Trabajo Fin de Máster

Predicción del consumo eléctrico de una fábrica para una planificación de la producción dada, partiendo del análisis estadístico de los consumos y producciones históricas

Pablo Álvarez González

Máster en Técnicas Estadísticas

Curso 2023-2024

Propuesta de Trabajo Fin de Máster

Título en galego: Predición do consumo eléctrico dunha fábrica para unha determinada planificación da produción, a partir da análise estatística dos consumos e producións históricas
Título en español: Predicción del consumo eléctrico de una fábrica para una planificación de la producción dada, partiendo del análisis estadístico de los consumos y producciones históricas
English title: Prediction of the electricity consumption of a factory for a given production planning, based on the statistical analysis of historical consumption and production
Modalidad: Modalidad B
Autor/a: Pablo Álvarez González, Universidade da Coruña
Director/a: Salvador Naya Fernández, Universidade da Coruña; Javier Tarrío Saavedra, Universidade da Coruña
Tutor/a: José Manuel Fernández Bouzo, Financiera Maderera, S.A.
Breve resumen del trabajo: Análisis de diferentes herramientas estadísticas con las que enfocar la tarea de predicción de consumos eléctricos para la empresa Finsa.
Recomendaciones:
Otras observaciones:

Don/doña Salvador Naya Fernández, categoría 1 de la Universidade da Coruña, don/doña Javier Tarrío Saavedra, categoría 2 de la Universidade da Coruñay don/doña José Manuel Fernández Bouzo, Energy Management en Finsa de Financiera Maderera, S.A., informan que el Trabajo Fin de Máster titulado

Predicción del consumo eléctrico de una fábrica para una planificación de la producción dada, partiendo del análisis estadístico de los consumos y producciones históricas

fue realizado bajo su dirección por don/doña Pablo Álvarez González para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En A Coruña, a 22 de julio de 2024.

NAYA FERNANDEZ SALVADOR - DNI 76353731X
Firmado digitalmente por NAYA FERNANDEZ SALVADOR - DNI 76353731X
Fecha: 2024.07.22 22:43:23 +02'00'

El/la director/a:
Don/doña Salvador Naya Fernández

TARRIO SAAVEDRA JAVIER - DNI 53164914T
Firmado digitalmente por TARRIO SAAVEDRA JAVIER - DNI 53164914T
Fecha: 2024.07.22 21:15:06 +02'00'

El/la director/a:
Don/doña Javier Tarrío Saavedra

El/la tutor/a:
Don/doña José Manuel Fernández Bouzo

El/la autor/a:
Don/doña Pablo Álvarez González

Declaración responsable. Para dar cumplimiento a la Ley 3/2022, de 24 de febrero, de convivencia universitaria, referente al plagio en el Trabajo Fin de Máster (Artículo 11, Disposición 2978 del BOE núm. 48 de 2022), **el/la autor/a declara** que el Trabajo Fin de Máster presentado es un documento original en el que se han tenido en cuenta las siguientes consideraciones relativas al uso de material de apoyo desarrollado por otros/as autores/as:

- Todas las fuentes usadas para la elaboración de este trabajo han sido citadas convenientemente (libros, artículos, apuntes de profesorado, páginas web, programas,...)
- Cualquier contenido copiado o traducido textualmente se ha puesto entre comillas, citando su procedencia.
- Se ha hecho constar explícitamente cuando un capítulo, sección, demostración,...sea una adaptación casi literal de alguna fuente existente.

Y, acepta que, si se demostrara lo contrario, se le apliquen las medidas disciplinarias que correspondan.

Índice general

Resumen

Resumen en español

Si se tuviese un conocimiento *a priori* del consumo que supondría llevar a cabo una producción dada, podría utilizarse esta información para optimizar la planificación, algo que resulta fundamental en el ámbito industrial. Esto, no solamente supondría un inmediato beneficio en términos de reducción de costes, sino que también aportaría un conocimiento más profundo sobre el propio producto.

En este trabajo, se parte de dos conjuntos de datos. El primero, contiene los datos históricos de producción de tablero, y cuenta con variables relacionadas con sus dimensiones, la cantidad de metros cuadrados producidos o el momento de inicio y fin de la producción. El segundo, contiene las medidas minutales históricas de consumo eléctrico de diversas variables en kilovatios hora. El objetivo propuesto es relacionar estas producciones y consumos y, con ello, tratar de predecir el consumo eléctrico de producciones futuras.

Para obtener dichas predicciones, se propone la utilización de diversos métodos de aprendizaje supervisado en los que se entrena un modelo con ejemplos u observaciones pasadas de los datos. Estos modelos, pueden medirse posteriormente en términos de error y variabilidad explicada de la respuesta al ser evaluados sobre datos nuevos, permitiendo así hacer después una comparación entre sus desempeños y escoger el que mejor se adapte a nuestros criterios.

English abstract

If there were *a priori* knowledge of the consumption that would be involved in carrying out a given production, this information could be used to optimize planning, something that is essential in the industrial field. This would not only have an immediate benefit in terms of cost reduction, but would also provide a deeper understanding of the product itself.

In this project, two data sets are used as a starting point. The first contains historical board production data, and includes variables related to its dimensions, the number of square meters produced or the start and end time of production. The second contains the historical minute measurements of electricity consumption for various variables in kilowatt hours. The proposed objective is to relate these productions and consumptions and, with this, try to predict the electricity consumption of future productions.

In order to obtain these predictions, we propose the use of various supervised learning methods in which a model is trained with examples or past observations of the data. These models can be subsequently measured in terms of error and explained variability of the response when evaluated on new data, thus allowing a subsequent comparison between their performances and choosing the one that best suits our criteria.

Capítulo 1

Introducción

1.1. La industria de la madera

El contexto en el que se enmarca esta memoria es el del sector de la madera. Se trata de una industria global que abarca desde la silvicultura[1] y la gestión forestal, hasta la producción y comercialización de productos de madera. Entre estos productos, puede hablarse también de la madera como combustible, un recurso utilizable estratégicamente como suministro de energía y que va en la dirección del acuerdo de Kioto de reemplazar los combustibles fósiles y mitigar los gases de efecto invernadero (Hillring, 2006).

En los últimos años, el sector ha enfrentado numerosos desafíos. En primer lugar, la pandemia de la COVID-19 causó grandes complicaciones en la cadena de suministro, afectando fundamentalmente en la disponibilidad y en el coste de los materiales. Sin embargo, las limitaciones de movilidad y ocio, junto con los confinamientos, provocaron que la construcción, rehabilitación y reforma del hábitat se convirtieran en un gasto prioritario (Financiera Maderera S.A., 2021). Posteriormente, la guerra en Ucrania no favorecería la situación y añadiría cierta incertidumbre, influyendo tanto en las dinámicas comerciales como en la estabilidad del mercado (Malliris, 2023). De hecho, en la segunda mitad del ejercicio 2022, se hace notar una tendencia de disminución de la demanda, fundamentalmente debido a la reducción de renta disponible en los hogares y consecuente caída del consumo en el sector en favor de otros productos y bienes de primera necesidad (Financiera Maderera S.A., 2022).

Una parte esencial de este sector es la sostenibilidad y el cumplimiento de las regulaciones ambientales. La gestión responsable de los bosques es garantizada por certificaciones como FSC (Forest Stewardship Council) y PEFC (Programme for the Endorsement of Forest Certification), y se preocupa y encarga de que la madera sea obtenida de manera sostenible, de modo que se protejan los ecosistemas y la biodiversidad. Las empresas en este sector también están invirtiendo en innovación para mejorar la eficiencia de sus procesos y desarrollar productos con menor impacto ambiental. Podría mencionarse, por ejemplo, la madera laminada cruzada o contralaminada (CLT), cada vez más popular en la construcción debido a sus beneficios ecológicos y estructurales, la cual consiste en tablones de madera aserrada y encolada, donde cada capa es orientada perpendicularmente a la capa anterior.

Para finalizar, se puede hablar de la industria de la madera como un sector imprescindible para la economía global. Destaca como fuente importante de generación de empleo a escala global considerando la amplia gama de trabajos que pueden realizarse, o las distintas etapas que componen cada uno de ellos. Además, la madera tiene el valor añadido de su versatilidad: una característica que hace que pueda ser utilizada tanto para la fabricación de muebles, la construcción, la elaboración de papel o incluso la elaboración de productos químicos.

[1] Del lat. *silva* 'selva, bosque' y *-cultura*: literalmente el cuidado o cultivo de los bosques.

1.2. Sobre Financiera Maderera, S.A.

Financiera Maderera S.A., popularmente conocida como Finsa, es una de las empresas líderes en el sector de la madera y derivados en Europa, destacando por su innovación, calidad y compromiso con la sostenibilidad. Fundada en 1931 en Santiago de Compostela, Finsa ha evolucionado de una pequeña serrería a un conglomerado industrial que desde los años setenta ya cuenta con presencia internacional. Tal y como se recoge en Financiera Maderera S.A. (2022), la compañía conserva su apuesta por dicha presencia y cuenta en la actualidad con 10 plantas productivas cercanas a los puertos de carga, 16 delegaciones comerciales propias en 10 países, 9 plataformas logísticas en diferentes puntos de Europa tales como Holanda, Irlanda, Polonia o Reino Unido y un departamento de exportaciones que opera con clientes de todo el mundo. En lo referente a sus factorías, Finsa cuenta con 10 fábricas repartidas por la Península Ibérica y Francia. De las 7 factorías que tiene en España, 6 pertenecen a la Comunidad Autónoma de Galicia (Santiago de Compostela, Padrón, Rábade, San Ciprián das Viñas, Coirós, Caldas de Reis) y la restante se encuentra en Cella, Teruel. Fuera de España cuenta con 2 factorías en Portugal (Gafanha de Nazaret y Nelas) y una en Francia (Ambarès-et-Lagrave) (Financiera Maderera S.A., 2022).

A lo largo de sus más de 90 años de historia, la empresa ha mantenido un enfoque constante en la diversificación y mejora de sus procesos productivos, cuidando cada detalle en la cadena de producción y fundamentando su proceso industrial en el sistema de economía circular. En un inicio, Finsa centraba su actividad en la producción de tableros de madera, lo cual a día de hoy sigue teniendo gran peso dentro de la empresa (Colaboradores de Wikipedia, 2024). Actualmente, se ocupa de diversos segmentos del sector maderero que incluyen la producción de tablero aglomerado de partículas, MDF (Medium Density Fiberboard), tablero contrachapado y componentes para la industria del mueble y la construcción. En definitiva, se ocupa de la transformación industrial de la madera, diseñando y fabricando soluciones técnicas para todo tipo de espacios.

La empresa se distingue por su capacidad de integrar la gestión forestal sostenible con tecnologías avanzadas de fabricación, de modo que garantiza productos de alta calidad a la vez que minimiza el impacto ambiental que pueda llegar a suponer. El compromiso de Finsa con la sostenibilidad es patente con su amplia colección de certificados que acreditan el seguimiento de unos estándares internacionales como los mencionados anterioremente (FSC y PEFC), así como también el cumplimiento de normativas ISO (International Organization for Standardization).

Para más información acerca de la empresa puede consultarse la propia web de Finsa (en la cual se pueden consultar los Estado de Información No Financiera revisados (Financiera Maderera S.A. 2021, 2022) o su entrada de la Wikipedia (Colaboradores de Wikipedia, 2024).

1.3. El problema propuesto

1.3.1. Planteamiento del problema

El objetivo principal del presente proyecto es encontrar el modo de predecir el consumo eléctrico requerido o que puede asociarse a una producción de tablero dada. Para ello, se hará uso tanto del histórico de datos de consumo eléctrico, como de los partes de producción facilitados por la empresa. En particular, el estudio se centrará en la fábrica de Finsa situada en San Ciprián das Viñas (Ourense), denominada Orember. Se escoge esta ubicación ya que se trata de una fábrica monoproducto, de modo que favorecerá la manipulación de los datos y análisis de resultados. El producto en cuestión es el tablero de MDF o tablero de fibra de densidad media, mencionado en la sección anterior. El tablero de MDF es un producto fabricado a partir de fibras de lignocelulosa obtenida de maderas seleccionadas, ligadas con resinas sintéticas bajo presión a altas temperaturas y que, junto con el tablero de partículas, conforman la principal actividad de Finsa (Financiera Maderera S.A. 2022).

1.3.2. El proceso de fabricación de MDF

Es fundamental, a la par que interesante, entender el proceso de fabricación de tablero de MDF para identificar los puntos clave y posteriormente realizar el análisis y empleo de técnicas estadísticas. El proceso principalmente consiste en las siguientes etapas:

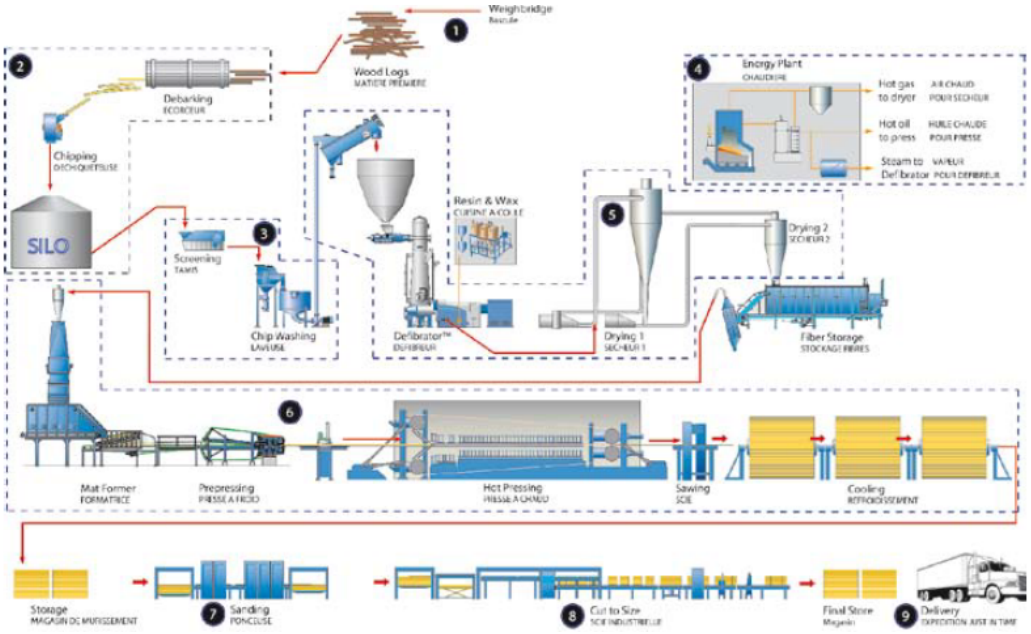


Figura 1.1: Proceso de producción de MDF (Fuente: MDF Brannhemmende EPD. https://lium.no/media/dokumentasjon/epd/mdf-brannhemmende-epd.pdf, Accedido 7 de julio de 2024)

- **Obtención de fibra.** El objetivo en esta fase es el de obtener una buena calidad de fibra, lo cual es esencial para conseguir un MDF de buenas propiedades. Se pueden diferenciar tres etapas:
  - Madera y astillado. La materia prima, es decir, los troncos de madera, debe ser descortezada y a continuación astillada. La astilla resultante debe tener un tamaño adecuado para lograrse una buena digestión. Por ese motivo, se realiza también un cribado en el que se elimina la astilla demasiado fina (genera polvo) o demasiado gruesa (no cuece bien).
  - Desfibrado. Se trata de un proceso termomecánico en el que se rompen las paredes celulares de la madera bajo la acción de la temperatura y el esfuerzo mecánico. Para ello se aplica previamente un prevaporizado, el cual es importante para el reblandecimiento de la astilla, su correcta digestión y preservación de longitud. A continuación, se hace pasar por un tornillo extrusor y finaliza la digestión de la astilla en el digestor.
  - Refino. El refino es también un proceso termomécanico en el cual la astilla termina de transformarse en fibra. Se trata de una etapa clave dentro del proceso de fabricación de tablero de MDF, pues las condiciones de refino determinarán la cantidad y calidad de fibra. Hay distintas variables y parámetros a tener en cuenta en este proceso:
    - La energía específica (SEC): es la potencia eléctrica aplicada por unidad de fibra.
    - Los discos de refino en términos de diseño, tamaño y aleación: consisten en un *rotor* y un *stator* enfrentados entre sí. Estos transforman la astilla en fibra mediante un proceso de corte —que puede ser mediante cizalla o compresión— y/o fibrilación. Son

unos elementos que afectan directamente a la calidad y producción de fibra, así como a la energía específica.

- Separación de los discos: debido al desgaste progresivo producido por los segmentos de refino, deben reajustarse muy a menudo para mantener las condiciones deseadas de SEC y calidad de fibra.
- Posición de la válvula de salida: es una posición regulable para poder obtener un compromiso entre cantidad y calidad de fibra, dado que afecta a la presión diferencial y, por tanto, al tiempo de estancia de fibra en la zona interdiscal.

Según se menciona en Benthien et al. (2014), las condiciones de refino afectan directamente a las características de la fibra y en consecuencia a las propiedades físicas y mecánicas del MDF. En dicho artículo, se llega a concluir que la especie de la madera, las condiciones del digestor y la distancia entre discos son los parámetros más influyentes en la calidad de la fibra.

- **Encolado.** Habitualmente se realiza en húmedo en la *blow-line*[2], antes del secadero. En el encolado, se dosifica una cantidad exacta de adhesivo que se distribuye de manera uniforme. Para obtener propiedades específicas del tablero, en esta fase pueden añadirse otro tipo de aditivos al adhesivo mediante bombas independientes. Estas sustancias mejorarían, por ejemplo, la resistencia a la humedad o al fuego.

- **Secado de fibra.** El secado de la fibra consiste en un transporte neumático con aire caliente, habitualmente dividido en dos etapas. En esta fase, se reduce la humedad de la fibra encolada desde un 80 - 130 % inicial hasta el 6 - 12 % final. Aproximadamente, el 80 % de la humedad es eliminado en la primera etapa. En cuanto a la temperatura de entrada, esta debe ser inferior a 200℃ en la primera etapa, y se deben evitar temperaturas superiores a 100℃ en la segunda, por riesgo de incendio.

- **Formación de la manta.** Después del secado, la fibra se almacena en un silo para abastecer la unidad de formación. Esta fibra se va depositando transversalmente sobre una cinta transportadora para formar la manta con una distribución de peso/densidad lo más homogénea posible. El peso de la manta es prefijado para cada espesor/calidad, y la velocidad de formación es marcada por la prensa. En esta etapa también se lleva a cabo un trabajo de preprensado mecánico de la manta, necesario para darle consistencia, eliminar el aire ocluido, reducir su espesor y aumentar su densidad, el cual facilitará el trabajo posterior de la prensa. Es habitual contar en este punto del proceso con un detector de metales —para prevenir posibles incidencias— y equipos adicionales de calentamiento de manta, como en el caso de Orember, que cuenta con microondas.

- **Prensa.** Posteriormente, la manta es dirigida a una prensa continua de platos calefactados con aceite térmico circulante en su interior, en un circuito cerrado. En la prensa, la cola comienza a endurecerse y la manta se compacta.

- **Salida de prensa y acabado.** Tras la salida de prensa, se realiza un corte de tablero con una medida bruta inicial. En esta fase, se llevan a cabo un control de defectos y características, y se cuenta con mesas de rechazo para tableros defectuosos. Los tableros, una vez formados, se dejan reposar en enfriadores que permiten reducir la temperatura de ambas caras, como se ve en la Figura 1.2. Posteriormente, se apilan los tableros en paquetes para su correcta estabilización y redistribución de humedades.

  En el paso siguiente, se realiza un lijado en serie en un sentido longitudinal consistente en la eliminación del sobreespesor o la cara quemada del tablero. Para ello, se debe optimizar en prensa el espesor bruto inicial: si los espesores brutos de salida de prensa son altos, generan un

[2]Conducto habitualmente utilizado en la industria que permite el transporte o el flujo continuo de materiales y consecuente mejora de la productividad.

mayor coste innecesario y una peor calidad en las caras; por el contrario, espesores muy bajos pueden generar defectos a la hora de realizar el lijado.

A continuación, se lleva a cabo un corte a medida final, seguido de una comprobación de tolerancias de longitud y escuadría. Por último, el producto es embalado y preparado para almacenaje y posterior distribución al cliente.

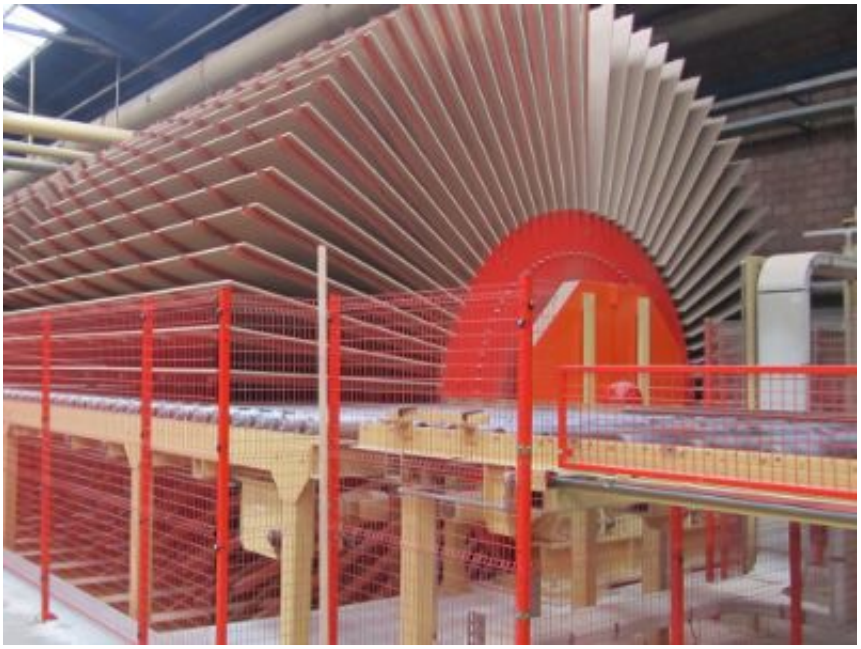


Figura 1.2: Sistema de enfriamiento de los tableros de MDF (Fuente: Cooling system for wood panels (2022) Schermesser Handing Systems https://www.schermesser.fr/en/cooling-system-for-wood-panels/, Accedido 7 de julio de 2024).

El producto final puede resultar muy diferente uno de otro, según los parámetros considerados en el proceso completo de fabricación. Por ejemplo, atendiendo a sus densidades, el producto puede ser estándar, de alta densidad o de baja densidad. A su vez, cada uno de estos productos puede ser ignífugo, hidrófugo, normal, para exterior... o una combinación de dichas propiedades y, dado que también se realiza un corte a medida final, las magnitudes de espesor, largo y ancho también variarán.

Capítulo 2

Descripción de los datos

2.1. Enfoque del problema

El objetivo principal de este proyecto es el de encontrar una relación entre el consumo eléctrico y la cantidad generada de producto. Concretamente se pretende, dada una planificación de la producción, ser capaces de predecir cuánto consumo eléctrico supondría si consideramos todos los puntos de consumo involucrados. Para ello, la empresa proporciona un dataset con información correspondiente a partes de producción históricos, y una plataforma digital en la que se puede consultar tanto el consumo eléctrico actual como el histórico.

El consumo eléctrico total de fabricación de tablero de MDF puede dividirse en las siguientes líneas de producción: astillado, MDF, lijado, escuadrado y texturizado. Al mismo tiempo, estas unidades de operación pueden desglosar su consumo en elementos más específicos. Por ejemplo, el consumo de la línea de MDF engloba otros 7 consumos: el consumo de las calderas, el consumo de las colas, el refino y el secado, el consumo de los transportes y las aspiraciones, el consumo de la formación, el consumo de la prensa y el estabilizado, el consumo del tratamiento de aguas y el consumo indirecto. En la plataforma en que se encuentran estos datos de consumo eléctrico, las medidas pueden obtenerse minutales, 15 minutales, horarias o diarias. En nuestro caso, dado que trataremos de asociar los kilovatios hora (kWh) consumidos durante el tiempo que se lleva a cabo una cierta producción con marca horaria, se tomarán las medidas de manera minutal para contar con una mayor rejilla de valores.

Los partes de producción con los que se cuenta únicamente tienen datos de la producción correspondiente a las líneas de MDF y lijado, a las que en lo sucesivo también puede referirse como 232 y 35, respectivamente. Estos datos abarcan desde inicios del año 2021 hasta abril del año 2024. No obstante, por diversos cambios en metodologías o sistemas de medición que hacen que los datos recopilados sean algo distintos, se opta por considerar solamente los datos que van desde enero de 2022 hasta abril de 2024 en el caso de la Línea 232.

2.2. Dos líneas de producción

Como se acaba de comentar, los partes de producción cuentan con un identificador de línea, según pertenezcan a la de MDF o a la de lijado. Las variables que se consideran son las siguientes:

- `linea`: variable de tipo factor con valores 232 o 35.
- `linea_txt`: variable de tipo carácter que contiene la etiqueta de la línea; TABLERO OREMBER-VI o TABLERO OREMBER-VI LIJAD.
- `fecha_produccion`: variable que indica el día, mes y año del parte de producción.

- `turno`: variable de tipo factor que indica el turno al que corresponde el parte de producción. Puede tomar los valores A, B, C, D y E.
- `hora_inicio`: variable numérica que indica la hora a la que comienza la producción. Se presenta en formato "hhmmss".
- `hora_fin`: variable numérica que indica la hora a la finaliza la producción. Se presenta en formato "hhmmss".
- `num_parte`: identificador del parte de producción.
- `material`: identificador del material.
- `modelo_obsydian`: tipo de modelo o gama de producto generado.
- `grueso_numer`: espesor del tablero en milímetros.
- `ancho`: ancho del tablero en milímetros.
- `largo`: largo del tablero en milímetros.
- `m2_lijado`: metros cuadrados totales lijados.
- `m3_lijado`: metros cúbicos totales lijados.
- `cant_m2_prod`: cantidad total producida en metros cuadrados.
- `cant_m2_prod_util`: cantidad producida útil en metros cuadrados.
- `cant_m3_prod`: cantidad total producida en metros cúbicos.
- `cant_m3_prod_util`: cantidad producida útil en metros cúbicos.

Cuadro 2.1: Ejemplos de partes de producción del dataset inicial de las líneas 232 y 35.

linea	linea_txt	fecha_produccion	turno	hora_inicio	hora_fin	num_parte	material	modelo_obsydian	grueso_mm	ancho	largo	m2_lijado	m3_lijado	cant_m2_prod	cant_m2_prod_util	cant_m3_prod	cant_m3_prod_util
232	TABLERO OREMBER-VI	2024-04-11	C	60000	71235	307297	40197192	FIBRAPAN S/L	19	2100	3660	0	0	2451.834	2451.834	46.584	46.584
35	TABLERO OREMBER-VI LIJAD	2024-04-10	B	0	140000	318854	40020781	FIBRAPAN	19	2100	5700	4764.06	90.517	4764.06	4764.06	90.517	90.517
35	TABLERO OREMBER-VI LIJAD	2024-04-10	B	0	0	318859	40020781	FIBRAPAN	19	2100	5700	11.97	0.227	11.97	0	0.227	0
35	TABLERO OREMBER-VI LIJAD	2024-04-10	D	0	0	318994	40021755	FIBRAPAN	18	2440	3050	2627.026	47.286	2627.026	2627.026	47.286	47.286
35	TABLERO OREMBER-VI LIJAD	2024-04-10	D	0	60000	319001	40021994	FIBRAPAN	22	2100	3660	722.484	15.894	722.484	722.484	15.894	15.894
35	TABLERO OREMBER-VI LIJAD	2024-04-10	A	0	220000	318912	40022694	FIBRAPAN	16	2440	4880	3226.851	51.629	3226.851	3226.851	51.629	51.629
35	TABLERO OREMBER-VI LIJAD	2024-04-10	A	0	220000	318922	40023231	FIBRAPAN FORMA	18	2440	3050	5961.042	107.298	5961.042	5961.042	107.298	107.298
35	TABLERO OREMBER-VI LIJAD	2024-04-10	D	0	60000	318985	40023276	FIBRAPAN FORMA	18	2440	4880	2667.212	48.009	2667.212	2667.212	48.009	48.009
232	TABLERO OREMBER-VI	2024-04-10	B	121607	135902	307118	40023302	FIBRAPAN FORMA S/L	18	2440	3050	0	0	5224.284	5224.284	94.037	94.037
232	TABLERO OREMBER-VI	2024-04-10	A	140000	141336	307127	40023302	FIBRAPAN FORMA S/L	18	2440	3050	0	0	751.642	751.642	13.529	13.529

Nótese la existencia de las variables `cant_m2_prod_util` y `cant_m3_prod_util`. Como su propio nombre indica, hacen referencia a la cantidad de metros cuadrados (o cúbicos) útiles producidos en cada parte de producción, que siempre será menor o igual que la cantidad bruta producida y almacenada en las otras variables `cant_m2_prod` y `cant_m3_prod`. A pesar de ser positivo poder contar con ambas variables, en este caso solamente se considerá la variable referente a la producción bruta, y los posteriores análisis se realizarán en función de esta.

Nótese también que en el caso de la Línea 232, las variables `m2_lijado` y `m3_lijado` tomarán siempre el valor 0. Por otro lado, si se considera la Línea 35, la variable `m2_lijado` tomará siempre el mismo valor que la variable `cant_m2_prod`, que es análogo a lo que ocurre con las variables `m3_lijado` y `cant_m3_prod`.

Para facilitar la manipulación de los datos, algunas de las variables son recodificadas o se opta por generar nuevas variables auxiliares:

- Se generan las variables `inicio` y `fin`, que no son más que las variables `hora_inicio` y `hora_fin` pero ahora con formato "hh:mm:ss".
- Se generan las variables `inicio_confecha` y `fin_confecha`: una combinación de la variable `fecha_produccion` con `inicio` y `fin`, respectivamente, de manera que el formato resultante sea "dd/mm/yyyy hh:mm:ss".
- Las variables auxiliares `inicio_confecha` y `fin_confecha` se transforman ahora a formato de tiempo Unix, esto es, segundos transcurridos desde el 1 de enero de 1970, y se almacenan en unas nuevas variables auxiliares denominadas `inicio_segundos` y `fin_segundos`, respectivamente.
- Una nueva variable auxiliar llamada `tiempo_prod` almacena ahora el tiempo correspondiente a cada parte de producción. Es el resultado de la diferencia de `fin_segundos` y `inicio_segundos`.

Un detalle que se ha tenido en cuenta a la hora de pulir los datos y para que estos tengan sentido, es la modificación apropiada de la variable `fecha_produccion`. Los partes de producción con hora de inicio antes de medianoche y hora de fin pasada la medianoche coincidían en la variable `fecha_produccion`, llevando a un error en variables como `tiempo_prod` si no se modificaba. Por ello, las nuevas variables `inicio_confecha` y `fin_confecha`, que utilizaban la variable `fecha_produccion` ya fueron corregidas en su proceso de creación. Dado que realmente la única fecha incorrecta sería la de `fin_confecha`, lo que se hace es comprobar con un bucle para cada parte de producción si el tiempo de producción es negativo, lo cual indicaría que el parte de producción coincide con el cambio de día. Así, si dicho tiempo es negativo, a la variable `fin_confecha` —recordemos, en formato de tiempo Unix—, se le suman 86400 segundos, que serían los correspondientes a adelantar un día.

Aunque ambas líneas cuentan esencialmente con las mismas variables, a continuación se verá que en la línea del lijado se presenta una dificultad adicional.

2.2.1. La Línea 232

Los partes de producción correspondientes a la Línea 232 (MDF) cuentan en su mayoría con marca temporal. No obstante, una gran cantidad de ellos tienen ceros tanto en la hora de inicio como en la hora de fin. Dado que carecen de esta información y no se puede identificar el momento concreto en el que se realiza la producción (para poder asociar con el consumo), estas observaciones son descartadas.

Para asociar consumos eléctricos (kWh) a partes de producción se hace uso de un bucle. La marca temporal con la que cuentan los consumos eléctricos minutales son convertidos también a tiempo Unix, para contar con un formato común al que ya tenían los partes de producción. Ahora, basta con hacer una suma acumulada de todos los consumos ocurridos en el intervalo de tiempo de cada parte de producción, y asociarla a dicho parte. El procedimiento es análogo con todas las variables de consumo eléctrico consideradas:

- `Medida_MDF`: consumo eléctrico total de la Línea 232 (kWh).
- `Medida_MW`: consumo eléctrico del microondas (kWh).
- `Medida_Refino`: consumo eléctrico correspondiente al refino (kWh).
- `Medida_Secado_y_Wesp`: consumo eléctrico correspondiente al secado (kWh)
- `Medida_Resto`: consumo eléctrico de la Línea MDF si no consideramos el microondas, el refino y el secado (kWh).
- `Medida_Fabrica`: consumo total de la fábrica (kWh).

Además, también se considera la variable `Hum_Astilla`, indicadora del porcentaje de humedad con que cuenta la astilla en la primera etapa del secado. Se incorpora esta nueva variable para estudiar si

existe una componente de tipo estacional que repercuta en los consumos. Dado que también se dispone de las mediciones minutales de esta variable, se puede incluir también a los partes de producción de una manera similar a las demás variables. En este caso, como la variable mide porcentajes de humedad, en lugar de hacer una suma acumulada, lo que se realiza es una media de todas las humedades registradas en el transcurso del parte de producción.

Por otro lado, la variable `modelo_obsydian`, indicadora de la gama del producto no será tenida en cuenta para realizar las predicciones pues, en un análisis previo, se consideró que tan solo algunas de estas gamas contaban con suficientes datos para realizar buenas predicciones. Específicamente, solo 20 de las gamas superaban las 50 observaciones, frente a 40 gamas que no (llegando varias de ellas a contar solamente con una observación), como se puede apreciar en la Figura 2.1. Como línea futura de investigación se puede plantear la creación de modelos de predicción para cada uno de los `modelo_obsydian` o tal vez buscar una solución basada en técnicas para datos no balanceados.

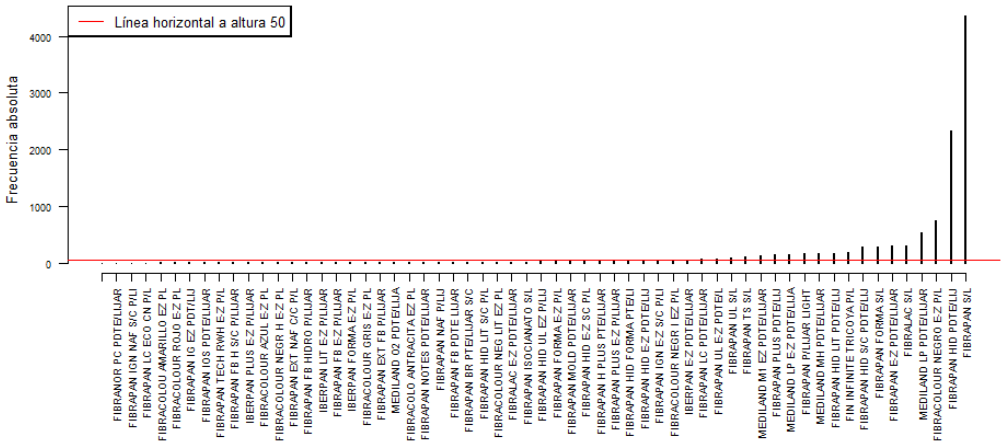


Figura 2.1: Frecuencia absoluta de cada uno de los `modelo_obsydian`.

2.2.2. La Línea 35

Si se consideran los partes de producción correspondientes a la Línea 35 (lijado), se puede ver que ocurre algo parecido a los de la Línea 232: muchos de ellos tienen ceros en las horas de inicio y en las horas de fin. No obstante, a diferencia de aquellos partes, en la Línea 35 *todos* tienen ceros en la hora de inicio y solo se informa de las horas de fin. Además, estas horas de fin siempre coinciden con una de las siguientes: 06:00, 14:00 o 22:00.

Esto, a priori, podría suponer un gran problema por contar con unos datos incompletos o poco prácticos, pues es precisamente la marca horaria la que nos permitirá ligar unos consumos con unas producciones. Aquí es cuando gana importancia la variable categórica `turno`.

Gracias a la variable `turno`, es posible completar aquellos valores faltantes en las variables `hora_inicio` y `hora_fin`. Esta variable indica, cada día, los tres turnos realizados. Estos consisten en lo siguiente:

- Turno de mañana de 06:00 a 14:00.
- Turno de tarde de 14:00 a 22:00.
- Turno de noche de 22:00 a 06:00.

En consecuencia, si cada turno (`turno`) tiene asociada una hora de fin (`hora_fin`), entonces todos los partes de producción que comparten ese mismo turno tendrán asociada esa misma hora de fin, siempre

que se fije el día de producción (`fecha_produccion`). Con este planteamiento, se puede realizar un bucle que complete las horas de fin de los partes de producción. Para completar las horas de inicio, basta con restar 8 horas a las horas de fin.

Dado que todos los partes de producción de un turno dado finalizan a la misma hora, no se puede conocer la hora real exacta a la que se realizaron ni discernir unos de otros a nivel de consumo. Es por eso que —en este caso de la Línea 35— todos los partes de producción pertenecientes a cada turno tendrán un único representante, para cada día, que contenga la totalidad de metros cuadrados producidos y cuyo tiempo de producción sea las 8 horas que dura el turno.

Finalmente, dado que cada día solamente va a contar con 3 "partes de producción", englobando cada uno de ellos todos los partes de producción de cada uno de los 3 turnos diarios diferentes, respectivamente, los datos son trasladados a un nuevo dataset creado convenientemente. El nuevo dataset, generado a partir de una rejilla de turnos, es similar al de partida y cuenta con lo siguiente:

- `fecha_produccion`: Variable que contiene los días repetidos tres veces desde el 2 de enero de 2021 hasta el 10 de abril de 2024.
- `inicio`: Variable con formato "yyyy-mm-dd hh:mm:ss" que, para cada `fecha_produccion`, se consideran las horas 06:00:00, 14:00:00 y 22:00:00.
- `fin`: Análoga a la variable `inicio` pero desplazada 8 horas hacia delante.
- `inicio_segundos`: Es el resultado de convertir la variable `inicio` a tiempo Unix.
- `fin_segundos`: Es el resultado de convertir la variable `fin` a tiempo Unix.
- `m_lijado`: Variable que almacena los metros lineales producidos en el turno de producción.
- `m2_lijado`: Variable que almacena los metros cuadrados producidos en el turno de producción.
- `m3_lijado`: Variable que almacena los metros cúbicos producidos en el turno de producción.
- `consumo`: Cantidad total de consumo eléctrico (kWh) durante el tiempo que dura el turno de producción.

Una última cuestión acerca de estos datos que cabe mencionar es que también se genera una nueva variable auxiliar, la de metros lineales lijados, que reemplazará a las variables de metros cuadrados y cúbicos y será en base a la que se realizarán las predicciones. Esta variable surge de la división entre la variable `m2_lijado` —es decir, largo por ancho— y la variable `ancho`.

Este planteamiento, aunque en cierta medida puede solventar la falta de información, lleva también a obtener predicciones menos precisas.

2.3. Metodología considerada

Con el propósito que corresponde a este trabajo, que es el de predecir unos consumos en base a un registro histórico, fundamentalmente se utilizarán técnicas de *Machine Learning*. Normalmente, se entrenarán modelos con los datos históricos disponibles. Para ello, lo habitual es dividir el conjunto de los datos en una muestra de entrenamiento, compuesta por un 80 % de los datos que servirán para *entrenar* el modelo, y una muestra de test, con el 20 % restante, con la que se validará el mismo. Posteriormente, se evaluará su precisión con alguna medida de error, que servirá para comparar también los distintos modelos generados con los diversos métodos (Fernández et al. 2021).

Por un lado tenemos la Línea 232, que dispone de partes de producción con variables dimensionales del producto como pueden ser el grueso, el ancho o el largo. También se cuenta con dos variables que miden la cantidad de producción en metros cuadrados y cúbicos, respectivamente. En cuanto a estas últimas variables, se cuestionará la necesidad de incluir simultáneamente a las dos en los

modelos, ya que se espera que exista una alta correlación entre ellas. Para finalizar, se tiene también la variable auxiliar `tiempo_prod`, que aunque no parezca un parámetro de producción a priori, sí es posible calcularlo ya que la empresa cuenta con un estándar de m^3/h, por lo que para un producto determinado se conoce cuantos metros cúbicos se fabrica por hora. Con toda esta información, junto con los datos históricos de consumos, se aspira a realizar predicciones individuales de cada una de las variables de consumo eléctrico involucradas.

Por otro lado, en la Línea 35 tan solo se cuenta con una variable predictora: los metros de lijado lineales. A esta situación además se añade la ausencia de una marca temporal precisa. Por estos motivos, no se emplearán modelos tan sofisticados como los de la Línea 232 y los resultados que se obtengan serán de menor calidad.

Pero antes de poner en práctica dichas técnicas, es preciso describirlas, por lo que el próximo capítulo servirá como introducción a este tipo de modelización algorítmica predictiva.

Capítulo 3

Fundamentos teóricos

Este capítulo consistirá en una introducción a las técnicas estadísticas basadas en el *Machine Learning*, disciplina enmarcada dentro de la Ciencia de datos. Estas técnicas suelen dividirse en dos enfoques diferentes: el aprendizaje no supervisado y el aprendizaje supervisado. Por un lado, el aprendizaje no supervisado incluye métodos exploratorios en los que no hay una variable respuesta explícita, como el análisis descriptivo, métodos de reducción de la dimensión, clúster o detección de datos atípicos. Por otro lado, el aprendizaje supervisado abarca los métodos predictivos, es decir, aquellos en los que una de las variables se define como variable respuesta. A su vez, según sea la naturaleza de la variable respuesta, los métodos pueden ser de clasificación, si la respuesta es categórica, o de regresión, si la respuesta es numérica. Dado que el presente trabajo consiste en la predicción de consumos eléctricos, este capítulo se centrará particularmente en la exposición de algunos métodos de regresión que serán puestos en práctica en el siguiente capítulo. Para ello, se revisará y tomará como punto de partida diversos trabajos que forman parte de la literatura de referencia en este campo, como son los estudios de James et al. (2021), Burkov (2019), Kuhn y Johnson (2018), Hastie et al. (2009) o Fernández et al. (2021), en los que se lleva a cabo un desarrollo más profundo y completo.

3.1. Terminología

Tal y como se expone en Kuhn (2018), una gran cantidad de ámbitos científicos han contribuido a este campo, por lo que pueden existir diferentes términos que hacen alusión a conceptos similares. Por mencionar algunos:

- Los términos muestra, dato puntual, observación o instancia se refieren a una única unidad de datos, como un cliente o un paciente. El término muestra puede también referirse a un subconjunto del conjunto de los datos, como puede ser la muestra de entrenamiento.
- La muestra de entrenamiento consiste en los datos utilizados para desarrollar modelos, mientras que la muestra de test se utiliza únicamente para evaluar el rendimiento de los modelos.
- Los predictores, variables independientes o atributos son los datos de *input* en la ecuación de predicción.
- Los términos resultado, variable dependiente, objetivo, clase o respuesta se refieren al resultado, acontecimiento o cantidad que se predice. En ocasiones también se utiliza la palabra *output*.
- La construcción de modelos, el entrenamiento de modelos y la estimación de parámetros hacen referencia al proceso de utilizar datos para determinar los valores de las ecuaciones del modelo.
- Los términos hiperparámetro, parámetro de ajuste, parámetro de tuneado o, en inglés, *tuning parameter*, se utilizan indistintamente.

3.2. Aprendizaje supervisado

Se denotará por $\mathbf{X} = (X_1, X_2, \ldots, X_p)$ al vector formado por las variables predictoras, pudiendo ser cada una de ellas tanto numérica como categórica. Se utilizará $Y(\mathbf{X})$ cuando nos refiramos a la variable objetivo o respuesta que, como ya se mencionó, puede ser numérica (regresión) o categórica (clasificación).

El objetivo de un algoritmo de aprendizaje supervisado es utilizar una muestra

$$\{(x_{1i}, \ldots, x_{pi}, y_i) : i = 1, \ldots, n\}$$

para generar un modelo que a partir de un vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ sea capaz de obtener predicciones $\hat{Y}(\mathbf{x})$ de la respuesta.

En regresión, consideraremos como base el modelo general

$$Y(\mathbf{X}) = m(\mathbf{X}) + \varepsilon,$$

donde $m(\mathbf{X}) = E(Y|_{\mathbf{X}=\mathbf{x}})$ es la media condicional, denominada función de regresión y ε es un error aleatorio de media cero y varianza σ^2, independiente de $\mathbf{X}$.

3.2.1. Construcción de los modelos

En las suposiciones anteriores, el aprendizaje supervisado trata de encontrar una función m en base a ejemplos. Se considera el sistema bajo estudio, compuesto por los *inputs* y los *outputs*, y se forma una muestra de entrenamiento $\{(x_{1i}, \ldots, x_{pi}, y_i) : i = 1, \ldots, N\}$. Los *inputs* observados $\mathbf{x_i}$ también se suministran a un sistema artificial, conocido como algoritmo de aprendizaje, que a su vez produce *outputs* $\hat{m}(\mathbf{x_i})$ como respuesta a los *inputs*. El algoritmo de aprendizaje tiene la particularidad de poder modificar la relación $\hat{m}$ establecida entre los *inputs* y los *outputs* en función de, por ejemplo, las diferencias $y_i - \hat{m}(\mathbf{x_i})$ entre la respuesta original y la generada por el modelo. Este proceso se conoce como aprendizaje por ejemplos (*learning by example*) y, una vez completado, se espera que las salidas generadas y las reales sean lo suficiente próximas como para ser útiles cuando se utilice un nuevo conjunto de datos desconocido para el modelo (Hastie et al. 2009).

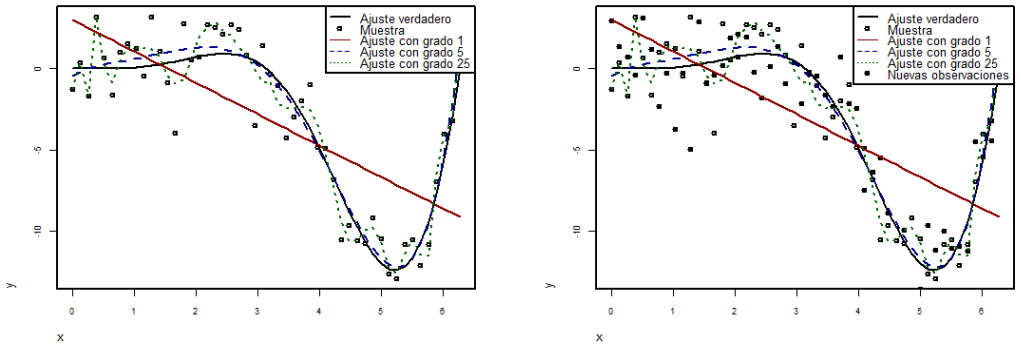
Alguno de los modelos existentes son muy flexibles, por lo que normalmente es necesario controlar el proceso de aprendizaje a través de unos parámetros de ajuste. Se pueden distinguir dos tipos de parámetros: los parámetros estructurales, que son los estimados al ajustar el modelo, y los hiperparámetros o parámetros de ajuste, que imponen restricciones al aprendizaje del modelo. Estos hiperparámetros deben seleccionarse de modo que produzcan modelos ni demasiado complejos, ni demasiado sencillos. Si se produce un modelo demasiado complejo, pueden aparecer problemas de sobreajuste[1] (*overfitting*) y, además, disminuiría la interpretabilidad del modelo. En caso contrario, cuando el modelo generado es demasiado sencillo, también es más fácil de interpretar, pero podrían aparecer problemas de infraajuste (*underfitting*) (Fernández et al. 2021).

Para medir la precisión al emplear los modelos en un nuevo conjunto de datos pueden calcularse medidas de bondad de ajuste como el error cuadrático medio (MSE),

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i).$$

Si consideramos el modelo descrito al comienzo de la sección y una función de pérdidas cuadrática, el predictor óptimo sería la media condicional $m(\mathbf{X}) = E(Y|_{\mathbf{X}=\mathbf{x}})$. De esta manera, las predicciones serían estimaciones de la función de regresión, es decir, $\hat{Y}(\mathbf{x}) = \hat{m}(\mathbf{x})$, y se podría expresar la media del error cuadrático de predicción en términos de sesgo y varianza, para una nueva observación $\mathbf{x_0}$. De este modo se tiene que, al aumentar la complejidad del modelo, disminuye el sesgo pero puede aumentar mucho la

[1] Ocurre cuando el modelo se ajusta demasiado bien a los datos empleados en el entrenamiento pero falla al utilizar un nuevo conjunto de datos.



(a) Muestra simulada y distintos ajustes polinómicos.

(b) Comportamiento frente a nuevas observaciones.

Figura 3.1: Grado del polinomio como parámetro de ajuste.

varianza. En caso contrario, los modelos sencillos no tienden a sobreajustar, pero si a infraajustar si no son lo suficientemente flexibles para modelar la relación subyacente, lo que implicaría un aumento en el sesgo. Además, predictores altamente correlacionados podrían acarrear problemas de colinealidad, lo que aumentaría la varianza del modelo. Se tratará, entonces, de encontrar unos hiperparámetros óptimos en términos del equilibrio entre el sesgo y la varianza (*bias-variance trade-off*) (Fernández et al. 2021; Kuhn y Johnson 2018),

$$\begin{aligned} E(Y(\mathbf{x_0}) - \hat{Y}(\mathbf{x_0}))^2 &= E(m(\mathbf{x_0}) + \varepsilon - \hat{m}(\mathbf{x_0}))^2 = E(m(\mathbf{x_0}) - \hat{m}(\mathbf{x_0}))^2 + \sigma^2 \\ &= E^2(m(\mathbf{x_0}) - \hat{m}(\mathbf{x_0})) + Var(\hat{m}(\mathbf{x_0})) + \sigma^2 \equiv sesgo^2 + varianza + error. \end{aligned}$$

Muestras de entrenamiento y test

Como ya se comentó anteriormente, para la construcción de estos modelos será necesario contar con una muestra de entrenamiento y otra de test. Para obtenerlas, se sigue el procedimiento tradicional en *Machine Learning* consistente realizar una partición disjunta de la base de datos original: los datos de entrenamiento o aprendizaje, para construir los modelos, y los datos de test, para evaluar el desempeño de los modelos.

En ocasiones, en lugar de considerar solamente los dos subconjuntos mencionados, se pueden considerar un tercero: la muestra de validación. En estos casos, se emplea la muestra de test para construir los modelos, la de validación para evaluarlos y seleccionar los valores óptimos de los hiperparámetros y, por último, se utiliza la de test para medir el rendimiento del modelo seleccionado.

Habitualmente, se selecciona al azar el 80 % de los datos para conformar la muestra de entrenamiento y el 20 % restante consistiría en la muestra de test. En caso de considerar tres muestras, la división suele ser del 70 % para los datos de test y a partes iguales para las muestras de validación y de test. No obstante, no existe una proporción óptima para dividir el conjunto de datos. De hecho, en la era del *big data* en que vivimos, a menudo los datasets cuentan con millones de observaciones y lo recomendado sería tomar el 95 % de los datos para entrenamiento y el 5 % restante para test (o 2.5 % para cada una de las muestras de validación y test, en caso de considerar las tres muestras) (Burkov 2019).

3.2.2. Evaluación de los modelos

Para evaluar la calidad predictiva de un modelo, es común emplear algún método de remuestreo como la validación cruzada. A continuación se describe la manera de proceder para dos modalidades de validación cruzada.

Leave-One-Out Cross-Validation

Leave-One-Out Cross-Validation (LOOCV; validación cruzada dejando uno fuera) es la versión más simple de la validación cruzada. En ella, es necesario separar el conjunto de datos en dos partes. Sin embargo, en lugar de separar los datos en conjuntos de tamaño similar, se selecciona una única observación, pongamos que se habla de $(\mathbf{x}_h, y_h)$, y se utiliza el resto de las observaciones $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{h-1}, y_{h-1}), (\mathbf{x}_{h+1}, y_{h+1}), \ldots, (\mathbf{x}_n, y_n)\}$ para realizar el ajuste del modelo. A continuación, se calcula el error de predicción en la observación seleccionada previamente, que es la única no utilizada para ajustar el modelo. Este procedimiento se repite con las $i \in \{1, \ldots, n\}$ observaciones de la muestra, combinándose todos los errores individuales para obtener medidas globales del error de predicción como

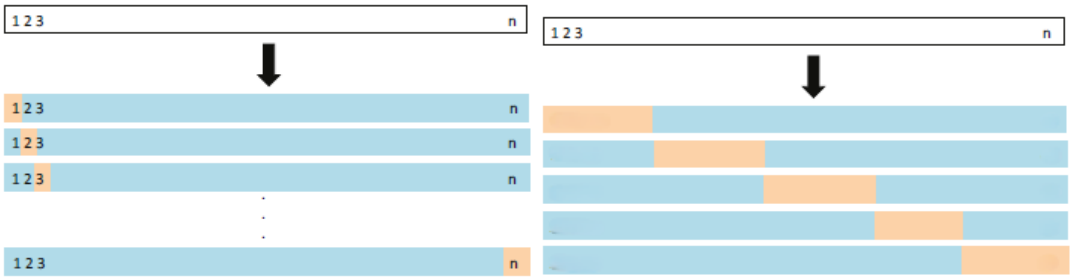
$$MSE_h = (y_k - \hat{y}_h)^2,$$

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} MSE_i.$$

k-Fold Cross-Validation

Puede verse que el método LOOCV supone la construcción de un modelo para cada una de las observaciones de la muestra, lo cual puede derivar en problemas computacionales si el conjunto de datos es demasiado grande. Como alternativa, surge el método *k-Fold Cross-Validation*, que requiere dividir aleatoriamente la muestra de observaciones en k grupos de aproximadamente el mismo tamaño. Análogamente al método anterior, uno de estos grupos es seleccionado y se utilizará para calcular los errores de predicción MSE_i una vez que se ajuste el modelo con los $k-1$ grupos restantes. Este proceso será repetido k veces y, cada una de las veces, se toma un grupo distinto para calcular los errores de predicción y los demás para ajustar el modelo. Así, se acaban obteniendo finalmente las k estimaciones del error $MSE_1, MSE_2, \ldots, MSE_k$, que combinados resultan en la expresión

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i.$$



(a) Esquema de LOOCV. (b) Esquema de k-Fold CV.

Figura 3.2: Métodos de validación cruzada para evaluar un modelo (Fuente: James et al. (2021)).

Como comentario final, se puede observar que LOOCV es un caso particular de k-fold CV en el cual se toma k igual a n, es decir, la cantidad total de observaciones. Para profundizar más en este tema puede consultarse James et al. (2021, pp. 200-205), de donde también se han tomado los esquemas de la Figura 3.2.

3.2.3. Otras medidas de error

Como se ha comentado anteriormente, para estudiar la precisión del modelo generado, este se evalúa en el conjunto de datos de test y se comparan las predicciones con los valores reales observados. Visualmente, esto podría hacerse representando en un gráfico de dispersión las observaciones frente a las predicciones y examinando si los puntos representados se encuentran en torno a la recta $y = x$. Si lo que queremos es cuantificar la precisión de los modelos para poder compararlos entre sí, se pueden emplear medidas de error como las siguientes:

- Error medio:
$$ME = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i).$$
- Raíz del error cuadrático medio:
$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{n}}.$$
- Error absoluto medio:
$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}.$$
- Error porcentual medio:
$$MPE = 100\frac{1}{n}\sum_{i=1}^{n}\frac{y_i - \hat{y}_i}{y_i}.$$
- Error porcentual absoluto medio:
$$MAPE = 100\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|.$$
- Pseudo R-cuadrado:
$$\tilde{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$
siendo $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

Este último valor, conocido como *pseudo R-cuadrado*, es el recomendado por Fernández et al. (2021) y mide la proporción de variabilidad de la respuesta en nuevas observaciones explicada por el modelo.

Gracias a este tipo de medidas de error es posible comparar la capacidad predictiva de modelos considerando diferentes combinaciones de hiperparámetros o, incluso, comparar modelos generados por una amplia variedad de métodos de aprendizaje estadístico. El motivo de que exista tal amplia variedad es que no existe ningún método que domine a los demás en cualquier conjunto de datos. Como se explica en James et al. (2021), puede que para un dataset particular sea un método específico el que mejor funcione, pero también puede ocurrir que para otro dataset similar (aunque algo diferente) tengan mejor rendimiento otros métodos. Es lo que se conoce como el *there is no free lunch in statistics*, acuñado en 1997 por David Wolpert and William Macready, que viene a decir que no hay atajos para alcanzar el éxito. En la siguiente sección se describirán algunos de estos métodos.

3.3. Métodos de regresión

3.3.1. Árboles de regresión

Los árboles de decisión son un método sencillo de implementar e interpretar y aplicable tanto en problemas de clasificación como de regresión. A pesar de no contar con una calidad predictiva excelente, sí pueden ser usados como métodos descriptivos o como base de otros métodos más competitivos. La idea consiste en particionar de manera disjunta el espacio predictor —conjunto de posibles valores para las variables predictoras $X_1, X_2, \ldots, X_p$— de modo que las subregiones resultantes sean tan simples que el proceso pueda representarse mediante un árbol binario. Básicamente:

1. Se divide el espacio predictor en una partición disjunta de J regiones $R_1, R_2, \ldots, R_J$.

2. Para cada observación situada en la región R_j se realiza la misma predicción, que consiste en el valor medio de las respuestas para las muestras de entrenamiento en la región R_j.

Teóricamente, las regiones R_j siendo $j \in \{1, \ldots, J\}$, podrían tener cualquier forma. Sin embargo, por simplicidad, se elige dividir el espacio predictor en rectángulos. Esto también facilitará la interpretación de los resultados del modelo predictivo. El objetivo es encontrar unas regiones $R_1, \ldots, R_J$ tales que minimicen la suma residual de cuadrados[2]:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_J})^2,$$

siendo $\hat{y}_{R_J}$ la media de las respuestas de la muestra de entrenamiento situada en la región j-ésima.

La vía anterior sería un tanto inviable, puesto que considerar todas las posibles particiones sería poco práctico computacionalmente. En su lugar, se considera un enfoque *greedy*[3] de arriba a abajo, conocido como *recursive binary splitting*. El algoritmo comienza en lo alto del árbol (aún todas las observaciones pertenecen a la misma región) y a continuación va dividiendo sucesivamente el espacio predictor, donde cada división genera dos nuevas ramas más profundas. El proceso iterativo es como sigue:

Se ha de seleccionar una variable explicativa X_j junto con un punto de corte s tal que al dividir el espacio predictor en dos regiones[4] $\{X|X_j < s\}$ y $\{X|X_j > s\}$ se produzca la máxima reducción de RSS.

1. Así, se consideran todas las variables explicativas $X_1, X_2, \ldots, X_p$ y todos los posibles puntos de corte s —para cada una de dichas variables—, y se toma aquella variable explicativa y aquel punto de corte que genera un árbol con el menor RSS. Formalmente, para cada j y s, se define el par de semiplanos

$$R_1(j, s) = \{X|X_j < s\} \; R_2(j, s)\{X|X_j > s\}$$

y se seleccionan los valores de j y s que minimizan la expresión

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2,$$

[2] En la fórmula, la expresión $i \in R_j$ hace referencia a aquellas observaciones $i \in \{1, \ldots, n\}$ tales que $x_i \in R_j$.

[3] En español: algoritmo *voraz*. Este algoritmo consiste en calcular un óptimo local en cada paso, con la esperanza de llegar a una solución general óptima. Cuando es de arriba a abajo, nunca se vuelve a la anterior elección aunque esta no sea la correcta.

[4] La notación $\{X|X_j < s\}$ significa la región del espacio predictor en la que la variable explicativa X_j toma un valor menor que el punto de corte s.

en la que, igual que antes, $\hat{y}_{R_1}$ es la media de las respuestas de la muestra de entrenamiento situada en la región $R_1(j, s)$ y $\hat{y}_{R_2}$ es la media de las respuestas de la muestra de entrenamiento situada en la región $R_2(j, s)$.

2. A continuación, se repite el paso anterior en cada una de las regiones R_1 y R_2.

Este proceso se repite sucesivamente hasta alcanzar una condición de parada como que el árbol alcance una profundidad máxima o que los nodos cuenten con una cantidad mínima de observaciones. Una vez que se tienen las regiones $R_1, \ldots, R_J$, como ya se ha mencionado antes, se predice la respuesta de la observación de test utilizando la media de las observaciones de entrenamiento de la región a la que pertenece dicha observación de test.

Podar el árbol

El proceso anterior puede llegar a producir predicciones razonables en el conjunto de entrenamiento, pero es posible que se produzca un sobreajuste de los datos y falle al predecir en datos aún desconocidos por el modelo, como los de la muestra de test. El motivo es que el árbol calculado es demasiado complejo. Un árbol más sencillo podría reducir la varianza y facilitar la interpretación a cambio, eso sí, de un aumento en el sesgo.

Una posible estrategia es la de *hacer crecer un árbol* T_0 lo suficientemente grande y después *podarlo* para obtener un *subárbol*. En vez de considerar todos los posibles subárboles, pues evaluarlos todos no sería una estrategia recomendada, se considera el hiperparámetro no negativo α. Para cada valor de α se corresponde un subárbol $T \subset T_0$ tal que

$$\sum_{j=1}^{|T|} \sum_{i:x_i \in R_j} (y_i - \hat{y}_{R_j})^2 + \alpha |T|$$

sea lo más pequeño posible. En este caso, $|T|$ indica el número de nodos terminales del árbol T, R_j es la región correspondiente al j-ésimo nodo terminal y $\hat{y}_{R_j}$ es la predicción de la respuesta asociada a dicha región.

El hiperparámetro α permite controlar el equilibrio entre la complejidad de un subárbol y su ajuste a los datos de entrenamiento. Cuando $\alpha = 0$, se tiene el árbol completo $T = T_0$. No obstante, a medida que α aumenta, los subárboles con muchos nodos terminales son penalizados, dando lugar a una solución más simple. Además, resulta que cuando aumentamos α desde cero, las ramas del árbol son podadas de un modo que se puede obtener una sucesión de subárboles en función de dicho hiperparámetro. Por último, el valor de α puede ser seleccionado, por ejemplo, mediante validación cruzada.

Una de las opciones más populares para la selección de α sería la de tomar el valor que minimice el error. No obstante, también es posible emplear la regla *one-standard-error*, que selecciona el árbol más pequeño que se encuentra a una distancia de un error estándar del árbol obtenido por la primera opción.

La metodología descrita en esta subsección lleva el nombre de CART (Classification and Regression Trees), es la desarrollada en Breiman et al. (1984) y también es la más popular. Para profundizar más acerca de los árboles de decisión, que incluyen regresión y clasificación, también puede consultarse Fernández et al. (2021) y James et al. (2021).

3.3.2. Bosques aleatorios

Los métodos *ensemble* son aquellos que combinan métodos de predicción sencillos, que por sí solos obtendrían predicciones mediocres, para obtener un único modelo más potente pero menos interpretable.

En este método, se emplea el *bootstrap* para generar muchas muestras a partir de la muestra de entrenamiento inicial y después se utiliza cada una de ellas como muestra de entrenamiento para

construir un modelo distinto. De este modo, se tienen tantas predicciones de la respuesta como modelos (y por ende como muestras de entrenamiento), y lo que se hace es promediarlas.

En los bosques aleatorios (*random forest*, RF), se construye un número de árboles de decisión a partir de las muestras bootstrap de entrenamiento. No obstante, al construir estos árboles de decisión, cada vez que se considera una división se escogen aleatoriamente m variables explicativas de las p totales como candidatas para generar la partición. Solamente una de estas m variables puede producir el corte y normalmente se suele considerar $m = \sqrt{p}$ (en clasificación) y $m = \frac{p}{3}$ (en regresión).

Si no se considerase solamente un subconjunto de las variables explicativas en cada uno de los cortes, podría darse el caso de la existencia de algún predictor muy fuerte: este sería escogido siempre en el primer corte y, por tanto, todos los árboles serían bastante similares o dependientes.

3.3.3. *Boosting*

La metodología *boosting* pertenece a los métodos *ensemble*, al igual que los bosques aleatorios. En efecto, el *boosting* trabaja de un modo similar al de los bosques, ya que también combina los resultados de varios modelos (más sencillos y a menudo denominados predictores débiles, como los árboles). No obstante, difieren en la manera de construir los árboles. En este caso, los árboles se hacen crecer de manera secuencial, es decir, cada nuevo árbol se hace crecer a partir de información de otros árboles ya crecidos. A diferencia de los bosques aleatorios, ahora no se necesita una muestra bootstrap para cada árbol, sino que cada árbol se ajusta a partir de una versión modificada del conjunto de datos inicial.

Entre los métodos *boosting* destaca *gradient boosting machine*, un método basado en un proceso iterativo de descenso de gradientes. El objetivo, en este caso, es encontrar un modelo aditivo que minimice una función de pérdida utilizando modelos con poca capacidad predictiva como los árboles de decisión. Si como función de pérdida se considera la suma residual de cuadrados (RSS), entonces la pérdida de emplear m para predecir y en la muestra de entrenamiento es

$$L(m) = \sum_{i=1}^{n} L(y_i, m(x_i)) = \sum_{i=1}^{n} L(y_i - m(x_i))^2.$$

Ahora, si $L(m) = \frac{1}{2}(y_i - m(x_i))^2$, entonces, minimizando $L(m)$ con el método de los gradientes, resulta que

$$-\frac{\partial L(y_i, m(x_i)}{\partial m(x_i)} = y_i - m(x_i) = r_i,$$

obteniéndose precisamente los residuos r_i.

El algoritmo *boosting* considerando árboles de regresión como predictor débil consistiría en los siguientes pasos:

1. Establecer una predicción inicial constante $\hat{m}(x) = 0$ y calcular los residuos $r_i = y_i$ para todo i de la muestra de entrenamiento.

2. Para $b = 1, 2, \ldots, B$, repetir:

   a) Ajustar un árbol $\hat{m}^b$ con d cortes ($d + 1$ nodos terminales) utilizando los residuos como respuesta en la muestra de entrenamiento (X, r).

   b) Calcular la versión regularizada del árbol:

   $$\lambda \hat{m}^b(x).$$

   c) Actualizar los residuos,

   $$r_i \leftarrow r_i - \lambda \hat{m}^b(x_i).$$

3. Calcular el modelo *boosting*:

$$\hat{m}(x) = \sum_{b=1}^{B} \lambda \hat{m}^b(x).$$

El método depende de tres parámetros de tuneado: B, λ y d.

- B es el número de árboles (o iteraciones). Si el valor es muy grande, podría llegar a producirse sobreajuste. Se utiliza validación cruzada para elegir un B apropiado.
- El parámetro de regularización $0 < \lambda < 1$. Controla la velocidad de aprendizaje del algoritmo. Valores demasiado pequeños de este parámetro pueden requerir utilizar una gran cantidad de árboles B para alcanzar una buena capacidad predictiva. Se suelen considerar valores como 0.01 o 0.001.
- d es el número de cortes en cada árbol y controla la complejidad del modelo conjunto. A menudo, se emplea el valor $d = 1$, es decir, se utiliza un solo corte en cada árbol. En este caso, $\hat{m}$ es un ajuste de un modelo aditivo ya que cada término involucra una única variable. De manera general, cuando $d > 1$, puede decirse que se trata de un parámetro que mide el orden de interacción entre las variables del modelo.

Existen variantes de este algoritmo como el *stochastic gradient boosting* (*SGB*), de las más utilizadas hoy en día, o el algoritmo *extreme gradient boosting* (*XGBoost*) que también está ganando popularidad. Este último método es más complejo, pues utiliza una función de pérdida con una penalización por complejidad y regulariza utilizando la hessiana de dicha función (que involucra el cálculo de las derivadas parciales de primer y segundo orden), además de otros parámetros de regularización adicionales (Fernández et al. 2021).

3.3.4. Modelos lineales

Remuestreo para seleccionar el modelo

En los modelos lineales, suponemos una función de regresión lineal

$$E(Y|\mathbf{X}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

y el efecto de las variables explicativas sobre la respuesta es proporcional a su valor. Cada coeficiente β_j representa el incremento medio de Y en aumentar en una unidad el valor de X_j, manteniendo fijas las demás variables. Así, tradicionalmente se considera el siguiente modelo para regresión lineal múltiple:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon.$$

Estos modelos requieren del cumplimiento de las hipótesis estructurales de linealidad, homocedasticidad, normalidad e independencia de los errores. No obstante, podrían considerarse técnicas de aprendizaje estadístico para seleccionar el modelo. Por ejemplo, podría considerarse como hiperparámetro la inclusión o no de cada una de las posibles variables predictoras, o el número máximo de estas, mediante remuestreo.

Métodos de regularización

Para ajustar un modelo de regresión lineal suele emplearse el método de mínimos cuadrados, esto es, utilizar como criterio de error la suma residual de cuadrados

$$RSS = \sum_{i=1}^{n} (y_i - \beta_0 - \beta^t \mathbf{x}_i)^2.$$

Uno de los problemas principales de estos modelos es cuando la cantidad de variables explicativas p es grande o cuando las correlaciones entre las variables predictoras son altas, pues las estimaciones del modelo tendrán mucha varianza y resultará en un modelo sobreajustado.

La forma de conseguir que el modelo reduzca su varianza es tratar de reducir su complejidad. Esto se puede lograr mediante regularización en los parámetros β_j, es decir, considerar todas las variables predictoras pero forzar a que alguno de los parámetros se estime con valores próximos a cero, en el caso de *ridge regression*, o directamente con ceros, en el caso de *LASSO*. Estos modelos emplean una penalización cuadrática o en valor absoluto, respectivamente, y su planteamiento es el siguiente:

- El objetivo de *ridge regression* —o regularización L_2— es

$$\min_{\beta_0,\beta} RSS + \lambda \sum_{j=1}^{p} \beta_j^2,$$

  o, equivalentemente,

$$\min_{\beta_0,\beta} RSS,$$

  sujeto a

$$\sum_{j=1}^{p} \beta_j^2 \leq s.$$

- El objetivo de *LASSO* (*least absolute shrinkage and selection operator*) —o regularización L_1— es

$$\min_{\beta_0,\beta} RSS + \lambda \sum_{j=1}^{p} |\beta_j|,$$

  o, equivalentemente,

$$\min_{\beta_0,\beta} RSS,$$

  sujeto a

$$\sum_{j=1}^{p} |\beta_j| \leq s.$$

Es posible considerar ambos problemas de manera unificada añadiendo un parámetro d. El problema ahora consistiría en

$$\min_{\beta_0,\beta} RSS + \lambda \sum_{j=1}^{p} |\beta_j|^d,$$

donde, si $d = 0$, la penalización consistiría en el número de variables empleadas; si $d = 1$, entonces estamos en el caso *LASSO*, y si $d = 2$, estaríamos en el caso *ridge*.

Ambos métodos dependen del hiperparámetro λ (o en su versión equivalente, s) y debe seleccionarse adecuadamente su valor empleando validación cruzada, como se recomienda en Fernández et al. (2021).

Una posible generalización de estos métodos es la conocida como *elastic net*, que combina las ventajas de ambos. El problema planteado en este caso sería el de

$$\min_{\beta_0,\beta} RSS + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right),$$

con $0 \leq \alpha \leq 1$ hiperparámetro adicional que permite la commbinación lineal de ambos métodos.

3.3.5. Regresión no paramétrica

Los métodos de regresión no parámetrica no hacen suposiciones explícitas sobre ninguna forma concreta de la media condicional, es decir, adoptan la forma

$$Y = m(X_1, \ldots, X_p) + \varepsilon,$$

con m una función cualquiera. Dado que en este caso el problema no reduce la estimación de m a unos pocos parámetros, será necesario contar con una gran cantidad de observaciones para obtener una estimación precisa de m (James et al. 2021).

La idea subyacente en la mayoría de estos métodos es la de ajustar localmente un modelo local de regresión, es decir, predecir la respuesta en función de lo que ocurre en observaciones *cercanas*.

k-vecinos más próximos

El método de los vecinos más próximos (en inglés, *k-nearest neighbors*; KNN) es uno de los más populares de regresión local.

El ajuste de un modelo de k-vecinos más próximos se define como

$$\hat{Y}(\mathbf{x}) = \hat{m}(\mathbf{x}) = \frac{1}{k} \sum_{i:\ \mathbf{x}_i \in N_k(\mathbf{x})} y_i,$$

siendo $N_k(\mathbf{x})$ un entorno (vecindario) de $\mathbf{x}$ definido por los k puntos más cercanos $\mathbf{x}_i$ en la muestra de entrenamiento. Dado que cercanía implica una métrica, es necesario especificar una distancia. Se puede considerar, por ejemplo, una métrica general

$$d(\mathbf{x}_0, \mathbf{x}_i) = \left(\sum_{j=1}^{p} |x_{j0} - x_{ji}|^d \right)^{\frac{1}{d}},$$

que para $d = 2$ se trataría de la distancia euclídea.

En resumen, este método selecciona las k observaciones con x_i más cercano a x en el espacio predictor y promedia sus respuestas. El hiperparámetro de estos modelos es el número de vecinos k y determina su complejidad de manera inversamente proporcional: valores pequeños de k producirán modelos complejos[5], mientras que valores altos de k resultarán en modelos más sencillos. En Fernández et al. (2021), la recomendación es la de seleccionar este hiperparámetro mediante validación cruzada con k grupos.

Projection pursuit

Projection pursuit es una técnica de análisis exploratorio de datos multivariantes que busca proyecciones lineales de los datos en espacios de dimensión baja. Inicialmente, se utilizaba como técnica gráfica y por ello las proyecciones eran en rectas o planos, resultando que las direcciones de interés eran aquellas con distribución no normal. Habitualmente, al realizar transformaciones lineales, el resultado suele tener la apariencia de una distribución normal, lo cual oculta las singularidades de los datos originales. Se supone que los datos originales son una transformación lineal de componentes no gaussianas y la idea es deshacer dicha transformación mediante la optimización de una función objetivo conocida como *projection index* (Fernández et al. (2021).

En el método de PPR (*projection pursuit regression*), se considera el modelo semiparamétrico

$$m(\mathbf{x}) = \sum_{m=1}^{M} g_m(\alpha_{1m} x_1 + \alpha_{2m} x_2 + \cdots + \alpha_{pm} x_p),$$

[5]Nótese la posibilidad de casos extremos: si $k = 1$, se interpolarían todas las observaciones, mientras que si $k = n$, se harían predicciones promediando la variable respuesta de la muestra de entrenamiento con lo que se tendría $\hat{y} = \bar{y}$

con $\alpha_m = (\alpha_{1m}, \alpha_{2m}, \ldots, \alpha_{pm})$ vectores de parámetros para $m \in \{1, \ldots, M\}$ y g_m funciones suaves en $\mathbb{R}^p$, llamadas funciones *ridge*. La variable escalar $\alpha_m^t \mathbf{x}$ es la proyección de $\mathbf{x}$ sobre el vector unitario α_m, y buscamos α_m de forma que el modelo se ajuste bien, de ahí el nombre de "búsqueda de proyecciones".

Se trata de un modelo muy general que, con M suficientemente grande y eligiendo adecuadamente las componentes podría aproximar cualquier función continua[6]. No obstante, el modelo así construido puede llegar a ser difícil de interpretar, ya que cada variable se introduce en el modelo de manera compleja y polifacética (Hastie et al. 2009). Una excepción sería el caso $M = 1$, modelo conocido como *single index model* en econometría, el cual es ligeramente más general que el modelo de regresión lineal y ofrece una interpretación similar.

Teóricamente, habría que estimar las funciones g_m con algún método de suavizado y los parámetros α_{im} (con $i = 1, \ldots, p$, $m = 1, \ldots, M$), utilizando como criterio de error RSS, pero en la práctica se emplea un proceso iterativo. Dicho proceso consistiría en ir fijando sucesivamente los valores de los parámetros y las funciones *ridge*.

Dada una muestra de entrenamiento $(\mathbf{x_i}, y_i)$, con $i = 1, 2, \ldots, n$, el modelo se ajusta con el procedimiento que se expone a continuación, que puede consultarse con más detalle en Hastie et al. (2009, pp. 390-391).

Se buscan los minimizadores de la función de error (aproximados)

$$\sum_{i=1}^{n} \left[y_i - g_m(\alpha_m^t \mathbf{x}_i) \right]^2, \tag{3.1}$$

en las funciones g_m y las direcciones α_m, siendo $m = 1, 2, \ldots, M$.

Considérese solamente un término ($M = 1$, se obvia ahora el subíndice). Dado el vector de dirección α, se forman las variables derivadas $v_i = \alpha^t \mathbf{x}_i$. Así, se tiene un problema de suavizado 1-dimensional se puede aplicar, por ejemplo, un spline de suavizado[7] para obtener la estimación de g

Por otro lado, dada g, queremos minimizar la Expresión 3.1 en α. Un método de búsqueda Gauss-Newton resultaría apropiado para esta tarea. Sea α_{old} la actual estimación de α, se aproxima:

$$g(\alpha^t \mathbf{x}_i) \approx g(\alpha_{old}^t \mathbf{x}_i) + g'(\alpha_{old}^t \mathbf{x}_i)(\alpha - \alpha_{old})^t \mathbf{x}_i,$$

para que resulte

$$\sum_{i=1}^{n} \left[y_i - g(\alpha^t \mathbf{x}_i) \right]^2 \approx \sum_{i=1}^{n} g'(\alpha_{old}^t \mathbf{x}_i)^2 \left[\left(\alpha_{old}^t \mathbf{x}_i + \frac{y_i - g(\alpha_{old}^t \mathbf{x}_i)}{g'(\alpha_{old}^t \mathbf{x}_i)} \right) - \alpha^t \mathbf{x}_i \right]^2.$$

Para minimizar el lado derecho, se puede realiza regresión de mínimos cuadrados con respuesta $\alpha_{old}^t \mathbf{x}_i + (y_i - g(\alpha_{old}^t \mathbf{x}_i))/g'(\alpha_{old}^t \mathbf{x}_i)$ a la entrada x_i y pesos $g'(\alpha_{old}^t \mathbf{x}_i)^2$, sin intercepto. Esto genera el vector de coeficientes α_{new} actualizado.

Estos dos pasos de estimación de g y α se iteran hasta la convergencia. En el caso en que $M > 1$, el modelo se construye de una forma progresiva por etapas, añadiendo un par (α_m, g_m) en cada etapa. El número de términos M puede determinarse utilizando validación cruzada. No obstante, es usual que M sea estimado en el desarrollo de la estrategia progresiva de etapas: la construcción del modelo finaliza una vez que el siguiente término no mejora el ajuste del modelo de manera significativa.

[6] Este tipo de modelos se conoce como *aproximadores universales.*

[7] Técnica consistente en trocear los datos en intervalos, fijando unos puntos de cortes, y ajustar un polinomio en cada segmento de forma que la conexión en los extremos de estos sea suave. Para más detalles, puede consultarse Fernández et. al (2021, pp. 170-176).

Capítulo 4

Resultados y discusión

En este capítulo se llevará a cabo el entrenamiento de los modelos de *Machine Learning* descritos para realizar las predicciones en las muestras de test, además de compararse el rendimiento y precisión de los distintos modelos estimados, discutiéndose posteriormente los resultados obtenidos. El objetivo es predecir los valores de la variable respuesta u objetivo de consumo eléctrico en función de una serie de variables independientes relacionadas con las dimensiones y estructura de los productos, cantidad producida y tiempo de producción.

Como se ha comentado en la Sección 2.2, contamos con dos bases de datos que se tratarán de manera separada: la Línea 232 y la Línea 35. En cuanto a los modelos empleados en la Línea 232, serán los expuestos en la Sección 3.3: árboles de regresión, bosques aleatorios, *gradient boosting machine* (*GBM*), *extreme gradient boosting* (*XGBoost*), métodos de regularización, k-vecinos más próximos y *projection pursuit*. Por otro lado, en la Línea 35 se empleará un modelo de regresión polinómico.

Tanto la manipulación de los datos como la construcción de los modelos se realiza con el del software estadístico y lenguaje de programación R (R Core Team, 2023).

4.1. Predicción en la Línea 232

La metodología considerada para llevar a cabo las predicciones de consumo en la línea de producción de MDF es la siguiente: para cada una de las variables respuesta (consumos) de interés, se construirán y evaluarán los modelos descritos en este trabajo, pudiendo comparar así el rendimiento o desempeño de los mismos. Dado que son 6 las variables de consumo, se explicará con detalle y a modo ilustrativo el procedimiento de construcción de los modelos con una de ellas. Para las otras 5 solamente se mostrarán los resultados obtenidos, ya que el procedimiento es análago. Finalmente, se valorará el interés de predecir el consumo de 5 líneas de producción diferentes frente a predecir solamente el consumo total.

4.1.1. Consumo total de la fábrica

La variable respuesta seleccionada para ilustrar el procedimiento es la referente al consumo eléctrico total de la fábrica (`Medida_Fabrica`). Las variables predictoras utilizadas en el modelo serán `grueso_numer`, `ancho`, `largo`, `cant_m3_prod` y `tiempo_prod`. Finalmente, se opta por no emplear la variable `cant_m2_prod` dada la alta correlación con la variable `cant_m3_prod`[1].

Para cada modelo construido, se calcularán medidas de la calidad predictiva y se utilizará alguna de ellas para comparar entre modelos. Las medidas consideradas son las descritas en la Subsección 3.2.3: ME, RMSE, MAE, MPE, MAPE y pseudo R-cuadrado. Serán precisamente estas dos últimas, que no dependen de la escala de la variable, las que se utilizarán para comparar los distintos modelos.

[1]Nótese que la variable `cant_m2_prod` puede obtenerse dividiendo las variables `cant_m3_prod` y `grueso_numer`.

Árbol de regresión

Para construir el modelo se hace uso de la metodología CART, que está implementada en el paquete `rpart`[2]. Como ya se comentó, el primer paso es separar los datos en una muestra aleatoria simple de entrenamiento —seleccionando aleatoriamente el 80 % de los mismos—, con la que se construirá el modelo, y una muestra de test —el 20 % restante—, con la que se medirá su precisión.

Se construyen dos modelos:

- `tree`: Modelo construido con los parámetros por defecto que se indican a continuación.
  - `minsplit`: mínimo de 20 observaciones en un nodo para dividirse.
  - `minbucket`: mínimo de observaciones en un nodo terminal igual a un tercio del parámetro anterior (se redondea al entero más cercano).
  - `cp`: parámetro de complejidad α' (0.01 por defecto), que se obtiene reescalando por la variabilidad total (suma residual de cuadrados) el α considerado en 3.3.1:

$$\alpha' = \alpha / \sum_{i=1}^{n} (y_i - \bar{y})^2.$$

  - `xval`: número de grupos para validación cruzada (10 por defecto).
  - `maxdepth`: profundidad máxima a la que puede llegar el árbol (30 por defecto).
- `tree2`: Modelo considerando inicialmente el árbol de regresión completo (`cp`=0). Posteriormente, con la regla *one-standard-error*[3] de Breiman et al. (1984) se selecciona el valor óptimo del hiperparámetro con el que se poda el árbol.

Podemos evaluar la precisión de ambos modelos en la muestra de test. Las métricas que obtenemos son las que se indican en el Cuadro 4.1.

Cuadro 4.1: Valores de diversas métricas que miden el desempeño en la predicción de dos modelos de árboles de decisión, `tree` y `tree2`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
tree	-8.219	2215.381	1431.169	-17.226	30.857	0.939
tree2	-18.519	1239.912	747.874	-2.482	9.969	0.981

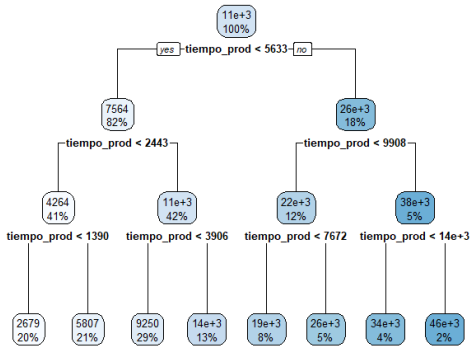
El primer árbol construido es un poco peor en términos de precisión que el segundo. El modelo consta de 8 nodos terminales. Las variables ordenadas de mayor a menor importancia son: `tiempo_prod` (61.7), `cant_m3_prod` (36.6), `grueso_numer` (0.8), `largo` (0.5) y `ancho` (0.4). No obstate, en cada división siempre se emplea la misma variable explicativa: `tiempo_prod`.

Si analizamos ahora el segundo árbol construido, las métricas que se obtienen son en general un poco mejores (se reduce el MAPE y aumenta el pseudo R-cuadrado). No obstante, el modelo también resulta mucho más complejo que el anterior y se pierde interpretabilidad. La importancia de las variables es similar al caso anterior: `tiempo_prod` (61.2), `cant_m3_prod` (36.8), `grueso_numer` (0.9), `largo` (0.6) y `ancho` (0.5) pero, en este caso, también se utlizan para realizar los cortes las variables referentes a los metros cúbicos y el largo (junto con el tiempo de producción).

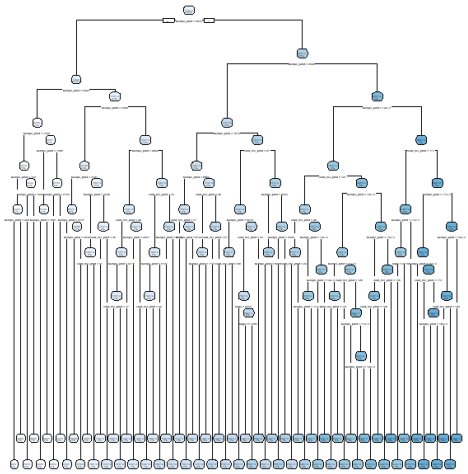
Este modelo ya anticipa la relevancia que tomará la variable `tiempo_prod` en los sucesivos modelos considerados.

[2]Therneau T, Atkinson B (2022). rpart: Recursive Partitioning and Regression Trees. R package version 4.1.19, https://CRAN.R-project.org/package=rpart.

[3]Como se menciona en Fernández et al. (2021), se seleccionaría el modelo más simple dentro de un error estándar de la precisión del modelo correspondiente al valor óptimo.



(a) Dendrograma del modelo `tree`.



(b) Dendrograma del modelo `tree2`.

Figura 4.1: Árboles de regresión construidos.

Bosques aleatorios

En esta ocasión se hace uso del paquete `randomForest` (Liaw y Wiener, 2002). Igual que antes, pondremos en práctica el método *random forest* entrenando dos modelos distintos.

- `rf`: El primer modelo considerado utilizará como hiperparámetro de complejidad `mtry` el recomendado para los modelos de regresión, es decir, `mtry`$= p/3$[4], siendo `mtry` el número de variables explicativas consideradas en cada corte y p la cantidad total de variables explicativas.

- `rf2`: En el segundo modelo, se establece una malla de valores para el parámetro `mtry`. El número de variables seleccionadas en cada división resulta `mtry`$= 3$, pues es la cantidad de variables seleccionada para aleatorizar en cada división que menor MSE produce.

A pesar de ambos modelos contar con la misma cantidad de árboles (500), se puede apreciar la mejora del modelo `rf2` respecto del modelo `rf`. Mientras el primer modelo tiene un $MSE_{rf} = 2556776$, el segundo modelo reduce esta cifra considerablemente, con un $MSE_{rf2} = 1187898$.

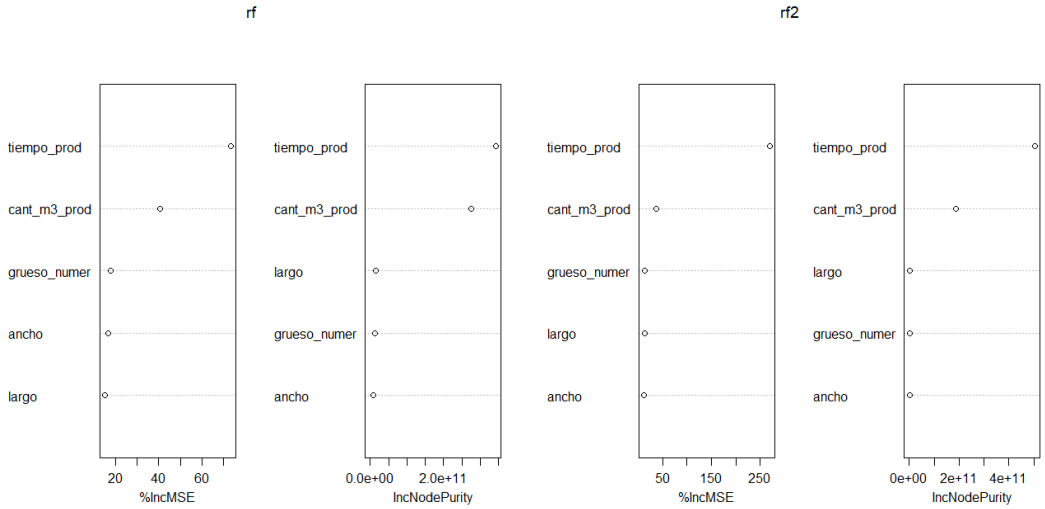
De nuevo, es posible evaluar la precisión de ambos modelos en la muestra de test mediante la serie de métricas indicadas en el Cuadro 4.2.

Cuadro 4.2: Valores de diversas métricas que miden el desempeño en la predicción de dos modelos de bosques aleatorios, `rf` y `rf2`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
rf	17.137	1598.754	863.910	-13.354	18.294	0.968
rf2	-7.469	1131.037	651.680	-0.860	6.545	0.984

[4]En realidad, se toma el máximo entre 1 y la parte entera por defecto de $p/3$. En nuestro caso, como $p = 5$, entonces `mtry`$= 1$.

En cuanto a la importancia de las variables explicativas, en este caso podemos emplear las medidas %IncMSE e IncNodePurity. La primera de ellas indica la pérdida de precisión del modelo al excluir cada variable, mientras que la segunda expresa cuánto aumenta el error del modelo cuando una variable concreta se permuta o baraja aleatoriamente.



(a) Importancia de las variables en `rf`.

(b) Importancia de las variables en `rf2`.

Figura 4.2: Métricas IncMSE e IncNodePurity de cada uno de los modelos considerados.

Podemos ver de nuevo la relevancia que toma la variable `tiempo_prod` en los modelos generados. Es la variable que más destaca en términos de %IncMSE e IncNodePurity.

Stochastic Gradient Boosting

El algoritmo *stochastic gradient boosting* está implementado en el paquete `gbm` (Greenwell et al. 2022) de R. Existen otros paquetes, como `caret` (Kuhn, 2008), que también permiten la construcción de modelos *SGB*. Para ello, habría que especificar el método con `method="gbm"` en la función `train()`.

Se construyen los dos modelos explicados a continuación:

- `gbm.fit`. El primer modelo se construye con el paquete `gbm`. Los hiperparámetros considerados son 1000 iteraciones (`ntrees` = 1000) y 10 grupos para validación cruzada (`cv.folds` = 1000).
- `gbm.fit2`. Para este modelo haremos uso del paquete `caret`. Se construye un primer modelo y se utiliza validación cruzada para la selección de hiperparámetros. Los parámetros seleccionados en este primer modelo en base a los que menor RMSE producen son: 100 árboles, 3 la profundidad de los árboles, 10 el número mínimo de observaciones en un nodo terminal y 0,1 el parámetro de regularización que se mantuvo fijo. Estos valores serán los empleados en el segundo modelo, en el cual se crea una malla del parámetro de regularización con los valores {0.001, 0.01, 0.5, 0.1, 0.3}. En el modelo resultante, el parámetro de regularización seleccionado resulta ser de nuevo 0.1.

Una vez construidos estos dos modelos, se puede evaluar su rendimiento en la muestra de test con el cálculo de las medidas de precisión habituales. En general, si comparamos a los dos, el segundo modelo mejora al primero en todas las métricas excepto en el MPE.

Si se calcula la importancia de las variables en este segundo modelo, de nuevo se vuelve a obtener que la más importante es la variable del tiempo de producción con un valor de 100. En orden, le

Cuadro 4.3: Valores de diversas métricas que miden el desempeño en la predicción de dos modelos de *SGB*, `gbm.fit` y `gbm.fit2`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
gbm.fit	-13.982	1193.361	684.229	-1.660	8.691	0.982
gbm.fit2	-6.496	1117.413	648.052	-2.335	7.803	0.985

siguen: `cant_m3_prod`, con 7.676; `grueso_numer`, con 0.04387 y `ancho`, con 0.008259. Se muestra en la Figura 4.3 el efecto parcial de las dos primeras variables sobre la variable respuesta de consumo.

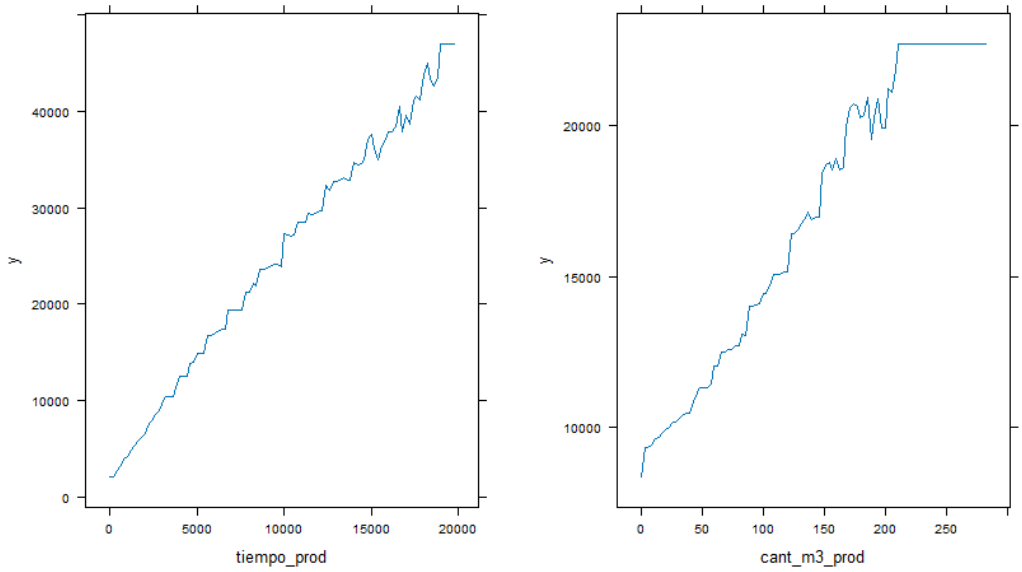


Figura 4.3: Efecto parcial del tiempo de producción y la cantidad de metros cúbicos producidos sobre el consumo.

Estos gráficos permiten visualizar el efecto marginal de variables explicativas sobre la variable respuesta. Como era de esperar, a medida que aumenta el tiempo de producción o los metros cúbicos producidos, aumenta el consumo. En ambos casos, el comportamiento es lineal, monótono y creciente.

Extreme Gradient Boosting

El algoritmo *extreme gradient boosting*, también conocido como *XGBoost*, es una de las técnicas de *boosting*. Este algoritmo se encuentra implementado en el paquete `xgboost` (Chen et al. 2024) de R. También se puede llevar a cabo con ayuda del paquete `caret`, especificando en las opciones `method="xgbTree"`.

Construiremos los dos modelos de *XGBoost* siguientes:

- `model_xgboost`. En el primer modelo se hace uso de la función `xgb.train()` del paquete `xgboost`. Este modelo se construye a partir de 200 iteraciones *boosting* y una profundidad máxima de árbol de 3. El RMSE más bajo se encuentra en la iteración número 140.

- `model_xgboost2`. En el segundo modelo se considera una malla de valores de los hiperparámetros. Para el número de iteraciones *boosting*, se consideran las cantidades {50, 100, 150, ... , 900, 950, 1000}. En cuanto al parámetro de regularización, se consideran los valores {0.025, 0.05, 0.1, 0.3}. Por último, para la profundidad máxima del árbol se estudian consideran las cifras {2, 3, 4, 5, 6}. Tras el entrenamiento del modelo, los hiperparámetros que minimizan la métrica considerada (RMSE) son los siguientes: 100 iteraciones *boosting*, 4 de profundidad máxima del árbol y 0.1 el parámetro de regularización.

Como en los dos métodos anteriores, se evalúa la precisión de ambos modelos en la muestra de test. En este caso, las métricas obtenidas son bastante similares: mientras el primer modelo tiene un menor RMSE y un mayor pseudo R-cuadrado que el segundo, este tiene un menor MAPE que el primero (ver Cuadro 4.4). No obstante, la diferencia en estas medidas es casi imperceptible, resultando en dos modelos casi idénticos en precisión al considerar la muestra de test (desconocida por los modelos hasta el momento de evaluarlos).

Cuadro 4.4: Valores de diversas métricas que miden el desempeño en la predicción de dos modelos de *XGBoost*, `model_xgboost` y `model_xgboost2`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
model_xgboost	-13.255	1120.670	643.660	-0.990	6.832	0.985
model_xgboost2	-18.475	1124.020	636.321	-1.257	6.613	0.984

En la Figura 4.4, se puede ver la evolución del RMSE de validación cruzada con 3-grupos según los parámetros de tuneado para el modelo `model_xgboost2`.

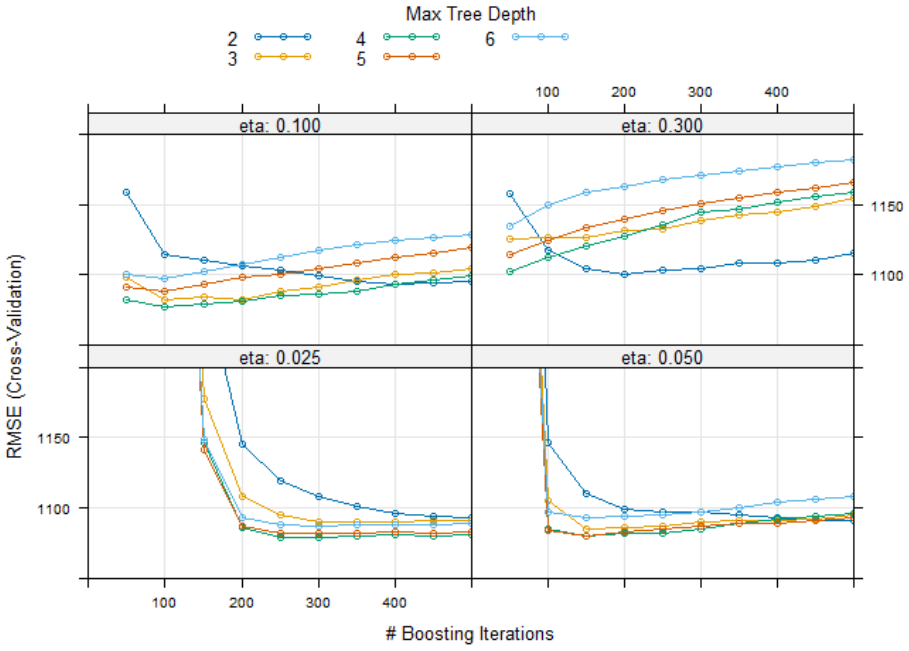


Figura 4.4: Evolución del RMSE de validación cruzada para el modelo `model_xgboost2`.

Métodos de regularización

En este apartado, se construirán modelos según los tres tipos de métodos de regularización descritos en la Subsección 3.3.4, que comprendía la regresión *ridge*, *LASSO* y *elastic net*.

En los dos primeros modelos, se hará uso del paquete `glmnet` (Friedman et al. 2010) de R. Para ajustar modelos de regresión *ridge* habrá que indicar en la función `glmnet()` el parámetro `alpha=0`, y para modelos *LASSO* el parámetro será `alpha=1`, aunque este ya es el empleado en la función por defecto.

El tercer modelo construido será empleando el método *elastic net*. Para ello, se hará uso del paquete `caret`.

- `fit.ridge`. En el modelo de regresión *ridge* se selecciona el parámetro de penalización por validación cruzada, empleando la función `cv.glmnet()`. Los coeficientes estimados por el modelo son:
  - `(Intercept)`: 1536.61395505
  - `grueso_numer`: -21.89189967.
  - `ancho`: -0.35327803.
  - `largo`: 0.06240894.
  - `cant_m3_prod`: 90.36709453.
  - `tiempo_prod`: 1.85525973.

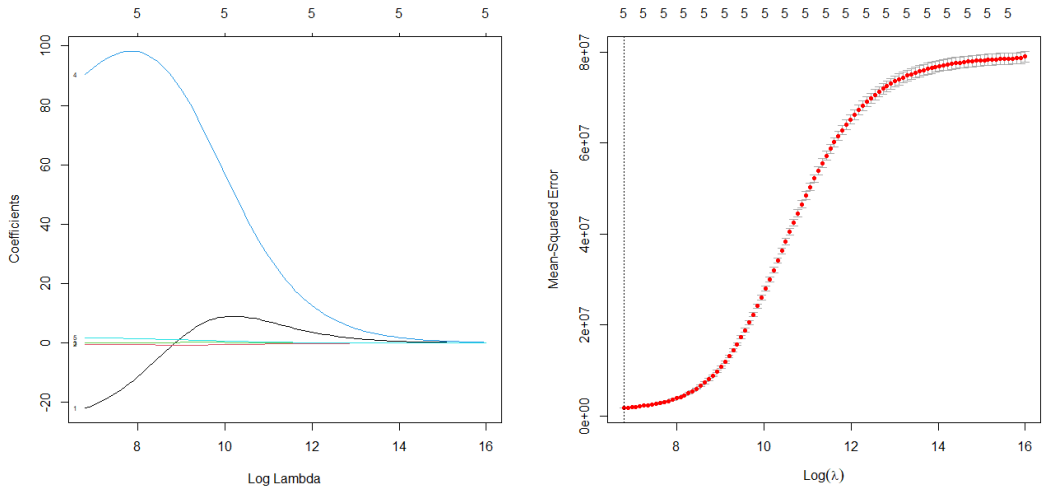


Figura 4.5: Evolución de los coeficientes en función del logaritmo del parámetro de penalización (izquierda) y evolución del MSE de validación cruzada en función del logaritmo del parámetro de penalización (derecha) en el modelo `fit.ridge`.

- `fit.lasso`. De manera análoga, en el modelo de regresión *LASSO* se selecciona el parámetro de penalización por validación cruzada. En este caso, también se hace uso de la función `cv.glmnet()`. Los coeficientes estimados por el modelo son:
  - `(Intercept)`: 343.616619.
  - `grueso_numer`: -4.027984.

- `ancho`: 0.
- `largo`: 0.
- `cant_m3_prod`: 61.628713.
- `tiempo_prod`: 2.259892.

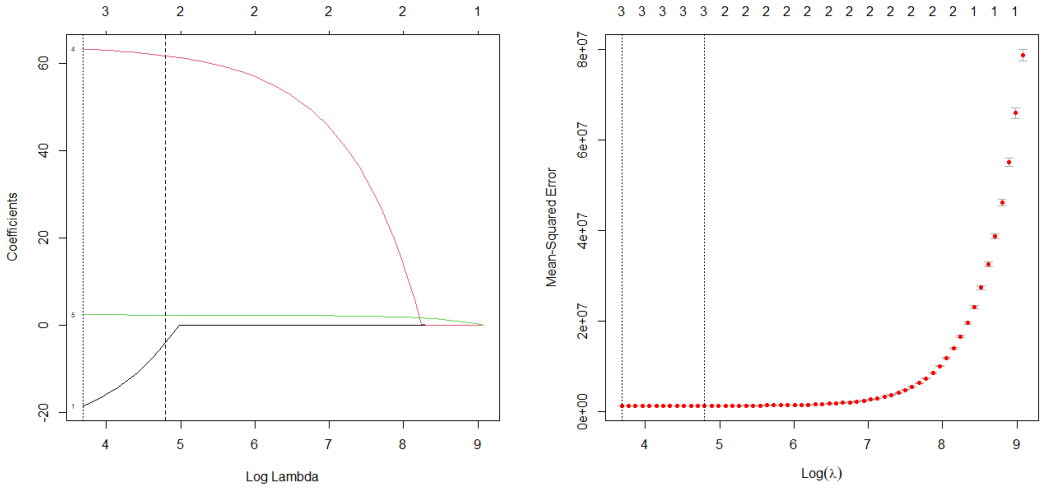


Figura 4.6: Evolución de los coeficientes en función del logaritmo del parámetro de penalización (izquierda) y evolución del MSE de validación cruzada en función del logaritmo del parámetro de penalización (derecha) en el modelo `fit.lasso`.

Como podemos observar en la Figura 4.6, el método *LASSO* fuerza a que las estimaciones de los coeficientes se vayan a cero cuando el valor del parámetro de penalización es lo suficientemente grande. De este modo, también se realiza una selección de las variables más importantes para el modelo. Por otro lado, el método *ridge* no fuerza a que las estimaciones de los coeficientes sean cero (aunque sí es posible que sean muy cercanas a cero en las variables menos importantes, como se ve en la Figura 4.5), y en el modelo final aparecerán todas las variables.

- `fit.glmnet`. Por último, se construye el modelo de *elastic net*. En este caso, es preciso llevar a cabo un preprocesado de las variables explicativas consistente en un centrado y escalado. Se considera validación cruzada de 5 grupos. Se utiliza el RMSE de validación cruzada (el menor) para seleccionar la combinación de los hiperparámetros `alpha` y `lambda` de regularización. Estos errores pueden verse en la Figura 4.7.

Una vez construido los tres modelos, podemos evaluar las predicciones en las muestras de test. De estos tres, el modelo `fit.ridge` presenta valores más elevados de error y un pseudo R-cuadrado ligeramente más bajo que los otros dos, como se puede ver en el Cuadro 4.5. Podríamos decir que, de los tres, es el que peor calidad predictiva presenta. Aunque realmente, en términos del pseudo R-cuadrado, en los tres modelos se obtienen prácticamente los mismos valores. El modelo de *elastic net* (que hemos llamado `fit.glmnet`) es el que presenta medidas más bajas de error, aunque no es demasiada la diferencia con respecto al modelo `fit.lasso`.

Vecinos más próximos

Para construir un modelo de regresión, empleando la técnica de k-vecinos más próximos, hacemos uso del paquete `caret` de R, especificando el método `method="knn"` en la función `train()`. El hiperparámetro del modelo es el número k de vecinos y para seleccionarlo realizamos validación cruzada

Cuadro 4.5: Valores de diversas métricas que miden el desempeño en la predicción de tres métodos de regularización, `fit.ridge`, `fit.lasso` y `fit.glmnet`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
fit.ridge	-13.558	1248.231	796.017	-9.814	14.283	0.981
fit.lasso	-4.992	1167.187	668.877	-4.091	8.718	0.983
fit.glmnet	-7.957	1145.753	653.495	-3.001	8.062	0.984

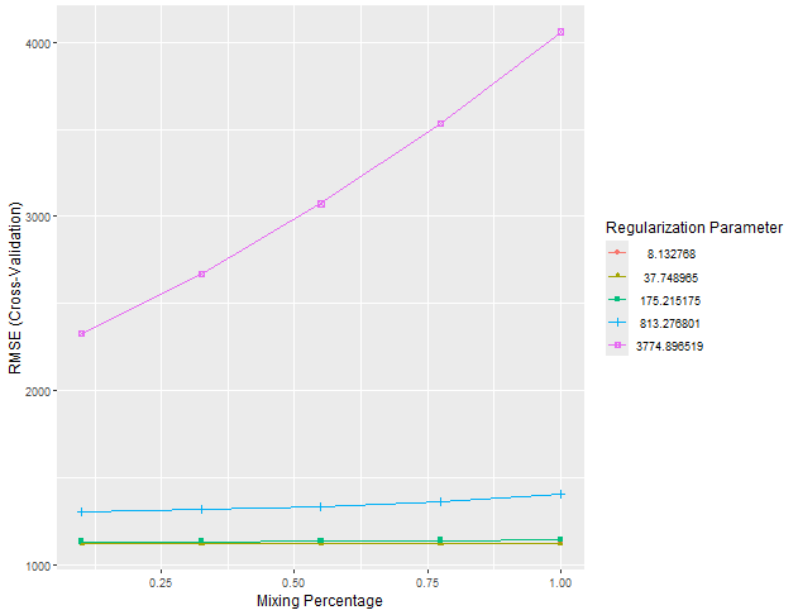


Figura 4.7: Errores de validación cruzada en función de los hiperparámetros del modelo `fit.glmnet`.

con 10 grupos. Finalmente, el parámetro que minimiza el error RMSE de validación cruzada obtenido se corresponde con $k = 6$. En la Figura 4.8 puede verse la evolución del RMSE de validación cruzada en función del número de vecinos considerados.

Si calculamos la importancia de las variables de este modelo, obtenemos el siguiente orden: `tiempo_prod` (100), `cant_m3_prod` (86.040), `largo` (1.272), `ancho` (0.210) y `grueso_numer` (0). Finalmente, también es posible evaluar la precisión de las predicciones con nuestros datos de la muestra de test. Las métricas obtenidas se recogen en el Cuadro 4.6.

Cuadro 4.6: Valores de diversas métricas que miden el desempeño en la predicción del modelo de k-vecinos más próximos `knn.fit`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
knn.fit	129.263	1381.448	839.876	-5.157	13.497	0.976

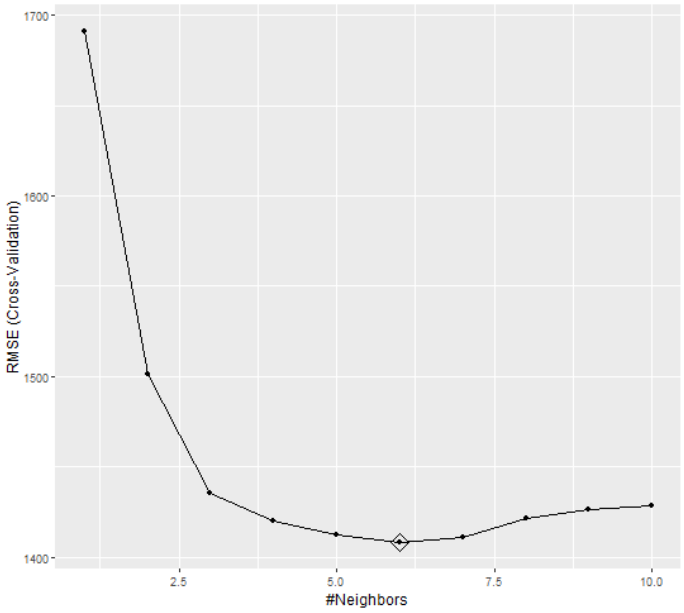


Figura 4.8: Evolución del error RMSE de validación cruzada en función del hiperparámetro k en el modelo `knn.fit`.

Projection pursuit

El último método considerado es el de búsqueda de proyecciones. Para implementarlo, de nuevo se hace uso del paquete `caret` de R. En este caso, habrá que seleccionar el método `"ppr"` dentro de las opciones de la función `train()`. Se emplea validación cruzada con 10 grupos para seleccionar el valor óptimo del hiperparámetro `nterms`, referente a los términos *ridge*.

El resultado que se obtiene es el de `nterms=3` y los coeficientes de los términos *ridge* son: 8923.8849, 261.2752 y 304.2447, respectivamente. Las estimaciones de las funciones *ridge* del modelo puede verse en la Figura 4.9.

Las medidas que se obtienen al evaluar las predicciones en la muestra de test se pueden ver en el Cuadro 4.7.

Cuadro 4.7: Valores de diversas métricas que miden el desempeño en la predicción del modelo de *projection pursuit* `ppr.fit`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
ppr.fit	-12.293	1085.297	631.630	-0.769	6.781	0.985

Comparación de los modelos

En este apartado, se pretende hacer una revisión global de todos los modelos construidos para tratar de predecir la variable de consumo eléctrico `Medida_Fabrica`, relativa al consumo total de la fábrica.

En primer lugar, se recogen de nuevo pero en una misma tabla las distintas medidas de error obtenidas de cada modelo considerado.

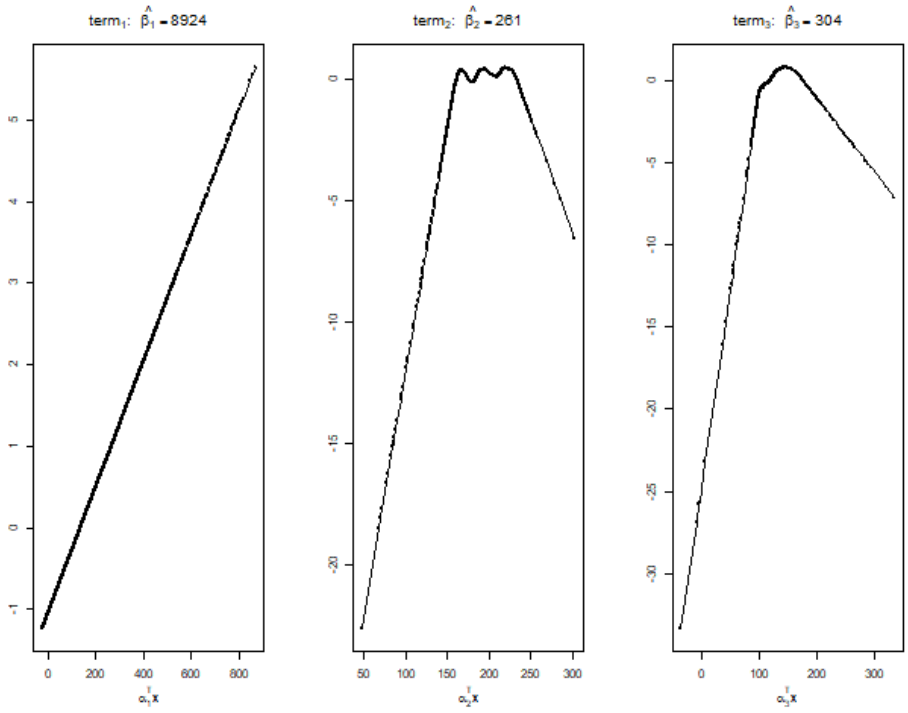


Figura 4.9: Funciones *ridge* estimadas del modelo `ppr.fit`.

Como se puede ver en el Cuadro 4.8, ningún modelo destaca demasiado sobre los demás. En general, todos tienen alrededor de un 0.98 de pseudo R-cuadrado —que, recordemos, mide la proporción de variabilidad de la respuesta en nuevas observaciones explicada por el modelo—. El modelo con menor RMSE, MAE y MPE es el de *projection pursuit*, además de tener el R-squared más alto. En cuanto al MAPE, están prácticamente a la par los modelos de *XGBoost* y el modelo `ppr.fit`. Por otro lado, los modelos `rf2` y `gbm.fit2` también tienen unas métricas similares. Tal vez, si solamente pudiésemos considerar un modelo, entonces tendríamos que escoger uno de los cinco que acabamos de mencionar. Sin embargo, dado que también entra en juego la aleatoriedad de la semilla a la hora de establecer la partición de los datos en muestra de test y entrenamiento, no podemos ser tajantes en nuestras afirmaciones.

A modo ilustrativo, podemos considerar uno de ellos y ver qué tal se ajustan las predicciones a los datos reales de la muestra de test. Por ejemplo, consideremos el modelo construido por el método de *projection pursuit*, `ppr.fit`.

Podemos ver en el Cuadro 4.9 de manera simbólica datos de diez partes de producción. A estos partes de producción se les agrega en la primera columna la predicción del consumo (correspondiente a la variable `Medida_Fabrica`).

Otra herramienta que podemos emplear es un gráfico de dispersión de las observaciones frente a las predicciones. Cuanto más cerca de la recta $y = x$ se sitúen los valores representados, mayor será la calidad del modelo predictivo considerado. En la Figura 4.10, se representan la recta $y = x$ en rojo y el ajuste lineal en azul.

4.1.2. Consumo de la variable `Medida_MDF`

Como hemos mencionado anteriormente, con las demás variables de consumo se obviará el proceso de desarrollar los modelos y solamente se presentará una tabla con las medidas de error obtenidas en

Cuadro 4.8: Medidas de error en los diferentes modelos construidos cometido al predecir la variable de consumo `Medida_Fabrica` en la muestra de test.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
tree	-8.219	2215.381	1431.169	-17.226	30.857	0.939
tree2	-18.519	1239.912	747.874	-2.482	9.969	0.981
rf	17.137	1598.754	863.910	-13.354	18.294	0.968
rf2	-7.469	1131.037	651.680	-0.860	6.545	0.984
gbm.fit	-13.982	1193.361	684.229	-1.660	8.691	0.982
gbm.fit2	-6.496	1117.413	648.052	-2.335	7.803	0.985
model_xgboost	-13.255	1120.670	643.660	-0.990	6.832	0.985
model_xgboost2	-18.475	1124.020	636.321	-1.257	6.613	0.984
fit.ridge	-13.558	1248.231	796.017	-9.814	14.283	0.981
fit.lasso	-4.992	1167.187	668.877	-4.091	8.718	0.983
fit.glmnet	-7.957	1145.753	653.495	-3.001	8.062	0.984
knn.fit	129.263	1381.448	839.876	-5.157	13.487	0.976
ppr.fit	-12.293	1085.297	631.630	-0.769	6.781	0.985

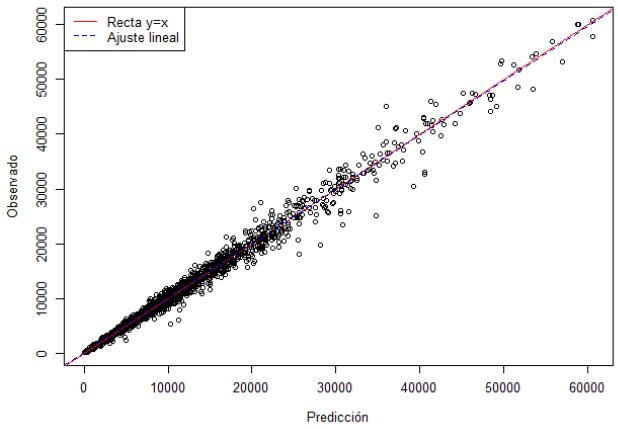


Figura 4.10: Gráfico de dispersión de los valores observados frente a las predicciones.

Cuadro 4.9: Ejemplos de partes de producción junto con la predicción de consumo estimada.

Predicción	Medida_Fabrica	grueso_numer	ancho	largo	cant_m3_prod	tiempo_prod
6251.800	6513	30	2440	4200	27.669	2038
7985.136	8328.25	28	2440	4500	34.740	2581
11608.270	11960.18	19	2070	5600	37.442	3926
11822.390	13105	16	2100	5700	41.559	3894
9306.315	10318	19	2440	4500	35.882	3027
10035.801	11459	15	2440	3050	35.498	3299
6063.579	6782	8	2520	4260	20.611	2016
12516.017	13378.50	8	2130	5058	39.129	4156
7568.273	8085.94	8	2130	5052	23.329	2494
10072.927	9770.87	16	2100	5700	39.261	3264

cada uno de ellos.

En este caso, destacan por tener un bajo RMSE el modelo de *random forest* `rf2`, los modelos *XGBoost*, y el modelo de *projection pursuit* `ppr.fit`. Además, estos modelos son también los que más bajo tienen el MAPE, y en términos del pseudo R-cuadrado son realmente muy parecidos. Podríamos considerar, por ejemplo, el modelo con menor MAPE, que es el bosque aleatorio `rf2`.

Del mismo modo que en la sección anterior, podemos representar mediante un gráfico de dispersión los valores observados frente a los predichos por el modelo `rf2`. En la Figura 4.11, se representan la recta $y = x$ en rojo y el ajuste lineal en azul.

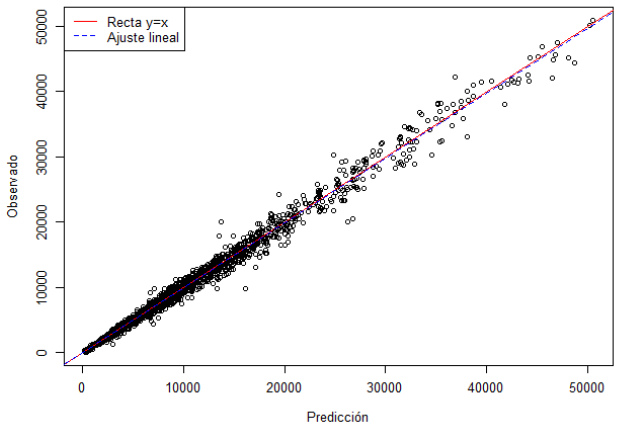


Figura 4.11: Gráfico de dispersión de los valores observados frente a las predicciones.

Cuadro 4.10: Medidas de error en los diferentes modelos construidos para predecir la variable de consumo `Medida_MDF`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
tree	-35.453	2109.351	1348.352	-22.776	35.847	0.929
tree2	-42.648	1031.333	641.559	-3.387	10.307	0.983
rf	4.178	1191.291	703.783	-17.641	22.164	0.977
rf2	-37.443	853.993	528.692	-1.657	6.635	0.988
gbm.fit	-63.509	1058.624	618.302	-3.422	9.327	0.982
gbm.fit2	-39.092	905.012	566.840	-4.228	9.446	0.987
model_xgboost	-42.292	843.724	528.566	-1.918	7.227	0.989
model_xgboost2	-29.626	833.518	512.807	-2.432	7.216	0.989
fit.ridge	-32.647	1082.921	680.940	-12.455	16.681	0.981
fit.lasso	-37.612	1034.919	607.755	-6.682	11.189	0.983
fit.glmnet	-49.315	1017.022	582.181	-4.292	9.445	0.983
knn.fit	-6.444	1122.755	687.408	-8.009	14.701	0.980
ppr.fit	-29.293	822.747	509.190	-2.320	7.540	0.989

4.1.3. Consumo de la variable `Medida_MW`

El tercer caso a considerar es el de la variable de consumo eléctrico del microondas. El procedimiento para entrenar los modelos es análogo a los anteriores. Cabe destacar que al calcular medidas de importancia, ya no era la variable `tiempo_prod` la de mayor valor (como en los modelos construidos para las dos variables de consumo precedentes). Esta pasaba a un segundo plano, y la primera era en todos los modelos la variable `cant_m3_prod`. Por ejemplo, en el modelo `ppr.fit`, el orden de importancia era: cantidad de metros cúbicos (100), tiempo de producción (91.733), largo (2.008), grueso (1.136) y ancho (0). En el modelo `model_xgboost2` es incluso más pronunciada esta diferencia: cantidad de metros cúbicos (100), tiempo de producción (20.0778), grueso (0.3869), ancho (0.2552) y largo (0).

Si nos fijamos en el Cuadro 4.11, podemos apreciar que los valores de MAPE son más altos y los valores del R-squared son más bajos que en los modelos para las dos variables anteriores. Si tuviésemos que considerar solamente uno para realizar predicciones, podría ser uno de los tres siguientes: `rf2`, `model_xgboost2` o `ppr.fit`, ya que son de los que menor RMSE y MAPE tienen simultáneamente. En general, todos los modelos tienen un R-squared similar, pero estos tres modelos son los que más alto lo tienen.

Cuadro 4.11: Medidas de error en los diferentes modelos construidos para predecir la variable de consumo `Medida_MW`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
tree	1.198	135.577	89.725	-18.474	34.091	0.891
tree2	2.500	107.732	69.686	-8.066	21.395	0.931
rf	3.607	94.967	61.551	-10.201	19.771	0.946
rf2	3.484	93.617	59.388	-2.910	13.814	0.948
gbm.fit	3.790	101.728	63.938	-7.171	18.562	0.939
gbm.fit2	2.732	95.160	61.538	-6.512	17.489	0.946
model_xgboost	2.858	93.525	59.950	-4.183	15.456	0.948
model_xgboost2	2.557	90.817	58.239	-4.457	14.877	0.951
fit.ridge	4.514	104.706	68.435	-12.849	22.189	0.935
fit.lasso	4.993	102.503	65.946	-9.526	19.270	0.938
fit.glmnet	4.890	98.487	62.341	-5.300	16.412	0.942
knn.fit	3.266	95.964	63.811	-9.515	20.747	0.945
ppr.fit	3.981	92.063	59.230	-4.104	15.935	0.950

4.1.4. Consumo de la variable `Medida_Refino`

La siguiente variable que se pretende predecir es la del consumo del refino. Como se puede ver en el Cuadro 4.12, aunque el valor R-squared de los modelos sigue siendo bastante elevado, también lo es el valor del error MAPE. El modelo con MAPE más bajo es `rf2`, con un valor de 15.698 y los que le siguen son de nuevo los modelos destacados en los casos anteriores: los dos modelos de *XGBoost* y el modelo de *projection pursuit*. De estos cuatro modelos, el que mayor pseudo R-cuadrado tiene es el modelo `ppr.fit`, pero tan solo es cuestión de milésimas.

4.1.5. Consumo de la variable `Medida_Secado_y_Wesp`

La variable respuesta referente al consumo eléctrico del secado es la penúltima que se considera. De nuevo, se vuelven a construir todos los modelos con la muestra de entrenamiento y posteriormente se evalúan en la muestra de test.

En el Cuadro 4.13 podemos ver que los modelos han mejorado ligeramente si los comparamos con los construidos para la variable `Medida_Refino`, tanto en términos de MAPE, como de R-squared. Los modelos con menor MAPE y mayor R-squared vuelven a ser dos de los modelos destacados anteriormente: `rf2` y `ppr.fit`. También podría destacarse aquí el modelo `fit.glmnet`, que obtiene el tercer MAPE más bajo y el segundo R-squared más alto.

Cuadro 4.12: Medidas de error en los diferentes modelos construidos para predecir la variable de consumo `Medida_Refino`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
tree	-26.306	916.064	606.552	-32.632	47.491	0.901
tree2	-0.390	705.862	440.117	-14.405	26.234	0.941
rf	-0.123	668.802	412.856	-23.422	31.484	0.947
rf2	7.145	648.919	381.748	-6.179	15.698	0.950
gbm.fit	7.024	705.962	404.293	-8.865	20.024	0.941
gbm.fit2	4.942	649.874	386.293	-9.441	19.090	0.950
model_xgboost	7.440	646.834	388.981	-7.183	17.401	0.951
model_xgboost2	5.115	640.549	376.054	-7.855	17.251	0.952
fit.ridge	7.609	716.868	446.630	-19.909	28.210	0.939
fit.lasso	6.816	704.885	423.607	-14.574	23.168	0.941
fit.glmnet	13.692	693.195	410.683	-9.193	20.111	0.943
knn.fit	25.577	706.109	430.286	-18.875	29.580	0.941
ppr.fit	17.974	628.658	374.012	-5.797	16.606	0.953

4.1.6. Consumo de la variable `Medida_Resto`

Por último, se considera predecir la variable de consumo `Medida_Resto`, que hacía referencia a aquellos otros elementos de la línea de MDF distintos del secado, el refino y el microondas (como por ejemplo las calderas).

Esta es la variable que obtiene un menor R-squared en todos los modelos considerados. El más alto de todos, se alcanza con el modelo de *projection pursuit*. Así mismo, los valores de MAPE son también bastante elevados en todos los modelos. El menor de ellos se alcanza con el método de bosques aleatorios, en el modelo `rf2`, y es bastante cercano al alcanzado con el modelo `ppr.fit`. En el Cuadro 4.14 pueden verse estas métricas.

4.1.7. Comentarios

En esta sección se ha llevado a cabo un análisis de los modelos de *Machine Learning* ajustados para tratar de predecir las distintas variables respuesta de manera independiente. Tras ver los resultados, se puede considerar hacer los siguientes comentarios:

- Antes de nada, debemos tener presente que las muestras se dividen en base a una semilla fijada. Por ese motivo, un cambio en la semilla podría producir un ligero cambio en los resultados

Cuadro 4.13: Medidas de error en los diferentes modelos construidos para predecir la variable de consumo `Medida_Secado_y_Wesp`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
tree	-4.716	365.224	233.093	-21.518	36.569	0.919
tree2	-6.603	275.058	158.595	-5.480	15.428	0.954
rf	-5.113	284.206	171.175	-16.767	23.899	0.951
rf2	-7.846	265.740	149.037	-3.626	11.785	0.957
gbm.fit	-10.779	270.160	148.053	-7.245	14.932	0.956
gbm.fit2	-8.105	263.190	146.147	-6.796	14.298	0.958
model_xgboost	2.098	266.581	148.342	-4.570	13.356	0.957
model_xgboost2	-8.203	262.902	145.143	-6.236	13.724	0.958
fit.ridge	-11.266	265.651	167.667	-13.169	19.676	0.957
fit.lasso	-10.117	265.176	150.881	-9.508	16.022	0.957
fit.glmnet	-10.090	264.236	145.669	-5.656	12.857	0.958
knn.fit	16.609	282.129	173.781	-8.527	19.228	0.952
ppr.fit	-10.816	257.220	143.733	-3.212	11.920	0.960

obtenidos. Para mayor robustez, podrían considerarse varias semillas y promediar los resultados obtenidos para mitigar el efecto de aleatoriedad.

- En general, parece darse el mismo patrón en cuanto a precisión en los modelos. Por ejemplo, cuando se construyen dos modelos empleando el mismo método, el que se construye en base a una rejilla de valores de los hiperparámetros (el segundo) suele ser mejor que el otro[5].
- De los trece modelos generados, hay cuatro que siempre destacan por tener unas mejores métricas: el modelo de *random forest* (`rf2`), los dos modelos de *extreme gradient boosting* (`model_xgboost` y `model_xgboost2`), aunque el segundo acostumbra ser un poco mejor, y el modelo de *projection pursuit* (`ppr.fit`).
- Los árboles de regresión `tree` y `tree2` son generalmente los que peores medidas proporcionan. Esto puede deberse a que son considerados predictores *débiles*. De hecho, se aprecia una notoria mejora cuando se combinan muchos árboles para producir un método *ensemble*.
- Las dos variables que mejores métricas obtienen son la medida total de fábrica (con alrededor de un pseudo R-cuadrado de 0.985 y valores inferiores a 10 unidades de MAPE) y la medida de

[5]Suponiendo que se puedan ordenar los modelos de mejor a peor. Por ejemplo, podríamos considerar que un modelo es *mejor* que otro si al evaluarlos en las muestras de test las medidas de error son más bajas y el pseudo R-cuadrado es mayor.

Cuadro 4.14: Medidas de error en los diferentes modelos construidos para predecir la variable de consumo `Medida_Resto`.

Modelos	ME	RMSE	MAE	MPE	MAPE	R-squared
tree	-13.800	1045.431	631.798	-17.497	32.994	0.851
tree2	-11.389	845.582	498.725	-8.291	21.219	0.902
rf	-5.186	815.957	480.053	-12.900	22.316	0.909
rf2	-16.021	801.846	462.167	-5.616	16.475	0.912
gbm.fit	-20.342	810.454	466.430	-7.767	18.532	0.910
gbm.fit2	-17.651	798.608	458.448	-7.777	18.135	0.913
model_xgboost	-9.996	792.807	460.507	-6.223	17.188	0.914
model_xgboost2	-10.926	788.323	453.700	-7.499	17.672	0.915
fit.ridge	-9.163	812.839	491.349	-13.653	22.455	0.910
fit.lasso	-12.966	807.446	478.380	-12.130	21.013	0.911
fit.glmnet	-15.216	800.013	461.561	-7.421	17.807	0.912
knn.fit	-0.077	825.027	491.010	-10.310	21.338	0.907
ppr.fit	-14.400	781.879	445.723	-5.291	16.555	0.916

MDF (también alrededor de 0.985 de pseudo R-cuadrado y con algunos valores inferiores a 10 unidades de MAPE).

- Cuando se calcula la importancia de las variables en los modelos, las dos variables que más portancia tienen a lo largo de todos los modelos son `tiempo_prod` y `cant_m3_prod`. Las otras tres variables predictoras —ancho, largo y grueso— apenas cuentan con importancia, llegando en alguna ocasión a tener un valor de 0.

Una observación sobre la variable `Hum_astilla`

En un principio, esta variable se consideraba en el dataset ya que se presuponía que la humedad de la astilla iba a ser un factor que influyese en el consumo eléctrico. También, dada la ubicación de la empresa en Galicia, el factor lluvia o humedad debería ganar relevancia. No obstante, al incluir esta variable en los distintos modelos construidos resulta que la precisión prácticamente se mantiene igual en todos, si no empeora. Además, cuando se calculan las medidas de importancia, en algunos casos resulta ser la última medida de importancia. Por ejemplo, la importancia de las variables obtenida con el modelo `rf` si también incluimos la humedad de la astilla es: tiempo de producción (57.79786), cantidad de metros cúbicos (39.05104), ancho (17.83508), grueso (17.31589) , largo (16.22928) y humedad (13.21575). Mientras que en otros, como en el modelo `model_xgboost`, se coloca de tercera aunque

su magnitud no sea muy grande: tiempo de producción (100), cantidad de metros cúbicos (7.57632), humedad (0.05098), grueso (0.04018), largo (0.01590) y ancho (0).

El dilema con esta variable es si incluirla o no, ya en ciertos periodos de tiempo no se recoge la medida. Entonces, en este escenario podría optarse trabajar solamente con los partes de producción que sí tengan valores en la variable `Hum_Astilla` o en emplear alguna técnica conveniente para el tratamiento de datos faltantes como la imputación, teniendo en cuenta que la proporción de datos faltantes es de 0.1376114 en este caso.

En nuestro caso, dada la prácticamente nula mejora (o alteración) de la precisión de los modelos al introducirla, junto con la gran cantidad de partes de producción que no tienen una medida de la humedad de la astilla, se opta por no considerarla.

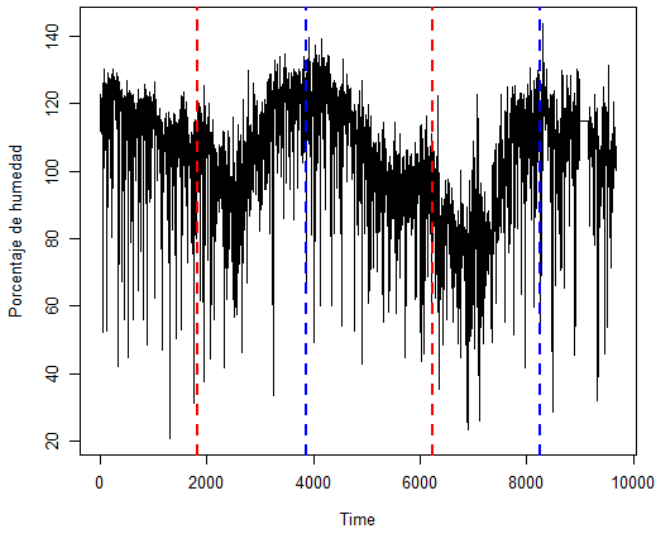


Figura 4.12: Representación de la serie temporal de la variable `Hum_Astilla`.

Una línea de investigación futura alternativa puede ser el uso de series de tiempo como herramienta predictiva y con las que se puede tratar de identificar la componente estacional. En la Figura 4.12, se representan las humedades medidas a lo largo de los dos años de tiempo considerados[6]. Empezando los datos en inicios de 2022, en líneas rojas verticales se marcan los días de inicio del verano y, en azul, los días de inicio del invierno. En esta figura se puede ver claramente la oscilación de la humedad a lo largo del año y la posible presencia de una componente estacional.

4.2. Predicción en la Línea 35

El caso considerado en la Línea 35 es algo más simple que el anterior. Como variable explicativa solamente contamos con los metros lijados (`m_lijado`) y trataremos de predecir el consumo eléctrico de la línea de lijado. Es por eso que se utilizará un método distinto a los puestos en práctica hasta el momento.

El método está basado en emplear remuestreo para la selección del modelo, descrito en la Subsección 3.3.4. De este modo no dependeremos del cumplimiento o no de las hipótesis estructurales de un modelo de regresión lineal. La idea es considerar como variables predictoras diferentes grados de la

[6]Para ser precisos, se representan las humedades de los partes de producción a lo largo de esos dos años.

variable (`m_lijado`) y mediante remuestreo seleccionaremos el número de predictores que se incluyen en el modelo.

Para construir este modelo, que llamaremos `fit35`, haremos uso del paquete `caret` de R y su función `train()`, en la que especificaremos que el método es `"leapSeq"`, es decir, empleando una búsqueda por pasos. Para escoger el hiperparámetro de complejidad, que en este caso es el número máximo de predictores, se emplea validación cruzada y toma el parámetro que minimiza el RMSE.

Finalmente, resulta que el mejor modelo se obtiene considerando cuatro variables explicativas: los términos hasta grado 4 de la variable (`m_lijado`). Los coeficientes estimados por el modelo son los siguientes:

- `(Intercept)`: 1304.081.
- `m_lijado`: 0.6046539.
- `I(m_lijado^2)`: -5.720427e-05.
- `I(m_lijado^3)`: 2.984327e-09.
- `I(m_lijado^4)`: -5.991741e-14.

Podemos evaluar la precisión del modelo final en la muestra de test. Las métricas se recogen en el Cuadro 4.15

Cuadro 4.15: Medidas de precisión al evaluar el modelo `fit35` sobre la muestra de test.

Modelo	ME	RMSE	MAE	MPE	MAPE	R-squared
fit35	2.137	336.680	256.190	-1.427	7.950	0.724

Si bien en esta situación el pseudo R-cuadrado es bastante menor a los obtenidos en los modelos de la Línea 232, las demás medidas son razonablemente buenas, destacando un MAPE inferior a 10 unidades. Además, si tenemos en cuenta como se han recogido las variables respuesta y explicativa —recordemos, juntando toda la producción de un turno de 8 horas en un solo parte de producción, y de manera similar con el consumo eléctrico—, de modo que no contamos con información del todo precisa, un pseudo R-cuadrado de casi el 0.75 es bastante bueno.

Para finalizar, se representan en la Figura 4.13 el ajuste del modelo a los datos de entrenamiento (a la izquierda) y el comportamiento frente a datos que el modelo desconoce, es decir, los de test (a la derecha).

A priori, podría pensarse que el efecto que produce la variable explicativa sobre la respuesta es proporcional y lineal, es decir, a más metros lijados más consumo. No obstante, también se plantean varios escenarios como:

- Es posible que otros factores no considerados interfieran, tales como la calidad del tablero o el estado en que se encuentran las máquinas involucradas.
- En caso de no ser continua la producción, encender la maquinaria podría suponer un gran consumo inicial (se puede relacionar esto con partes de producción de pocos metros lijados), pero luego, una vez que la maquinaría estuviese caliente, el consumo se amortiguaría.

Por otro lado, podemos ver que en la frontera, donde casi no hay valores, el modelo no se ajusta demasiado bien a la tendencia que describen los datos.

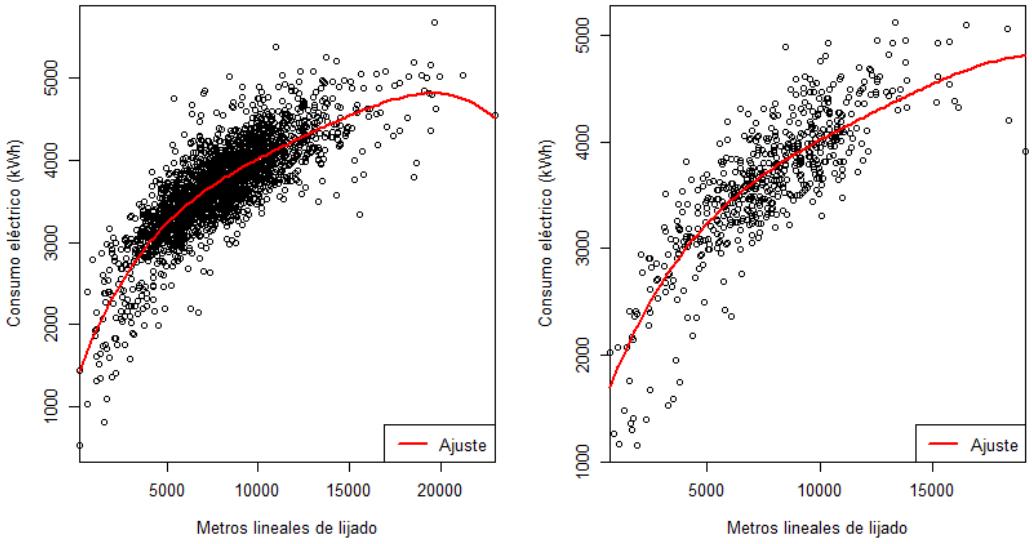


Figura 4.13: Ajuste del modelo final a los datos de entrenamiento (izquierda) y comportamiento frente a los datos desconocidos de test (derecha).

Capítulo 5

Conclusiones y líneas futuras

El objetivo de este proyecto ha sido el desarrollo de técnicas de predicción que permitiesen estimar el consumo eléctrico asociado a una planificación de la producción dada. En los dos primeros capítulos, se hace una composición de lugar para comprender el problema planteado por la empresa. Posteriormente, en el Capítulo 3, se realiza una introducción teórica a los fundamentos del aprendizaje supervisado y se describe alguno de sus métodos. Para finalizar, en el Capítulo 4, se lleva a cabo la construcción de los modelos contemplados en el capítulo anterior y se estudia y compara su desempeño al ponerlos en práctica.

Los modelos estimados para predecir consumos en la Línea 232 obtienen medidas de error similares y relativamente bajas al ser evaluados en la muestra de entrenamiento, mientras que los pseudo R-cuadrado que se obtienen oscilan entre el 0.85 y el 0.98. En general, los modelos que peor calidad presentan son los árboles de decisión. En cambio, al combinar árboles de regresión en los métodos de *random forest* y *boosting*, los resultados mejoran considerablemente. De hecho, los modelos que mayor calidad predictiva han demostrado son estos últimos, junto con el de *projection pursuit*.

En cambio, en la Línea 35 surge la problemática de contar con unos datos sin marca temporal precisa. El planteamiento en este caso ha sido el de llevar a cabo una reestructuración de los datos, agrupando en un parte de producción todos los partes de producción correspondientes a ese mismo turno. De este modo, aunque no tuviésemos la misma precisión temporal que en el caso de la Línea 232, se lograría relacionar el consumo eléctrico total de las 8 horas de duración de un turno con la producción total de ese mismo turno. El modelo entrenado en estas circunstancias no obtiene unas métricas de precisión tan buenas como las de la Línea 232, pero puede deberse a factores como el recientemente descrito, o el hecho de emplear solamente una variable explicativa. A pesar de todo, este modelo presenta un MAPE relativamente bajo (inferior al 10 %) y un pseudo R-cuadrado cercano a 0.75 al evaluarlo en nuevas observaciones.

En Finsa, actualmente el problema se aborda mediante unos ratios promedios de consumo por metro cúbico. Por parte de la empresa, el grado de satisfacción con los resultados obtenidos es alto, ya que el estudio demuestra que hay relación entre las variables respuesta y las explicativas, lo cual permite realizar previsiones fiables de producción. Además, en el análisis se ha demostrado la gran importancia de contar con datos que tengan una marca temporal adecuada.

Durante la elaboración de este trabajo, surgieron ideas que han quedado pendientes de desarrollo. Por ejemplo, podrían barajarse métodos alternativos de *Machine Learning* distintos a los considerados en este trabajo, como máquinas de soporte vectorial, regresión spline adaptativa multivariante o redes neuronales. Además, podría plantearse la construcción de un modelo que también considerase la calidad del producto, teniendo en cuenta que los datos están desbalanceados.

Entre las ideas de desarrollo futuro, también está la de tener en cuenta las paradas de producción. Es decir, tiempos cortos del periodo que duran los partes de producción en que la producción está detenida. Es algo que tal vez podría adaptarse fácilmente a los modelos que se han considerado en este trabajo: en lugar de utilizar el tiempo de producción —desde que inicia hasta que finaliza—,

considerar el tiempo real, que sería el resultado de sustraerle a aquel el tiempo que duran las paradas de producción.

Como última línea de investigación futura, se podría sopesar el empleo de herramientas de series temporales. Con estas, podría tratarse de identificar, por un lado, la parte de consumo eléctrico que se corresponde al producto y, por otro, la parte que se debe a condiciones estacionales como la humedad o la temperatura. Estas últimas condiciones afectarían al estado de la materia prima, provocando cambios en los parámetros considerados a la hora de procesarla y, consecuentemente, en el consumo.

Bibliografía

[1] Benthien J , Bähnisch C, Heldner S, Ohlmeyer M (2014) Effect of Fiber Size Distribution on Medium-Density Fiberboard Properties Caused by Varied Steaming Time and Temperature of Defibration Process. Wood and fiber science: journal of the Society of Wood Science and Technology 46: 175-185.

[2] Breiman L, Friedman JH, Stone, CJ, Olshen, RA (1984) Classification and Regression Trees. Taylor; Francis

[3] Burkov A (2019) The Hundred-Page Machine Learning Book.

[4] Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, Zhou T, Li M, Xie J, Lin M, Geng Y, Li Y, Yuan J (2024) xgboost: Extreme Gradient Boosting. R package version 1.7.7.1, https://CRAN.R-project.org/package=xgboost.

[5] Colaboradores de Wikipedia (2024) Finsa. Wikipedia, La enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Finsa&oldid=158000421. Accedido 5 de julio de 2024.

[6] Fernández Casal R, Costa Bouzas J, Oviedo de la Fuente M (2021). Aprendizaje Estadístico. https://rubenfcasal.github.io/aprendizaje˙estadistico.

[7] Financiera Maderera S.A. (2021) Estado de información no financiera 2021, Finsa. https://www.finsa.com/documents/20121/d25c43c2-4df5-1690-17b4-6c0710d9b579. Accedido 1 de julio de 2024.

[8] Financiera Maderera S.A. (2022) Estado de información no financiera 2022, Finsa. https://www.finsa.com/documents/20121/4e1ce942-944b-ae6c-d4c0-50c88f89ccdc. Accedido 1 de julio de 2024.

[9] Friedman J, Tibshirani R, Hastie T (2010) Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, *33*(1), 1-22. doi:10.18637/jss.v033.i01https://doi.org/10.18637/jss.v033.i01.

[10] Greenwell B, Boehmke B, Cunningham J, Developers G (2022) gbm: Generalized Boosted Regression Models. R package version 2.1.8.1, https://CRAN.R-project.org/package=gbm.

[11] Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer.

[12] Hillring B (2006) World trade in forest products and wood fuel. Biomass and Bioenergy, 30(10), 815-825. https://doi.org/10.1016/j.biombioe.2006.04.002.

[13] James G, Witten D, Hastie T, Tibshirani R (2021) An Introduction to Statistical Learning: With Applications in R, Second Edition. Springer.

[14] Kuhn, M (2008) Building Predictive Models in R Using the caret Package. Journal of Statistical Software, 28(5), 1–26. https://doi.org/10.18637/jss.v028.i05.

[15] Kuhn M, Johnson K (2018). Applied predictive modeling. Springer.

[16] Liaw A, Wiener M (2002) Classification and Regression by randomForest. R News 2(3),18–22. https://cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf.

[17] Malliris P (2023) The economy and war top lumber market challenges heading into 2023, Fastmarkets. https://www.fastmarkets.com/insights/economy-war-top-lumber-market-challenges-2023/. Accedido 1 de julio de 2024.

[18] R Core Team (2023) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria https://www.R-project.org/.

Apéndice A

Análisis descriptivo de las variables de la Línea 232

Antes de comenzar a hacer un análisis descriptivo de los datos referentes a la Línea 232, a mayores del acomodamiento de los datos descrito en la Sección 2.2, se realiza también una segunda depuración de los datos. Básicamente consiste en descartar los partes de producción en los que algún contador individual marca más consumo eléctrico que la variable `Medida_Fabrica` que, recordemos, mide el consumo total de la fábrica. La mayor parte de estos casos son partes de producción con un consumo total de fábrica asociado de 0 kWh, lo cual resulta poco razonable. Además, también se eliminan algunos partes de producción con valores atípicos en el consumo eléctrico (demasiado grandes): la causa era que durante un tiempo la medida era constante de 0 kWh y cuando se corrigió el error, la medida de consumo pasó a marcar —se entiende— el acumulado de los días que no tenían registro.

A.1. Análisis descriptivo

En este apartado se realizará un breve análisis descriptivo de aquellas variables de los datos que tienen mayor interés y que serán utilizadas para la construcción de los modelos.

Las variables `linea`, `linea_txt`, `fecha_produccion`, `turno`, `hora_inicio`, `hora_fin`, `num_parte`, `material` y `modelo_obsydian` ya fueron descritas previamente en la Sección 2.2. Por otro lado, consideraremos solamente las variables de cantidad de producción total (`cant_m2_prod` y `cant_m3_prod`), y no la útil (`cant_m2_prod_util` y `cant_m3_prod_util`).

- En la figura A.1 se describe la variable `grueso_numer`.
- En la figura A.2 se describe la variable `ancho`.
- En la figura A.3 se describe la variable `largo`.
- En la figura A.4 se describe la variable `cant_m2_prod`.
- En la figura A.5 se describe la variable `cant_m3_prod`.
- En la figura A.6 se describe la variable `tiempo_prod`.
- En la figura A.7 se describe la variable `Medida_Secado_y_Wesp`.
- En la figura A.8 se describe la variable `Medida_Resto`.
- En la figura A.9 se describe la variable `Medida_Refino`.
- En la figura A.10 se describe la variable `Medida_Fabrica`.

- En la figura A.11 se describe la variable `Hum_Astilla`.

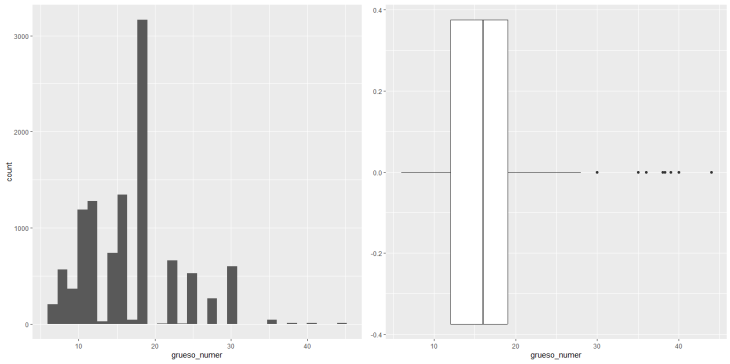


Figura A.1: Histograma y boxplot de la variable `grueso_numer`.

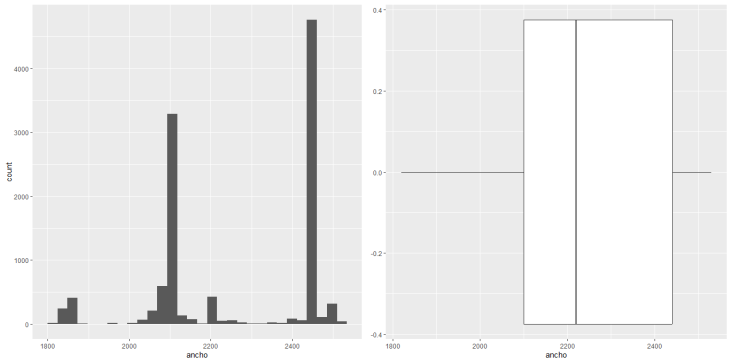


Figura A.2: Histograma y boxplot de la variable `ancho`.

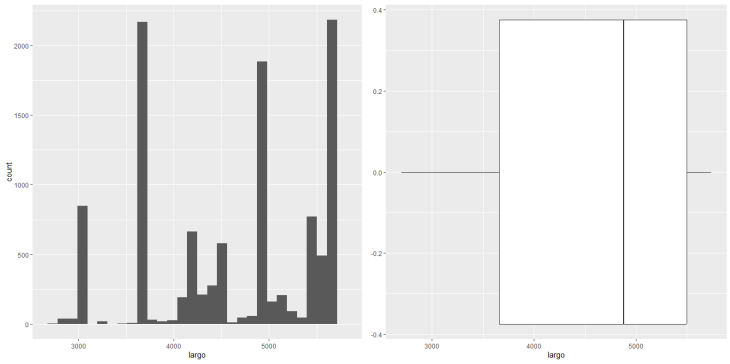


Figura A.3: Histograma y boxplot de la variable `largo`.

También podemos construir una matriz de correlaciones para ver la relación entre cada par de variables. En la figura A.12, las correlaciones positivas se muestran en color azul y las correlaciones negativas en color rojo. Además, la intensidad del color es proporcional a los coeficientes de correlación.

Por otro lado, en la tabla A.1 se recoge un resumen numérico de las variables utilizadas. Las unidades de las variables grueso, ancho y largo son los milímetros; las del tiempo de producción, los

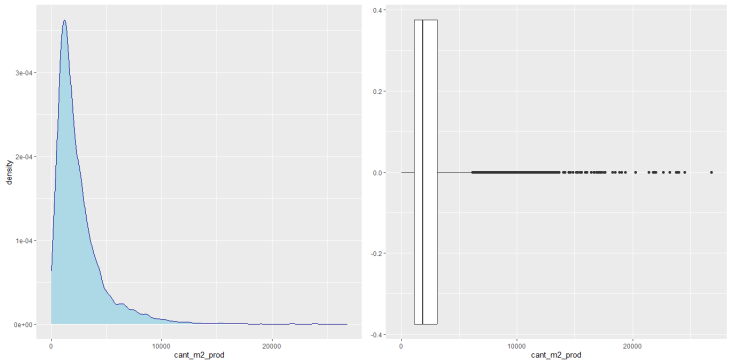


Figura A.4: Función de densidad y boxplot de la variable `cant_m2_prod`.

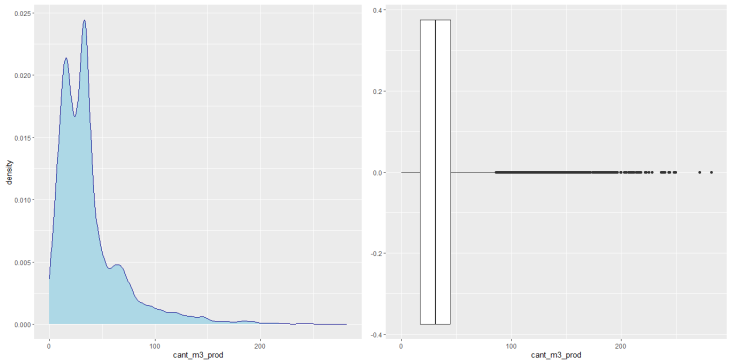


Figura A.5: Función de densidad y boxplot de la variable `cant_m3_prod`.

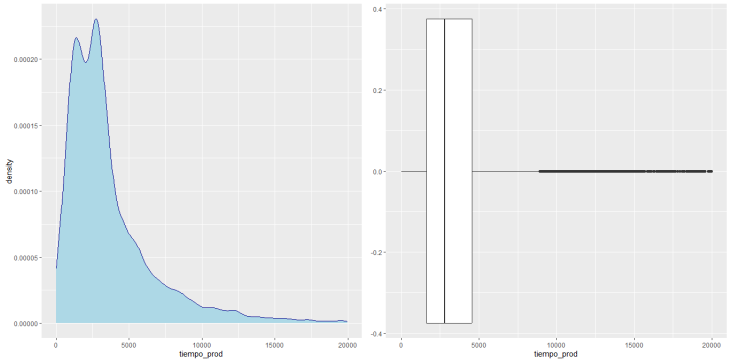


Figura A.6: Función de densidad y boxplot de la variable `tiempo_prod`.

segundos y las medidas de consumo eléctrico son kilovatios hora (kWh). En cambio, la humedad de la astilla no tiene unidades, sino porcentajes.

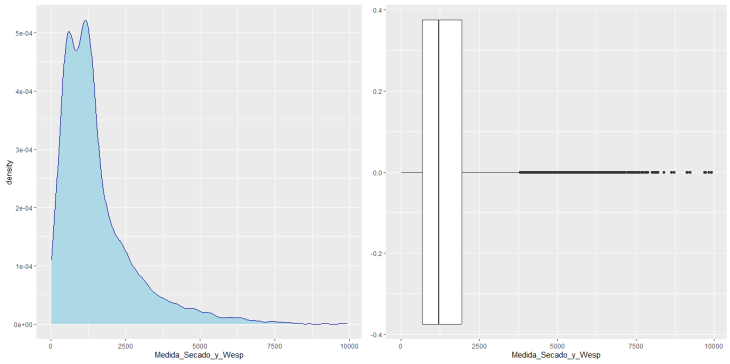


Figura A.7: Función de densidad y boxplot de la variable `Medida_secado`.

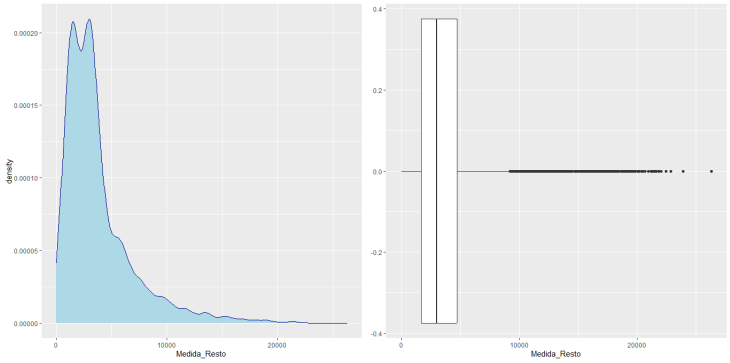


Figura A.8: Función de densidad y boxplot de la variable `Medida_Resto`.

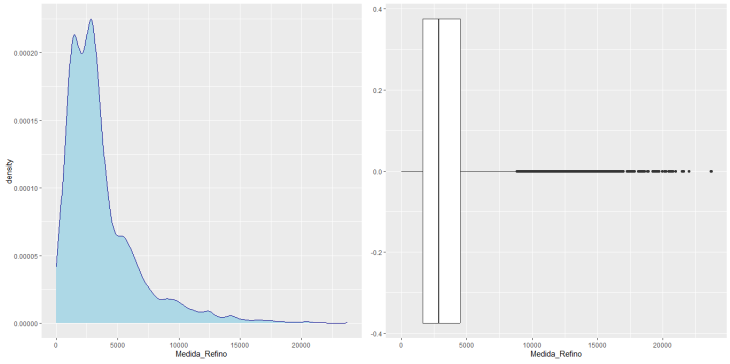


Figura A.9: Función de densidad y boxplot de la variable `Medida_Refino`.

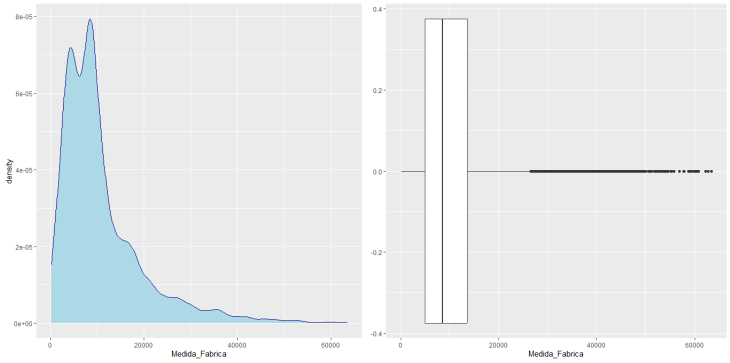


Figura A.10: Función de densidad y boxplot de la variable `Medida_Fabrica`.

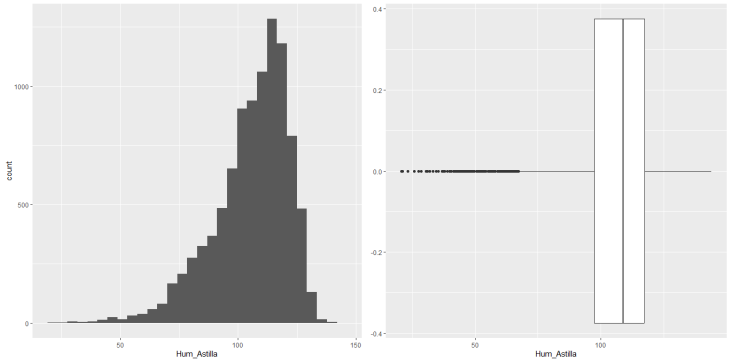
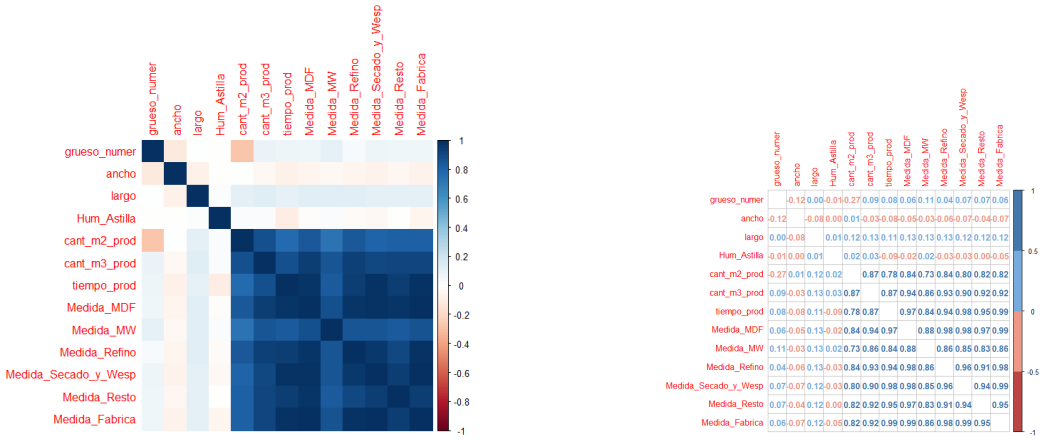


Figura A.11: Histograma y boxplot de la variable `Hum_Astilla`.



(a) Intensidad según el color

(b) Coeficientes de correlación

Figura A.12: Matriz de correlaciones entre las variables de la Línea 232

	Min.	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
grueso_numer	6.00	12.00	16.00	16.76	19.00	44.00
ancho	1820	2100	2220	2256	2440	2530
largo	2700	3660	4880	4621	5500	5740
cant_m2_prod	6.688	1097.085	1829.268	2500.306	3116.062	26818.674
cant_m3_prod	0.075	17.146	31.106	37.998	44.780	282.013
tiempo_prod	8	1609	2803	3646	4539	19979
Medida_MDF	67.47	4312.65	7564.80	9492.84	11715.27	57794.35
Medida_MW	0.9	220.0	388.4	487.8	592.5	17478.4
Medida_Refino	0.51	1653.89	2872.51	3649.43	4515.88	23730.30
Medida_Secado_y_Wesp	19.3	691.8	1205.2	1548.9	1938.6	9912.0
Medida_Resto	1.78	1682.79	2973.09	3817.31	4692.64	26310.40
Medida_Fabrica	141.5	4889.5	8530.0	10860.0	13553.5	63444.0
Hum_Astilla	20.88	97.34	108.78	105.65	117.31	143.80

Cuadro A.1: Resumen numérico de las variables de la Línea 232.

Apéndice B

Análisis descriptivo de las variables de la Línea 35

En lo referente a la Línea 35, recordemos que en la Subsección 2.2.2 habíamos creado un nuevo dataset con las variables de interés y la rejilla de los tres turnos diarios. Estas variables eran `fecha_produccion`, `inicio`, `fin`, `inicio_segundos`, `fin_segundos`, `m_lijado`, `m2_lijado`, `m3_lijado`, `consumo`.

B.1. Análisis descriptivo

Se muestran a continuación unos descriptivos de las variables utilizadas para la construcción de los modelos en la Línea 35: `m_lijado` y `consumo`.

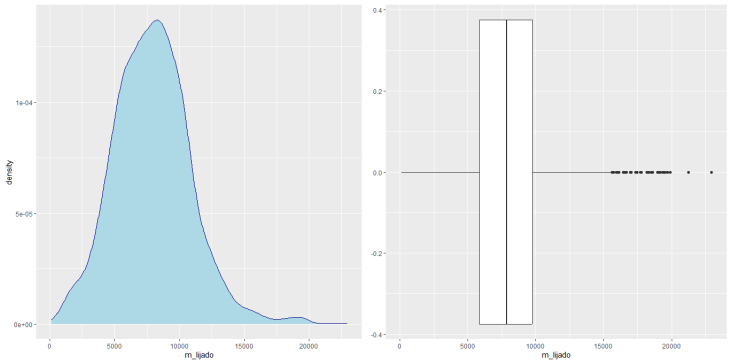


Figura B.1: Histograma y boxplot de la variable `m_lijado`.

Así mismo, en la figura B.3 podemos ver la densidad de metros lineales producidos y el consumo de kWh según el turno: en rojo, los turnos de noche; en verde, los turnos de mañana; en azul, los turnos de tarde.

Para finalizar, recopilamos en el Cuadro B.1 un pequeño resumen número de las variables empleadas para la construcción de modelos de la Línea 35.

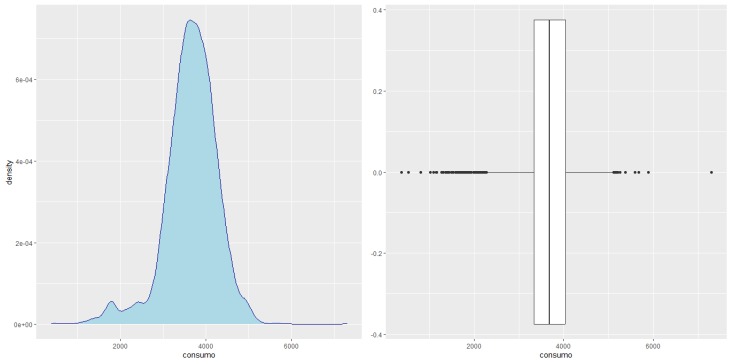


Figura B.2: Histograma y boxplot de la variable `consumo`.

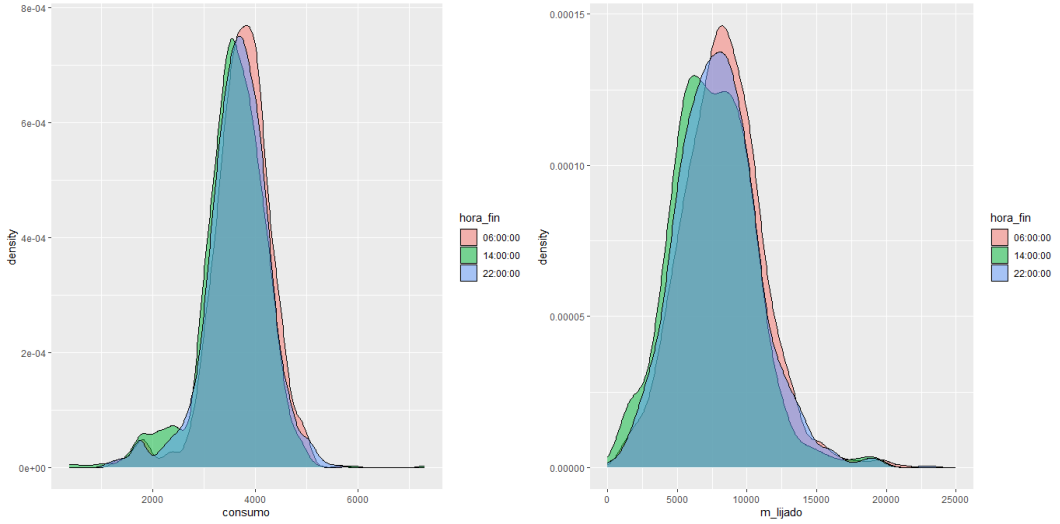


Figura B.3: Densidades de las variables `m_lijado` y `consumo` según el turno de producción.

	Min.	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
consumo	378.3	3336.6	3688.7	3649.4	4039.1	7304.9
m_lijado	116.7	5849.2	7881.3	7925.7	9737.3	22937.0

Cuadro B.1: Resumen numérico de las variables de la Línea 35.