que se utilizó para elegir el algoritmo de clasificación de la aplicación.

Por practicidad, en estos experimentos se utilizó una agrupación anual de datos.

A continuación, se describen los experimentos y su efecto en la precisión de la predicción de franjas de uso. Posteriormente se recogen los resultados en una tabla resumen, tanto para este caso como para la predicción de número de encendidos en los dos electrodomésticos analizados, lavadora y lavavajillas.

1. Extreme Gradient Boosting Classification (XGBoost)

Se partió del algoritmo utilizado como Caso Base durante los experimentos del punto 4.1, *XGBoost* con la configuración descrita en la sección 3.7, y se introdujeron variaciones en algunos de sus parámetros, como la función de coste, la medida de error objetivo, o el número de estimadores (o árboles).

En la tabla 33 se muestran los resultados para diferentes funciones de coste[24], especificada a través del parámetro *eval_metric*. Por defecto se empleó la función logarítmica multiobjetivo para clasificación multiclase, *'mlogloss'*, que penaliza los resultados erróneos en base a la probabilidad de clasificación, pero también se utilizó *root mean square error* (RMSE), con el valor de parámetro: *'rmse'* y *multiclass error rate*

[24] *Cost function* o *loss function* en inglés.

(MCER) con el valor de parámetro: *'merror'*. Los resultados de precisión promedio se situaron entre el 45% y el 55%.

Algoritmo	XGBOOST eval_metric = 'mlogloss' number_of_estimators = 200, parallel_trees = 3, depth = 10, learning objective = MSE		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	48%	3	56%	3
RUN2	46%	3	51%	3
RUN3	55%	3	48%	4
RUN4	51%	3	54%	3
RUN5	48%	3	51%	3
PROMEDIO	50%		52%	

Algoritmo	XGBOOST eval_metric = 'rmse' number_of_estimators = 200, parallel_trees = 3, depth = 10, learning objective = MSE		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	46%	3	52%	3
RUN2	56%	3	46%	4
RUN3	54%	3	49%	3
RUN4	50%	3	52%	3
RUN5	51%	3	52%	3
PROMEDIO	52%		50%	

Algoritmo	XGBOOST eval_metric = 'merror' number_of_estimators = 200, parallel_trees = 3, depth = 10, learning objective = MSE		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	54%	3	43%	3
RUN2	55%	3	46%	3
RUN3	55%	3	40%	4
RUN4	59%	3	49%	3
RUN5	50%	3	46%	3
PROMEDIO	55%		45%	

Tabla 33. Precisión obtenida con XGBoost para distintas funciones de coste.

En la tabla 34 se muestran los resultados para una variación en el número de árboles, primero utilizando 400 y después 800, frente a los 200 del caso base. En el caso de 400 se ve una pequeña mejora de la precisión, pero no tiene el mismo efecto si se aumenta a 800, además un mayor número de árboles requiere más tiempo de cálculo.

Algoritmo	XGBOOST eval_metric = 'mlogloss' number_of_estimators = 200, parallel_trees = 3, depth = 10, learning objective = MSE		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	48%	3	56%	3
RUN2	46%	3	51%	3
RUN3	55%	3	48%	4
RUN4	51%	3	54%	3
RUN5	48%	3	51%	3
PROMEDIO	50%		52%	

Algoritmo	XGBOOST eval_metric = 'mlogloss' number_of_estimators = 400, parallel_trees = 3, depth = 10, learning objective = MSE		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	49%	3	46%	4
RUN2	56%	3	48%	3
RUN3	56%	3	52%	4
RUN4	53%	3	44%	3
RUN5	55%	3	52%	3
PROMEDIO	54%		49%	

Algoritmo	XGBOOST eval_metric = 'mlogloss' number_of_estimators = 800, parallel_trees = 3, depth = 10, learning objective = MSE		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	54%	3	48%	3
RUN2	43%	3	56%	3
RUN3	48%	3	48%	3
RUN4	53%	3	40%	4
RUN5	51%	3	44%	3
PROMEDIO	50%		47%	

Tabla 34. Precisión obtenida con XGBoost según la cantidad de estimadores (árboles).

Por último, se realizó un análisis de la función objetivo. En la tabla 35 se muestran los resultados, con un mejor comportamiento de la función MSE (*mean square error*).

Algoritmo	**XGBOOST** **eval_metric = 'mlogloss'** **number_of_estimators = 400,** **parallel_trees = 3,** **depth = 10,** **learning objective = MSE**		**number_of_key_features = 30,** **size (train/total)= 0.15,** **cross_validation_number = 10**	
	Lavadora		***Lavavajillas***	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	48%	3	56%	3
RUN2	46%	3	51%	3
RUN3	55%	3	48%	4
RUN4	51%	3	54%	3
RUN5	48%	3	51%	3
PROMEDIO	50%		52%	

Algoritmo	**XGBOOST** **eval_metric = 'mlogloss'** **number_of_estimators = 400,** **parallel_trees = 3,** **depth = 10,** **learning objective = Squared log error**		**number_of_key_features = 30,** **size (train/total)= 0.15,** **cross_validation_number = 10**	
	Lavadora		***Lavavajillas***	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	49%	3	44%	4
RUN2	60%	3	49%	3
RUN3	45%	3	48%	4
RUN4	49%	3	43%	3
RUN5	55%	3	49%	3
PROMEDIO	52%		47%	

Tabla 35. Resultados de precisión con XGBoost en función de la función objetivo.

A la vista de los resultados anteriores, con el método *Extreme Gradient Boosting* se alcanzaron precisiones promedio de entre 47% y 54%.

2. Gradient Boosting Classification

En este segundo experimento, se probó un nuevo tipo de algoritmo y se variaron algunos parámetros, como el criterio de partición de nodos y el número de árboles.

En la tabla 36 se muestran los resultados utilizando dos criterios de partición, que son similares, aunque mejoran ligeramente para el criterio de Friedman.

Algoritmo	Gradient Boosting Classifier loss = 'log_loss' number_of_estimators = 200, parallel_trees = 3, depth = 10, criterion = 'friedman_mse'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	55%	3	44%	3
RUN2	58%	3	56%	3
RUN3	49%	3	49%	4
RUN4	56%	3	49%	2
RUN5	43%	2	49%	4
PROMEDIO	52%		50%	

Algoritmo	Gradient Boosting Classifier loss = 'log_loss' number_of_estimators = 200, parallel_trees = 3, depth = 10, criterion = 'mse'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	46%	3	49%	4
RUN2	50%	3	48%	3
RUN3	54%	3	43%	4
RUN4	50%	3	49%	3
RUN5	55%	3	48%	4
PROMEDIO	51%		47%	

Tabla 36. Precisión obtenida con el clasificador Gradient Boosting para dos criterios de error en la partición.

En la tabla 37 se muestran los resultados para una variación en el número de árboles, pasando de los 200 a los 400, pero no se apreció diferencia en los resultados.

Algoritmo	Gradient Boosting Classifier loss = 'log_loss' number_of_estimators = 200, parallel_trees = 3, depth = 10, criterion = 'friedman_mse'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	46%	3	49%	4
RUN2	50%	3	48%	3
RUN3	54%	3	43%	4
RUN4	50%	3	49%	3
RUN5	55%	3	48%	4
PROMEDIO	51%		47%	

Algoritmo	Gradient Boosting Classifier loss = 'log_loss' number_of_estimators = 400, parallel_trees = 3, depth = 10, criterion = 'friedman_mse'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	53%	3	44%	3
RUN2	56%	3	46%	3
RUN3	45%	3	51%	3
RUN4	46%	3	51%	4
RUN5	55%	3	41%	3
PROMEDIO	51%		47%	

Tabla 37. Precisión obtenida con el clasificador Gradient Boosting según el número de árboles empleados.

A la vista de los resultados, con el método *Gradient Boosting* se alcanzaron precisiones promedio de entre 47% y 52%.

3. Random Forest

En las ejecuciones llevadas a cabo con el algoritmo *Random Forest* se modificaron los parámetros que controlan la división de los elementos en los nodos de los árboles, (parámetro *criterio)*, el número de árboles (parámetro *n_estimators*) y la aleatoriedad de los datos muestreados (parámetro *random_state*).

En la tabla 38 se muestran los resultados para parámetro *criterion*. Los resultados mejoran en el caso de seleccionar la función '*entropy'* hasta llegar a un 57% de precisión para ambos electrodomésticos, frente a la función '*Gini*' que es algo más baja en el lavavajillas.

Algoritmo	Random Forest criterio = 'gini', number_of_estimators = 200, max depth = 10, random_state = 0, max_features = 'sqrt',		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	55%	3	54%	3
RUN2	55%	3	56%	3
RUN3	56%	2	54%	3
RUN4	60%	3	56%	4
RUN5	59%	3	44%	4
PROMEDIO	57%		53%	

Algoritmo	Random Forest criterio = 'entropy', number_of_estimators = 200, max depth = 10, random_state = 0, max_features = 'sqrt',		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	56%	2	59%	4
RUN2	58%	3	57%	4
RUN3	58%	3	54%	3
RUN4	56%	3	52%	3
RUN5	56%	2	62%	4
PROMEDIO	57%		57%	

Tabla 38. Precisión obtenida con Random Forest según la medida de bondad en la partición.

En la tabla 39 se presentan los resultados obtenidos al variar el número de árboles, donde se ve que la precisión es mejor para 200 árboles.

Algoritmo	**Random Forest** **criterio = 'entropy',** **number_of_estimators = 100,** **max depth = 10,** **random_state = 0,** **max_features = 'sqrt',**		**number_of_key_features = 30,** **size (train/total)= 0.15,** **cross_validation_number = 10**	
	Lavadora		***Lavavajillas***	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	56%	2	46%	3
RUN2	60%	3	57%	4
RUN3	53%	2	56%	4
RUN4	58%	3	56%	3
RUN5	56%	2	49%	3
PROMEDIO	57%		53%	

Algoritmo	**Random Forest** **criterio = 'entropy',** **number_of_estimators = 200,** **max depth = 10,** **random_state = 0,** **max_features = 'sqrt',**		**number_of_key_features = 30,** **size (train/total)= 0.15,** **cross_validation_number = 10**	
	Lavadora		***Lavavajillas***	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	56%	2	59%	4
RUN2	58%	3	57%	4
RUN3	58%	3	54%	3
RUN4	56%	3	52%	3
RUN5	56%	2	62%	4
PROMEDIO	57%		57%	

Algoritmo	**Random Forest** **criterio = 'entropy',** **number_of_estimators = 400,** **max depth = 10,** **random_state = 0,** **max_features = 'sqrt',**		**number_of_key_features = 30,** **size (train/total)= 0.15,** **cross_validation_number = 10**	
	Lavadora		***Lavavajillas***	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	55%	3	51%	4
RUN2	55%	3	52%	4
RUN3	61%	2	54%	3
RUN4	55%	2	52%	3
RUN5	60%	2	56%	3
PROMEDIO	57%		53%	

Tabla 39. Resultados de precisión con Random Forest según el número de árboles utilizados.

En la tabla 40 se muestran los resultados obtenidos al variar la forma de aleatoriedad a partir del parámetro *Random State*, pero el caso de 0 sigue siendo el de mayor precisión.

Algoritmo	Random Forest criterio = 'entropy', number_of_estimators = 200, max depth = 10, random_state = 0, max_features = 'sqrt',		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	56%	2	59%	4
RUN2	58%	3	57%	4
RUN3	58%	3	54%	3
RUN4	56%	3	52%	3
RUN5	56%	2	62%	4
PROMEDIO	57%		57%	

Algoritmo	Random Forest criterio = 'entropy', number_of_estimators = 200, max depth = 10, random_state = 42, max_features = 'sqrt',		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	59%	3	52%	4
RUN2	51%	3	48%	4
RUN3	56%	3	56%	4
RUN4	54%	3	59%	3
RUN5	60%	3	49%	3
PROMEDIO	56%		53%	

Tabla 40. Precisión obtenida con Random Forest según aleatoriedad indicada.

Con el método *Gradient Boosting* se alcanzan precisiones promedio de entre 53% y 57%.

4. Support Vector Classification

Los métodos de Máquinas de Vectores Soporte pueden ser programados en Python utilizando el paquete SVC de *scikit-learn* para clasificación multiclase.

Para este tipo de algoritmo es importante definir bien la función *kernel* así como el procedimiento de cálculo en el análisis multiclase, *one versus one*, 'ovo', o *one versus the rest*, 'ovr'.

En la tabla 41 se muestra el resultado de utilizar diversas funciones de *kernel*. El comportamiento de este algoritmo fue bastante exitoso, llegando a valores de precisión de entre 55% y 58% con la función *rbf*. Según especifica Hsu [2016], esta función *kernel* es especialmente recomendable como primer intento, debido a que puede manejar muestras en un espacio dimensional mayor y soporta posibles no-linealidades.

Algoritmo	SVM kernel_function = 'rbf', decision_function = 'ovo', random_state = 0,		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	58%	2	56%	4
RUN2	59%	2	56%	4
RUN3	56%	2	56%	4
RUN4	58%	2	56%	4
RUN5	58%	2	54%	4
PROMEDIO	58%		55%	

Algoritmo	SVM kernel_function = 'sigmoid', decision_function = 'ovo', random_state = 42,		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	51%	2	48%	3
RUN2	56%	3	52%	3
RUN3	55%	3	54%	3
RUN4	56%	3	56%	3
RUN5	49%	3	52%	3
PROMEDIO	54%		52%	

Algoritmo	SVM kernel_function = 'poly', decision_function = 'ovo', random_state = 42,		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	**Precisión Usuario objetivo**	**Precisión General**	**Precisión Usuario objetivo**
RUN1	59%	2	54%	4
RUN2	60%	2	56%	4
RUN3	58%	2	56%	4
RUN4	56%	2	56%	4
RUN5	56%	2	54%	4
PROMEDIO	58%		55%	

Tabla 41. Precisión obtenida con Support Vector Classification según variación de la función kernel.

En el caso del tipo de secuencia de cálculo multiclase, *ovo* y *ovr* dan resultados similares, como se recoge en la tabla 42.

Algoritmo	SVM kernel_function = 'rbf', decision_function = 'ovo', random_state = 42,		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	58%	2	56%	4
RUN2	59%	2	56%	4
RUN3	56%	2	56%	4
RUN4	58%	2	56%	4
RUN5	58%	2	54%	4
PROMEDIO	58%		55%	

Algoritmo	SVM kernel_function = 'rbf', decision_function = 'ovr', random_state = 42,		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	56%	2	56%	4
RUN2	58%	2	56%	4
RUN3	58%	2	56%	4
RUN4	58%	2	52%	4
RUN5	56%	2	56%	4
PROMEDIO	57%		55%	

Tabla 42. Precisión obtenida con Support Vector Classification según variación del método de cálculo multiclase.

A la vista de los resultados, en la clasificación con *Support Vector Classification* se alcanzaron precisiones promedio de entre 52% y 58%.

5. *K- Nearest Neighbor* (KNN)

En el experimento con este algoritmo, se probaron dos diferentes métodos de computación, como se muestra en la tabla 43, en la que se observa que las precisiones son buenas para la lavadora, pero no tanto para el lavavajillas.

Algoritmo	K- Nearest Neighbours neighbors = 5 metric_distance = 'minkowski' power distance parameter = 2 algorithm = 'ball_tree'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10	
	Lavadora		*Lavavajillas*	
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	54%	3	33%	3
RUN2	53%	3	46%	3
RUN3	51%	3	40%	3
RUN4	54%	3	49%	3
RUN5	58%	3	52%	3
PROMEDIO	54%		44%	

Algoritmo	K- Nearest Neighbours neighbors = 5 metric_distance = 'minkowski' power distance parameter = 2 algorithm = 'kd_tree'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10		
	Lavadora		*Lavavajillas*		
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo	
RUN1	51%	3	44%	3	
RUN2	56%	3	37%	3	
RUN3	48%	3	49%	3	
RUN4	53%	3	43%	3	
RUN5	51%	3	44%	3	
PROMEDIO	52%		44%		

Tabla 43. Precisión obtenida con K- Nearest Neighbor según variación del algoritmo.

También se realizaron simulaciones variando el número de vecinos, elevándolo primero a 10 y después a 50 para cada muestra, como se ve en la tabla 44. Las precisiones fueron mayores de lo esperado, alcanzándose valores de 58% en el caso de la lavadora.

Algoritmo	K- Nearest Neighbours neighbors = 10 metric_distance = 'minkowski' power distance parameter = 2 algorithm = 'ball_tree'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10		
	Lavadora		*Lavavajillas*		
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo	
RUN1	58%	2	52%	4	
RUN2	51%	3	51%	4	
RUN3	55%	2	56%	4	
RUN4	55%	3	56%	4	
RUN5	55%	3	51%	4	
PROMEDIO	55%		53%		

Algoritmo	K- Nearest Neighbours neighbors = 50 metric_distance = 'minkowski' power distance parameter = 2 algorithm = 'ball_tree'		number_of_key_features = 30, size (train/total)= 0.15, cross_validation_number = 10		
	Lavadora		*Lavavajillas*		
	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo	
RUN1	58%	2	55%	4	
RUN2	58%	2	55%	4	
RUN3	58%	2	55%	4	
RUN4	58%	2	55%	4	
RUN5	58%	2	55%	4	
PROMEDIO	58%		55%		

Tabla 44. Precisión obtenida con K- Nearest Neighbor según variación del número de vecinos originales.

A la vista de los resultados anteriores, las precisiones obtenidas utilizando el método *K-Nearest Neighbor* fueron de entre el 44% y el 58%.

6. Artificial Neural Network (ANN)

En el último experimento, se configuró una red neuronal artificial y se probaron variaciones en el número de capas, así como en el lote (o *batch*) y en el número de épocas.

Los resultados se muestran en la tabla 45 y alcanzaron precisiones cercanas al 65%.

Algoritmo ANN
activation_function = 'relu'
activation_function_out = 'softmax'
number of layers = 16
batch = 10
number_of_epochs = 10000

	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	59%	2	62%	3
RUN2	63%	3	60%	4
RUN3	62%	3	64%	3
RUN4	63%	2	61%	4
RUN5	64%	2	59%	4
PROMEDIO	62%		61%	

Algoritmo ANN
activation_function = 'relu'
activation_function_out = 'softmax'
number of layers = 100
batch = 10
number_of_epochs = 10000

	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	62%	2	66%	3
RUN2	65%	3	68%	3
RUN3	64%	2	66%	3
RUN4	64%	2	63%	4
RUN5	66%	2	64%	4
PROMEDIO	64%		65%	

Algoritmo ANN
activation_function = 'relu'
activation_function_out = 'softmax'
number of layers = 100
batch = 10
number_of_epochs = 50000

	Precisión General	Precisión Usuario objetivo	Precisión General	Precisión Usuario objetivo
RUN1	66%	2	63%	3
RUN2	66%	2	68%	4
RUN3	64%	2	66%	4
RUN4	66%	2	64%	4
RUN5	64%	3	65%	4
PROMEDIO	65%		65%	

Tabla 45. Precisión obtenida con una Red Neuronal Artificial según variación del número de capas y número de épocas.

Resumen de resultados

A continuación, se presenta la tabla 46, que resume los resultados obtenidos en la predicción de franjas con los algoritmos utilizados. Aquellos para los que se obtiene mayor precisión fueron *Random Forest*, *Support Vector Classification* (SVC), y sobre todo la red neuronal artificial.

Algoritmo	Métrica evaluación	Función objetivo	Número estimadores	Depth	Precisión General Lavadora	Precisión General Lavavajillas	Incremento de precisión sobre el caso base	
XGBoost	mlogloss	Squared Loss	200	10	50%	52%	Caso Base	
XGBoost	rmse	Squared Loss	200	10	52%	50%	2%	-1%
XGBoost	ıltiobjective err	Squared Loss	200	10	55%	45%	5%	-7%
XGBoost	mlogloss	Squared Loss	400	10	54%	49%	4%	-3%
XGBoost	mlogloss	Squared Loss	800	10	50%	47%	0%	-5%
XGBoost	mlogloss	squared log	400	10	52%	47%	2%	-5%
XGBoost	mlogloss	absolute error	400	10	51%	47%	1%	-5%
XGBoost				**Promedio**	**52%**	**48%**	**2%**	**-4%**
Algoritmo	**Criterio error partición**	**Función objetivo**	**Número estimadores**		**Lavadora**	**Lavavajillas**		
Gradient Boosting	log_loss	friedman mse	200	10	52%	50%	2%	-2%
Gradient Boosting	log_loss	mse	200	10	51%	47%	1%	-4%
Gradient Boosting	log_loss	friedman mse	400	10	51%	47%	2%	-5%
Gradient Boosting				**Promedio**	**51%**	**48%**	**2%**	**-4%**
Algoritmo	**Criterio Partición**	**Número estimadores**	**Random State**		**Lavadora**	**Lavavajillas**		
Random Forest	Gini	200	0		57%	53%	8%	1%
Random Forest	Entropy	200	0		57%	57%	7%	5%
Random Forest	Entropy	100	0		57%	53%	7%	1%
Random Forest	Entropy	400	0		57%	53%	8%	1%
Random Forest	Entropy	200	42		56%	53%	7%	1%
Random Forest				**Promedio**	**57%**	**54%**	**7%**	**2%**
Algoritmo	**Función Kernel**	**Función decisión**			**Lavadora**	**Lavavajillas**		
SVC	poly	ovo			58%	55%	8%	3%
SVC	rbf	ovo			58%	55%	8%	3%
SVC	sigmoid	ovo			54%	52%	4%	1%
SVC	rbf	ovr			57%	55%	8%	3%
SVC				**Promedio**	**56%**	**54%**	**7%**	**3%**
Algoritmo	**Algoritmo**	**Número de estimadores**			**Lavadora**	**Lavavajillas**		
KNN	ball tree	5			54%	44%	4%	-8%
KNN	kd tree	5			52%	44%	2%	-8%
KNN	ball tree	10			55%	53%	5%	1%
KNN	ball tree	50			58%	55%	8%	3%
KNN				**Promedio**	**54%**	**49%**	**5%**	**-3%**
Algoritmo	**Número de capas**	**Número de épocas**			**Lavadora**	**Lavavajillas**		
ANN	16	10000			62%	61%	13%	9%
ANN	100	10000			64%	65%	15%	14%
ANN	100	50000			65%	65%	16%	13%
ANN				**Promedio**	**64%**	**64%**	**14%**	**12%**

Tabla 46. Resumen de los resultados de precisión en la predicción de franjas para todos los algoritmos empleados en la segunda tanda de experimentos.

También se realizó este análisis para la predicción de encendidos.

Como se ha especificado, existen algunos procedimientos para optimizar hiperparámetros de forma automatizada; llevan a cabo pruebas masivas para los diferentes parámetros que toma cada algoritmo. Algunos de estos métodos, como expone Syarif [2016] serían los algoritmos genéticos, la validación cruzada o *Gridsearch*, aunque

es recomendable tener cuidado con los métodos automáticos debido a que cada ronda puede dar lugar a resultados ligeramente diferentes.

Para esta segunda parte de esta tanda de experimentos se utilizó el método de optimización *gridsearch* incluido en el paquete *model_selection*, de *sklearn*:

class `sklearn.model_selection.`**`GridSearchCV`**(*estimator, param_grid, *, scoring=None, n_jobs=None, refit=True, cv=None, verbose=0, pre_dispatch='2*n_jobs', error_score=nan, return_train_score=False*)

Por ejemplo, para el caso de *Random Forest*, se incluyeron los siguientes grupos de parámetros:

```
parameters = {'criterion': ['giny', 'entropy', 'log_loss'],
                  'random_state' : [0, 42],
                  'n_estimators' : [100,500,1000],
                  'max_depth'    : [4,6,10]}
```

Y el resultado fue el siguiente:

```
Results from Grid Search

 The best estimator across ALL searched params:
 RandomForestClassifier(criterion='entropy', max_depth=6, random_st
ate=42)
 The best score across ALL searched params:
 0.5055953715566944
 The best parameters across ALL searched params:
 {'criterion': 'entropy', 'max_depth': 6, 'n_estimators': 100, 'ran
dom_state': 42}
```

Los resultados en la predicción de número de encendidos, se muestran en la tabla 47.

Algoritmo	Métrica evaluación	Función objetivo	Número estimadores	Depth	Precisión General Lavadora	Precisión General Lavavajillas	Incremento de precisión sobre el caso base	
XGBoost	mlogloss	Squared Loss	200	10	47%	46%	Caso Base	
XGBoost	rmse	Squared Loss	200	10	48%	48%	1%	2%
XGBoost				**Promedio**	**48%**	**47%**	**1%**	**2%**
Algoritmo	**Criterio error partición**	**Número estimadores**	**Profundidad**		**Lavadora**	**Lavavajillas**		
Gradient Boosting	log_loss	friedman mse	100	8	52%	50%	5%	4%
Gradient Boosting					**52%**	**50%**	**5%**	**4%**
Algoritmo	**Criterio Partición**	**Número estimadores**	**Random State**		**Lavadora**	**Lavavajillas**		
Random Forest	Entropy	100	42		49%	45%	2%	-1%
Random Forest					**49%**	**45%**	**2%**	**-1%**
Algoritmo	**Función Kernel**	**Función decisión**			**Lavadora**	**Lavavajillas**		
SVC	rbf	ovo			49%	48%	2%	2%
SVC					**49%**	**48%**	**2%**	**2%**
Algoritmo	**Algoritmo**	**Número de vecinos**			**Lavadora**	**Lavavajillas**		
KNN	auto	20			44%	46%	-3%	0%
KNN					**44%**	**46%**	**-3%**	**0%**
Algoritmo	**Número de Capas**	**Número de épocas**			**Lavadora**	**Lavavajillas**		
ANN	100	50000			62%	69%	15%	23%
ANN					**62%**	**69%**	**15%**	**23%**

Tabla 47. Resumen de los resultados de precisión en la predicción de encendidos para todos los algoritmos empleados en la segunda tanda de experimentos.

Para todos estos casos se estimó también el *Mean Absolute Percentual Error* (MAPE), calculado a partir de la siguiente expresión:

```
MAPE = np.mean(abs(y_pred - y_test)/y_pred)
```

Los resultados del MAPE estuvieron en el entorno de 20-25%, lo que quiere decir que la aproximación a la clase correcta correspondería a un entorno de 75-80%.

5. Conclusiones y líneas futuras

A la vista de los resultados se pueden extraer las siguientes conclusiones:

El análisis de datos y la selección de características resultaron ser parte importante en la resolución de este problema de clasificación de consumo en hogares.

Las características más relevantes obtenidas de una curva general de consumo fueron las variables sintéticas creadas para recoger indicadores de presencia en hogar o de teletrabajo, así como indicadores de consumo en fin de semana.

Las características obtenidas de una curva general de consumo basadas en estadísticos, como la desviación típica o los valores de máximos y mínimos, tuvieron mayor impacto en el análisis que los valores de media o suma, que presentan más redundancias y puede ser recomendable no incluirlos en exceso.

Las variables asociadas a picos de consumo son de utilidad, pero no tan importantes como se esperaba frente a las variables estadísticas o las mencionadas previamente, debido al caso concreto que nos ocupa: a la variabilidad de los datos, a la presencia de más de un pico en la curva general total diaria, o al camuflaje de picos debido a la base de agrupación elegida (mensual, anual).

Para el orden de centenares de usuarios, la ampliación de las franjas de consumo utilizadas, lo que supone una reducción del número de franjas, por ejemplo, de 7 a 4, mejora sustancialmente la precisión de la clasificación, incrementándola en órdenes de hasta el 20%. Una solución para mejorar los resultados podría ser aumentar el número de datos de usuarios y comprobar si resultados similares pueden ser obtenidos utilizando las 7 franjas.

La interpretación de las horas límite de las franjas propuestas en las preguntas del cuestionario parece un factor que puede influir en algunos puntos porcentuales sobre el resultado final de la clasificación, por lo que se recomienda matizar bien esos límites en este tipo de preguntas.

Los usuarios que dan como opción una franja no fija de consumo (del tipo “sin un horario fijo”) pueden distorsionar la precisión de la clasificación en varios puntos porcentuales, por lo que se recomienda evitar dar esa opción en los cuestionarios.

Los mejores resultados para el problema de clasificación se alcanzaron con el empleo de redes neuronales artificiales con incrementos de precisión de entre 15 y 23% sobre las del Caso Base, que utilizaba *XGBoost*.

El uso de algoritmos como *Random Forest* o *Support Vector Classification*, logró un incremento de la precisión en la detección de franja de hasta el 7% en la predicción de franjas para la clasificación sobre las del Caso Base.

En la predicción del número de encendidos, el algoritmo *Gradient Boosting* mostró consiguió incrementos de precisión de hasta el 5% sobre la del Caso Base.

Los mejores resultados de precisión estuvieron en torno al 64% en la predicción de franja de uso de lavadora y lavavajillas a partir de los datos de consumo. En cuanto a la predicción del número de encendidos se alcanzaron cifras de entre el 62% y el 69% de precisión.

Estos resultados de precisión, que podrían parecer insuficientes en otros problemas, internamente se consideraron aceptables para el problema actual, debido a la gran complejidad de cálculo mencionada; por un lado, las variables explicativas y las variables objetivo no tienen una relación tan directa entre sí, ya que se parte de curvas de consumo total horario de hogares que incluyen consumos de otros electrodomésticos además de los analizados (lavadora y lavavajillas). En los estudios analizados como referencia, algunos resultados, como los de Yu, arrojan precisiones más altas, pero también emplean un volumen de hogares mayor y datos de más frecuencia. Los estudios de Beckel y Sondenkamp tienen precisiones entre los 50% y los 70%, pero también con un volumen de hogares superior a los 3000 y varían según la característica analizada. Por otra parte, los consumos para todo el territorio español presentan variabilidad que añade dificultad adicional.

El *Mean Absolute Percentual Error* o MAPE fue cercano al 20%, lo que permitió pronósticos con desviaciones menores de este porcentaje.

Como posibles líneas futuras de trabajo para mejorar el comportamiento de la aplicación descrita en este proyecto se sugieren:

- Ampliar el volumen de datos.
- Incluir nuevas características, a partir de datos adicionales de usuario disponibles para todos los usuarios objetivo de la aplicación, como datos de temperatura de la zona, datos de número de personas en el hogar, tamaño del piso, potencia contratada, datos de eficiencia de los equipos obtenidos del cuestionario, etc.

- Completar los datos de franjas correspondientes a "sin un horario fijo" con métodos promedio que aproximasen a valores cercanos de las demás etiquetas.
- Matizar mejor las franjas horarias o las preguntas del cuestionario.
- Completar datos nulos con valores promedio u otros métodos de rellenado de datos vacíos.
- Utilizar métodos de *clustering* aplicados a estas características para estimar posibles segmentaciones adicionales de clientes.

6. Bibliografía

1. Erin, A., Grøgaard B. The long-term energy transition: Drivers, outcomes, and the role of the multinational enterprise. Journal of International Business Studies (2021).

2. Utkucan Ş. Future of renewable energy consumption in France, Germany, Italy, Spain, Turkey and UK by 2030 using optimized fractional nonlinear grey Bernoulli model. Sustainable Production and Consumption. Volume 25, January 2021, Pages 1-14.

3. Doh J., Budhwar P., Wood G. Long-term energy transitions and international business: Concepts, theory, methods, and a research agenda. Journal of International Business Studies (2021).

4. Reinisch, Kofler, Iglesias, Kastner. ThinkHome Energy Efficiency in Future Smart Homes. EURASIP Journal on Embedded Systems volume 2011, Article number: 104617 (2011).

5. Ibarra, D. (03/12/2019). "Repsol hará un ajuste contable de 4.800 millones para lograr cero emisiones netas de CO2". Página web de Energía 16, la sección de energía del diario Cambio 16 [diario web]. www.cambio16.com. Recuperado de: https://www.cambio16.com/repsol-avanza-hacia-las-cero-emisiones-netas-de-co2/ [06/07/2022].

6. Marchal, L. (2021, 30 de marzo). Repsol se transforma en su camino hacia las cero emisiones netas. *El Economista.* https://www.eleconomista.es/ESP500/noticias/11111164/03/21/Repsol-se-transforma-en-su-camino-hacia-las-cero-emisiones-netas.html

7. Sharma R, Mithas S, Kankanhalli A. (2014). Transforming decision-making processes: a research agenda for understandingthe impact of business analytics on organisations. Eur J Inf Syst 23(4):433–441.

8. Jacobs, G., Maun J.C. (2019). Identifying washing machine consumption in supervised global electric consumption. IEEE Milan PowerTech.

9. Michel, A. Bush, E., Nipkow J. (2017). Topten International Group. Dishwashers: Testing the energy variability related to different uses. https://storage.topten.eu/source/files/Topten_Dishwashers_May17.pdf

10. Kwac, J. Household Energy Consumption Segmentation Using Hourly Data. IEEE TRANSACTIONS ON SMART GRID, VOL. 5, NO. 1, JANUARY 2014.

11. Cai, J., Luo, J., Wang, S., Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing Journal.

12. Zhou K, Fu C, Yang S (2016) Big data driven smart energy management: From big data to big insights. Renewable and Sustainable Energy Reviews 56:215–225. 00052.

13. Beckel, C., Sadamori, L., & Santini, S. (2013). Automatic socio-economic classification of households using electricity consumption data. In Proceedings of the fourth international conference on Future energy systems (pp.75-86). Berkeley, CA: ACM.

14. Yu, J., Xu, C., Zhao, J., Xia, X. (2020). Electricity User Consumption Feature Selection and Behavior Portraying. 4th IEEE Conference on Energy Internet and Energy Systems Integration.

15. Sodenkamp M., Hopf K, Staake T. Using Supervised Machine Learning to Explore Energy Consumption Data in Private Sector Housing. Handbook of Research on Organizational Transformations through Big Data Analytics. Chapter 19, January 2014, Pages 320-333.

16. Natekin A., Knoll A., Gradient boosting machines, a tutorial. Front Neurorobot. 2013; 7: 21. Published online 2013 Dec 4. doi: 10.3389/fnbot.2013.00021.

17. Chen, T, Guestrin C. (2016). XGBoost: A Scalable Tree Boosting System.

18. Muhamedyev, R(2015). Machine learning methods: An overview. COMPUTER MODELLING & NEW TECHNOLOGIES 2015 19(6) 14-29.

19. Kotsiantis, S.B. (2007). Supervised Machine Learning: A Review of Classification Techniques.

20. Soofi, A., Awan, A. Classification Techniques in Machine Learning: Applications and Issues, 13, 459-465 (2017) Journal of Basic & Applied Sciences.

21. Schölkopf, B., Christianini, (2002). N. Support Vector Machines and Kernel Methods. The New Generation of Learning Machines. AI Magazine Volume 23 Number 3.

22. Han, J. Pei, J. (2012). Data Mining Concepts and Techniques. Morgan Kaufman. 3rd edition.

23. Dietterich, T. Ensemble Methods in Machine Learning. Multiple Classifier Systems. 2000. WWW home page http://wwwcsorstedu/~tgd.

24. Bauer, E. Kohavi, R. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning 36, 105–139 (1999) Kluwer Academic Publishers. Manufactured in The Netherlands.

25. IDAE. Consumos del Sector Residencial en España. Resumen de Información Básica https://www.idae.es/uploads/documentos/documentos_Documentacion_Basica_Residencial_Unido_c93da537.pdf

26. Ayub, N., Usman, A. (2020). Big Data Analytics for Short and Medium-Term Electricity Load Forecasting Using an AI Techniques Ensembler. Article in Energies.

27. Hino H., Shen H., Murata N., Wakao S., Hayashi Y. A Versatile Clustering Method for Electricity Consumption Pattern Analysis in Households. IEEE Transactions on smart grid, June 2013. Vol 4. No. 2.

28. Hedén W. Predicting Hourly Residential Energy Consumption using Random Forest and Support Vector Regression. An Analysis of the Impact of Household Clustering on the Performance Accuracy. 2016.

29. Kesornsit, W., Sirisathitkul Y.. Hybrid Machine Learning Model for Electricity Consumption Prediction Using Random Forest and Artificial Neural Networks. Applied Computational Intelligence and Soft Computing. Volume 2022, Article ID 1562942, 11 pages. Hindawi. 2022.

30. Nordahl, C, Boeva, B., Grahn, H., Person Netz, M.(2019). Profiling of Household Residents' Electricity Consumption Behavior using Clustering Analysis. Blekinge Institute of Technology, 371 79 Karlskrona, Sweden.

31. Hsu, C., Chang, C., Lin, C. (2016). A Practical Guide to Support Vector Classification. Department of Computer Science. National Taiwan University, Taipei 106, Taiwan. http://www.csie.ntu.edu.tw/~cjlin.

32. Syarif, I., Wills, G., Prugel-Bennet, A. (2016) SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. Article in TELKOMNIKA (Telecommunication Computing Electronics and Control). DOI: 10.12928/telkomnika.v14i4.3956

Anexo: Estructura del código

El desarrollo del código se llevó a cabo utilizando el esquema que se muestra en la figura 30.

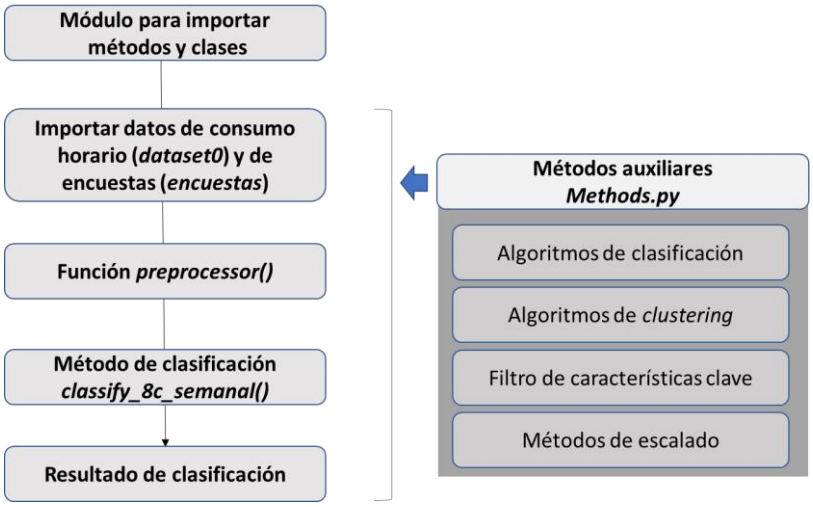

Figura 30. Esquema de la estructura del código.

En primer lugar, se incluyeron módulos para importar los métodos, clases y procedimientos necesarios para el desarrollo del código, tal como se muestra en la figura 31.

```
import requests
import pandas as pd
from datetime import date, timedelta
import datetime
import numpy as np

import os

from sklearn import datasets
from sklearn.mixture import GaussianMixture
from sklearn.model_selection import StratifiedKFold
from sklearn import mixture
from scipy.stats import kstest
from scipy import stats
from scipy.spatial import distance
from scipy.stats import kstest,chisquare
from sklearn.metrics import silhouette_samples, silhouette_score

from scipy.signal import find_peaks

from sklearn.cluster import KMeans
from sklearn.impute import SimpleImputer
from sklearn.mixture import GaussianMixture

import requests
import pandas as pd
from datetime import date, timedelta
import datetime
import numpy as np
from numpy import max
#import ruptures as rpt
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import pyplot
from mpl_toolkits.mplot3d import Axes3D
```

Figura 31. Fragmento del módulo de importación de métodos.

Entre ellos, el archivo *methods.py* (figura 32), que contenía el código de los algoritmos de clasificación y *clustering* utilizados, como *Gradient Boosting*, *K-means* y *XGBoost*, entre otros, para varios tipos de clase o categorías analizadas (número de franjas etc), así como métodos adicionales auxiliares.

```
import datetime
import numpy as np
#import ruptures as rpt
import matplotlib
import matplotlib.pyplot as plt


# General

def intersection(lst1, lst2):
    lst3 = [value for value in lst1 if value in lst2]
    return lst3

def precios_omie():
    files = os.listdir('/deploy3/data_jmpalomar/TFM/DATA/marginalpdbc_2021/')
    numero_files = len(files)
    dataframe = pd.DataFrame()
    for f in files:
    #for f in range(1,numero_files):
        dataset_i = pd.read_csv('/deploy3/data_jmpalomar/TFM/DATA/marginalpdbc_2021/'+ f, sep = ';', skipinitialspace=True, skiprows =
[0,25], header = 0, names = ['year', 'month', 'day', 'hour', 'price', 'price2', 'other'])
        #encoding='utf-8'
        dataframe_i = pd.DataFrame(data = dataset_i)
        dataframe = dataframe.append(dataframe_i)
    dataframe2 = dataframe.copy()
    dataframe2.reset_index()
    dataframe2 = dataframe2.drop(columns = ['price2', 'other','year'])
    ##dataframe2 = dataframe.iloc[:,0:-2]
    dataframe2 = dataframe2.reset_index(drop = True)
    return dataframe2

# Unsupervised clustering execution

def unsupervised(dataset):
    dataset = dataset0.copy()
    dataset = dataset.dropna() #eliminamos valores nulos antes de aplicar las métricas
    dataset['CUPS_20'] = dataset['email']
    dataset = dataset.drop(columns = ['email'])
```

Figura 32. Fragmento del módulo methods.py donde se programaron los algoritmos.

Después se incluyó la lectura de los datos en formato **.cvs*, tanto para consumo horario, que se almacenan en el *dataframe* 'dataset0', como para las encuestas, que se almacenan en el *dataframe* 'encuestas'. Se muestra en la figura 33.

```
dataset0 = pd.read_csv('/deploy3/data_jmpalomar/TFM/DATA/consumos_total.csv', encoding='utf-8', sep = ';')
encuestas = pd.read_csv('/deploy3/data_jmpalomar/TFM/DATA/respuestas.csv', encoding='utf-8', sep = ';')
```

Figura 33. Fragmento de código correspondiente a la lectura de datos de consumo y del cuestionario.

Los datos del 'dataset0' se estructuraron en 26 columnas, 24 columnas correspondientes a las 24 horas, más dos adicionales para el indicador de usuario y la fecha del día. De esta forma para cada usuario, cada día se representó como un vector de 24 consumos horarios como fila del *dataframe*.

Se realizó un preprocesado del 'dataset0' para poder estructurarlo de esta forma. En la figura 34 se muestra un fragmento de la función utilizada para ello, *preprocessor(dataset, encuesta)*, que toma como argumentos los datos de consumo y los datos de la encuesta y retorna el *dataset* procesado y listo para aplicar el procedimiento de clasificación, así como los *dataset* separados para la clasificación de la lavadora y el lavavajillas. En esta función se realizaron las siguientes operaciones:

1. Preprocesado de los datos para eliminar datos ausentes.
2. Cálculo de características.
    a. Incorporar las franjas horarias. Inicialmente 7 y posteriormente 4.
    b. Incorporar características asociadas a picos y adyacentes al pico de consumo.
    c. Incorporar nivel de renta, obtenido a partir del código postal.
3. Estructurar los datos disponiendo un usuario por fila y una característica por columna. Para ello se realizaron bucles para incorporar las características a partir del *dataframe* original.
4. Combinación o *merge* del *dataset* de consumo y del *dataset* de los datos de la encuesta, garantizando la no repetición de líneas de usuarios.
5. Separación de características generales con característica objetivo de la hora de la lavadora para conformar el *dataset* de la lavadora.
6. Separación de características generales con característica objetivo de la hora del lavavajillas para conformar el *dataset* del lavavajillas.

```
def preprocessor (dataset0, encuestas):

############ ADAPTAMOS EL DATASET Y DEFINIMOS 4 FRANJAS HORARIAS: MADRUGADA MAÑANA TARDE Y NOCHE ################

dataset = dataset0.copy()
dataset = dataset.dropna() #eliminamos valores nulos antes de aplicar las métricas
dataset['CUPS_20'] = dataset['email']
dataset = dataset.drop(columns = ['email'])
#estimamos fecha y mes
dataset['date'] = pd.to_datetime(dataset['fecha'], format = "%d/%m/%Y")
#dataset['month'] = dataset['date'].dt.strftime('%b')
dataset['month'] = dataset['date'].dt.month
dataset['week'] = dataset['date'].dt.isocalendar().week
min_month = dataset['month'].min()
max_month = dataset['month'].max()

lista_usuarios = pd.unique(dataset['CUPS_20'])

#Eliminamos fechas >20/11 para evitar nan adicionales
#dataset = dataset.drop(dataset[dataset['date']>'2021-11-14 00:00:00'].index)

#Eliminamos la columna date
dataset = dataset.drop(columns = ['date'])

#agrupamos las categorías 1,2,3,4,5,6,7
#dataset['madrugada'] = dataset.iloc[:,[1,2,3,4,5,6,23,24]].sum(axis=1) # de 01 a 06 incluida + de 23 a 24 incluida
#dataset['mañana'] = dataset.iloc[:,7:13].sum(axis=1) # de 07 a 12 incluida

#Añadimos nuevas features obtenidas como sumas de cada franja horaria
dataset['amanecer_s'] = dataset.iloc[:,7:10].sum(axis=1) # de 07 a 09 incluida
dataset['mañana_s'] = dataset.iloc[:,10:13].sum(axis=1) # de 10 a 12 incluida
dataset['mediodía_s'] = dataset.iloc[:,13:16].sum(axis=1) # de 13 a 15 incluida
```

Figura 34. Fragmento de código de la función preprocessor() que sirvió para adecuar los datos a la forma de trabajo del método de clasificación.

A continuación, se programaron los diversos métodos de clasificación. Como se muestra en la figura 35, este tipo de métodos toman como argumentos el identificador del usuario objetivo, el *dataset* y el número de características clave requeridas.

```
def classify_8c_semanal(usuario_objetivo, dataset0, number_of_key_features):
    usuario_obj = usuario_objetivo
    dataset = dataset0.copy()
    dataset['date'] = pd.to_datetime(dataset['fecha'], format = "%d/%m/%Y")
    dataset['week'] = dataset['date'].dt.week
    min_week = dataset['week'].min()
    max_week = dataset['week'].max()
    #print('min_week',min_week)
    #print('max_week',max_week)
    numero_semanas = max_week-min_week
    dataset_obj = dataset[dataset['email']==usuario_obj]
    dataset_wo_obj = dataset[dataset['email']!=usuario_obj]
    #print('dataset_obj', dataset_obj)
    #print('dataset_wo_obj', dataset_wo_obj)

    prediccion_wm = []
    prediccion_dw = []
    precision_wm = []
    precision_dw = []
    validacion_wm = []
    validacion_dw = []

    #for m in range(min_month,6):
    #for w in range(min_week,max_week):
    for w in range(39,44):
        try:
            print('##### SEMANA %i #####' %w)
            dataset_semanal = dataset_wo_obj[dataset_wo_obj['week'] == w]
            dataset_semanal = dataset_semanal.drop(columns = ['date','week'])
            #print('dataset_semanal', dataset_semanal)
            ### elabora un dataset con todos los usuarios para esa semana y las características definidas en preprocessor (suma,
            C_wm, C_dw, dataset_enriched = preprocessor(dataset_semanal, encuestas)
            #print('dataset_semanal',dataset_semanal)
            #print('dataset_enriched',dataset_enriched)
```

Figura 35. Fragmento de código del método de clasificación.

El método realiza las siguientes operaciones:

- Extrae del *dataset* el usuario objetivo.
- Agrupa los datos del *dataset* en la base temporal deseada, semanal, mensual, o anual.
- Aplica la función *preprocessor()* al *dataset* para obtener el *dataset preprocesado.*
- Aplica el algoritmo de clasificación al *dataset preprocesado* y almacena el clasificador obtenido en la base temporal especificada (semana, mes, año), a la vez que arroja un valor de la precisión tras la validación cruzada.
- Procesa los datos del usuario objetivo para adaptarlos al formato del clasificador.
- Aplica el clasificador al usuario objetivo.
- Arroja valores de precisión del algoritmo para cada unidad de la base tomada (semana, mes, año) y la franja para el usuario objetivo para esas mismas unidades de base.