Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Monitorización de indicadores clave energéticos en una unidad de destilación de una refinería de petróleo mediante minería de datos

Trabajo Fin de Máster

Máster Universitario en Gestión de la Información

Autor: Pedro Miguel Villalba Torán

Tutores: Cèsar Ferri Ramírez

Alberto J. Ferrer Riquelme

Curso académico 2015-2016

Resumen

El principal objetivo de la monitorización de indicadores clave energéticos (o KEI, del inglés, *Key Energy Indicator*) es el de proporcionar una mejor comprensión de cómo se está utilizando la energía para ayudar a la toma de decisiones en la optimización de costes. En esta tesina de máster se compararán diferentes técnicas de minería de datos (métodos estadísticos de análisis multivariante y métodos de aprendizaje automático) para la monitorización de un indicador energético del consumo de fuelóleo en el horno de una unidad de destilación de una refinería de petróleo de la empresa Repsol, S.A. Se establecerá una metodología lo más automatizada posible, con la intervención mínima del usuario. La metodología consta de dos fases. La primera fase (*offline*) se basa en la construcción de un modelo a partir de datos históricos, mediante el uso de diferentes técnicas de minería de datos. La segunda fase (*online*) consiste en la explotación del modelo para predecir el KEI a partir de las variables más importantes (o *drivers*) seleccionadas durante la fase de construcción del modelo. Además, el sistema de monitorización incorporará un diagnóstico de fallos, para lo cual se establecerán unos límites de control, fuera de los cuales se considerará que el consumo energético es diferente al esperado. En tal caso, se dispondrá de un mecanismo que permitirá conocer qué variables del proceso son responsables de la situación, lo que será de gran ayuda para un mejor conocimiento del proceso y para la toma de decisiones que permitan una operación más eficiente del proceso.

Palabras clave: aprendizaje automático, métodos estadísticos multivariantes, monitorización de procesos, indicador clave de energía, refinería de petróleo

Abstract

The main objective of KEI (*Key Energy Indicators*) monitoring is to get a better understanding about the use of energy to help in the decision making process during cost optimization. In this master thesis several data mining tools (multivariate statistical methods and machine learning methods) will be compared to monitorize fuel oil consumption in a crude distillation unit of a petroleum refinery property of Repsol, S.A. A fully automated methodology with minimal user intervention will be developed. We employ a two-phase methodology. In phase I (*offline*), a model is trained from historical data using several data mining tools. The second phase (*online*) is based on model exploitation to predict KEI from the most important variables (*drivers*) selected during phase I. The monitoring system will include a failure diagnosis tool. An upper control limits will be set in order to detect abnormal situations. The system will include a tool to determine which process variables are responsible of such abnormal situations. This will be very helpful for better process understanding and for decision making for a more efficient process performance.

Keywords : machine learning, multivariate statistical methods, process monitoring, key energy indicator, petroleum refining

Tabla de contenidos

1 Introducción

Las industrias de proceso se encuentran altamente automatizadas y disponen de una serie de medidores y sistemas de control que proporcionan las calidades deseadas de los productos a la vez que aseguran el cumplimiento de las normas de seguridad y medio ambiente. Los sistemas de monitorización se encuentran por encima de estos sistemas de control y permiten realizar un seguimiento de ciertas variables o indicadores clave del proceso que permiten evaluar la eficiencia del proceso y determinar las causas de posibles situaciones anómalas.

En las plantas industriales, los sistemas de información recogen cientos e incluso miles de mediciones cada segundo, por lo que el volumen de datos es muy elevado. En este sentido, las técnicas de minería de datos son especialmente útiles para desarrollar modelos que permitan monitorizar los indicadores clave del proceso.

1.1 Sistemas de control de procesos

El objetivo principal del control de procesos es mantener el sistema en las condiciones de operación deseadas, de manera eficiente y segura, a la vez que se satisface la calidad de los productos y los requisitos medioambientales (Seborg et al. 2004).

En los últimos años, el aumento de la competitividad, las normativas medioambientales y de seguridad cada vez más restrictivas y los cambios bruscos de la economía han hecho cada vez más difícil mantener las especificaciones de calidad de los productos de forma rentable. En el caso de las refinerías de petróleo, este problema se agrava más, debido a la creciente complejidad e integración de sus procesos. Por ello es imprescindible disponer de sistemas de control computerizados.

En la Figura 1.1, las actividades del control de procesos se organizan de manera que las funciones imprescindibles se encuentran en los niveles más bajos, mientras que las funciones deseables, aunque opcionales, están en los niveles superiores. La escala de tiempo para cada actividad se encuentra en el margen derecho de la figura. Puede comprobarse que la frecuencia de ejecución es mucho menor para las funciones de los niveles superiores.

Las actividades del Nivel 1 corresponden a aparatos de medición (sensores) y equipo de actuación (como, por ejemplo, válvulas de control) y se utilizan para medir las variables de proceso e implementar las acciones de control. Estas funciones son básicas para cualquier sistema de control de procesos.

Las funciones del Nivel 2 (seguridad y protección de equipos y medio ambiente) aseguran que el proceso opere de manera segura para el personal de la planta y los equipos y se cumpla la normativa medioambiental. Entre estas funciones se encuentran, por ejemplo, la gestión de alarmas durante situaciones anómalas o el control de sistemas de seguridad durante paradas de emergencia.

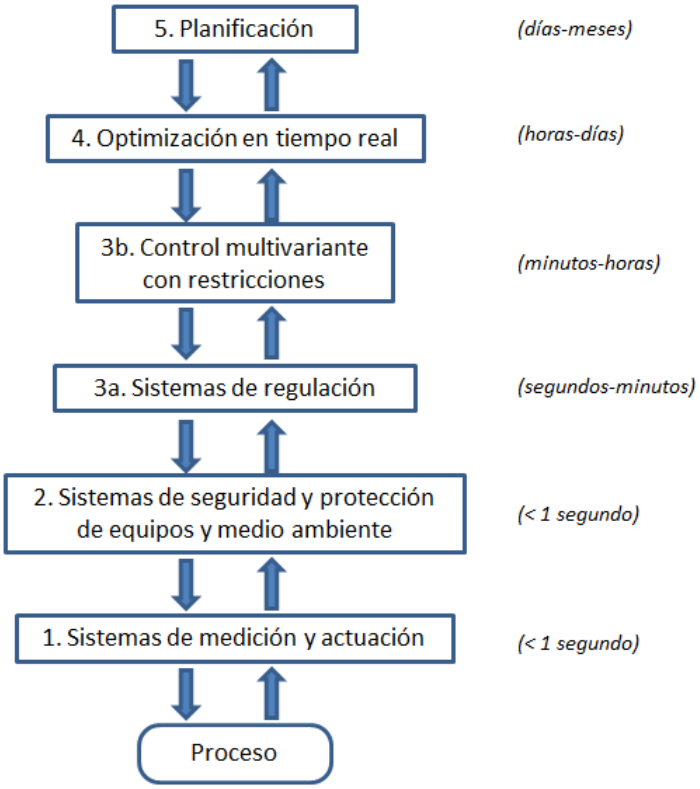


Figura 1.1.- Jerarquía de las actividades de control de procesos

Los sistemas de regulación (Nivel 3a) aplican diversas técnicas de control para mantener las variables de proceso (temperaturas, caudales, presiones o composiciones) cerca de sus puntos de operación, de manera que se cumplan las especificaciones de calidad de los productos y las normas medioambientales y se asegure la integridad de la planta.

Las técnicas de control estándar del Nivel 3a no son adecuadas para controlar problemas complejos en los que existen, por ejemplo, fuertes interacciones entre las variables de proceso. Para estas otras situaciones son necesarias técnicas más avanzadas de control de procesos (Nivel 3b), como el control predictivo basado en modelos o MPC (del ingés, *Model Predictive Control*).

Las condiciones de operación óptimas de la planta son parte del proceso de diseño de la misma, pero, al operar la planta, las condiciones óptimas de operación pueden cambiar frecuentemente debido a cambios en la disponibilidad de los equipos, perturbaciones del proceso o condiciones económicas. Por ello, es necesario recalcular las condiciones óptimas de operación ante las nuevas situaciones. Esto se lleva a cabo en el Nivel 4 (optimización en tiempo real), que calcula las nuevas condiciones óptimas de operación a partir de un modelo de los procesos de la planta y de datos económicos (como el coste de los materiales o el precio de los productos). Los objetivos típicos de esta optimización son minimizar los costes de operación o maximizar los beneficios.

El nivel más alto en la jerarquía de control de procesos de la planta (Planificación, Nivel 5) hace referencia a la planificación global de la planta y se encarga de calcular los niveles de producción tanto de los productos finales como de los productos intermedios en base a la disponibilidad de los equipos, de la capacidad de almacenamiento y de la previsión de ventas.

Son actividades que forman parte de la estrategia de la empresa y que suponen problemas de optimización complejos que incluyen cuestiones de ingeniería y de negocio. Por ello son decisiones que se toman a medio-largo plazo (días, semanas e incluso meses).

1.1.1 Industria 4.0

La primera revolución industrial tuvo lugar a finales del siglo XVIII y utilizaba máquinas de vapor para los procesos productivos.

La segunda apareció a principios del siglo XX con el uso de la energía eléctrica en las cadenas de producción.

La tercera supuso el cambio de los componentes analógicos y mecánicos por los componentes digitales y electrónicos, a partir de los años 70. La industria química ha sido pionera en el uso de ordenadores para el control de procesos. En 1980 Cutler y Ramaker presentaron una primera versión del *Dynamic Matrix Control* (DMC) que supuso las bases para el control avanzado de procesos y el uso de ordenadores para optimizar procesos en tiempo real.

En 2011 apareció el término "Industria 4.0" para hacer referencia a la cuarta revolución industrial, que consiste en una fabricación altamente informatizada en la que todos los procesos están interconectados entre sí. Este nuevo concepto ha dado lugar a lo que se conoce como *smart factories*, en las que los sistemas de almacenamiento y los de producción son capaces de realizar tareas complejas intercambiando información entre ellos sin necesidad de la intervención humana.

La Industria 4.0 se basa en gran parte en el IoT (del inglés, *Internet Of Things* o Internet de las Cosas) que hace referencia al uso de componentes que poseen una tecnología que les permite comunicarse con los Sistemas de Información (SI) y ser detectados por sensores (The Economist 2015). Este tipo de sistemas generan una gran cantidad de datos que deben ser analizados en tiempo real (concepto conocido como *Big Data*) para aumentar la producción y optimizar costes. Por ello, es necesario disponer de una infraestructura que permita analizar este gran volumen de información. En este sentido, muchas empresas están utilizando *cloud computing* (computación en la nube) para desplegar sistemas avanzados de computación paralela y distribuida, especialmente útiles para la explotación de grandes volúmenes de datos. La tecnología *cloud computing* permite contratar servicios a terceros para implementar proyectos de *Big Data* de manera virtual, sin necesidad de poseer un centro de datos físico propio.

1.2 Sistemas de monitorización de energía

Los sistemas de monitorización se encuentran por encima de los sistemas de control de procesos vistos anteriormente. Concretamente, el objetivo de los sistemas de monitorización de energía es el de proporcionar un mejor conocimiento sobre el uso de la misma y ayudar en el proceso de toma de decisiones para mejorar la eficiencia energética. Estos sistemas calculan

el consumo actual de energía y lo comparan con el valor estimado, permitiendo, además, conocer las causas de posibles comportamientos anómalos (Abonyi et al. 2014).

Los métodos para calcular el consumo esperado pueden estar basados en primeros principios o en datos. Los primeros utilizan ecuaciones termodinámicas, derivadas de los balances de energía y de materia de la planta o de posibles reacciones químicas, entre otros. Es decir, intentan modelizar el proceso mediante ecuaciones matemáticas.

Los métodos basados en datos históricos del proceso pueden ser más o menos complejos. Entre las implementaciones más sencillas se encuentran los esquemas basados en periodos anteriores. Por ejemplo, es habitual la comparación del mes actual con el mismo mes del año pasado. Esta aproximación tiene numerosos inconvenientes. El principal problema es que las condiciones de proceso entre un mes y el otro pueden haber cambiado. Además, puede haberse dado condiciones de proceso anómalas que hayan disparado los consumos energéticos, con lo que deberían descartarse dichas situaciones a la hora de realizar la comparación.

Existen otros sistemas que trabajan con intervalos menores y que obtienen un consumo promedio (cada hora o cada media hora, por ejemplo) derivado de los últimos días de proceso. Aun así, el problema es el mismo que en el caso anterior, dado que pueden haberse contabilizado situaciones anómalas, que requerirán de la supervisión de algún experto para que no sean tenidas en cuenta a la hora de obtener el modelo.

Las anteriores propuestas son demasiado simplistas. Una mejor solución son los modelos basados en la actividad, sobre todo cuando se conocen las causas de los cambios en el consumo energético como, por ejemplo, cambios en la producción. En este caso los modelos se obtienen en base a los factores clave o *drivers* del proceso. Una vez se dispone de un modelo que es capaz de estimar el consumo energético de un proceso a partir de sus factores clave, es posible simular diferentes escenarios que permitan estudiar proyectos para mejorar la eficiencia energética.

En el presente trabajo se desarrollarán modelos derivados de datos históricos de la planta mediante el uso de técnicas de minería de datos (métodos estadísticos de análisis multivariante y métodos de aprendizaje automático). Estas técnicas permiten identificar situaciones anómalas, que serán descartadas a la hora de crear los modelos definitivos. Estos modelos contienen la información histórica de la planta y son capaces de calcular el consumo energético esperado para las condiciones de proceso actuales. Además, como se verá más adelante, permiten conocer las variables más importantes a la hora de determinar el consumo (los factores clave o *drivers*, comentados anteriormente), así como identificar situaciones anómalas y conocer sus causas.

1.2.1 Indicadores clave de energía

Los indicadores clave de energía o KEI (*Key Energy Indicators*) son una parte esencial de un sistema de monitorización de energía. En muchas ocasiones, a pesar de que la planta puede estar aparentemente bajo control, el funcionamiento puede mejorarse desde el punto de vista

de la eficiencia energética, pero los ingenieros y operarios de planta no son conscientes de tal situación. Los KEI dan solución a este problema ya que permiten caracterizar el consumo energético de una unidad de proceso en base a un número reducido de variables de operación o incluso una única variable. En cualquier caso, la variable o variables utilizadas para el cálculo del KEI deben tener un efecto significativo en el rendimiento energético del proceso. Por ello, a la hora de definir un KEI, es de vital importancia entender las interacciones entre las diferentes variables del proceso y el consumo energético (Zhu 2014).

1.3 Detección y diagnóstico de fallos

De cara a la monitorización, no sólo es importante predecir el consumo de energía esperado, sino establecer unos límites de control que permitan saber si la diferencia entre el valor observado y el predicho es estadísticamente significativa o no. En caso de detectar una situación anómala (un valor observado fuera de los límites de control), es interesante determinar las causas de tal situación. Este concepto se conoce con el nombre de detección y diagnóstico de fallos.

Un fallo puede definirse como un comportamiento anómalo que provoca que los procesos se desvíen inaceptablemente de sus condiciones de operación normales. En las plantas de proceso como las refinerías de petróleo, los fallos pueden categorizarse según diferentes causas (Aldrich & Auret 2013, p.6):

- Fallos de los instrumentos de medición (sensores) que pueden provocan errores en los cálculos de las acciones de control.
- Fallos en los actuadores que provocan errores en la operación del proceso.
- Fallos en los equipos.
- Fallos del personal.

Estos fallos pueden aparecer bruscamente, como un fallo en una bomba, o lentamente, como fallos debidos al ensuciamiento o corrosión de los equipos.

El principal objetivo del diagnóstico de fallos es la rápida detección de estas situaciones anómalas, la identificación de sus causas y, a ser posible, su corrección con la menor repercusión posible en el proceso. Como se ha comentado en el apartado anterior, esto se consigue con la obtención de un modelo que representa el comportamiento deseado del proceso. La detección de fallos se basa en monitorizar la desviación entre el proceso actual y el predicho por el modelo y compararla con unos límites preestablecidos. Una vez se detecta un fallo, para establecer las causas generalmente se usan gráficos de contribución, en los que se representan las desviaciones entre los valores observados para las variables predictoras y los valores predichos mediante el modelo. La corrección del problema dependerá de la experiencia de los ingenieros y no suele estar automatizada, como ocurre con la detección.

1.4 Minería de datos

La minería de datos (*data mining*) es un campo de las ciencias de la computación que trata de identificar patrones o relaciones en grandes conjuntos de datos mediante el uso de técnicas estadísticas y de aprendizaje automático (*machine learning*).

El origen de la estadística se sitúa en el siglo XVIII, cuando se desarrolló un sistema para la recolección de datos demográficos y económicos por parte de los gobiernos. En el siglo XIX, la recolección de datos se intensificó y el concepto de estadística se amplió para definir la recolección, resumen y análisis de los datos.

El término "aprendizaje automático" es más reciente y nace como un subcampo dentro de la inteligencia artificial para el reconocimiento de patrones (*pattern recognition*) y el desarrollo de algoritmos que "aprenden" a partir de unos datos y pueden realizar predicciones a partir de estos datos.

Es difícil establecer diferencias entre todas estas disciplinas ya que, en muchas ocasiones, se solapan y, básicamente, lo que cambian es la terminología (Sharp Sight Labs 2016). Así, por ejemplo, en estadística se habla de "ajustar" un modelo, mientras que en aprendizaje automático se usa el término "aprender".

Volviendo al objetivo de la tesina, para poder desarrollar sistemas de detección y diagnóstico de fallos es necesario disponer de modelos. En el presente trabajo, los modelos se obtendrán a partir de datos históricos mediante el uso de técnicas de minería de datos. Estas técnicas proporcionan flexibilidad respecto a cómo pueden estimarse los modelos. En el caso de que se dispongan de los valores de la variable respuesta se habla de aprendizaje supervisado. Si la variable respuesta es categórica se habla de clasificación y si es continua, de regresión. La mayoría de técnicas de clasificación pueden adaptarse para abordar problemas de regresión. Es el caso de las tres técnicas que se usarán en este trabajo: *Partial Least Squares* (PLS), *Random Forest* (RF) y *Support Vector Machine* (SVM). Todas ellas pueden utilizarse para tratar tanto problemas de clasificación como de regresión.

Cuando no existe una variable respuesta, el objetivo, entonces, es obtener patrones a partir de los datos que permitan la creación de subconjuntos de datos con un comportamiento similar. Este segundo caso se conoce como aprendizaje no supervisado e incluye técnicas como *Principal Component Analysis* (PCA), *clustering* y *k-medias*, entre otras.

1.5 Objetivos

El principal objetivo del presente trabajo es la implementación de un sistema para la monitorización de Indicadores Clave de Energía o KEI (*Key Energy* Indicator) para ayudar a los ingenieros de planta a optimizar el consumo de energía. El estudio se realizará sobre el consumo específico de combustible en el horno de calentamiento de crudo de la unidad de destilación atmosférica de una refinería de petróleo de la empresa Repsol, S.A.

Para ello se desarrollará una metodología para la creación de modelos mediante minería de datos a partir de datos históricos de proceso de la planta, que permitirán monitorizar el consumo de combustible en el horno, siguiendo los siguientes objetivos específicos:

1. Limpiar los datos no válidos e imputar los posibles datos faltantes.
2. Obtener las condiciones normales de operación (NOC, del inglés *Normal Operating Conditions*) a partir de los datos históricos, descartando situaciones anómalas.
3. Desarrollar modelos a partir de los datos NOC que permitan monitorizar el KEI y establecer límites de control para detectar situaciones anómalas.
4. Implementar técnicas de detección de fallos para determinar las causas de las situaciones anómalas detectadas. Esta información será de gran utilidad para los ingenieros para la toma de decisiones en el mantenimiento preventivo de la planta.
5. Crear un cuadro de mandos con la información necesaria para poder monitorizar el KEI y diagnosticar posibles situaciones anómalas de consumo energético.

2 Refinerías de petróleo

Las refinerías de petróleo modernas son sistemas altamente complejos e integrados que permiten separar y transformar el crudo en una gran variedad de productos, como gasolinas, gasóleos, queroseno, lubricantes, ceras o alquitranes, entre otros (Aitani 2004).

El primer paso es separar el crudo en fracciones mediante destilación. A partir de aquí, se han desarrollado diversos sistemas para convertir estas fracciones destiladas del petróleo en una gran variedad de productos.

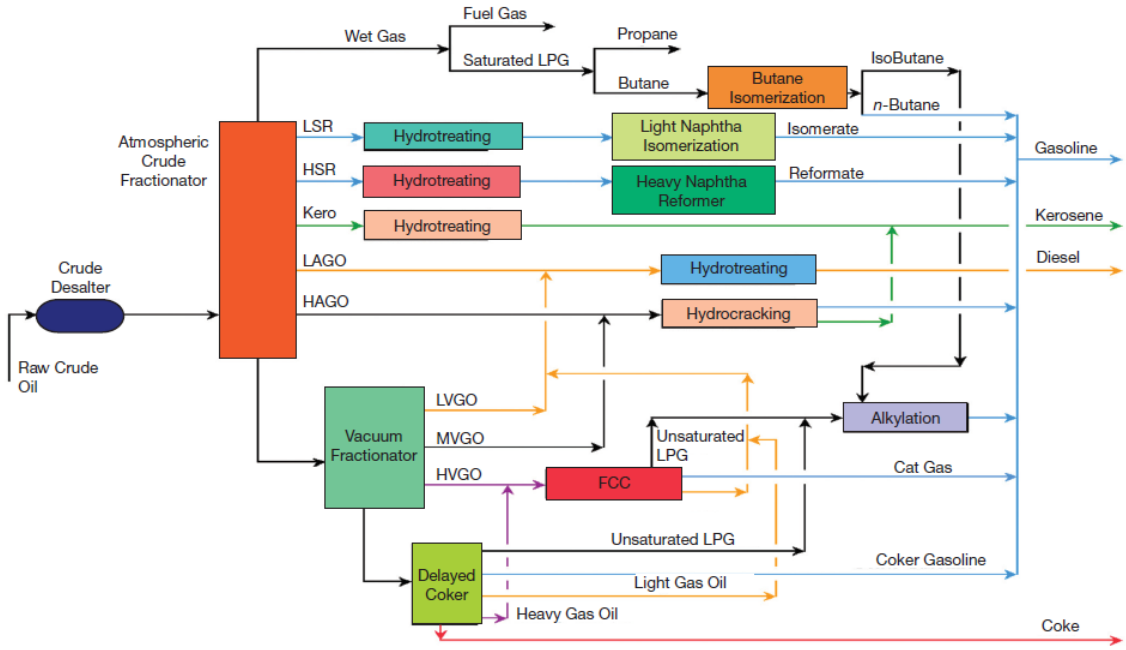


Figura 2.1.- Esquema simplificado de los procesos de refino y flujo de productos (LSR=nafta ligera; HSR=nafta pesada; LAGO=gasóleo atmosférico ligero; HAGO=gasóleo atmosférico pesado; LVGO=gasóleo de vacío ligero; MVGO=gasóleo de vacío medio; HVGO=gasóleo de vacío pesado) (Olsen 2014)

La Figura 2.1 presenta un diagrama simplificado de una refinería (Olsen 2014). Una vez desalado el crudo, éste es precalentado y entra en la columna de destilación atmosférica. En la destilación, los productos más ligeros salen por cabezas (la zona alta de la columna de destilación), mientras que los más pesados se obtienen por fondos. A lo largo de la columna existen diferentes corrientes laterales que contienen diversas fracciones del petróleo (como, por ejemplo, queroseno y naftas). Los fondos de la destilación atmosférica contienen las fracciones más pesadas del petróleo, que son destiladas al vacío. La destilación al vacío permite trabajar con temperaturas más altas y conseguir la separación de los productos más pesados (como los alquitranes).

El siguiente paso es convertir estas fracciones destiladas del petróleo en productos de mayor valor añadido. Así por ejemplo, el reformado catalítico convierte las naftas pesadas en gasolina, mientras que el craqueo catalítico en lecho fluidizado o FCC (*Fluid Catalytic Cracking*) transforma las fracciones más pesadas resultantes de la destilación al vacío. Otros procesos

más recientes, como el hidrocraqueo, están orientados a producir productos más ligeros a partir de los productos de fondos de la destilación.

Todos los productos deben ser tratados para aumentar su calidad y disminuir su efecto contaminante. Así, por el ejemplo, la unidad de hidrotratamiento se encarga de reducir el contenido en azufre, que es uno de los principales contaminantes de los combustibles derivados del petróleo.

Finalmente, existen una serie de procesos auxiliares que se encargan de generar los productos necesarios para los procesos de conversión (como el hidrógeno) o la energía y el vapor necesarios para calentar los productos.

El presente trabajo se centrará en la unidad de destilación atmosférica y, concretamente, estudiará el consumo específico de combustible en el horno de precalentamiento de crudo, previo a la destilación atmosférica.

2.1 Unidad de destilación atmosférica

El esquema típico de una unidad de destilación atmosférica puede observarse en la Figura 2.2 (Jones & Pujadó 2006).

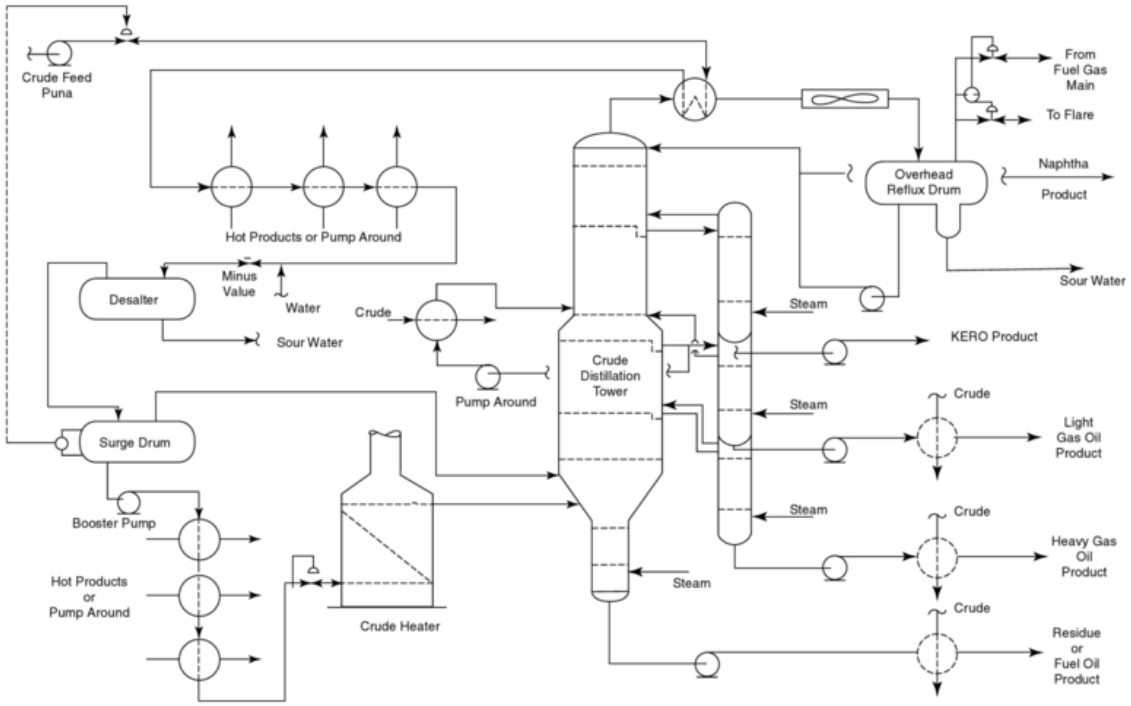


Figura 2.2.- Diagrama típico de una unidad de destilación atmosférica de crudo (Jones & Pujadó 2006).

Como se ha comentado, el objetivo principal de la destilación atmosférica es el de separar el petróleo en fracciones o cortes, en base a los rangos de puntos de ebullición de sus componentes (también conocidos como puntos de corte). Las fracciones más ligeras, como los gases licuados del petróleo, se obtienen por cabezas de la columna, mientras que los componentes más pesados (como el gasóleo) se obtienen por fondos. En las corrientes

laterales de la columna se obtienen productos con diferentes puntos de corte (queroseno, naftas ligeras, naftas pesadas, …). Cada una de estas corrientes laterales pasa a unas columnas de separación que reducen la presión parcial de la mezcla de hidrocarburos, restableciendo su equilibrio líquido-vapor. Estas corrientes laterales se utilizan para precalentar el crudo antes de entrar al horno. Esto es importante, dado que da idea del grado de integración existente entre los diferentes componentes de la unidad. El horno es el encargado de calentar el crudo antes de entrar en la columna de destilación, consumiendo gran parte de la energía del proceso. Como se verá más adelante, el indicador clave energético que se estudiará en el presente trabajo está basado en el consumo de fuel en el horno de la unidad de destilación de crudo.

2.2 La energía en las refinerías de petróleo

Como puede deducirse a partir de lo comentado anteriormente, las refinerías de petróleo son grandes consumidoras de energía. Al igual que muchas otras industrias, las refinerías se encuentran en un mercado global cada vez más competitivo, por lo que las empresas buscan reducir sus costes de producción sin afectar al rendimiento de sus procesos o la calidad de sus productos. Una de las formas de conseguirlo es mediante la reducción de la energía consumida en los procesos. Las inversiones realizadas en este sentido no repercuten únicamente en los costes de producción, sino que también reducen las emisiones de las plantas (Worrell et al. 2015).

Las tecnologías encargadas de eliminar los contaminantes al final de proceso (*end-of-pipe solutions*) resultan caras y, a veces, ineficientes. Sin embargo, la inversión en la mejora de la eficiencia energética a lo largo de las diferentes etapas del proceso permite reducir costes y emisiones. Esta idea forma parte de una estrategia medioambiental hacia lo que se conoce como “triple bottom line”, concepto creado por el *World Business Council on Sustainable Development* (WBCSD) y que engloba a la sociedad, la economía y el medio ambiente. Estos tres aspectos están interconectados entre sí, dado que la sociedad depende de la economía y ésta de la salud del ecosistema global, el cual se basa en los aspectos sociales, económicos y medioambientales de un negocio. Por ello, de cara a un desarrollo sostenible, la mayoría de empresas actuales deberían incluir las inversiones en eficiencia energética como parte de su estrategia de negocio.

3 Minería de datos

3.1 Introducción

El objetivo de la minería de datos (*data mining*) es el de identificar patrones o relaciones en grandes conjuntos de datos mediante el uso de técnicas estadísticas y de aprendizaje automático (*machine learning*).

En el campo del aprendizaje automático suele distinguirse entre aprendizaje supervisado y aprendizaje no supervisado.

Cuando se trata de predecir el valor de una o más variables (denominadas variables respuesta) a partir de otras variables (denominadas variables predictoras) se habla de aprendizaje supervisado. El aprendizaje supervisado puede dividirse en dos grandes grupos: clasificación y regresión, dependiendo del tipo de variable respuesta.

En la clasificación, la variable respuesta es de tipo categórico (o lógico). Una variable categórica es aquella que se mide mediante atributos o categorías. Algunos de los algoritmos que pueden incluirse en esta categoría son las Máquinas de Vector Soporte o los Vecinos más Próximos.

En los problemas de regresión, la variable respuesta es cuantitativa, es decir, que puede medirse mediante valores continuos. En esta categoría estarían técnicas como la Regresión Lineal o el Partial Least Squares. La mayoría de algoritmos que permiten tratar problemas de clasificación pueden adaptarse para resolver también problemas de clasificación. Así, por ejemplo, existen Máquinas de Vector Soporte de Regresión.

En el aprendizaje no supervisado no existe una variable respuesta y el objetivo es describir las asociaciones y patrones entre un determinado conjunto de variables. Entre este tipo de técnicas están el Principal Component Analysis o las k-Medias.

En el presente trabajo, se utilizarán técnicas de regresión para predecir un indicador clave energético a partir de un conjunto de variables de proceso de una unidad de destilación. También se utilizarán técnicas de clasificación en modo no supervisado para identificar posibles observaciones anómalas.

Por otro lado, una de las principales críticas que puede aplicarse a algunas técnicas de aprendizaje automático, como las Máquinas de Vector Soporte o las Redes Neuronales, es el hecho de comportarse como cajas negras. Este concepto expresa la imposibilidad de interpretar los resultados de estas técnicas, más allá de las predicciones que puedan ofrecer. Como se ha comentado, en el caso de la monitorización de procesos, no sólo interesa obtener un buen modelo que permita comparar el valor real con el esperado para saber si el proceso se comporta según las condiciones normales de operación, sino que interesa saber cuáles son las causas de tal comportamiento.

En los siguientes apartados se comentarán los métodos de minería de datos que han sido seleccionados para realizar el presente estudio, haciendo hincapié en sus funcionalidades de cara a la monitorización de procesos y diagnóstico de fallos.

Por una lado se utilizará el PLS (ver apartado 3.2), que es un método estadístico multivariante que permite no sólo detectar situaciones anómalas en un proceso, sino también establecer sus causas.

Por otro lado, se utilizarán dos métodos de aprendizaje automático: Random Forest (ver apartado 3.4) y Máquinas de Vector Soporte (ver apartado 3.5).

3.2 Lenguajes de programación

Existen diversos lenguajes científicos de programación para desarrollar proyectos de minería de datos. En esta tesina, los cálculos se realizarán mediante dos de los lenguajes de programación más utilizados, como son R[1] y Python[2].

R es un lenguaje de código abierto para realizar cálculos estadísticos que proporciona una gran variedad de técnicas de minería de datos, así como amplias capacidades gráficas. Incluye, entre otras, las siguientes características:

- Gestión de datos eficiente.
- Operaciones matriciales.
- Gran colección de herramientas para el análisis de datos.
- Lenguaje de programación que permite trabajar con condiciones, bucles, funciones y clases definidas por el usuario.

La mayor parte del sistema está escrito en el propio lenguaje R, pero para tareas con gran carga computacional puede desarrollarse código en Fortran, C o C++, que puede ser llamado desde R en tiempo de ejecución.

R puede extenderse fácilmente a través de sus librerías, disponibles en un repositorio denominado CRAN[3], que cubre innumerables librerías con funciones matemáticas y estadísticas.

Como entorno de desarrollo se ha utilizado RStudio[4], también de código abierto, que incluye una consola y un editor con resaltado de sintaxis que permite ejecución directa de código, así como herramientas gráficas, historial de código, depuración y gestión del espacio de trabajo.

La comunidad que utiliza R es muy activa y numerosa y dispone de varios foros donde se resuelven todo tipo de dudas.

[1] "The R Project for Statistical Computing" https://www.r-project.org/
[2] "Python Software Foundation" https://www.python.org/
[3] "The Comprehensive R Archive Network" https://cran.r-project.org/
[4] "RStudio – Open Souce and enterprise-ready professional software for R" https://www.rstudio.com/

Por otro lado, Python es un lenguaje de programación en código abierto cuyo principal objetivo es la legibilidad y simplicidad de código. Se trata de un lenguaje interpretado, existiendo intérpretes para la mayoría de Sistemas Operativos. Es un lenguaje de alto nivel que permite programación orientada a objetos, entre otras funcionalidades.

Al igual que R, Python dispone de una amplia comunidad de desarrolladores que han implementado gran cantidad de librerías para temas muy diversos. En cuanto a la interfaz utilizada, se ha optado por Spyder[5], que posee grandes similitudes con el entorno de desarrollo de Matlab.

3.3 Partial Least Squares

3.3.1 Introducción

Partial Least Squares (PLS) es un método desarrollado en 1975 por Herman Wold y permite relacionar dos matrices, **X** e **Y**, mediante un modelo lineal multivariante (Wold et al. 2001). A diferencia de la Regresión Lineal Múltiple, permite analizar datos fuertemente correlacionados, con ruido y numerosas variables predictoras, a la vez que permite, simultáneamente, modelizar varias variables respuesta. Esta situación es muy frecuente en la industria química y, por supuesto, en las refinerías de petróleo.

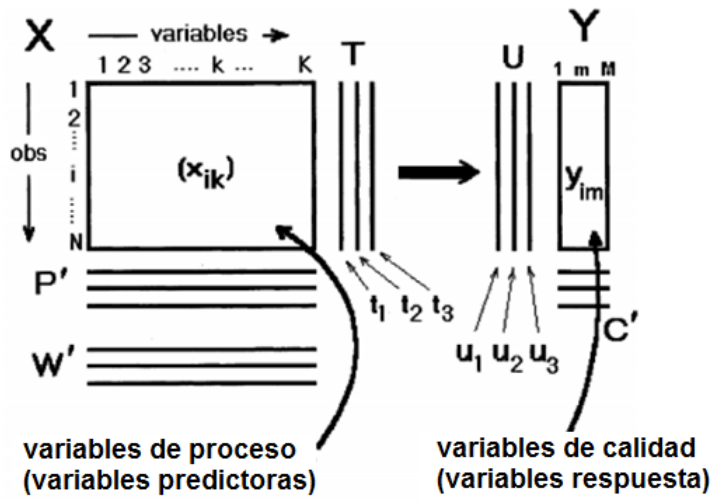


Figura 3.1.- Esquema del modelo PLS (Wold et al. 2001)

En la Figura 3.1 puede verse un esquema del modelo PLS. Los datos pueden representarse mediante dos conjuntos **X** e **Y**. Como norma general, las variables **X** suelen ser variables de proceso a partir de las cuales se desean predecir una o más variables de calidad **Y**. Sin entrar en consideraciones matemáticas, el PLS encuentra unas pocas variables (denominadas variables latentes o *scores*) que permiten predecir tanto la matriz de variables respuesta **Y** como modelizar la propia matriz **X**. Es conveniente recalcar que el número de variables

[5] "The Scientific PYthon Development EnviRonment": https://pythonhosted.org/spyder/

latentes encontradas (A) suele ser muy inferior al número de variables X originales, por lo que, en ocasiones, el PLS es reconocido como una técnica de reducción de la dimensionalidad. Los scores de **X** (**T**) son estimados como combinación lineal de las variables originales mediante una serie de coeficientes o *weights* (**W**).

Por otro lado, **Y** también posee sus scores (**U**), que multiplicados por sus correspondientes pesos (**C**) son unos buenos estimadores de la propia matriz respuesta **Y**.

Finalmente, los coeficientes de la regresión mediante PLS (**B**), que relacionan **X** con **Y**, se obtienen por combinación lineal de los anteriores coeficientes: **B = W * C'**

El estadístico R^2 permite estimar la bondad del ajuste obtenido y se calcula como:

$$R_A^2 = 1 - \frac{SCR_A}{SC_Y}$$

Donde,

R_A^2 es la R^2 para el modelo PLS obtenido con A componentes principales

SCR_A es la Suma de Cuadrados de los residuos de **Y** tras ajustar un modelo PLS con A componentes principales

SC_Y es la Suma de Cuadrados de la matriz de datos **Y**

La bondad de predicción del PLS se determina mediante validación cruzada (ver apartado 4.1.9.1). El estadístico Q^2 permite estimar la bondad de la predicción:

$$Q_A^2 = 1 - \frac{PRESS}{SC_Y}$$

Donde,

Q_A^2 es la Q^2 para el modelo PLS obtenido con A componentes principales

PRESS (*PRediction Error Sum of Squares*) es la Suma de Cuadrados de los residuos obtenidos mediante validación cruzada

SC_Y es la Suma de Cuadrados de la matriz de datos **Y**

Existen muchos criterios para conocer el número de componentes (A) adecuado para un modelo PLS. En este trabajo se utilizará un criterio basado en el estadístico Q^2 (obtenido a través de un proceso de validación cruzada). El método de construcción del modelo PLS es iterativo: empieza calculando una componente y posteriormente añade más componentes sucesivamente. Para conocer el número de componentes adecuado, en cada paso, se hace una doble comprobación:

1. Se comprueba si el valor del estadístico Q^2 aumenta en un 1% o más al añadir la nueva componente.
2. Se comprueba si el valor del estadístico Q^2 aumenta en un 5% o más para alguna de las variables.

Si se cumple alguna de las anteriores condiciones, la componente se añade al modelo. Si no, se detiene el proceso, obteniendo así el número de componentes adecuado.

En la Figura 3.2 puede verse un ejemplo de un gráfico que suele acompañar a los resultados del PLS, que representa el valor de R^2 y Q^2 acumulado según el número de componentes. En este caso, el criterio de selección del número de componentes se cumple al llegar a 9 componentes.

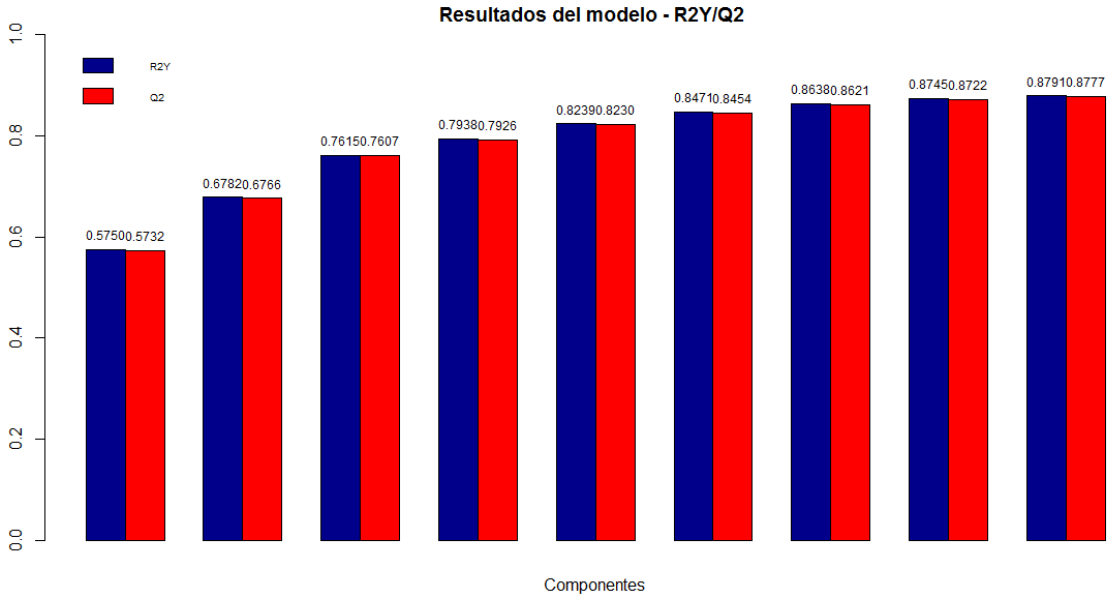


Figura 3.2.- Ejemplo de gráfico R^2/Q^2 para un modelo PLS

El PLS permite calcular dos estadísticos que son de gran importancia para la detección y diagnóstico de fallos (Kourti & MacGregor 1995), como se verá más adelante (ver apartado 4.2.2):

1. *Squared Prediction Error* (SPE): representa la Suma de Cuadrados residual de cada observación. Mide la distancia euclídea al cuadrado de cada observación original al subespacio de proyección.
2. T^2 de Hotelling: representa la distancia al cuadrado en el sentido de Mahalanobis de la proyección de las observaciones en el subespacio de componentes principales respecto al centro de gravedad de la nube de datos.

El PLS es una técnica muy versátil que puede ser aplicada a numerosos problemas como clasificación, discriminación y detección y diagnóstico de fallos, con estructuras de datos muy diversas (datos faltantes, colinealidad o deficiencia de rango, es decir, con más variables que observaciones) (Ferrer et al. 2008).

3.3.2 Librerías de R

Se han estudiado diferentes librerías que implementan la técnica de PLS: "pls"[6], "plsdepot"[7], "mixOmics"[8] y "plsgenomics"[9].

[6] Librería "pls": https://cran.r-project.org/web/packages/pls/pls.pdf
[7] Librería "plsdepot": https://cran.r-project.org/web/packages/plsdepot/plsdepot.pdf
[8] Librería "mixOmics": https://cran.r-project.org/web/packages/mixOmics/mixOmics.pdf

Aparte de que ninguna de ellas incorpora ningún tipo de selección del número de componentes, el principal problema es que no están orientadas hacia la monitorización y diagnóstico de fallos, por lo que no calculan los límites de control de los estadísticos comentados anteriormente. La que proporciona mayor variedad de información en sus resultados es la librería "plsdepot" implementada por Gastón Sánchez, por lo que se ha tomado como punto de partida sobre la que programar todo lo necesario para poder crear un sistema de detección y diagnóstico de fallos:

1. Selección automática del número de componentes.
2. Gráfico de R^2 y Q^2 acumulado según el número de componentes.
3. Cálculo del estadístico SPE y T^2 de Hotelling y de sus correspondientes límites de control.
4. Gráficos de control: SPE y T^2 de Hotelling.
5. Contribuciones a SPE y a la T^2 de Hotelling.
6. Coeficientes del modelo, tanto escalados como desescalados.
7. Cálculo del VIP.

3.4 Random forest

3.4.1 Introducción

Breinman propuso los *Random Forests* (RF) en 2001 cambiando la forma en la que se construyen los árboles de regresión o clasificación (Breinman 2001). En los árboles estándar, cada nodo se divide usando la mejor división posible entre todas las variables predictoras disponibles. En un RF, sin embargo, cada nodo se divide usando la mejor de las variables a partir de un subconjunto de las variables disponibles, que se selecciona aleatoriamente en cada nodo. Esta idea, en principio un poco ilógica, funciona muy bien en comparación con otros clasificadores como las Máquinas de Vector Soporte (o SVM, *Support Vector Machines*) o las Redes Neuronales. También se trata de una técnica robusta frente al sobreajuste. Además, es una técnica fácil de usar dado que principalmente utiliza dos parámetros (el número de variables que serán seleccionadas en cada división, m_{try}, y el número de árboles en el bosque, n_{tree}).

El proceso de creación de un RF empieza por seleccionar n_{tree} muestras de los datos originales mediante *bootstrapping* (técnica de muestreo aleatorio con reposición, es decir, en la que algunos elementos no serán seleccionados y otros lo podrán ser más de una vez en cada muestreo). Para cada muestra, se creará un árbol con la modificación comentada anteriormente: para cada nodo, en lugar de coger la mejor división de entre todos los predictores posibles, se seleccionarán aleatoriamente m_{try} predictores y se seleccionará el que proporcione una mejor división. Finalmente, se creará la predicción agregando las predicciones de los n_{tree} árboles (el criterio para generar la predicción del RF será la media de las predicciones en el caso de la regresión y la clase más votada en el caso de la clasificación).

[9] Librería "plsgenomics": https://cran.r-project.org/web/packages/plsgenomics/plsgenomics.pdf

La técnica de muestreo mediante *bootstrapping* omite, en promedio, un 37% de las observaciones para cada árbol de decisión del RF. Estas observaciones se denominan *out-of-bag* (OOB) y permiten calcular lo que se conoce como error OOB. Dicho error se calcula a partir de la predicción OOB: para cada observación, la predicción OOB es un promedio de las predicciones de todos los árboles del RF para los que dicha observación no ha sido tomada en cuenta (es decir, es una observación OOB). El error OOB se calcula comparando la predicción OOB de cada una de las observaciones con el valor real. Este error es un estimador insesgado del error real del RF.

Los RF poseen dos características adicionales muy interesantes (Liaw & Wiener 2002):

1. **Importancia de las variables**. Tal y como se verá más adelante (ver apartado 4.1.7.1), este concepto es difícil de definir, dado que la importancia de una variable puede estar condicionada en mayor o menor medida por su posible interacción con otras variables. El algoritmo del RF estima la importancia de la variable comprobando la variación en el error de la predicción cuando los datos de esta variable son permutados mientras los del resto de las variables permanecen fijos. Los cálculos se realizan árbol tras árbol, a medida que se construye el modelo.
2. **Medida de proximidad**. El elemento (*i*,*j*) de la matriz de proximidad generada por el RF es la fracción de árboles cuyos elementos *i* y *j* caen el mismo nodo final. La idea es que las observaciones similares deberían estar en los mismos nodos con mayor frecuencia que las observaciones diferentes.

3.4.2 Modo no supervisado para la detección de anómalos

El mayor inconveniente de cara a utilizar RF para la detección y diagnóstico de fallos es que se trata de un sistema supervisado. En el caso de la detección de fallos esto significa que, para poder detectar fallos, sería necesario disponer de unos datos de entrenamiento en los que las observaciones estuviesen clasificadas previamente en función de si representan o no un fallo. En este sentido, el PLS sería un método no supervisado, ya que el propio método es capaz de discernir qué situaciones son anómalas y cuáles no, sin necesidad de una clasificación previa de las observaciones que representan un fallo.

Disponer de un histórico de fallos en un entorno como el que se está trabajando tiene múltiples inconvenientes. Por un lado, sería necesario que los expertos de planta revisasen los históricos de datos para detectar situaciones anómalas. Esto es viable cuando se trabaja con pocas variables, pero este no es el caso. Hay situaciones en las que los ingenieros y los operarios de planta reconocen cuándo el consumo de combustible en el horno es mayor de lo que debería, pero hay muchos otros casos en los que no es así. Por otro lado, sería imposible disponer de un histórico que recogiese todas las posibles situaciones anómalas de consumo de combustible, porque son ilimitadas.

Sin embargo, existe la posibilidad de construir un RF en modo no supervisado (Liaw & Wiener 2002). Para ello pueden etiquetarse los datos originales como “clase 1” y crear un segundo conjunto de datos del mismo tamaño que el primero, que se etiquetará como “clase 2”. Este segundo conjunto de datos se generará aleatoriamente a partir de las distribuciones

univariantes de las variables originales. Dada la matriz de variables predictoras $\mathbf{X}_{NxK}$ (con *N* observaciones y *K* variables), se muestreará un valor de la primera variable (vector columna x_{i1}, donde i=1,…,N), otro de x_{i2}, y así sucesivamente hasta x_{iK}. El proceso se repetirá *N* veces hasta tener un conjunto de igual tamaño que el primero y que se etiquetará como "clase 2". Esta segunda clase tendrá una distribución de variables aleatorias independientes, cada una con la misma distribución univariante que la correspondiente variable del conjunto de datos original. De esta forma, la clase 2 destruye la estructura de dependencia de los datos originales y representaría un conjunto de datos anómalos. Ahora es posible entrenar un RF para que clasifique el conjunto de datos según estas dos clases y calcular la proximidad entre las observaciones. La matriz de proximidades o matriz de similaridad **S** tendrá dimensiones *N*x*N*, donde *N* es el número de observaciones. Se trata de una matriz simétrica con unos en la diagonal. Los valores de los elementos de esta matriz pueden ir entre 0 y 1 de manera que, cuanto más cercano a 1 es el valor, más próximas son las observaciones. De esta manera, será posible detectar observaciones anómalas, que serán aquellas cuya proximidad sea cercana a 0. En algunos casos, se trabaja sobre la matriz de disimilaridad **D**, que no es más que **D** = 1-**S**.

Debe recalcarse que en todo este proceso, la variable respuesta no se ha tenido en cuenta en ningún momento y que la medida de proximidad entre observaciones se ha efectuado únicamente a partir de las variables predictoras.

3.4.3 Librerías

3.4.3.1 R

Se han probado varias librerías para la obtención de RF. La primera de ellas es la librería "randomForest"[10]. Se trata de un interfaz de R sobre la librería original de Fortran implementada por los propios inventores de los Random Forest, Leo Breinman y Adele Cutler. Esta librería no implementa el cálculo en paralelo, aunque es posible utilizar la librería "foreach" para paralelizarla. Permite calcular la importancia de las variables durante la fase de entrenamiento, pero no en las predicciones.

Otra librería probada ha sido "ranger". Esta librería sí está diseñada para computación paralela, lo que acelera notablemente el tiempo necesario para el entrenamiento. Lamentablemente, no calcula la importancia de las variables, por lo que ha sido descartada.

La última librería estudiada ha sido "randomForestSRC", implementada por Ishwaran y Kogalur (Ishwaran & Kogalur 2007). Esta librería está especialmente diseñada para abordar temas de análisis de supervivencia aunque también permite construir RF para clasificación y regresión. Al igual que "ranger", permite computación paralela. Lo más interesante es que incorpora la posibilidad de calcular la matriz de proximidad y la importancia de las variables, tanto en la fase de entrenamiento como a la hora de realizar las predicciones.

[10] Librería "randomForest": https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

3.4.3.2 Python

Como se demostrará en los siguientes apartados, los parámetros devueltos por las librerías de R para la estimación de la importancia de las variables y la proximidad entre las observaciones no son útiles en el caso de utilizar los RF para la monitorización y detección de fallos.

Sin embargo, la librería de Python "TreeInterpreter"[11] desarrollada por Ando Saabas permite calcular las contribuciones de las variables en las predicciones devueltas por el RF.

Un árbol de decisión con M nodos divide el espacio de los datos de entrenamiento en M regiones R_m, $1 \leq m \leq M$. Así, la función de predicción de un árbol de decisión puede definirse como:

$$f(x) = \sum_{m=1}^{M} c_m \, I(x, R_m)$$

Donde,

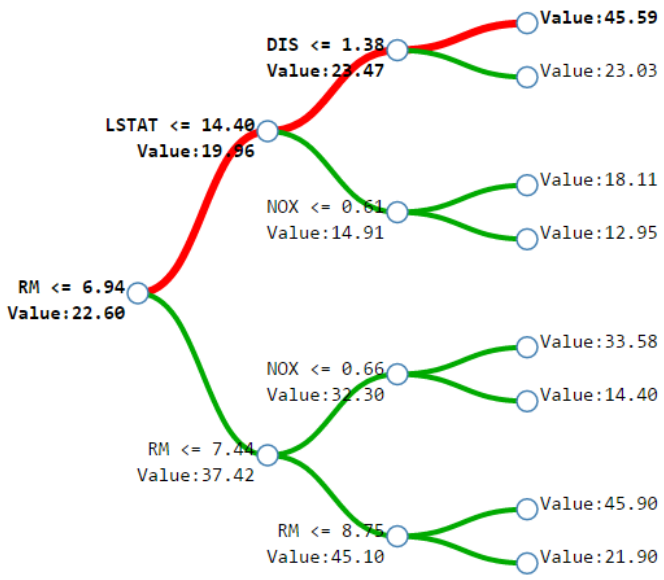
M es el número de nodos del árbol
c_m es una constante que depende de la región m
R_m es una región del espacio de datos
$I = \begin{cases} 1, & x \in R_m \\ 0, & \text{en caso contrario} \end{cases}$

El valor de c_m se determina en la fase de entrenamiento del árbol, que en el caso de los árboles de regresión corresponde a la media de la variable respuesta de las observaciones que pertenecen a la región R_m.

La ecuación anterior devuelve la predicción para la variable respuesta, pero no tiene en cuenta el camino seguido hasta llegar a dicha predicción. La idea de Ando Saabas es la de descomponer la predicción teniendo en cuenta las contribuciones de cada uno de los nodos hasta llegar al nodo final (Saabas 2014). Esto se verá mediante un ejemplo. Se ha construido un árbol de regresión que predice el valor de una casa en Boston en los años 70' en función de una serie de parámetros: distancia al centro (DIS), número de habitaciones promedio por vivienda (RM), porcentaje de población con bajo estatus (LSTAT) y ppm de óxidos de nitrógeno del aire (NOX).

En el ejemplo de la Figura 3.3, el precio predicho para la vivienda es de 45.59 miles de dólares. Puede verse el camino de decisión seguido hasta llegar al valor predicho. Al pie del árbol se muestra la ecuación de la predicción, que se basa en sumar las contribuciones de cada una de las variables a lo largo del camino de decisión.

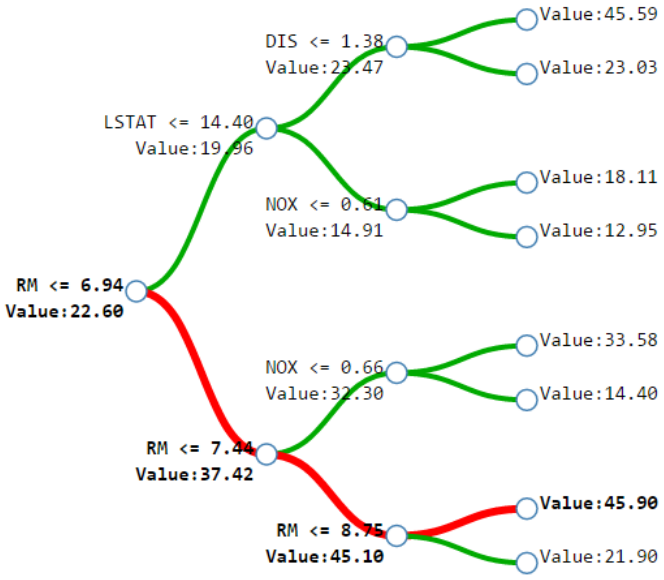
[11] "Random forest interpretation with scikit-learn": http://blog.datadive.net/random-forest-interpretation-with-scikit-learn/



Prediction: **45.59** ≈ **22.60** (trainset mean) **- 2.64**(loss from RM) **+ 3.52**(gain from LSTAT) **+ 22.12**(gain from DIS)

Figura 3.3.- Ejemplo A del camino de decisión (Saabas 2014)

En la Figura 3.4 se muestra un segundo ejemplo. En ambos casos, las predicciones son muy parecidas, sin embargo puede comprobarse que el camino seguido en el caso B es muy diferente al del caso A.



Prediction: **45.90** ≈ **22.60** (trainset mean) **+ 14.82**(gain from RM) **+ 7.68**(gain from RM) **+ 0.80**(gain from RM)

Figura 3.4.- Ejemplo B del camino de decisión (Saabas 2014)

Así pues, la ecuación de predicción puede escribirse ahora como:

$$f(x) = c_{full} + \sum_{k=1}^{K} contrib(x,k)$$

Donde,

K es el número de variables predictoras

c_{full} es el valor promedio de la variable respuesta (en la raíz del árbol, es decir, sin tener en cuenta ninguna división)

contrib(x,k) es la contribución de la *k*-ésima variable predictora a la variable respuesta *x*

La anterior ecuación puede extenderse fácilmente al caso de un RF, dado que la predicción de un RF es el promedio de las predicciones de cada uno de sus árboles, en este caso, de sus contribuciones:

$$F(x) = \frac{1}{J}\sum_{j=1}^{J} c_{j\,full} + \sum_{k=1}^{K}\left(\frac{1}{J}\sum_{j=1}^{J} contrib_j(x,k)\right)$$

Donde J es el número de árboles del RF.

3.5 Support Vector Machines (SVM)

3.5.1 Introducción

Las Máquinas de Vector Soporte o SVMs (*Support Vector Machines*) son un conjunto de algoritmos de aprendizaje supervisado usados para clasificación de datos. Básicamente, una SVM busca un hiperplano que separa de forma óptima los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior. Su nombre es debido al vector formado por los puntos más cercanos al hiperplano de separación que se denomina vector de soporte (*support vector*, en inglés) (Cortes & Vapnik 1995).

Matemáticamente, dado un conjunto de datos de validación determinado (x_i,y_i), i=1,…,l, donde $x_i \in R^n$ son los atributos e $y_i \in \{1,-1\}$ son las categorías, se pretende hallar un hiperplano que divida los datos con categoría y_i = 1 de aquellos con categoría y_i = -1, y que puede representarse mediante la siguiente ecuación:

$$\boldsymbol{w^T} \cdot \boldsymbol{x} - b = 0$$

Donde **w** es el vector normal (perpendicular) al hiperplano y **x** es el vector de valores de una observación.

Si los datos de entrenamiento pueden separarse linealmente, es posible seleccionar dos hiperplanos ($\boldsymbol{w^T} \cdot \boldsymbol{x} - b = 1$ y $\boldsymbol{w^T} \cdot \boldsymbol{x} - b = -1$) que separen dichos datos de manera que se maximice la distancia entre ambos:

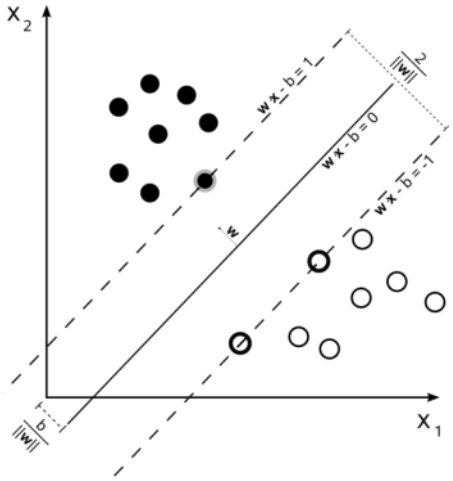


Figura 3.5.- Hiperplano de separación entre dos clases (Cortes & Vapnik 1995).

La distancia entre ambos hiperplanos es $2/\|\boldsymbol{w}\|$, por lo que, en definitiva, lo que se desea es minimizar $\|\boldsymbol{w}\|$, según el siguiente problema de optimización:

$$\min_{w,b} \ \|\boldsymbol{w}\|$$

$$\text{sujeto a } y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} - b) \geq 1, \forall\, i = 1, \dots, l$$

Se pretende que los individuos del grupo 1 (y=+1) caigan por encima del margen definido por el hiperplano $H = \boldsymbol{w^T} \cdot \boldsymbol{x} - b = 1$, y que los individuos del grupo 2 (y=-1) caigan por debajo del margen definido por el hiperplano $H = \boldsymbol{w^T} \cdot \boldsymbol{x} - b = -1$

El anterior problema de optimización es difícil de resolver porque depende de la norma del vector **w**, que implica una raíz cuadrada. Sin embargo, es posible convertir dicho problema en un problema de optimización cuadrática, minimizando $\frac{1}{2}\|\boldsymbol{w}\|^2$ en lugar de $\|\boldsymbol{w}\|$:

$$\min_{w,b} \ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$\text{sujeto a } y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} - b) \geq 1, \forall\, i = 1, \dots, n$$

Esto se conoce como formulación primal. Existe una segunda formulación del problema (formulación dual) que se basa en el uso de multiplicadores de Lagrange en las restricciones y que reduce el coste computacional.

Idealmente, el modelo basado en SVM debería producir un hiperplano que separe completamente los datos en dos categorías. Sin embargo, una separación perfecta no siempre es posible y, si lo es, el resultado del modelo no puede ser generalizado para otros datos (está sobreajustado). Cuando los datos no pueden ser separados completamente se utiliza el método denominado *soft margin*, que incorpora unas variables de holgura ξ_i que evalúan el error de clasificación de los datos. Estas variables están ponderadas por un parámetro de penalización *C* que controla la compensación entre los errores de entrenamiento y la holgura de las restricciones (ξ_i), creando así un margen (*soft margin*) que permite algunos errores en la clasificación a la vez que los penaliza (Hastie et al. 2013):

$$\min_{w,b,\xi} \ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i$$

$$\text{sujeto a } \ y_i(\boldsymbol{w}\cdot\boldsymbol{x_i} - b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

En la Figura 3.5 se muestra un ejemplo de separación lineal. Al igual que sucede en otros métodos lineales, es posible hacer el método más flexible incorporando funciones *kernel*, que permiten proyectar los datos en un nuevo espacio de *N* dimensiones de manera que un conjunto de datos no separable linealmente pueda ser separado en este nuevo espacio. Las funciones *kernel* más utilizadas habitualmente son las siguientes:

- Polinomial: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\gamma \boldsymbol{x}_i{}^T \boldsymbol{x}_j + r)^d, \ \gamma > 0$
- RBF (del inglés, *radial basis function*): $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right)$
- Sigmoidal o MLP (del inglés, *multilayer perceptron*):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\alpha \boldsymbol{x}_i \boldsymbol{x}_j{}^T + \beta), \qquad \alpha > 0, \qquad \beta < 0$$

Donde γ, *r*, *d*, σ, α y β son parámetros del *kernel*.

El concepto de las SVM expuesto para el problema de clasificación puede extenderse para el caso de la regresión (Smola & Bernhard 1998). En este caso se habla de *Support Vector Regression* (SVR).

En el caso de la clasificación, el modelo generado sólo depende de un subconjunto de los datos de entrenamiento, dado que la función de coste no tiene en cuenta los datos que están más allá del margen establecido por el parámetro *C*. De manera análoga, en el caso de la regresión se introduce un nuevo parámetro ε que establece un límite máximo para las desviaciones del modelo:

$$\min_{w,b,\xi} \ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i$$

$$\text{sujeto a } \ y_i(\boldsymbol{w}\cdot\boldsymbol{x_i} - b) \geq \varepsilon - \xi_i$$
$$y_i(b - \boldsymbol{w}\cdot\boldsymbol{x_i}) \geq \varepsilon - \xi_i^*$$
$$\xi_i \geq 0$$

En la Figura 3.6 puede verse un ejemplo de los márgenes de tolerancia ε. Los puntos dentro de estos márgenes no son tenidos en cuenta por la función de coste a la hora de entrenar el modelo.

Finalmente cabe decir que las SVM sufren de un fenómeno conocido con el nombre de *curse of dimensionality*. Este concepto hace referencia a los problemas que aparecen al aumentar el número de dimensiones de los datos a estudio y que no estaban presentes para dimensiones bajas. En el caso de las SVM, si el número de variables es muy alto y la separación de las clases puede realizarse mediante unas pocas variables, será difícil que el *kernel* encuentre la solución, al tener que explorar entre un número muy elevado de posibilidades. En este trabajo, este

problema intentará solucionarse mediante una selección previa de las variables más importantes (ver apartado 4.1.7).

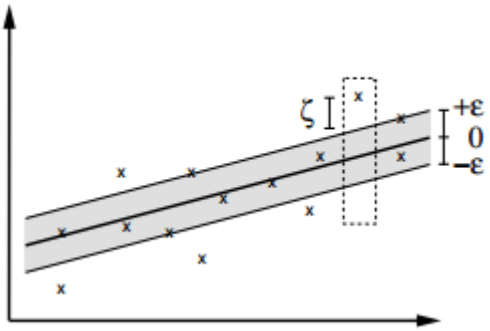


Figura 3.6.- Margen de tolerancia (ε) en el caso de la SVR (Smola & Bernhard 1998).

3.5.2 Modo no supervisado

La misma idea de Breinman & Cutler para entrenar un RF en modo no supervisado puede aplicarse al caso de las SVM, para detectar observaciones anómalas (ver apartado 3.4.2).

3.5.3 Librerías de R

Para crear los modelos de SVM (tanto los de clasificación como los de regresión) se ha utilizado la librería "e1071" de R[12], que proporciona una interfaz en R para la conocida librería "libsvm" implementada en C++ (Meyer 2015).

Esta librería permite obtener tanto modelos de clasificación como de regresión. También incorpora el caso especial de clasificación de una clase mediante SVM (*one-class SVM*) para la detección de observaciones anómalas (ver apartado 4.1.6.4).

Las funciones *kernel* incorporadas en esta librería son la lineal, polinomial, RBF (*radial basis function*) y sigmoidal.

[12] Librería "e1071": https://cran.r-project.org/web/packages/e1071/e1071.pdf

4 Metodología

4.1 Construcción de modelos

El esquema general para el diagnóstico de fallos mediante modelos basados en datos se basa en dos etapas. En la primera fase, se construye el modelo en modo *offline* a partir de los datos históricos. En la segunda, se implementa el modelo de manera *online*, conectándolo a los sistemas de información de la planta para recoger los datos y creando una interfaz de usuario para mostrar los resultados (Aldrich & Auret 2013, p.221-78).

Los pasos para la creación de un modelo a partir de datos históricos del proceso son los siguientes:

4.1.1 Análisis Exploratorio de Datos

El Análisis Exploratorio de Datos o EDA (*Exploratory Data* Analysis) es un término introducido por John W. Tukey en 1977 para describir el conjunto de técnicas que permiten analizar un conjunto de datos y determinar sus principales características, con el objetivo de seleccionar las técnicas estadísticas más apropiadas para analizar dichos datos.

Los estadísticos propuestos por Tukey inicialmente fueron: mínimo, máximo, mediana, cuantiles, media y desviación estándar. Existen otros, como el coeficiente de asimetría (*skewness*), que permite evaluar el grado de asimetría que presenta la distribución de una variable, o la curtosis, que da idea de la forma de la distribución (respecto a la distribución Normal).

También existen métodos gráficos, como los siguientes:

- Histogramas: es una representación gráfica de una variable en forma de barras, donde cada barra indica la frecuencia de los valores representados. La principal desventaja de los histogramas es que proporcionan una función de densidad de probabilidad discreta que suele ser muy sensible a la elección de los parámetros (punto inicial y anchura de las clases), así como al número de datos.
- KDE (*Kernel Density Estimation*): representan una alternativa a los histogramas basada en distribuciones *kernel*, que suma las funciones de suavizado de cada valor para generar una curva de probabilidad continua (Bowman & Azzalini 1997).
- Gráfico de caja y bigotes: está basado en cuartiles y está compuesto por un rectángulo, la "caja", y dos brazos, los "bigotes". Proporciona información sobre los valores mínimo y máximo, los cuartiles Q1, Q2 o mediana y Q3, y sobre la existencia de valores atípicos y la simetría de la distribución.
- Papel probabilístico normal: permite contrastar la normalidad de un conjunto de datos, comparando la distribución empírica de una muestra de datos con la distribución Normal. Consiste en representar, en un mismo gráfico, los datos empíricos

observados frente a los datos que se obtendrían en una distribución Normal teórica. Si la distribución de la variable es Normal, los puntos quedarán cerca de una línea recta.

Un tema de interés, sobretodo en el caso que nos ocupa (el de la industria química) es la correlación entre variables. Existen diversos métodos para medir la correlación entre variables, según la naturaleza de los datos. El más conocido es el coeficiente de correlación de Pearson (r), que puede adoptar valores entre +1 (correlación positiva perfecta) y -1 (correlación negativa perfecta). Como norma general, se establecen los siguientes rangos de valores (en valor absoluto):

- $|r| < 0.25$: correlación muy débil
- $0.25 \leq |r| < 0.5$: correlación débil
- $0.5 \leq |r| < 0.75$: correlación moderada
- $|r| \geq 0.75$: correlación alta

Cuando se desea analizar la correlación en un conjunto de variables puede construirse la matriz de correlación. Se trata de una matriz simétrica, con unos en la diagonal y que contiene los coeficientes de correlación de cada pareja de variables.

El coeficiente de correlación de Pearson calcula la correlación entre parejas de variables, pero también es posible utilizar técnicas multivariantes como las siguientes:

- Escalado Multidimensional: permite calcular y visualizar el nivel de similaridad de las observaciones de un conjunto de datos para n variables, en base a una matriz de distancias. Sus principales aplicaciones están en el campo del marketing y las ciencias sociales.
- *Principal Component Analysis* (PCA): proyecta el conjunto de datos en un espacio de variables latentes obtenidas a partir de la matriz de covarianzas. Los ejes de este espacio están formados por los vectores propios de la matriz de covarianzas y están ordenados de mayor a menor según la varianza de las variables latentes obtenidas. Esta técnica es especialmente útil en el caso de que exista una elevada correlación entre variables (como es el caso de la industria química), permitiendo conocer la estructura de correlación latente de los datos.

4.1.2 Limpieza de datos

La limpieza de datos (*data cleansing* o *data scrubbing*) permite detectar y corregir (o eliminar) datos faltantes, incompletos, anómalos o irrelevantes de un conjunto de datos.

En el caso de las plantas de proceso como las refinerías de petróleo, hay dos causas principales en la generación de datos sucios:

a) Fallo en el sistema de información de la planta o PI (*Plant Information system*) encargado de capturar y almacenar la información de los diferentes medidores de la

planta. Dicho fallo puede ser debido, a su vez, a un fallo en la red de comunicación o en el sistema de almacenamiento, que provocará datos faltantes.

b) Fallo de medición en el sensor, que generará mediciones imprecisas o sesgadas.

Un Análisis Exploratorio de Datos o EDA (*Exploratory Data Analysis*) puede ayudar a detectar ambas situaciones.

El tratamiento de datos faltantes depende del porcentaje presente en el conjunto de datos. Por norma general, variables con más de un 50% de datos faltantes no son aconsejables para obtener modelos. De igual forma, observaciones con un porcentaje elevado de datos faltantes deberían ser descartadas. Como se ha comentado antes, las situaciones con un porcentaje elevado de datos faltantes suelen ser debidas a fallos en las comunicaciones o en los sistemas de almacenamiento (servidores).

Existe una extensa bibliografía sobre el tema de imputación de datos. En general, imputar la media o la mediana de una variable no es una práctica recomendable. Entre los mejores métodos para la imputación de datos se encuentran los basados en las técnicas de *k-nearest neighbor* o *Principal Component Analysis* (Folch-Fortuny et al. 2015).

4.1.3 Escalado

El escalado es necesario para asegurar que las variables con valores altos no dominen sobre el modelo. Por ejemplo, una variable de proceso podría ser la temperatura del reactor con valores alrededor de los 100ºC, mientras que otra variable podría ser la concentración del producto con valores de 0.1. Es decir, existe un factor de mil entre ambas variables de proceso. Si no se escalan, la temperatura del reactor contribuirá de manera desproporcionada al modelo, aun cuando capture menos información que la otra variable.

El escalado de una variable se obtiene restando la media y dividiendo por la desviación estándar. Tanto la media como la desviación estándar deben ser calculadas a partir de datos correspondientes a condiciones normales de operación (o datos NOC, *Normal Operating Conditions*).

La media $\bar{x}_i$ y la desviación estándar s_i para una variable de proceso no escalada $\boldsymbol{x}_i^{(no\ esc)}$ con *N* observaciones se calculan como:

$$\bar{x}_i = \frac{1}{N}\sum_{j=1}^{N} x_{i,j}^{(no\ esc)}$$

$$s_i = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N}\left(x_{i,j}^{(no\ esc)} - \bar{x}_i\right)^2}$$

La variable escalada $x_{i,j}$ para la *j*-ésima observación de la variable *i* se calcula como:

$$x_{i,j} = \frac{x_{i,j}^{(no\ esc)} - \bar{x}_i}{s_i}$$

Durante la fase de implementación *online*, los nuevos valores de las variables de proceso se escalan con las medias y las desviaciones estándar calculadas a partir de los datos NOC durante la fase *offline* de construcción del modelo.

4.1.4 Muestreo

Las variables de proceso que se utilizarán para obtener el modelo deben muestrearse con una determinada frecuencia para asegurar que existen suficientes datos de condiciones normales de operación (NOC) y que las situaciones anómalas serán detectadas a tiempo. Es muy importante disponer de un conjunto representativo de datos NOC, dado que los sistemas de detección de fallos se basan en estos datos para capturar la estructura latente del proceso y poder identificar situaciones anómalas que rompan dicha estructura. Pero, a la vez, es necesario saber si la frecuencia de muestreo tiene el tamaño suficiente como para detectar a tiempo dichas situaciones anómalas. En el caso de las refinerías de petróleo, las dinámicas suelen ser muy lentas. En el presente trabajo, teniendo en cuenta la experiencia de los ingenieros de planta, se ha decidido muestrear los datos con periodos de 1 hora. En este caso, al recuperar los datos del sistema de información, el propio sistema realiza un promedio de los datos registrados en el sistema durante cada hora y devuelve dicha media.

4.1.5 Retardo

Al aplicar un retardo l a una variable se obtiene la misma variable desplazada l instantes previos. Existen varias técnicas para saber si las mediciones de una variable de proceso son estadísticamente independientes de las mediciones en momentos previos. Dichas técnicas incluyen la autocorrelación simple, la correlación parcial y la correlación cruzada (Peña 2005).

El coeficiente de autocorrelación simple de orden l, $\rho(l)$, mide la correlación entre observaciones de la variable retardadas l unidades de tiempo. Este coeficiente mide tanto la relación directa entre el instante *t* y el instante *t+l,* como la relación indirecta, es decir, transmitida a través de los instantes intermedios. Sin embargo, para la identificación de la dinámica del proceso interesa estimar sólo los efectos directos entre dos instantes *t* y *t+l* (eliminando el efecto transmitido a través de los instantes intermedios). Para ello se utiliza el coeficiente de autocorrelación parcial.

La correlación cruzada es un concepto similar a los anteriores pero, en este caso, se comparan dos variables de proceso en dos instantes de tiempo determinados. En la industria química suele existir correlación entre variables de proceso pero, en ocasiones, dicha correlación está "retardada" en el tiempo dado que el efecto de una variable en una zona de la unidad puede no verse reflejado en otra zona hasta que haya pasado un cierto tiempo. Por ello es necesario estudiar la correlación entre variables retardadas en el tiempo.

Mediante el uso de las anteriores técnicas es posible encontrar los instantes de tiempo (retardos) en los que las observaciones de las variables de proceso están correlacionadas. Cuando se detecta esta situación, puede crearse una matriz de datos ampliada $\mathbf{X}^{*(l)}$ a partir de la matriz de datos original, pero que incluya, además, las l mediciones previas de las variables

correlacionadas. Al retardar una variable x_t un retardo l se obtiene la variable x_{t-l}. En definitiva, se trata de desplazar l veces los valores de *x*, de manera que al final se dispone de una matriz con *N* - l observaciones, donde l es el mayor de todos los retardos aplicados a la variable en cuestión (Figura 4.1).

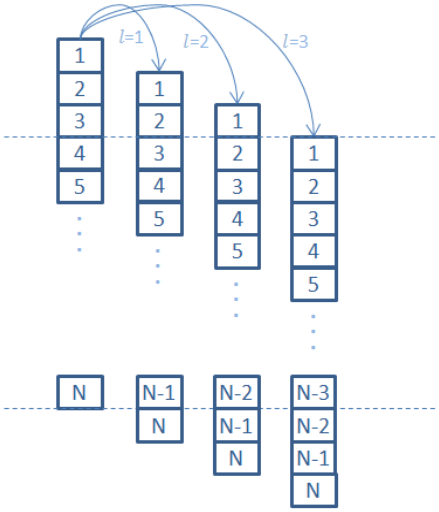


Figura 4.1.- Creación de la matriz de datos ampliada $\mathbf{X}^{*(l)}$

Un ejemplo de una técnica que usa esta matriz ampliada para estudiar la correlación a lo largo del tiempo entre variables es el *Dynamic PCA* (*Dynamic Principal Component Analysis*) o el *Dynamic PLS* (*Dynamic Partial Least Squares*). Realmente no son más que el uso de PCA o PLS sobre la matriz ampliada. Estas técnicas también permiten estudiar si existe correlación entre variables (Ku et al. 1995).

Para poder conocer los retardos que son importantes, es necesario establecer un retardo l_{max} lo suficientemente grande. El siguiente paso es construir la matriz ampliada con todos los retardos desde 1, ..., l_{max}, para todas las variables. Finalmente, mediante PCA pueden conocerse qué variables están más correlacionadas entre sí.

4.1.6 Segmentación de datos

4.1.6.1 Introducción

Debido a los grandes avances de los últimos años en cuanto a la recolección y almacenamiento de datos, en los procesos industriales altamente sensorizados diariamente se recogen grandes cantidades de datos. En concreto, en las grandes plantas industriales como las refinerías de petróleo, existen sensores que miden miles de variables de proceso prácticamente cada segundo. Otros parámetros, como la calidad de los productos, se miden con menos frecuencia ya que, en la mayoría de los casos, requieren ensayos de laboratorio que no pueden llevarse a cabo en tiempo real. Las bases de datos de estas plantas almacenan millones de registros históricos sobre el funcionamiento de la misma, pero no todos estos datos son válidos (Shardt & Shah 2014). Como se ha visto, puede haber datos erróneos o faltantes debido a un mal

funcionamiento de los sensores o de las comunicaciones, o datos anómalos debidos a periodos de actividad anormales de la planta. Estos datos deben ser descartados antes de obtener los modelos que serán utilizados en aplicaciones industriales. El proceso de obtención de los datos "normales"(o *Normal Operating Conditions*, NOC) se conoce como "segmentación" y se lleva a cabo mediante una medida del grado de anomalía del dato. En este sentido, las técnicas de detección de valores atípicos (*outliers*) tienen un papel fundamental.

La detección de valores atípicos se refiere al problema de encontrar patrones que no están en concordancia con el comportamiento generalizado esperado (Aggarwal 2013). El principal problema es establecer un límite para decidir qué observaciones son normales y cuáles son atípicas. Muchas veces los datos están acompañados de ruido que no interesa en la construcción del modelo. En el ejemplo de la Figura 4.2, los principales patrones (*clusters* o conglomerados) de los datos son idénticos en ambos casos, aunque hay diferencias significativas fuera de estos *clusters*. En el caso de la Figura 4.2(a) el punto etiquetado con la letra "A" parece muy distinto del resto de datos y, por lo tanto, es anómalo. Sin embargo, la situación de este mismo punto en la Figura 4.2(b) no está tan clara. A pesar de que el punto sigue estando en una región poco poblada, es más difícil determinar con seguridad que corresponda a una desviación respecto al resto del conjunto de datos. Parece más bien que este punto forma parte del ruido de los datos.

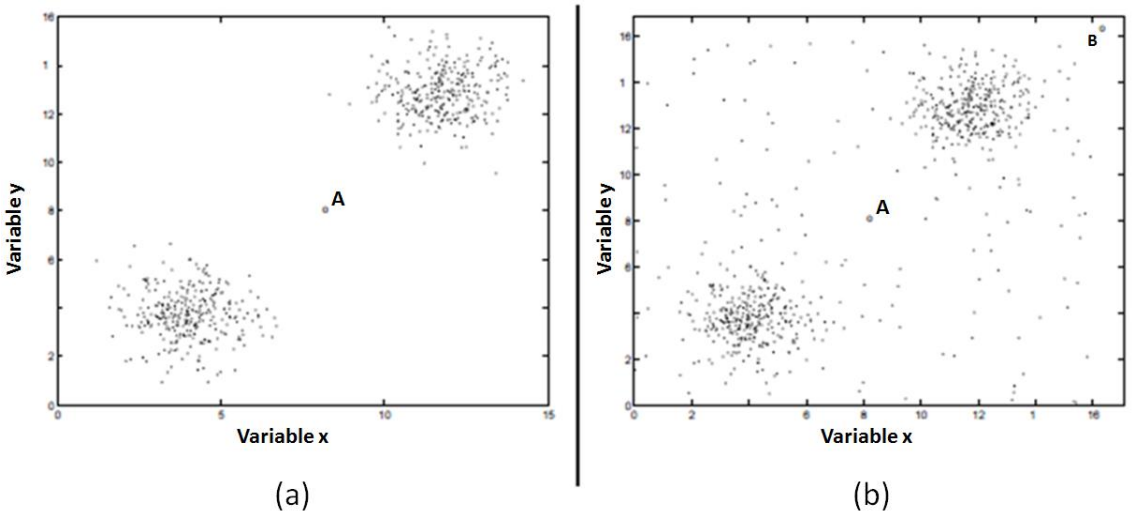


Figura 4.2.- Diferencia entre ruido y valores atípicos (Aggarwal 2013).

Algunos autores distinguen entre valor atípico y valor anómalo, siendo éste último un valor atípico que, por algún motivo, tiene especial interés para el analista de datos.

En el presente trabajo se aplicarán técnicas no supervisadas para la detección de valores atípicos (Malik et al. 2014). Este tipo de métodos parten de la premisa de que las observaciones normales son mucho más frecuentes que las observaciones atípicas, lo que es una situación habitual en las bases de datos de históricos de plantas industriales.

Una vez eliminadas las observaciones atípicas, pueden quedar "islas" de datos en el histórico que, en algunos casos, pueden contener pocos datos. Además, este número puede reducirse todavía más al incluir variables decaladas, ya que al aplicar el retardo se "pierden" algunas observaciones. Otro tema a tener en cuenta es que se está trabajando con datos que tienen

una cierta dinámica, por lo que, al validar los modelos, es necesario seleccionar conjuntos de datos contiguos (no aleatorios). Por todo ello, se tomarán sólo subconjuntos de datos con un número de observaciones lo suficientemente grande como para estudiar posibles retardos. En este trabajo se ha decido trabajar con subconjuntos de 100 observaciones, como mínimo (es decir, poco más de 4 días, dado que los datos son horarios).

Finalmente, comentar que la obtención de datos NOC se aplicará únicamente sobre los datos de entrenamiento, que son los que se utilizarán para construir el modelo, y no sobre los de validación.

En los siguientes apartados se comentará cómo se utilizarán las 3 técnicas seleccionadas en este trabajo para segmentar datos: PLS, Random Forest y Support Vector Machines.

4.1.6.2 Partial Least Squares

Como se ha comentado (ver apartado 3.4.2), el PLS sería un método no supervisado por lo que se refiere a la detección de anómalos para la segmentación de datos, ya que el propio método es capaz de discernir qué situaciones son anómalas y cuáles no, sin necesidad de una clasificación previa de las observaciones que representan un fallo. La técnica estadística de PLS permite detectar situaciones anómalas a partir de los gráficos de los estadísticos SPE y T^2 de Hotelling (ver apartado 4.2.2). Los límites de control (UCL_{SPE} para el estadístico SPE y UCL_{T2} para el estadístico T^2 de Hotelling) están calculados en base al percentil 95% de la distribución, por lo que es esperable que aproximadamente sólo el 5% de las observaciones supere dicho límite, bajo condiciones NOC. De cara a la segmentación de los datos, se eliminarán únicamente aquellas observaciones que estén claramente fuera de los límites de control. Por ello, se establece un criterio que consiste en eliminar aquellas observaciones que tengan un valor de SPE o de T^2 superior a 3 veces el correspondiente UCL (UCL_{SPE} o UCL_{T2}, respectivamente).

4.1.6.3 Random forest

A partir de la matriz de disimilaridad **D** (ver apartado 3.4.2) es posible aplicar técnicas de *clustering*, como el escalamiento multidimensional, para realizar diferentes estudios como, por ejemplo, visualizar en un gráfico la capacidad de separación del RF o detectar observaciones anómalas. El escalamiento multidimensional es una técnica que sitúa cada observación en un espacio de *N* dimensiones de manera que las distancias entre observaciones se conservan de la mejor manera posible.

Respecto a la detección de observaciones anómalas existe otra posibilidad que es la proporcionada por la función “outlier” de la librería ”randomForest” de R (ver apartado 3.4.3.1). Esta opción obtiene una medida del grado de anomalía de cada observación a partir de la matriz de similaridad **S**. Dicho grado de anomalía se calcula como el inverso de la suma de cuadrados de las proximidades entre cada observación y el resto de las observaciones. Cuando este valor es superior a 10 se considera que la observación es anómala (Breiman 2001).

4.1.6.4 Support Vector Machines

La clasificación de una clase mediante SVM (*one-class SVM*) fue introducida por Schölkopf en 1999 y ha sido utilizada para la detección de observaciones atípicas (Schölkopf 1999). La técnica se basa en la creación de una esfera mediante una serie de vectores soporte que delimitan el conjunto de datos normales, excluyendo los datos atípicos. Ello conlleva la modificación de la función objetivo del problema de optimización de la SVM, añadiendo un parámetro ν que se utiliza para controlar el volumen de la esfera y, por lo tanto, el límite que separa las condiciones normales de los datos atípicos. El principal problema es que este límite no se conoce a priori, por lo que es necesario supervisar los resultados por un experto que permita identificar si se trata realmente de observaciones anómalas o no.

Por otro lado, la misma técnica que se aplica para los Random Forest para la detección de observaciones anómalas en modo no supervisado es aplicable también al caso de las SVM, entrenando una SVM que sea capaz de discernir las observaciones anómalas a partir de un conjunto de datos creados artificialmente (ver apartado 3.4.2).

4.1.7 Selección de variables

4.1.7.1 Introducción

La selección de variables (*feature selection*) se refiere al proceso de obtención de un subconjunto de variables que son útiles para construir un buen predictor. Estas variables "útiles" no son necesariamente relevantes. Una variable relevante puede ser redundante en presencia de otra variable relevante con la que está fuertemente correlacionada. Y al contrario, un subconjunto de variables útiles pueden excluir variables redundantes, aunque relevantes. El objetivo de la selección de variables es el de excluir aquellas variables que son tanto redundantes como irrelevantes y que, por lo tanto, pueden ser eliminadas sin provocar demasiada pérdida de información (Guyon & Elisseeff 2003).

La selección de variables es de gran interés en áreas donde los conjuntos de datos contienen cientos e incluso miles de variables como, por ejemplo, el procesamiento de texto en documentos de internet, el análisis de expresión génica y la química combinatoria. En el caso que nos ocupa, la selección de variables permitirá conocer cuáles son los factores clave o *drivers* del proceso.

Los principales objetivos de la selección de variables son los siguientes:

- Mejorar la capacidad predictiva del modelo.
- Reducir el número de variables y, por tanto, el coste computacional del modelo.
- Proporcionar una mejor comprensión del proceso.

Los algoritmos para la selección de variables pueden dividirse en los siguientes grupos:

4.1.7.2 Filtros

Los métodos de filtrado analizan las propiedades de las variables sin tener en cuenta la técnica que se usará posteriormente para crear el modelo final. La mayoría de estos métodos pueden seleccionar y, simultáneamente, ordenar las variables según un determinado criterio de importancia.

Existen innumerables métodos para la selección de variables basados en filtros. Algunos de ellos están disponibles en la librería "mlr"[13] de R. Esta librería incorpora métodos y algoritmos de otras librerías de R, unificándolos bajo un mismo formato de manera que es posible utilizar de forma similar algoritmos de diferentes librerías. La librería "mlr" aborda diversos temas relacionados con la minería de datos, como el preprocesamiento e imputación de datos, muestreo (validación cruzada, doble validación, *bootstrap*, …), computación paralela, evaluación del rendimiento del modelo, análisis de curvas ROC, *tuning* de parámetros o *ensemble learning*, entre otros.

Respecto a la selección de variables, "mlr" permite comparar diversos métodos simultáneamente. Como norma general, los métodos que suelen dan mejores resultados son el método *information gain* (ganancia de información) y el estadístico del test chi-cuadrado de independencia entre las variables predictoras y la variable respuesta.

La ganancia de información (IG, del inglés, *Information Gain)* es una medida de la reducción de la entropía que, a su vez, es una medida de la incertidumbre asociada a una variable aleatoria. Se calcula como (Roobaert et al. 2006):

$$IG(\mathbf{X}, x_i) = H(\mathbf{X}) - \sum_{v \in valores(x_i)} \frac{|x_i = v|}{N} H(x_i = v)$$

Donde,

X es la matriz de datos de dimensiones *N* x *k* (N observaciones y *k* variables)

x_i es el vector columna *i*-ésimo de la matriz **X**. Representa a la variable *i*-ésima (*i*=1,…,*k*)

$|x_i = v|/N$ es la fracción de observaciones de la variable *i*-ésima que contienen el valor *v*

$H(\mathbf{X})$ es la entropía asociada a la matriz de datos **X**, definida como:

$$H(\mathbf{X}) = \sum_i - p_i \, log_2 \, p_i$$

Donde p_i es la probabilidad de la clase *i*, calculada como la proporción de la clase *i* en el conjunto de datos **X**.

$H(x_i = v)$ es la entropía asociada a la variable x_i que contiene el valor *v*

[13] "Integrated Filter Methods": http://mlr-org.github.io/mlr-tutorial/release/html/filter_methods/index.html

La idea básica es que cuanto mayor es la entropía, mayor es el contenido en información. El objetivo principal de la ganancia de información es determinar qué atributos de un determinado conjunto de variables son más útiles a la hora de discriminar entre las clases de la variable respuesta. Este concepto es extrapolable para variables continuas, en cuyo caso será necesario discretizar.

4.1.7.3 Partial Least Squares

El PLS permite seleccionar las variables más importantes del modelo gracias al parámetro *Variable Importance in the Projection* (VIP), que resume la importancia de las variables en **X**. El parámetro VIP fue desarrollado por Wold y se basa en una suma ponderada de los pesos **w*** teniendo en cuenta la cantidad de varianza explicada de Y en cada dimensión. Matemáticamente, se define como (Wold et al. 1993):

$$VIP_{Ak} = \sqrt{\sum_{a=1}^{A} \left(w_{ak}^2 * (SSY_{a-1} - SSY_a) \right) * \frac{K}{(SSY_0 - SSY_A)}}$$

Donde,

A es el número de componentes
K es el número de variables en **X**
W_{ak} es el peso de la variable *k* para la componente *a*
SSY_a es la suma de cuadrados de los residuos del modelo con *a* componentes

La suma de cuadrados de todos los VIP es igual al número de términos en el modelo dado que el VIP promedio es igual a 1. El VIP es especialmente útil para interpretar un modelo PLS con muchos componentes y multitud de respuestas. También es posible comparar el VIP de un término con el resto para determinar su importancia relativa. Los términos con VIP superior a 1 son los más importantes a la hora de explicar la variabilidad de **Y**.

Utilizando el VIP como criterio de selección es posible realizar lo que se conoce como *variable prunning* (o poda del modelo). Esta poda consiste en seleccionar únicamente aquellas variables cuyo VIP sea mayor que un determinado VIP de corte. Por regla general hay un VIP de corte óptimo, que produce un modelo con el mejor ajuste. La idea general de esta técnica es que, al prescindir de variables poco relevantes, se elimina ruido del modelo, mejorando el ajuste.

4.1.7.4 Random Forest

Existe un parámetro devuelto por los RF que permite conocer la importancia relativa de las variables. Mediante este parámetro es posible identificar las variables que tienen mayor contribución a la hora de predecir las observaciones. Dicho de otra forma, si se detectan variables con una importancia muy baja significa que estas variables prácticamente no han sido utilizadas por los árboles para realizar las particiones de los diferentes nodos, por lo que, si se prescinde de ellas, la bondad de ajuste del modelo no se verá afectada.

4.1.7.5 Problema mínimo-óptimo

Cuando el método no posee un sistema de selección de variables propio, se pueden usar métodos de búsqueda para encontrar el mínimo subconjunto de variables que proporcionan las mejores predicciones. Este problema, conocido como el problema mínimo-óptimo (Nilsson et al. 2007), ha sido muy estudiado y existen gran cantidad de algoritmos. Todos ellos buscan repetidamente el espacio de variables con diferentes subconjuntos de predictores, utilizando como criterio de selección del mejor subconjunto aquél que proporcione la mejor precisión al modelo en base a un determinado parámetro (como pueden ser el estadístico R^2 o el porcentaje de aciertos). Ejemplos de estos métodos son los algoritmos genéticos, *simmulated annealing* y métodos de selección tipo *forward*, *backward* o *stepwise*.

Uno de estos métodos es el denominado *Recursive Feature Elimination* (RFE), que se basa en una selección de variables tipo *backward*, es decir, que parte del conjunto de variables original y va descartando de forma secuencial las menos relevantes. El algoritmo se basa en seleccionar subconjuntos de variables de diferentes tamaños y calcular el ajuste obtenido en el modelo en base a un determinado criterio de selección. Una descripción detallada del algoritmo puede encontrarse en Kuhn 2016.

Otro método de este tipo está basado en el parámetro de importancia de variables que calculan los RF y está disponible en la librería "varSelRF" de R[14]. Este método lleva a cabo la eliminación de variables tipo *backward* a partir de la importancia de las variables calculada por el RF, tomando como criterio de minimización el error *out-of-bag* (ver apartado 3.4).

4.1.8 Ajuste de parámetros

Existen métodos, como las SVM o las Redes Neuronales, que incorporan una serie de parámetros que es necesario ajustar para obtener los mejores resultados posibles. A esta técnica se le conoce con el nombre de "ajuste de parámetros" (*hyperparameter tuning*).

El ajuste de parámetros se resuelve mediante un proceso de optimización en el que la función objetivo suele ser una medida de la bondad de ajuste del modelo obtenido (precisión, R^2, …). En el apartado 5.9.3 se aplicará esta técnica al caso de las SVM.

4.1.9 Validación del modelo

4.1.9.1 Validación cruzada

La validación se utiliza para evaluar la capacidad predictiva del modelo evitando los problemas de sobreajuste que aparecen cuando el modelo se valida contra los mismos datos con los que se ha obtenido el propio modelo.

[14] The "varSelRF" package, https://cran.r-project.org/web/packages/varSelRF/varSelRF.pdf

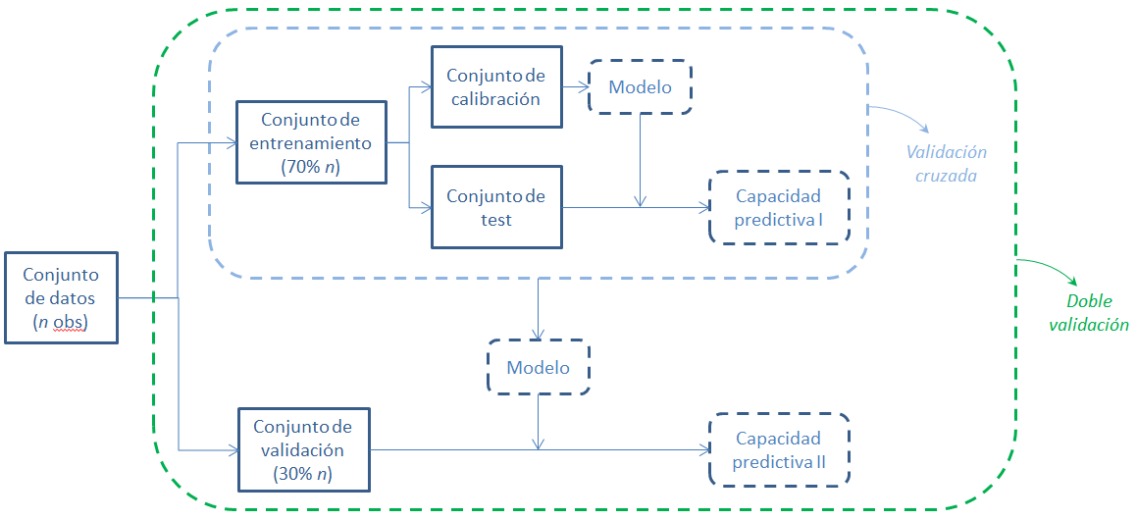


Figura 4.3.- Validación cruzada y doble validación

Para ello se ha implementado lo que se conoce con el nombre de "doble validación" (ver Figura 4.3), que consiste en lo siguiente:

1. Separar el conjunto de datos originales en un conjunto de entrenamiento y otro de validación según el ratio de entrenamiento indicado por el usuario (normalmente el ratio de entrenamiento es de 0.7, es decir el 70% de los datos se dedican a entrenar el modelo, mientras que el 30% restante se usa para validarlo).
2. Realizar la validación cruzada sobre los datos de entrenamiento.

Esta metodología es especialmente útil en algunas técnicas de aprendizaje automático como las SVM que son bastante susceptibles a sufrir sobreajuste, es decir, que ajustan bien los datos de entrenamiento pero obtienen malas predicciones en nuevos datos de similares características (conjunto de validación).

El objetivo principal de la doble validación es el de evaluar la capacidad predictiva del modelo a partir de un conjunto de datos "externo" al modelo, es decir, que no ha intervenido en la construcción del mismo, lo que evita los posibles problemas de sobreajuste comentados anteriormente.

Por otro lado, sobre el conjunto de datos de entrenamiento se realizará una validación cruzada. La validación cruzada se basa en dividir el conjunto de entrenamiento en un conjunto de calibración con el que se construirá un modelo y un conjunto de test sobre el que se aplicará el modelo anterior para poder evaluar su capacidad predictiva. El proceso puede repetirse varias veces, dependiendo del criterio utilizado para obtener los conjuntos de calibración/test. Los métodos más utilizados son los siguientes:

a) *Leave-one-out*: el conjunto de test está formado por una única observación y el conjunto de calibración, por el resto de observaciones. Este proceso se repite tantas veces como observaciones tenga el conjunto de datos.
b) *k-fold*: en este caso, el conjunto de datos se divide aleatoriamente en *k* conjuntos donde *k*-1 de ellos constituyen el conjunto de calibración (con los que se ajustará el modelo) y el conjunto restante se usará como conjunto de test (con el que se evaluará

la capacidad predictiva del modelo ajustado). Este proceso se repite *k* veces de forma que cada conjunto forma parte del conjunto de test sólo una vez. Cabe decir que, en el caso de variables respuesta categóricas o lógicas, la selección de los conjuntos se hace de manera que se mantiene la proporción de clases presente en el conjunto de datos original, para que sean lo más representativas posible.

Sea cual sea el criterio elegido para dividir la muestra en conjuntos de calibración/test el proceso de validación cruzada se basa en los siguientes pasos:

1. Obtener un modelo a partir del conjunto de calibración.
2. Predecir la variable respuesta en el conjunto de test aplicando el modelo obtenido en el paso anterior.
3. Repetir los pasos 1 y 2 para cada conjunto de calibración/test.

Tanto si el método elegido para dividir la muestra es el *leave-one-out* como si es *k-fold*, al final del proceso de validación cruzada se dispondrá de una predicción para cada una de las observaciones del conjunto de datos, con la que podrá calcularse la bondad del ajuste.

Cabe comentar que, en este trabajo, debido a la dinámica existente en el proceso (correlación entre variables decaladas), la selección de los subconjuntos de entrenamiento y validación no se realizará de manera aleatoria (como suele hacerse en este tipo de métodos), sino que los subconjuntos contendrán datos correlativos.

4.1.9.2 Validación mediante ventanas

Como se ha comentado, existe una cierta dinámica en el proceso a estudio, por lo que se utilizará una técnica de validación útil para este tipo de casos que está basada en la creación de ventanas. Esta técnica consiste en utilizar un conjunto de datos correlativos como entrenamiento (ventana de entrenamiento) y otro, inmediatamente posterior al anterior, como validación (ventana de validación).

4.2 Monitorización de procesos y diagnóstico de fallos

4.2.1 Introducción

En el apartado anterior (apartado 4.1) se ha visto cómo construir un modelo a partir de unos datos históricos. La monitorización de procesos se basa en utilizar este modelo para comparar el valor real con el esperado. En caso de que el proceso no se comporte según las condiciones normales de operación, el diagnóstico de fallos permitirá conocer cuáles son las causas de ese comportamiento inesperado.

4.2.2 Partial Least Squares

La monitorización de procesos y diagnóstico de fallos mediante PLS se basa en la construcción de gráficos de control a partir de los estadísticos SPE y T^2 de Hotelling (ver apartado 3.3.1). Estos gráficos representan el valor del correspondiente estadístico para cada una de las observaciones y una línea que indica el límite de control superior o UCL (del inglés, *upper control limit*, Figura 4.4). Los límites de control (UCL_{SPE} para el estadístico SPE y UCL_{T2} para el estadístico T^2 de Hotelling) están calculados en base al percentil 95% de la distribución, por lo que es esperable que aproximadamente sólo el 5% de las observaciones supere dicho límite, bajo condiciones NOC. Los valores por encima de este límite indican posibles situaciones anómalas.

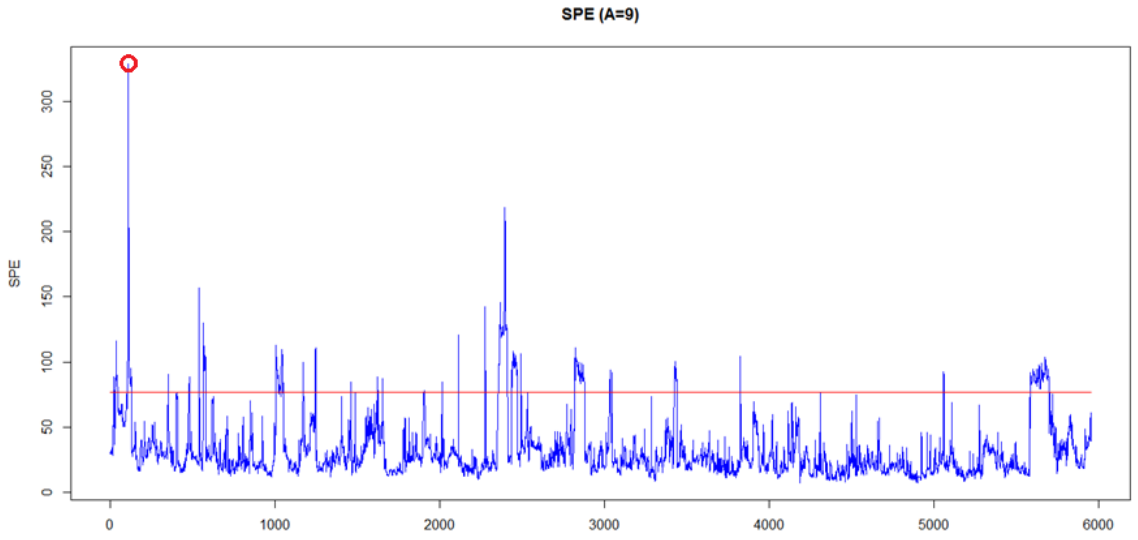


Figura 4.4.- Ejemplo de gráfico de control basado en el estadístico SPE

El gráfico SPE es el que suele comprobarse primero, dado que detecta situaciones atípicas, que rompen la estructura de correlación del modelo. En segundo lugar se comprueba el gráfico T^2 de Hotelling que permite comprobar situaciones extremas, es decir, observaciones que si bien siguen la estructura de correlación del modelo, sus valores están fuera de lo esperado, bajo condiciones NOC.

Una vez identificado un posible fallo, es posible determinar cuál es su causa a través de los gráficos de contribución del SPE o de la T^2 de Hotelling. Estos gráficos permiten conocer qué variables del proceso son las que contribuyen más a la situación anómala detectada.

En el ejemplo de la figura anterior se observa un valor de SPE muy alto para la observación número 109 (Id="7020"). Si se calculan las contribuciones al SPE de dicha observación se obtiene el gráfico de la Figura 4.5, donde la mayor contribución corresponde a la variable "PRESION_05". Si se comprueba esta variable se observa que, efectivamente, sufre un aumento que está fuera de lo esperado (ver Figura 4.6). Esta observación será considerada como atípica y deberá descartarse a la hora de construir el modelo final con datos NOC (ver apartado 4.1.6).

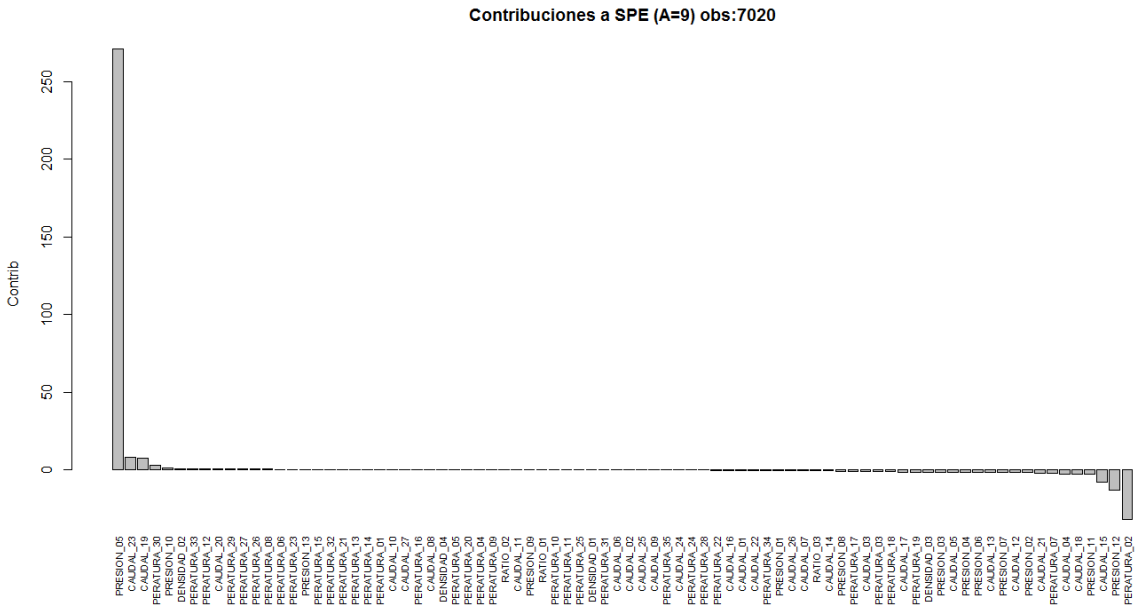


Figura 4.5.- Contribuciones a SPE de la observación "7020"

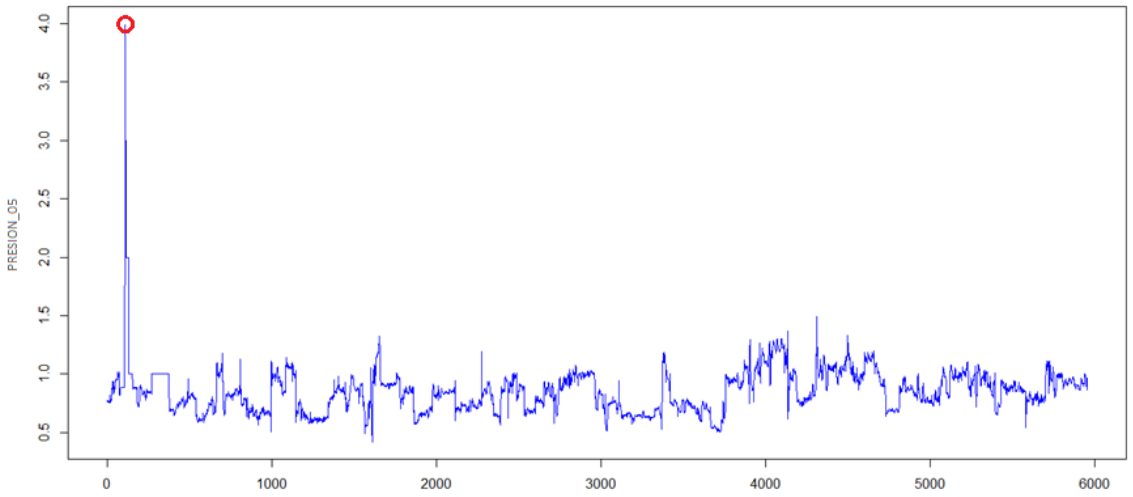


Figura 4.6.- Variable " PRESION_05"

4.2.3 Random Forests

La selección de las variables más importantes se realizará con el modelo de regresión. Estas variables seleccionadas se utilizarán después para ajustar el modelo de clasificación. Podría haberse buscado el número de variables más importantes por separado (regresión y clasificación), pero, por coherencia con la selección de variables que se hace en el PLS, se utilizará el mismo conjunto de variables para ambos casos. Esto es debido a que, en el caso del PLS, tanto la predicción de la variable respuesta como la detección de situaciones anómalas se realiza con un único modelo y, por lo tanto, con el mismo conjunto de variables.

Así pues, teniendo en cuenta todo lo visto hasta el momento, se propone la siguiente metodología para la monitorización de procesos y diagnóstico de fallos mediante el uso de Random Forests:

A. Fase *offline* (construcción del modelo):
    1. Obtener los datos NOC mediante segmentación de datos (ver apartado 4.1.6.3).
    2. A partir de los datos NOC, entrenar un RF de regresión (RFR) para conocer qué variables son las más importantes (ver apartado 4.1.7.4). En este punto puede interesar ampliar el estudio mediante la incorporación de variables decaladas (ver apartado 4.1.5).
    3. Entrenar un RF de clasificación (RFC) con el conjunto de datos NOC etiquetados como clase 1 y el conjunto de datos sintéticos (obtenidos según lo comentado en el apartado 3.4.2) etiquetados como clase 2. En esta etapa se utilizarán las variables predictoras más importantes obtenidas en el paso anterior.
B. Fase *online* (explotación del modelo): a la hora de predecir una nueva observación se hará lo siguiente:
    1. Se utilizará el RFR para calcular la predicción de la variable respuesta.
    2. Se utilizará el RFC para conocer la probabilidad de que la nueva observación corresponda a un dato NOC (clase 1) o sea un dato anómalo (clase 2). Si la nueva observación es un dato anómalo, la predicción del RFR no se tendrá en cuenta. En este caso, el diagnóstico de fallos se encarga de establecer cuáles son las causas.

Respecto al diagnóstico de fallos, en el caso de los RF, la importancia de cada variable se estima comprobando la variación en el error de la predicción cuando los datos de esta variable son permutados mientras los del resto de las variables permanecen fijos (ver apartado 3.4.1). Como se ha comentado, este parámetro da una idea del peso o la contribución de la variable en la predicción, lo cual ayudaría al diagnóstico de fallos. Sin embargo, para poder calcularlo es necesario conocer el valor real de la variable respuesta, el cual no está disponible durante la fase de explotación del modelo, es decir, la fase de predicción de nuevas observaciones.

Sin embargo existe una librería implementada en Python denominada “TreeInterpreter” que permite calcular las contribuciones de las variables predictoras, tanto en la fase de construcción del modelo como en la fase de explotación (ver apartado 3.4.3.2) y será la utilizada en este caso.

4.2.4 Support Vector Machines

La metodología para el caso de las SVM es muy similar a la anterior descrita para los RF. La principal diferencia es que las SVM son muy sensibles a la selección de parámetros por lo que debe incluirse una etapa de ajuste de parámetros. También es necesario incluir un proceso de selección de variables dado que las SVM pueden resultar impracticables cuando el número de variables es muy alto. La solución para abordar ambos problemas simultáneamente es plantear una optimización multiobjetivo, en la que las variables de decisión serán los parámetros de la SVM y el número de variables a seleccionar, y las funciones objetivo serán la minimización del número de variables y la maximización del ajuste del modelo.

Al igual que en el caso de los RF (ver apartado 4.2.3), la selección de las variables más importantes se realizará con el modelo de regresión. Las variables seleccionadas se utilizarán

después para ajustar el modelo de clasificación. Así pues, para el caso de las SVM se propone la siguiente metodología de trabajo para la fase *offline* (construcción del modelo):

1. Entrenar una SVM de regresión (SVR) con los datos de entrenamiento. Este paso incluye la optimización multiobjetivo comentada anteriormente, por lo que se obtendrán tanto el número de variables óptimo como los parámetros óptimos del modelo SVR que permitirá predecir la variable respuesta.
2. Entrenar una SVM de clasificación (SVC) a partir del conjunto de datos creados para trabajar en modo no supervisado (ver apartado 3.5.2), utilizando el conjunto de variables predictoras halladas en el paso anterior.

En ambos casos (clasificación o regresión) los modelos se validarán mediante validación cruzada (ver apartado 4.1.9.1).

La fase *online* (explotación del modelo) será idéntica a la planteada en el caso de los RF:

1. Se utilizará la SVR para calcular la predicción de la variable respuesta.
2. Se utilizará la SVC para conocer la probabilidad de que la nueva observación corresponda a un dato NOC (clase 1) o sea un dato anómalo (clase 2). Si la nueva observación es un dato anómalo, la predicción de la SVR no se tendrá en cuenta. En este caso no se ha encontrado ninguna posibilidad de realizar un diagnóstico de fallos.

5 Caso de estudio: Unidad de destilación de crudo

5.1 Software y hardware utilizados

Para realizar los cálculos se ha utilizado el software R y Python (ver apartado 3.2).

Respecto al hardware, se ha utilizado un PC con placa base Gigabite GA-990XA-UD3 y procesador AMD FX-6200 de 6 núcleos a 3.80 GHz, 16 GB de RAM y disco duro SSD de 500 GB.

5.2 Datos históricos

Por motivos de confidencialidad, las 82 variables predictoras disponibles han sido anonimizadas. Sin embargo, puede comentarse que, entre estas 82 variables, se dispone de 35 mediciones de temperatura en diferentes puntos de la unidad a estudio, 27 mediciones de caudal, 13 mediciones de presión, 4 mediciones de densidades de productos y 3 ratios (relaciones entre variables de proceso que se consideran de interés para el control de la unidad).

Los datos históricos están disponibles desde el 1 de enero de 2014 hasta el 5 de octubre de 2015, con un periodo de muestreo de 1 hora.

El sistema de información de la planta almacena los cambios en el valor de la medición de cada sensor. Durante el periodo de muestreo seleccionado para este trabajo (1 hora) pueden producirse varios cambios en el valor de un determinado sensor. Sin embargo, debido a la dinámica lenta del proceso, la variabilidad de estos cambios es pequeña, por lo que la media horaria es un dato suficientemente fiable para obtener los modelos que se usarán en este trabajo para monitorizar el proceso.

La variable a estudio (variable respuesta) es el consumo específico de combustible en el horno, definido como:

$$Cons\_esp = \frac{FOE}{CPre}$$

Donde,

Cons_esp: consumo específico de combustible en el horno (adimensional)
CPre: crudo a preflash (t/h)
FOE: Flow of Oil Equivalent (t/h) del horno, definido como:

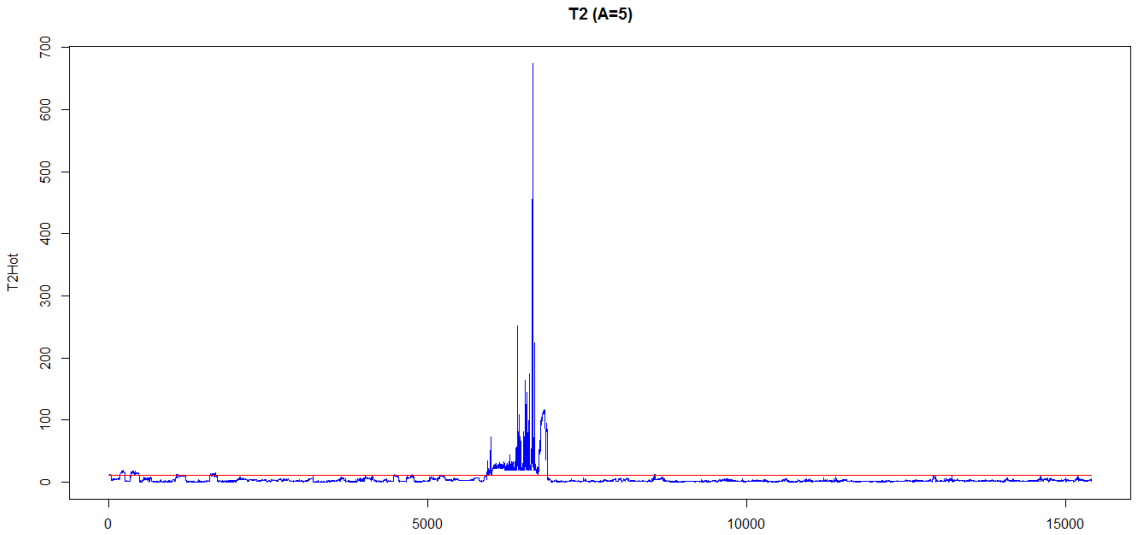
$$FOE = \frac{FGB * HC_{FG} + 1000 * FOL * HC_{FO}}{1000 * 40.2}$$

Donde,

FOE: Flow of Oil Equivalent (t/h)

FGB: fuel gas a quemador másico (kg/h)
HC_{FG}: poder calorífico del fuel gas (MJ/kg)
FOL: fuel oil a mecheros C-H1 (t/h)
HC_{FO}: poder calorífico del fuel oil (MJ/kg)
40.2: poder calorífico promedio del fuel-oil pesado (MJ/Kg)

Se ha realizado un primer PLS con todos los datos disponibles. Como se ha comentado en el apartado 3.2, el PLS permite detectar situaciones anómalas a través de los estadísticos SPE y T^2 de Hotelling. En este caso, gráfico de la T^2 de Hotelling (para 5 componentes) es el siguiente:



Se detecta una zona en la que los valores de la T^2 están claramente por encima del límite de control (línea horizontal roja). Esta zona corresponde al periodo que va desde el 05/09/2014 hasta el 15/10/2014, lo que coincide con una parada de planta para realizar tareas de mantenimiento. Según han indicado los ingenieros de planta, se aprovechó también para realizar modificaciones en el proceso que podrían haber afectado al rendimiento del horno. Este es un tema importante, dado que el modelo se obtendrá a partir de unos datos en condiciones normales de operación (NOC) que corresponden a una determinada configuración del proceso, de manera que cualquier cambio en el mismo invalidaría el modelo obtenido hasta el momento.

En resumen, de cara al trabajo de los siguientes apartados se utilizarán los datos históricos posteriores a la parada, es decir, desde 16/10/2014 hasta 05/10/2015, con un total de N=8509 observaciones.

5.3 Análisis Exploratorio de Datos

El análisis se ha realizado sobre los datos a partir de los que se construirá el modelo (es decir, el periodo comprendido entre 16/10/2014 y 05/10/2015).

El porcentaje de datos faltantes se ha calculado tanto para las observaciones como para las variables. En el periodo estudiado únicamente existe una observación con un porcentaje de datos faltantes superior al 50%. En este caso han fallado todas las mediciones, seguramente

debido a un fallo en el sistema de comunicaciones o en el sistema de almacenamiento de datos.

Respecto a las variables con mayor porcentaje de datos faltantes, son las siguientes:

Tabla 5.1.- Variables con mayor porcentaje de datos faltantes

Variable	% Faltantes
TEMPERATURA_02	3.69
PRESION_05	2.29
TEMPERATURA_33	1.62
CAUDAL_20	1.42
TEMPERATURA_15	1.25

Ninguna de ellas posee un porcentaje de datos faltantes alarmante.

También se ha calculado la matriz de correlación de todas las variables (tanto las 82 variables predictoras como la variable respuesta). Se ha utilizado una función[15] de R para buscar las variables que están correlacionadas entre sí con un coeficiente de Pearson superior a un determinado valor de corte. Los resultados para diferentes valores de corte son los siguientes:

Tabla 5.2.- Distribución de los coeficientes de correlación.

Coeficiente de correlación (r)	Nº Variables
$\|r\| \geq 0.75$	27 (32.53%)
$0.5 \leq \|r\| < 0.75$	44 (53.01%)
$\|r\| < 0.5$	64 (77.11%)

Se comprueba que casi un tercio de las variables tienen una fuerte correlación, confirmando la idea de que las variables de proceso en la industria química suelen estar fuertemente correlacionadas.

También sería conveniente realizar un PCA para estudiar la estructura latente de correlación, pero esto se hará más adelante, de una forma similar, mediante PLS.

5.4 Limpieza de datos

Anteriormente se ha comentado que se dispone de 82 variables predictoras, aunque cabe decir que el volcado original contenía 94 variables. Ello es debido a que 12 de estas variables han sido descartadas por no contener datos o no presentar variabilidad (es decir, contener siempre el mismo dato).

Como se ha comprobado en el apartado anterior, existen una serie de variables con datos faltantes. La imputación de datos faltantes se realizará mediante dos técnicas:

[15] "findCorrelation (caret package)" http://www.inside-r.org/packages/cran/caret/docs/findCorrelation

- Cuando se trabaje con PLS, esta técnica es capaz de trabajar con datos faltantes e imputar sus valores mientras ajusta el modelo.
- En el resto de casos, la imputación se realizará mediante *k-Nearest Neighbors* (utilizando *k*=5 e imputando la mediana de los vecinos más próximos).

5.5 Variables decaladas

Para estudiar el efecto del decalaje en el proceso, se ampliará la matriz de datos con las variables decaladas hasta un total de 10 horas (l = 10), lo cual generará un matriz de 912 variables, dado que se decalarán tanto las 82 variables predictoras como la propia variable respuesta. Las variables decaladas se identificarán mediante un sufijo que indicará cuál es el decalaje. Así, por ejemplo, la variable "VarA_3" corresponderá a la variable original "VarA" decalada 3 horas.

5.6 Validación del modelo

Para validar las diferentes técnicas que se aplicarán en los siguientes apartados se usará la doble validación (ver apartado 4.1.9.1). Ello implica que las 8508 observaciones originales disponibles se dividirán en dos conjuntos, uno de entrenamiento y otro de validación con un ratio de 0.7, con lo cual el tamaño de los conjuntos será:

- Entrenamiento (70%): 5956 observaciones
- Validación (30%): 2552 observaciones

De ahora en adelante, cualquier estadístico o parámetro que haga referencia al conjunto de datos de entrenamiento se identificará con el sufijo "train" y el de validación con el sufijo "test". Así, por ejemplo, la bondad de ajuste de un modelo PLS sobre el conjunto de entrenamiento será R^2_{train} y sobre el conjunto de validación, R^2_{test}

5.7 Partial Least Squares

5.7.1 Introducción

A continuación se presentarán todos los modelos que se han ajustado mediante la técnica de PLS. Inicialmente, se ajustará el modelo con todas las variables disponibles. La segmentación de datos para obtener los datos NOC se realizará mediante el criterio comentado en el apartado 4.1.6.2.

Posteriormente, se ampliará la matriz original con variables decaladas hasta 10 retardos y se volverá a ajustar un PLS. También se aplicarán las técnicas de segmentación para obtener los datos NOC.

En otro apartado, se seleccionarán las variables más importantes y se comprobará el efecto que tienen sobre la capacidad predictiva del modelo PLS.

Finalmente, se mostrarán algunos ejemplos que ilustran cómo el PLS puede utilizarse para detectar fallos y diagnosticar sus causas.

5.7.2 Variables originales

Los resultados del modelo sobre las 82 variables son los siguientes:

- A = 9 (número de componentes)
- R^2_{train} = 0.8790 (Figura 5.1)
- R^2_{test} = 0.7000 (Figura 5.2)

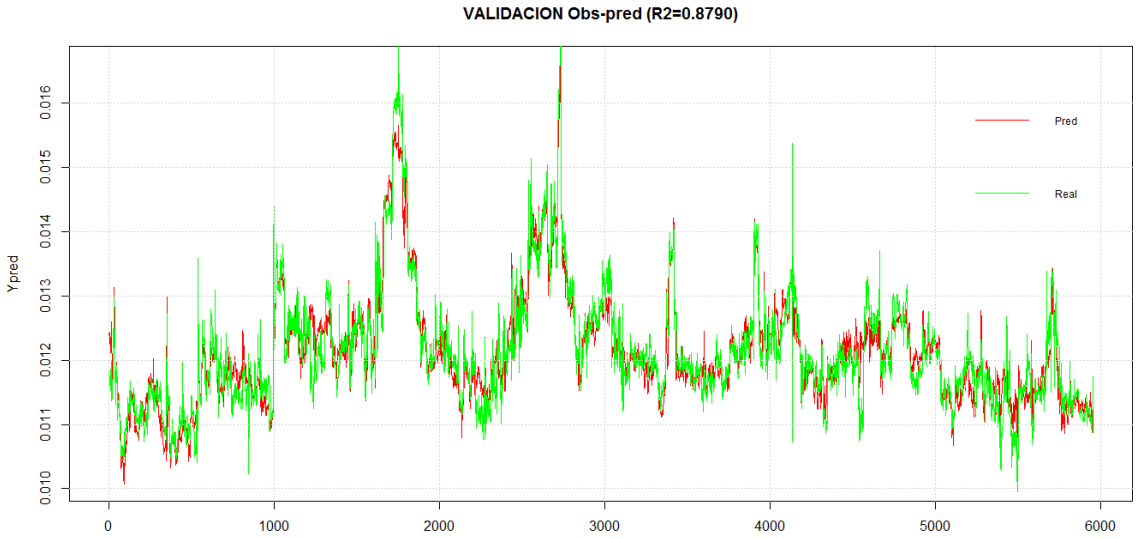


Figura 5.1.- Gráfico de observados frente a predichos para los datos de entrenamiento (variables originales).

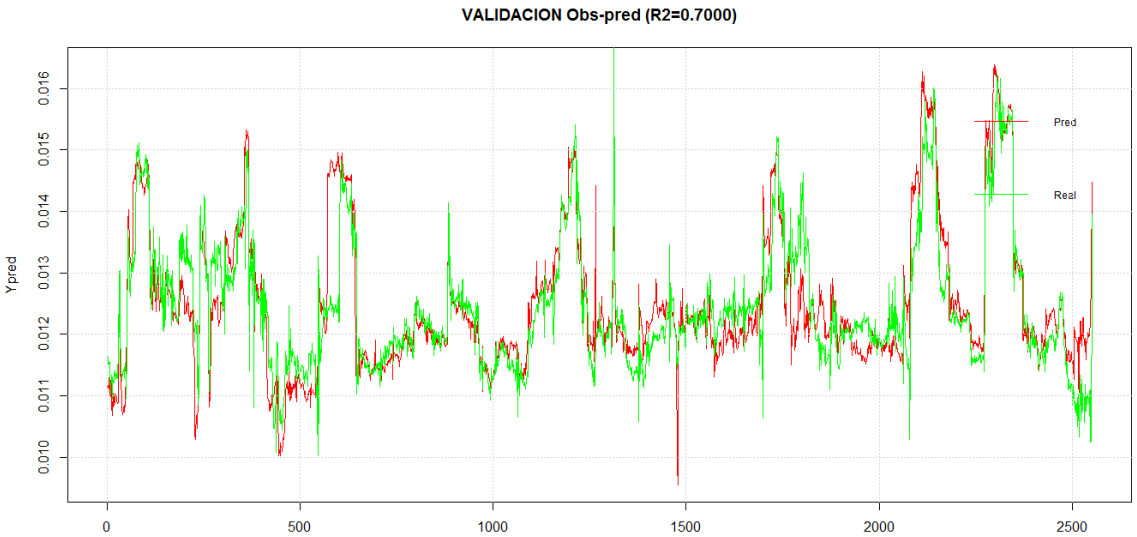


Figura 5.2.- Gráfico de observados frente a predichos para los datos de validación (variables originales).

Los gráficos de control son los siguientes:

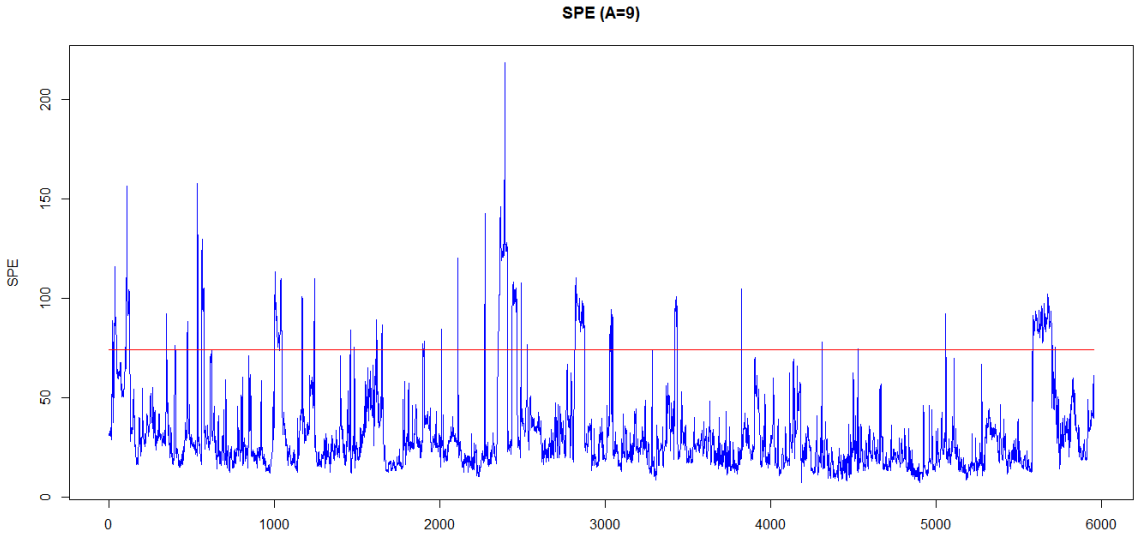


Figura 5.3.- Gráfico SPE para el modelo PLS con las variables originales (datos de entrenamiento).

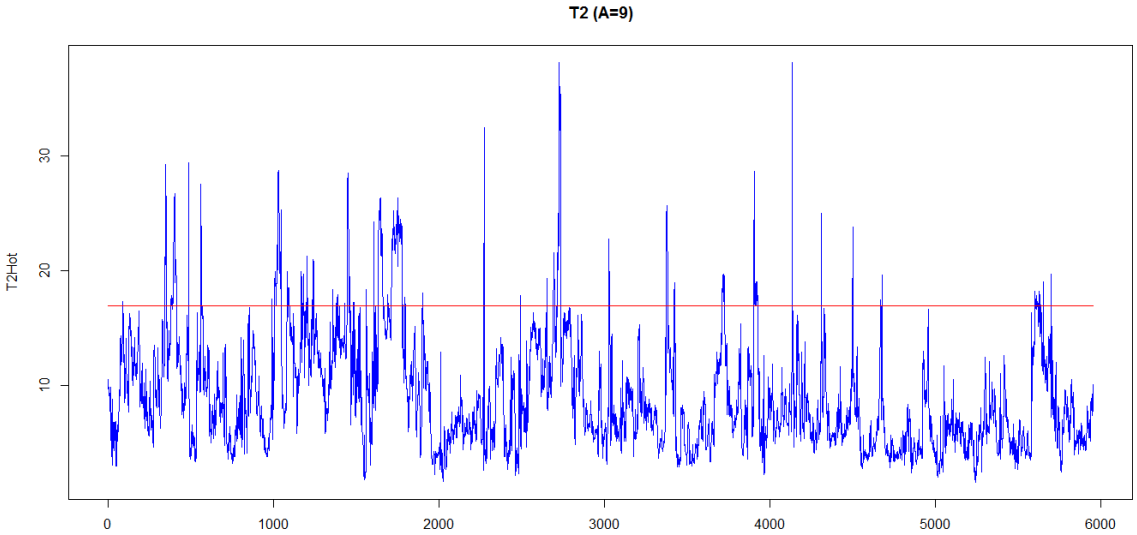


Figura 5.4.- Gráfico T^2 de Hotelling para el modelo PLS con las variables originales (datos de entrenamiento).

En este caso no hay observaciones que cumplan el criterio de segmentación de datos (ver apartado 4.1.6.2), por lo que todos los datos de entrenamiento se considerarán datos NOC.

5.7.3 Variables decaladas

Ahora se ajustará un modelo PLS sobre la matriz ampliada con variables decaladas hasta 10 retardos, por lo que se dispondrá de un total de 912 variables. Los resultados del modelo PLS son los siguientes:

- A = 9 (número de componentes)
- R^2_{train} = 0.9276 (Figura 5.5)
- R^2_{test} = 0.8526 (Figura 5.6)

El modelo sigue necesitando 9 componentes, pero los ajustes son ahora mucho mejores, tanto en el conjunto de entrenamiento como en el de validación. Esto corrobora la idea de que existe dinámica entre las variables de proceso de la unidad de destilación de crudo.

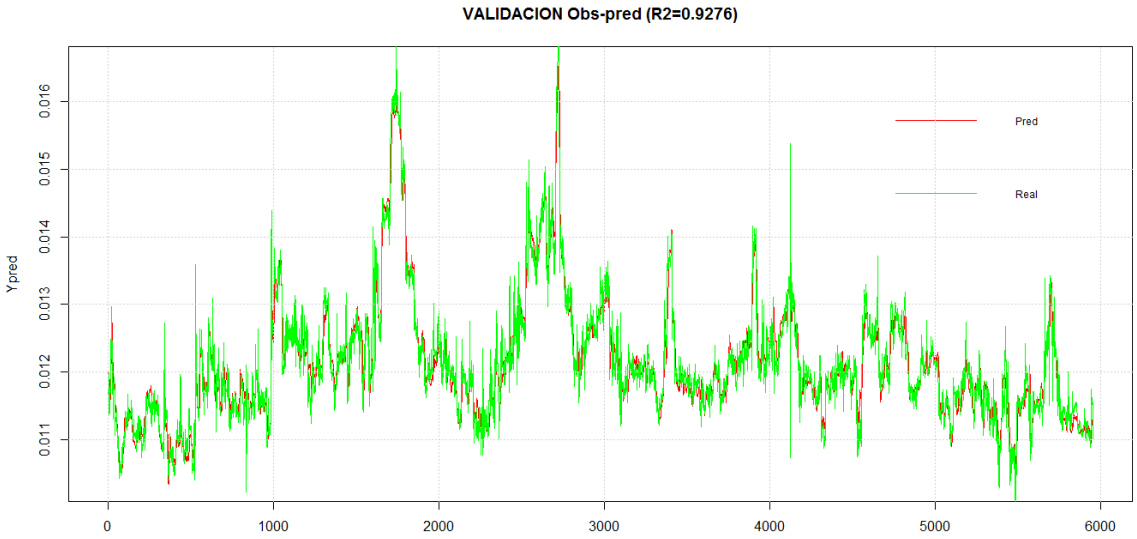


Figura 5.5.- Gráfico de observados frente a predichos para los datos de entrenamiento (variables decaladas).

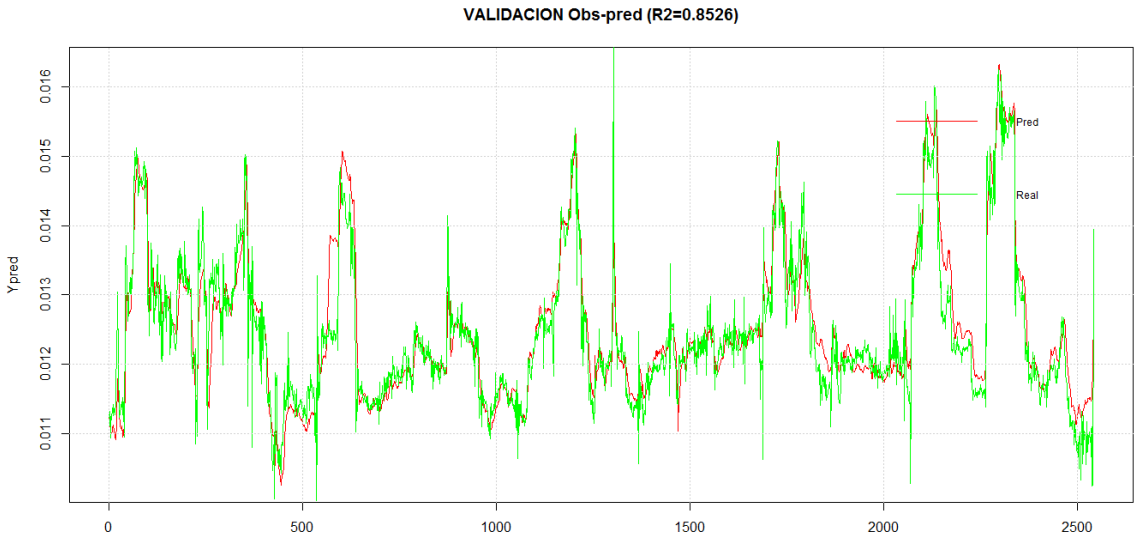


Figura 5.6.- Gráfico de observados frente a predichos para los datos de validación (variables decaladas).

Observando los gráficos de control (Figura 5.7 y Figura 5.8), se comprueba nuevamente que no hay observaciones que cumplan el criterio de segmentación de datos (ver apartado 4.1.6.2), por lo que todos los datos de entrenamiento se considerarán datos NOC.

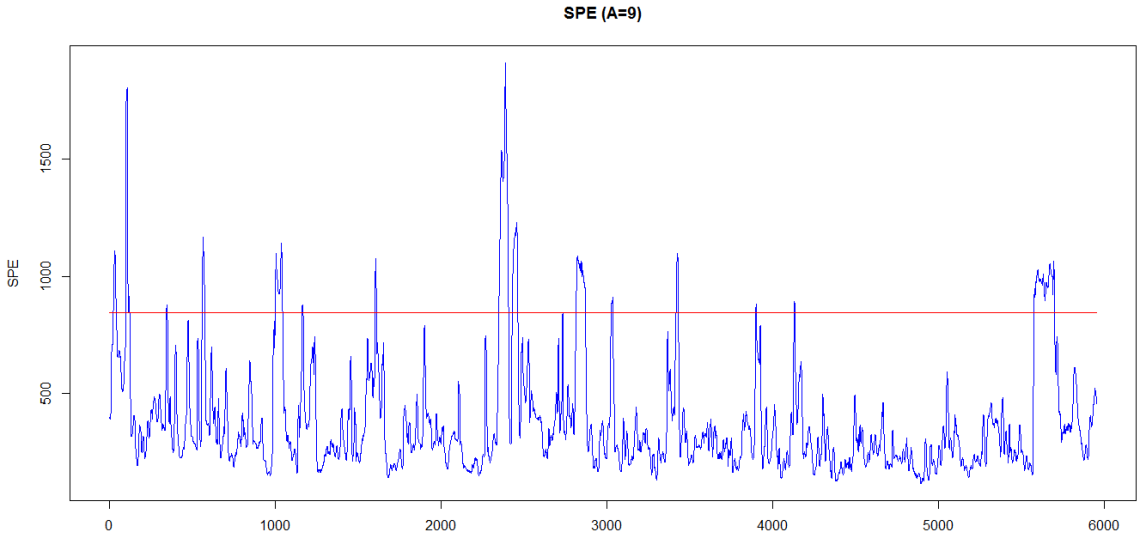


Figura 5.7.- Gráfico SPE para el modelo PLS con las variables decaladas (datos de entrenamiento).

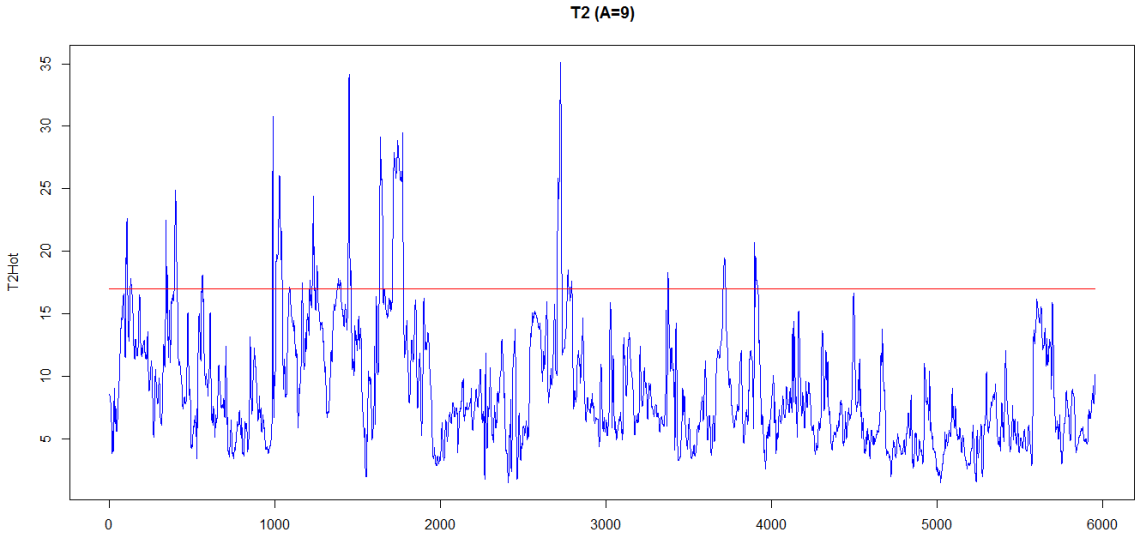


Figura 5.8.- Gráfico T^2 de Hotelling para el modelo PLS con las variables decaladas (datos de entrenamiento).

5.7.4 Selección de variables

Se ha comprobado que al utilizar retardos, ha mejorado la bondad del ajuste del modelo. Por un lado, las variables decaladas pueden aportar información al modelo, mejorando el ajuste, pero, por otro, pueden aportar ruido. Para conocer cuáles de estas variables son realmente importantes en el modelo se utilizará el VIP (ver apartado 4.1.7.3).

La Figura 5.9 muestra el gráfico VIP para el modelo con variables decaladas. Únicamente se muestran las variables con VIP superior a 1.5.

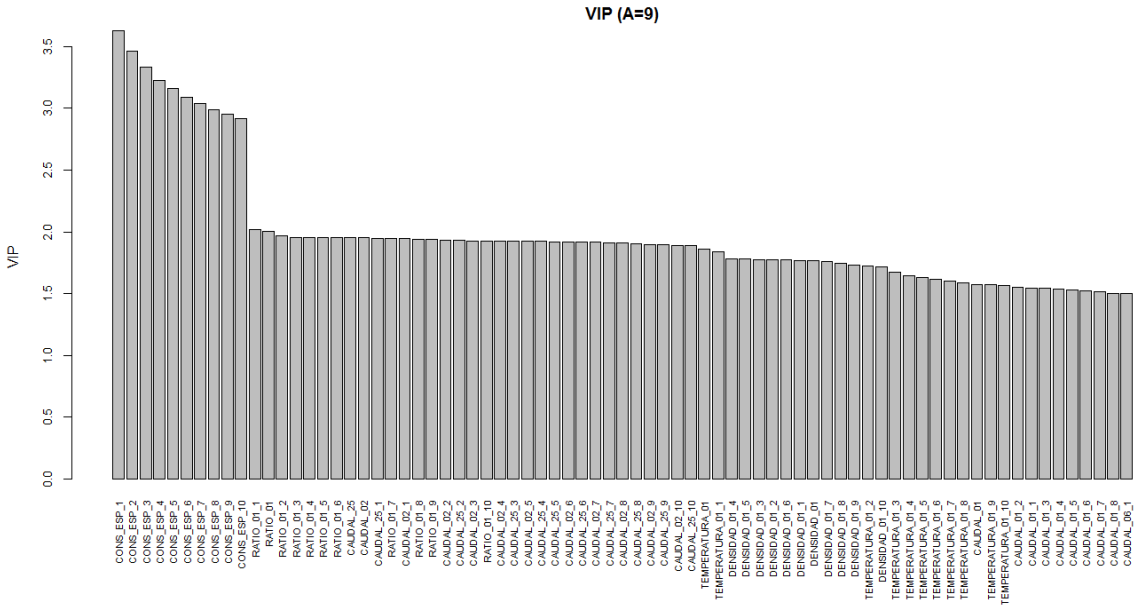


Figura 5.9.- Gráfico VIP para el modelo con las variables decaladas

Se han ajustado modelos PLS para diferentes valores de corte de VIP, desde 1 hasta 2, para ver el efecto en la bondad del ajuste. Los resultados son los siguientes:

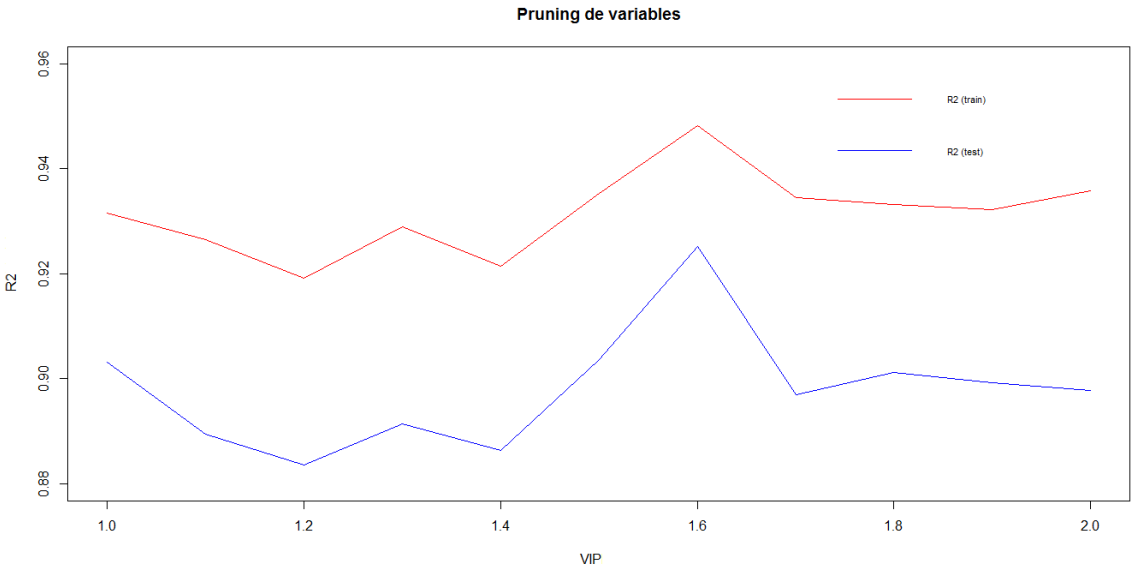


Figura 5.10.- Efecto de la poda de variables mediante el VIP en la bondad de ajuste del modelo PLS.

Se comprueba que el mejor modelo se obtiene para un VIP de corte de 1.6, que selecciona 62 variables (de un total de 912) y que mejora notablemente tanto los resultados del modelo completo como los del modelo con las variables originales, como puede verse en la Tabla 5.3.

Tabla 5.3.- Comparativa de los diferentes modelos PLS

Modelo	Nº var	R^2_{train}	R^2_{test}
Variables originales	82	0.8790	0.7000
Variables decaladas	912	0.9276	0.8526
Variables decaladas + pruning	62	0.9482	0.9252

5.7.5 Monitorización de procesos y diagnóstico de fallos

Para demostrar cómo se realiza el diagnóstico de fallos mediante PLS, se estudiarán los gráficos de control obtenidos ajustando el modelo anterior sobre conjunto de datos de validación (ver apartado 5.7.4). Primero se comprueba el gráfico SPE (Figura 5.11), que sirve para detectar observaciones atípicas, y después el gráfico T^2 de Hotelling (Figura 5.12), que identifica observaciones extremas.

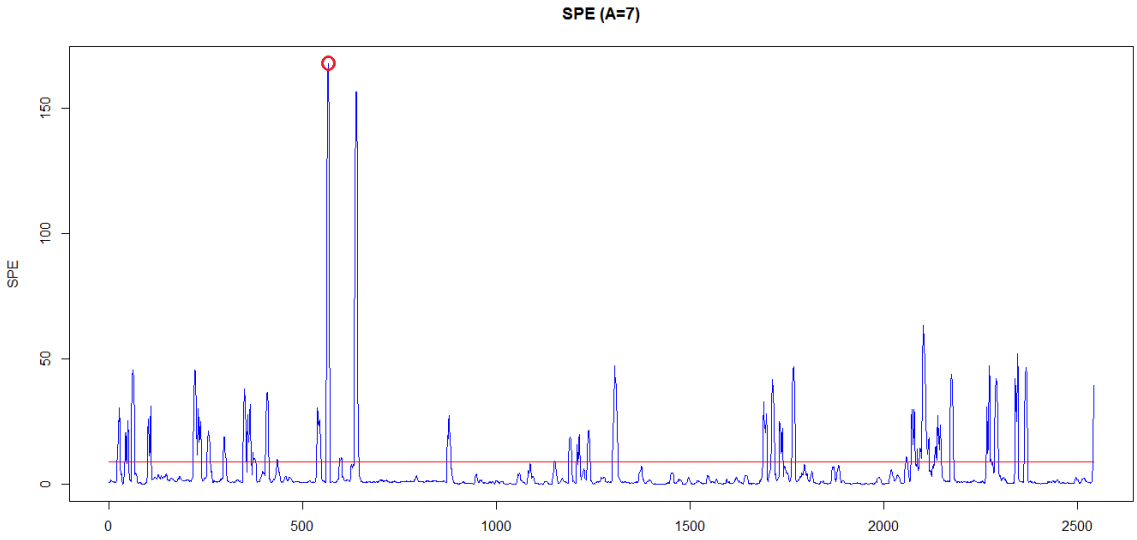


Figura 5.11.- Gráfico SPE para el modelo PLS con las variables decaladas (datos de validación).

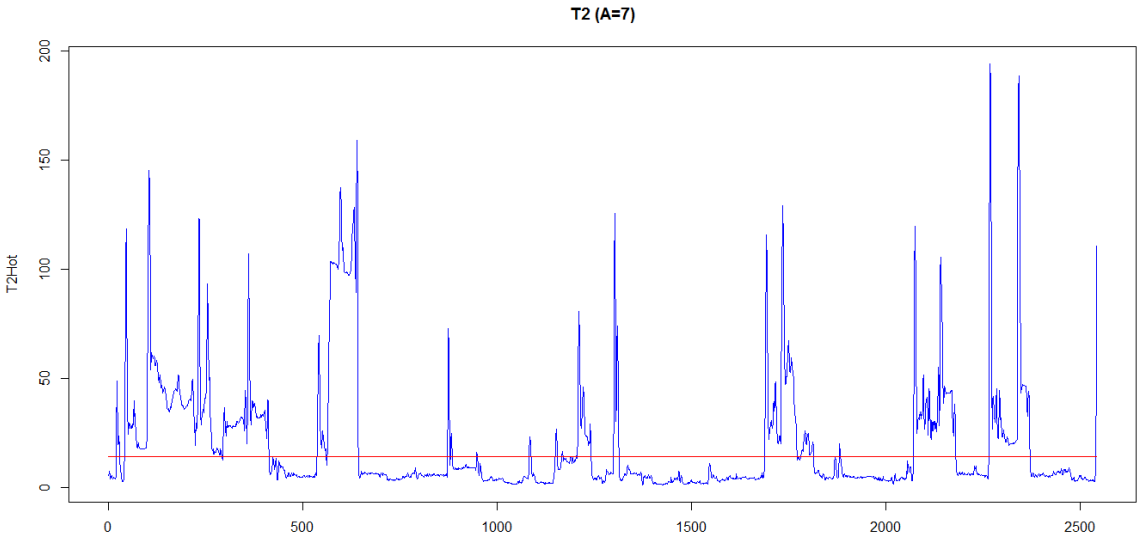


Figura 5.12.- Gráfico T^2 de Hotelling para el modelo PLS con las variables decaladas (datos de validación).

Como ejemplo se tomará la observación con mayor SPE (marcada mediante un círculo rojo en la Figura 5.11), que corresponde a la observación etiquetada como “13444”. En la Figura 5.13 se muestran las contribuciones al SPE para dicha observación. Se observa que la mayor contribución es debida a la variable “DENSIDAD_01” y a ella misma decalada en el tiempo. En

la Figura 5.14 puede verse el gráfico de la variable “DENSIDAD_01” para los datos de validación. La línea vertical roja indica la posición de la observación “13444”. Se comprueba cómo, en este punto, se produce un gran cambio en la variable “DENSIDAD_01”, que es responsable de la observación anómala detectada en el gráfico SPE.

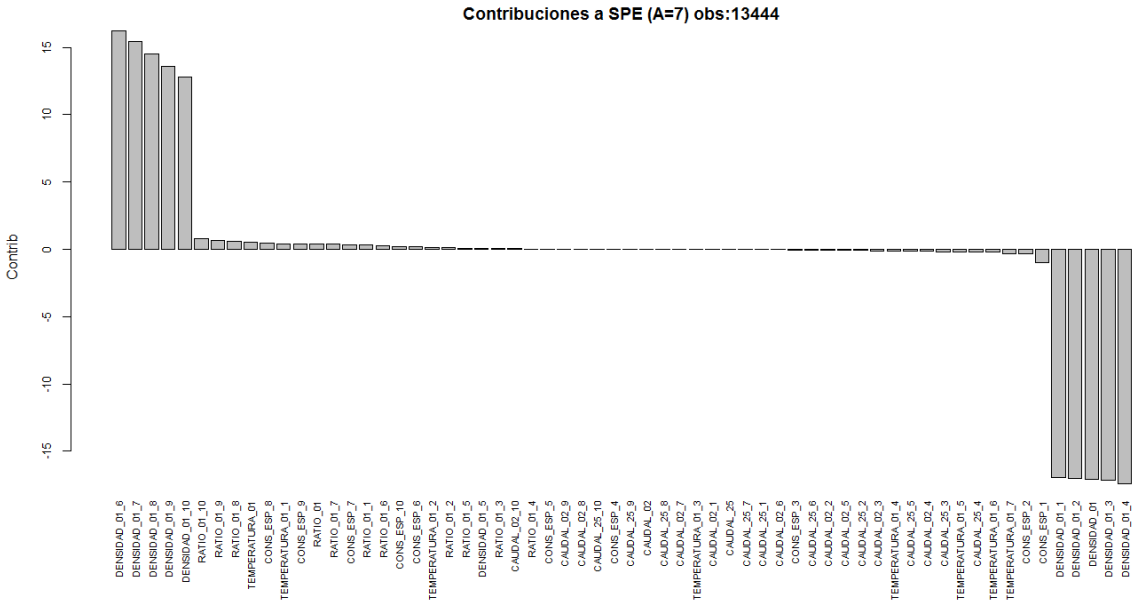


Figura 5.13.- Gráfico de contribuciones a SPE para la observación “13444”

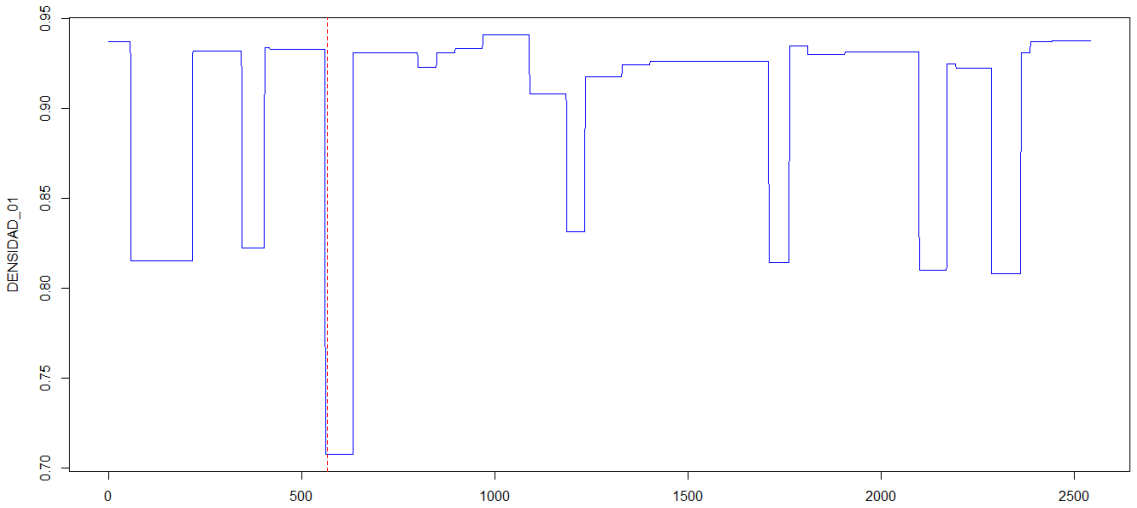


Figura 5.14.- Variable “DENSIDAD_01” (datos de validación)

5.8 Random forest

5.8.1 Introducción

En una primera parte se realizarán una serie de estudios previos:

- Se comprobará el efecto del tamaño del horizonte de predicción (ver apartado 4.1.9.2) sobre la bondad de ajuste del modelo, primero sobre las 82 variables originales y, posteriormente, sobre las variables decaladas.
- Se estudiará cómo afecta la selección de parámetros (*hyperparameter tuning*) y la poda de variables sobre la bondad de ajuste de los modelos.

Posteriormente se aplicará la metodología para la monitorización y diagnóstico de fallos comentada en el apartado 4.2.3.

5.8.2 Variables originales

Se ha entrenado un RF con las variables originales utilizando los parámetros por defecto. La bondad de ajuste para los datos de entrenamiento es R^2=0.9912 (Figura 5.15), mientras que para los datos de validación es de R^2=0.5724 (Figura 5.16).

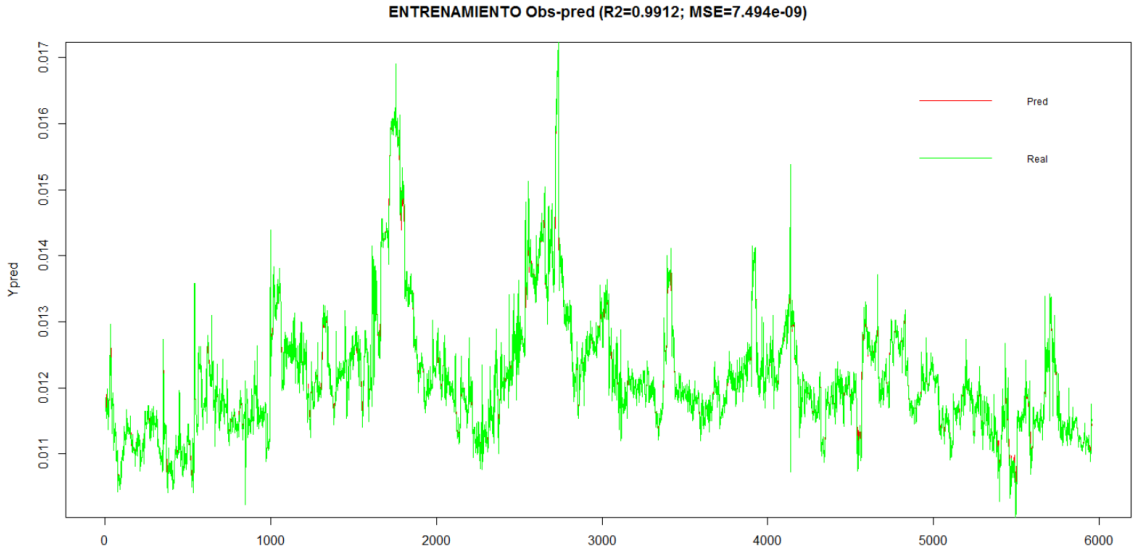


Figura 5.15.- Gráfico de observados frente a predichos para los datos de entrenamiento (variables originales).

Teniendo en cuenta que el proceso posee una cierta dinámica, se ha realizado un estudio del efecto del tamaño del horizonte de predicción (ventana de validación, ver apartado 4.1.9.2) sobre la bondad de ajuste del modelo a partir de las 82 variables originales. Para ello se han entrenado modelos con diferentes tamaños de ventanas de entrenamiento y diferentes tamaños de ventanas de predicción. Los resultados están en la Tabla 8.1. La librería utilizada ha sido "randomForestSRC" y el número de árboles utilizado en todos los casos ha sido de 1000. Se han utilizado todas las observaciones disponibles, que en este caso son de 8508.

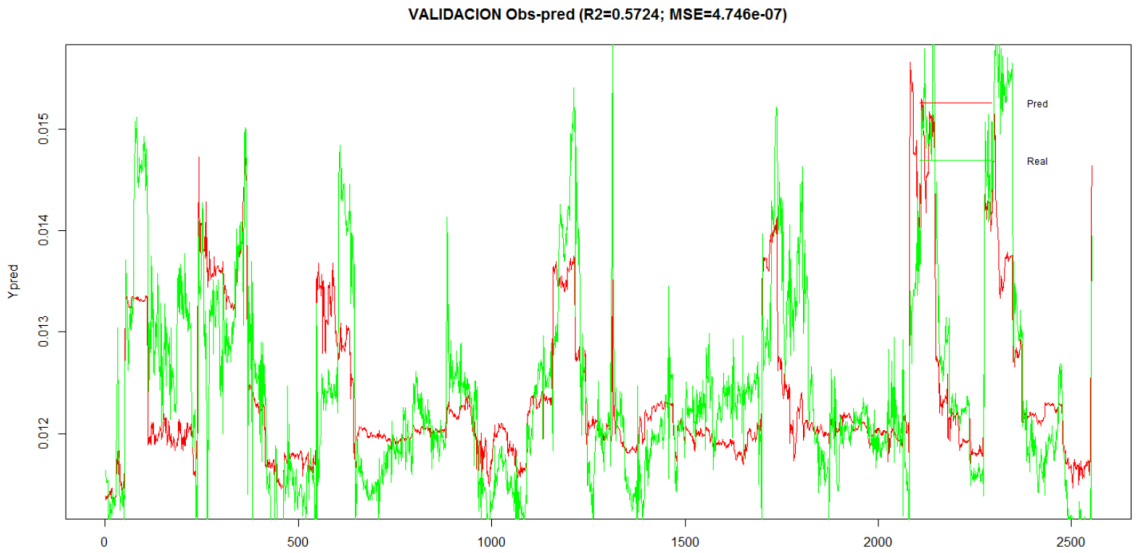


Figura 5.16.- Gráfico de observados frente a predichos para los datos de validación (variables originales).

Se han escogido diferentes tamaños para la ventana de entrenamiento (columna N_{train} de la Tabla 8.1): desde 500 hasta 6000 observaciones, con un intervalo de 500. El tamaño de la ventana de predicción se ha definido como un porcentaje del tamaño de la ventana de entrenamiento (columna P_{pred}), de manera que el número de observaciones de la ventana de predicción es N_{train} x P_{pred} / 100. Concretamente se han probado estos porcentajes: 1%, 5%, 10%, 20%, 30% y 40%. Se ha escogido este límite en base a la proporción típica de la validación de tipo *hold out*, en la que se usa un 70% de los datos para entrenamiento y un 30% para validación. Ello supone que el tamaño del conjunto de validación es 30*100/70 = 43% respecto al de entrenamiento. Finalmente, también se ha calculado la bondad de ajuste sobre los datos de validación utilizados hasta ahora, que son el 30% de los datos disponibles.

Todos los datos se han seleccionado de manera consecutiva. Así por ejemplo, si N_{train} = 500 y P_{pred} = 10%, entonces las primeras 500 observaciones serán la ventana de entrenamiento y las siguientes 500 x 10 / 100 = 50 observaciones serán la ventana de predicción. Respecto a los datos de validación, siempre son los mismos: las últimas observaciones equivalentes al 30% de los datos disponibles.

El parámetro de bondad de ajuste calculado ha sido el MSE (*Mean Squared Error*). En la Tabla 8.1, el sufijo “train” hace referencia a la ventana de entrenamiento, el sufijo “pred” hace referencia a la ventana de predicción y el sufijo “test” hace referencia al 30% de las observaciones totales (datos de validación).

Para comprobar el efecto de estos factores se ha realizado un ANOVA, tomando como variable a estudio el MSE_{pred} y, como factores, el tamaño del conjunto de entrenamiento (N_{train}) y el tamaño de la ventana de predicción (P_{test}). Los resultados (Figura 5.17) indican que únicamente el tamaño del conjunto de entrenamiento (N_{train}) es estadísticamente significativo con un nivel de confianza del 95% (p-valor <= 0.05). Es decir, el tamaño del conjunto de entrenamiento (N_{train}) influye sobre la media de MSE_{pred}.

```
Model:
MSEpred ~ Ntrain + Ppred
       Df  Sum of Sq        RSS     AIC F value   Pr(>F)
<none>                 7.3234e-12 -2120.0
Ntrain 11 5.9512e-12 1.3275e-11 -2099.2  4.0631 0.000229 ***
Ppred   5 1.3703e-12 8.6938e-12 -2117.7  2.0583 0.084674 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 5.17.- Resultados del ANOVA (variables originales)

5.8.3 Variables decaladas

Se ha entrenado un RF con las variables originales utilizando los parámetros por defecto. La bondad de ajuste para los datos de entrenamiento es R^2=0.9902 (Figura 5.18), mientras que para los datos de validación es de R^2=0.8787 (Figura 5.19).

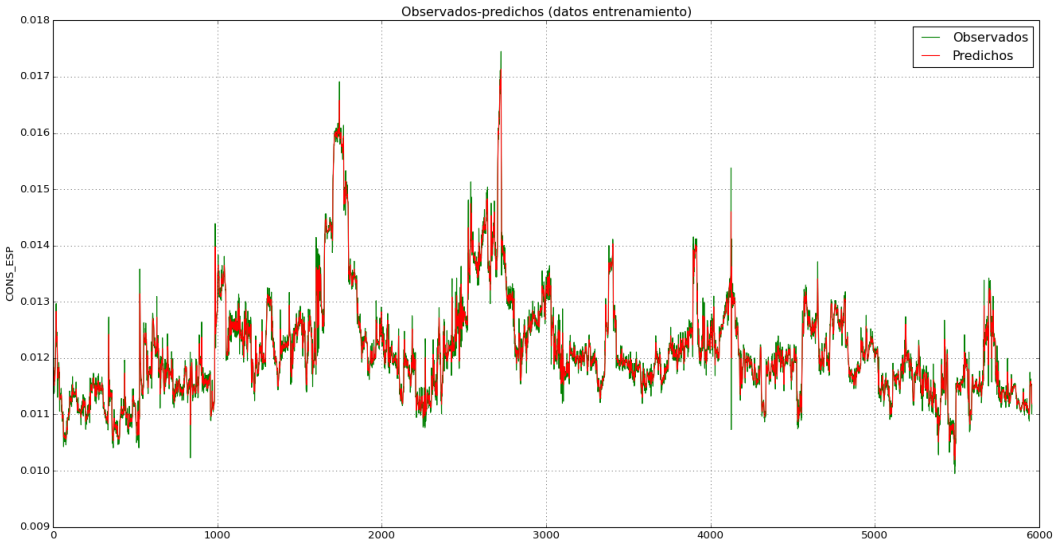


Figura 5.18.- Gráfico de observados frente a predichos para los datos de entrenamiento (variables originales).

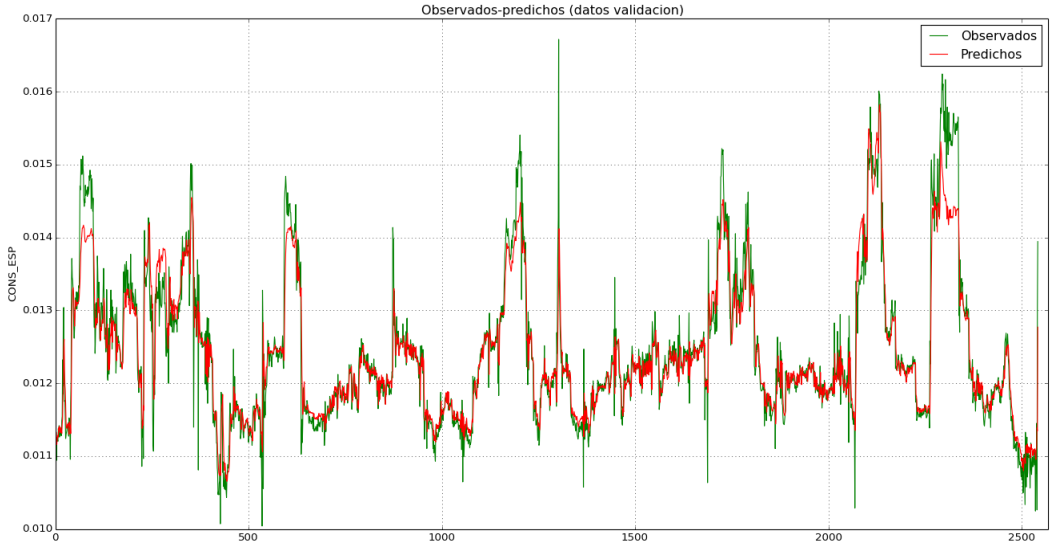


Figura 5.19.- Gráfico de observados frente a predichos para los datos de validación (variables originales).

Se realiza un estudio similar al anterior pero sobre la matriz ampliada con variables decaladas hasta 10 retardos, por lo que se dispondrá de un total de 912 variables. Los resultados están en la Tabla 8.2. En este caso, los errores disminuyen considerablemente respecto al caso anterior: a partir de un tamaño de la ventana de entrenamiento de más de 2000 observaciones los resultados del MSE_{valid} están entorno al 1E-07 (mientras que para el caso de utilizar variables originales los valores eran del orden de 4E-07).

Por otro lado, también se ha realizado un ANOVA. Al igual que en el caso anterior, únicamente el tamaño del conjunto de entrenamiento (N_{train}) es estadísticamente significativo con un nivel de confianza del 95% (p-valor <= 0.05) sobre la media de MSE_{pred} (Figura 5.20).

```
Model:
MSEpred ~ Ntrain + Ppred
       Df  Sum of Sq        RSS     AIC F value    Pr(>F)
<none>                6.1302e-12 -2132.8
Ntrain 11 4.8838e-12 1.1014e-11 -2112.6  3.9834 0.0002791 ***
Ppred   5 4.9190e-13 6.6222e-12 -2137.2  0.8827 0.4989131
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 5.20.- Resultados del ANOVA (variables decaladas)

5.8.4 Ajuste de parámetros

En el caso de los RF existen principalmente tres parámetros que pueden ajustarse durante la fase de entrenamiento del modelo (ver apartado 3.4):

- n_{tree}: número de árboles en el bosque.
- m_{try}: número de variables seleccionadas en cada división de los árboles.
- *Tamaño nodo*: número mínimo de observaciones en un nodo terminal.

La librería “randomForestSRC” propone 1000 como valor por defecto para el número de árboles. En cuanto al parámetro m_{try} propone los mismos valores que Breinman:

- Para casos de regresión, el valor por defecto es $p/3$
- Para el resto de casos, el valor por defecto es $\sqrt{p}$

Donde p es el número de variables.

Respecto al tamaño del nodo, los valores por defecto son: 1 para clasificación, 3 para análisis de supervivencia y 5 para regresión, aunque se recomienda experimentar con diferentes valores.

Teniendo en cuenta lo anterior, se realiza un estudio para comprobar el efecto de estos parámetros en la bondad de ajuste del RF. Con los datos NOC se entrenarán diferentes RF modificando los anteriores parámetros y se calcularán tanto la R^2 como el MSE de las predicciones sobre el conjunto de validación. Los valores seleccionados para los parámetros son los siguientes:

- n_{tree}: 100, 200, 500 y 1000
- m_{try}: $p/3$, $p/5$ y $p/10$
- *Tamaño nodo*: 1, 2, 3, 4, 5 y 6.

Los resultados se muestran en la Tabla 8.3. Los valores de R^2 oscilan entre 0.82224 y 0.8818 y los de MSE entre 1.31E-07 y 1.97E-07. Los mejores resultados se muestran en la Tabla 5.4. El mayor efecto en el ajuste está provocado por el número de variables.

Tabla 5.4.- Mejores resultados del ajuste de parámetros del RF.

n_{tree}	m_{try}	Tamaño nodo	R^2	MSE	Tiempo (min)
200	304	5	0.8818	1.31E-07	6.5
500	304	6	0.8815	1.32E-07	16.1
500	304	4	0.8809	1.32E-07	16.4
200	304	4	0.8803	1.33E-07	6.6
100	304	5	0.8799	1.33E-07	3.3
1000	304	3	0.8799	1.33E-07	33.5
1000	304	6	0.8786	1.35E-07	32.5
1000	304	1	0.8781	1.35E-07	34.1
500	304	1	0.8765	1.37E-07	17.1
500	304	5	0.8765	1.37E-07	16.3

Se ha realizado un ANOVA con los datos de la Tabla 8.3. La variable a estudio ha sido la R^2 y los factores han sido n_{tree} (número de árboles en el bosque), m_{try} (número de variables seleccionadas en cada división de los árboles) y *Tamaño nodo* (número mínimo de observaciones en un nodo terminal). Los resultados del ANOVA (Figura 5.21) muestran que tanto el factor m_{try} como el n_{tree} son estadísticamente significativos con un nivel de confianza del 95% (p-valor <= 0.05). Es decir, tanto el número de variables seleccionadas en cada división como el número de árboles influyen sobre la media de R^2.

```
Model:
R2 ~ ntree + mtry + node.size
          Df Sum of Sq       RSS     AIC  F value    Pr(>F)
<none>                 0.0011554 -772.88
ntree      3 0.0004820 0.0016375 -753.77   8.4827 8.541e-05 ***
mtry       2 0.0159395 0.0170950 -582.89 420.7560 < 2.2e-16 ***
node.size  5 0.0001187 0.0012741 -775.84   1.2533    0.2957
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figura 5.21.- Resultados del ANOVA (ajuste de parámetros)

Por otro lado, se observa que, cuando el número de árboles en el RF es bajo, los mejores resultados se obtienen aumentando la "complejidad" de los árboles (aumentando el número de nodos). Para valores altos del número de árboles, el tamaño del nodo no parece tener demasiado efecto. En cualquier caso, escogiendo el número de variables recomendado en el caso de la regresión ($m_{try} = p/3 = 304$, en este caso), se comprueba que el efecto de los parámetros no es demasiado importante y que los RF son poco sensibles por lo que se refiere al ajuste de parámetros, siempre dentro de unos límites razonables, a diferencia de otras técnicas de aprendizaje automático como las SVM o las Redes Neuronales.

5.8.5 Selección de variables

Aprovechando la capacidad de los RF para calcular la importancia de las variables (ver apartado 3.4), se realizará un estudio sobre *variable pruning* (poda de variables). Este término hace referencia a la obtención de un segundo modelo utilizando únicamente las variables más importantes obtenidas en un primer análisis. Este tema es especialmente importante, sobre todo en el caso de trabajar con variables decaladas, ya que el número de variables crece considerablemente y sólo unas pocas de ellas pueden ser importantes de cara a la modelización.

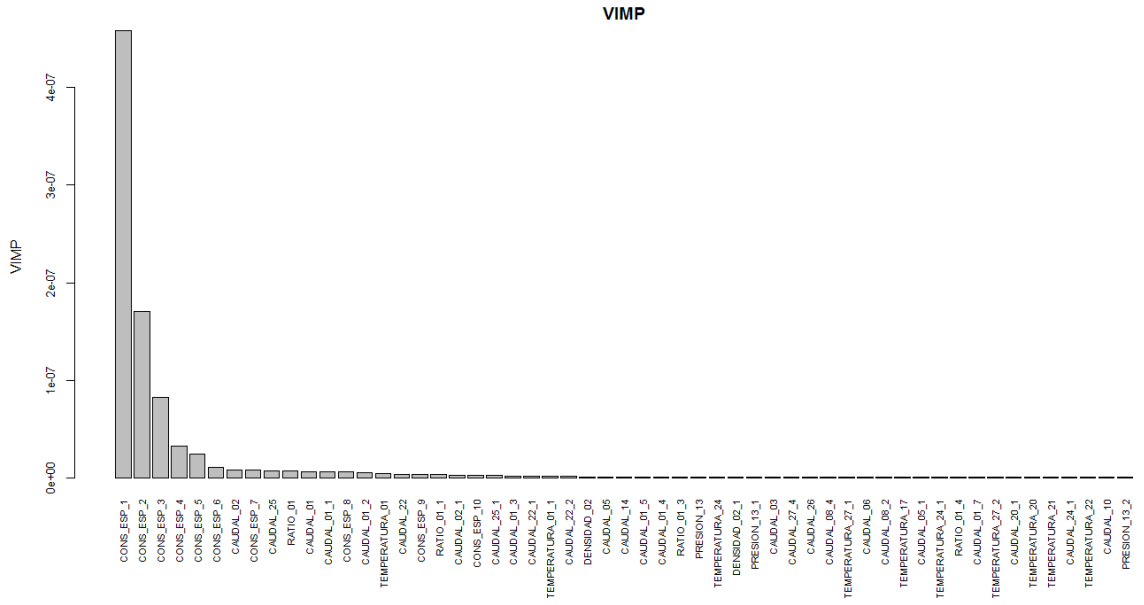


Figura 5.22.- VIMP de las primeras 55 variables

El primer paso es crear un RF sobre el conjunto de datos de entrenamiento con todas las variables y calcular su importancia. Para este estudio se utilizarán las variables decaladas. En este tipo de estudios normalmente se dibuja el denominado *Variable Importance Plot* o VIMP, que consiste en un diagrama de barras en el que se muestran las variables ordenadas decrecientemente según su importancia. En este caso, debido al elevado número de variables se realiza un filtro previo. Se comprueba que la importancia de la primera variable supera en un factor de 1000 al resto de variables a partir de la variable 55, por lo que se reduce el gráfico a estas primeras 55 variables (Figura 5.22).

El proceso posee una gran inercia, dado que las primeras 7 variables corresponden a la propia variable respuesta (“CONS_ESP”) decalada hasta 7 retardos. El peso de estas variables es muy alto comparado con el resto. Para poder apreciar el punto de corte seleccionado en el gráfico, se dibuja un VIMP con las primeras 200 variables (excluyendo las 7 primeras variables). El punto de corte se ha dibujado mediante una línea horizontal roja (Figura 5.23) que indica el valor de la importancia de la primera variable (“CONS_ESP_1”) dividido por 1000. A partir de este valor, la importancia de las variables es prácticamente despreciable. Este es el criterio que se ha seguido para seleccionar las variables que se usarán posteriormente en el estudio. El punto de corte corresponde a un conjunto de 55 variables.

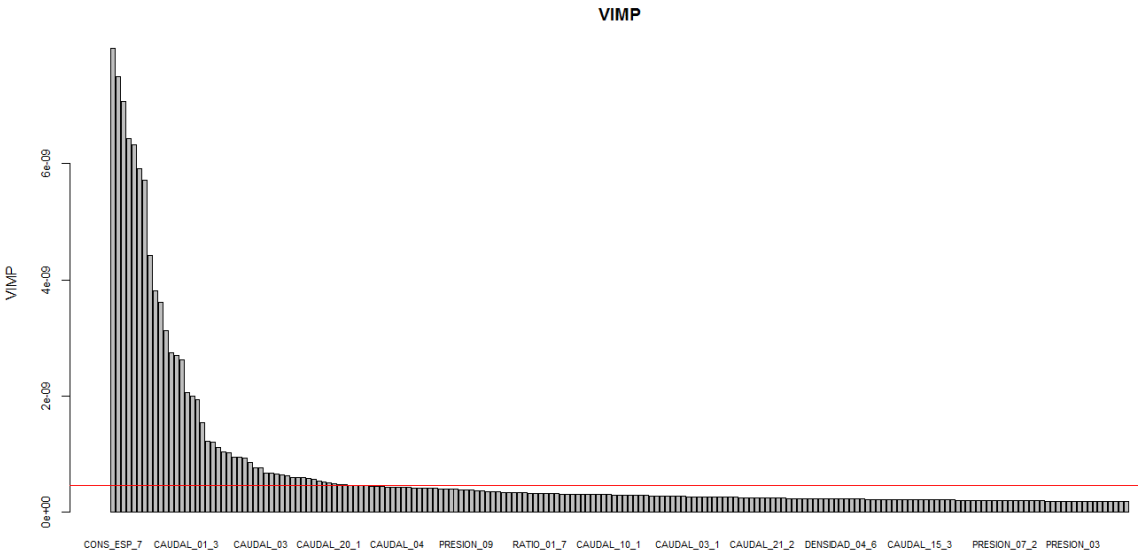


Figura 5.23.- VIMP de las primeras 300 variables.

Por otro lado, se ha entrenado un RF con las primeras n_{VIMP} variables para ver el efecto en las predicciones del conjunto de validación. Se han seleccionado diferentes valores para n_{VIMP}. Los resultados se muestran en la Tabla 5.5. Puede comprobarse cómo la bondad del ajuste aumenta con el número de variables hasta llegar a un máximo (en torno a 10 variables), a partir del cual empieza a disminuir. A modo informativo, la tabla también muestra el tiempo necesario para entrenar el modelo (t_{train}, en minutos) y para realizar las predicciones (t_{valid}, en segundos). La mayoría de los resultados están entre 0.88 y 0.90, que son bondades de ajuste muy altas.

Tabla 5.5.- Ajuste de RF con diferente número de variables

n_{VIMP}	R^2_{train}	MSE_{train}	t_{train} (min)	R^2_{valid}	MSE_{valid}	t_{valid} (s)
1	0.9791	1.775E-08	2.55	0.8566	1.593E-07	0.4479
2	0.9883	9.919E-09	2.69	0.8972	1.141E-07	0.4349
3	0.9889	9.387E-09	2.64	0.9016	1.092E-07	0.4649
4	0.9889	9.372E-09	2.68	0.9035	1.071E-07	0.4338
5	0.9890	9.334E-09	2.72	0.9030	1.077E-07	0.4384
6	0.9891	9.247E-09	2.67	0.9012	1.097E-07	0.4223
7	0.9890	9.298E-09	2.98	0.9027	1.081E-07	0.4247
8	0.9894	8.953E-09	3.01	0.9045	1.060E-07	0.4224
9	0.9897	8.693E-09	3.00	0.9094	1.006E-07	0.4493
10	0.9897	8.708E-09	3.31	0.9102	9.967E-08	0.4176
20	0.9902	8.304E-09	4.29	0.9029	1.078E-07	0.4276
30	0.9903	8.226E-09	5.18	0.8999	1.111E-07	0.4396
40	0.9903	8.255E-09	6.43	0.9007	1.103E-07	0.5417
50	0.9902	8.281E-09	7.32	0.8982	1.131E-07	0.4323
100	0.9904	8.130E-09	12.71	0.8901	1.220E-07	0.4149
150	0.9904	8.097E-09	17.92	0.8853	1.273E-07	0.4153
200	0.9905	8.095E-09	23.15	0.8871	1.253E-07	0.4232
250	0.9904	8.120E-09	28.47	0.8843	1.284E-07	0.4248
300	0.9903	8.189E-09	33.64	0.8833	1.296E-07	0.5382
912	0.9902	8.337E-09	97.41	0.8787	1.347E-07	0.4357

5.8.6 Monitorización de procesos

Siguiendo la metodología expuesta en el apartado 4.2.3, el primer paso es obtener los datos NOC. A partir de la matriz de similaridad **S** se ha calculado el grado de anomalía de cada una de las observaciones (ver apartado 4.1.6.3). En este caso, no se han obtenido observaciones con anomalías superiores a 10, por lo que todos los datos disponibles se considerarán datos NOC.

Paralelamente, se ha aplicado la técnica de escalamiento multidimensional (MDS) sobre la matriz de disimilaridad **D** (=1-**S**) para detectar posibles observaciones anómalas. La Figura 5.24 representa cada una de las observaciones proyectadas en las dos primeras dimensiones obtenidas mediante MDS. Si bien se observan zonas de mayor densidad que otras, no parece haber *clusters* de observaciones que estén claramente separados del resto, lo cual confirma que no existen observaciones anómalas.

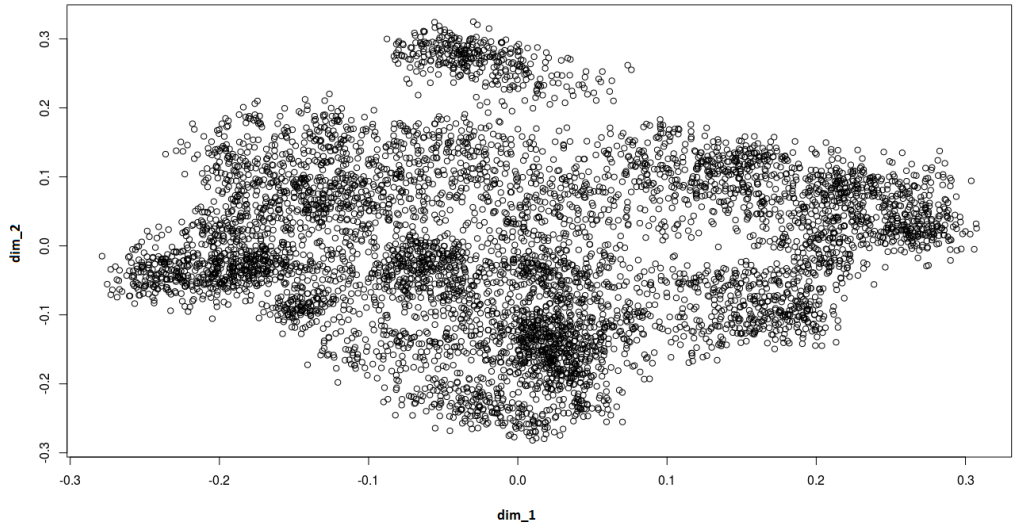


Figura 5.24.- Escalamiento multidimensional de la matriz de disimilaridad.

Lo siguiente es entrenar un RF de regresión (RFR) a partir de los datos NOC (que son todos los datos disponibles, dado que no se han encontrado observaciones anómalas). Podría realizarse un estudio para ajustar los parámetros del RF (como el que se ha hecho en el apartado 5.8.4) pero, como se ha visto, los RF suelen ser poco sensibles a la modificación de los parámetros, por lo que se usarán los parámetros por defecto.

Siempre es recomendable comprobar las variables más importantes, tanto para un mejor conocimiento del proceso como para mejorar la bondad del ajuste y simplificar el modelo. Como se ha visto (ver apartado 5.8.5), las 7 primeras variables más importantes corresponden a la propia variable respuesta decalada en el tiempo, lo cual permite obtener unas buenas predicciones pero no aporta nada de cara al diagnóstico de fallos. Por ello, se decide escoger las primeras 55 variables para disponer de más información a la hora de determinar las causas durante la etapa de diagnóstico de fallos (el criterio de corte para seleccionar estas 55 variables más importantes está explicado en el apartado 5.8.5). La capacidad predictiva del modelo con estas 55 variables está en torno a un 99% de la variabilidad total (R^2=0.9902) para los datos de entrenamiento (Figura 5.25). Este valor tan elevado suele ser indicador de que el modelo está sobreajustado. Los RF no suelen ser muy susceptibles al sobreajuste, aunque en este caso habría sido conveniente utilizar validación cruzada para estimar la capacidad

predictiva del modelo. De todas formas, como se verá más adelante, se dispone del conjunto de datos de validación para evaluar la capacidad predictiva (ver Figura 5.29).

Figura 5.25.- Gráfico observados frente a predichos (datos de entrenamiento)

Por otro lado, la poda de variables puede afectar a la capacidad de predicción de datos NOC. Esto se comprobará en el siguiente paso, que consiste en entrenar un RF de clasificación (RFC) con el conjunto de datos NOC etiquetados como clase 1 y el conjunto de datos sintéticos (obtenidos según lo comentado en el apartado 3.4.2) etiquetados como clase 2. Para entrenar este RFC se ha utilizado las 55 variables más importantes encontradas en la etapa de regresión (RFR). Al realizar un escalamiento multidimensional sobre la matriz de disimilaridad de los datos predichos mediante este RFC, se observa que el modelo obtenido separa perfectamente ambas clases (en la Figura 5.26, los puntos en negro corresponden a la clase 1 y los rojos, a la clase 2).

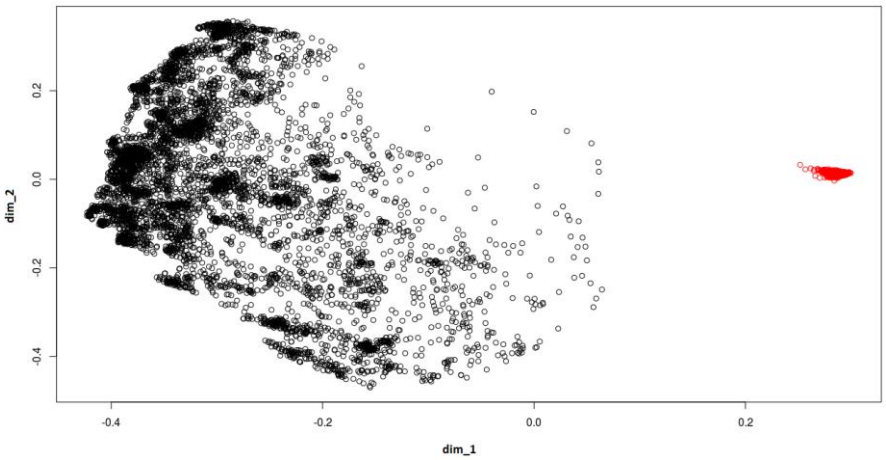


Figura 5.26.- Escalamiento multidimensional de la matriz de disimilaridad (clase1 y clase 2).

Otra forma de comprobar la capacidad predictiva del RFC obtenido es dibujando la probabilidad predicha sobre el conjunto de datos de entrenamiento. Las primeras 5951 observaciones corresponden a la clase 1 (datos NOC originales, no anómalos) y las 5951 restantes, a la clase 2 (datos sintéticos, anómalos). En la Figura 5.27 se ha dibujado la

probabilidad de que la observación sea considerada un dato anómalo, de manera que las probabilidades inferiores 0.5 (línea continua roja) corresponden a observaciones normales (no anómalas) y las superiores a 0.5 indican posibles observaciones anómalas. Puede comprobarse cómo los valores son bastante extremos, con lo que la separación entre los datos NOC y los datos anómalos es clara.

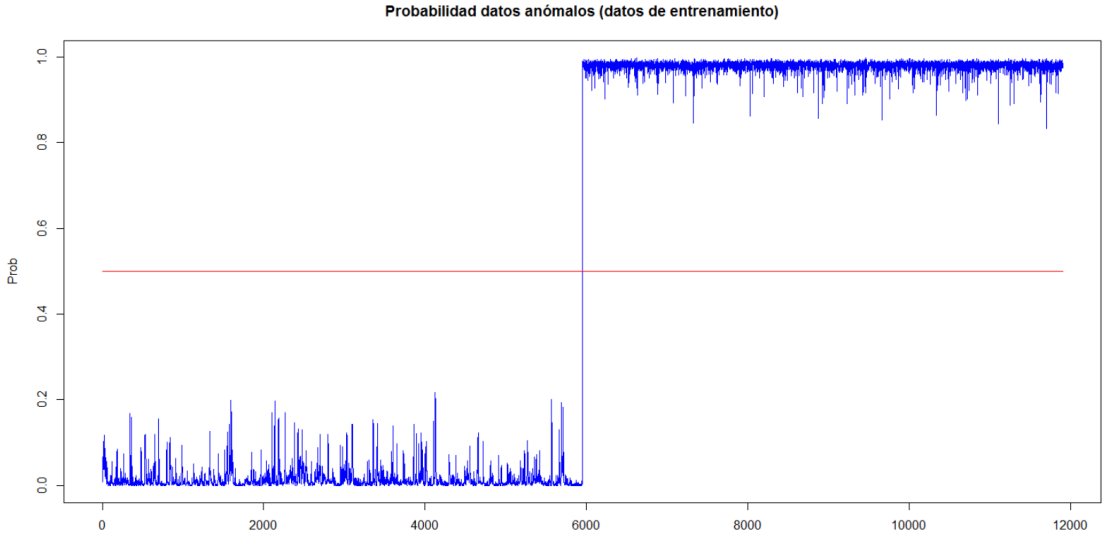


Figura 5.27.- Probabilidad de observaciones anómalas (datos de entrenamiento)

El siguiente paso es la explotación del modelo, es decir, la aplicación del modelo a las nuevas observaciones. En este caso, se aplicará el modelo al conjunto de datos de validación. En realidad se dispone de dos modelos:

- Un RF de clasificación (RFC) que calculará la probabilidad de que la nueva observación pertenezca a los datos NOC o bien se trate de una observación anómala.
- Otro RF de regresión (RFR) que calculará la predicción de la variable respuesta.

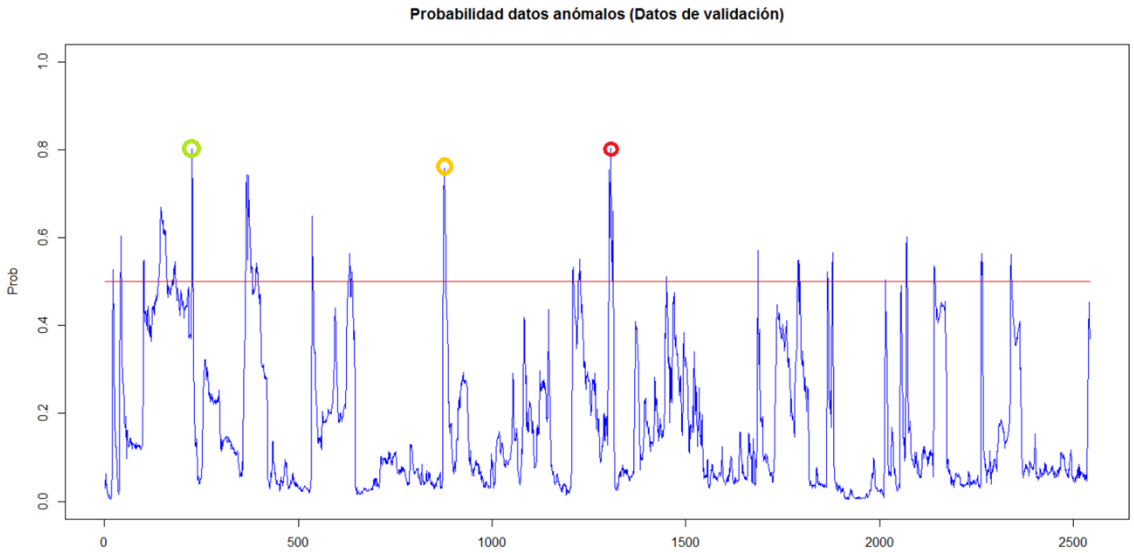


Figura 5.28.- Probabilidad de observaciones anómalas (datos de validación)

En la Figura 5.28 se muestra la probabilidad calculada mediante el RF de clasificación para cada una de las observaciones del conjunto de validación. La probabilidad dibujada corresponde a la clase 2, es decir, a las observaciones anómalas, por lo que valores superiores a 0.5 (línea

horizontal roja en la Figura 5.28) indican observaciones que pueden ser anómalas. Cuanto más cercana a 1 sea la probabilidad, mayor será el grado de anomalía.

La Figura 5.29 muestra el gráfico de valores observados frente a los valores predichos por el RFR para los datos de validación. En este caso, el modelo explica un 89% de la variabilidad total (R^2=0.8944).

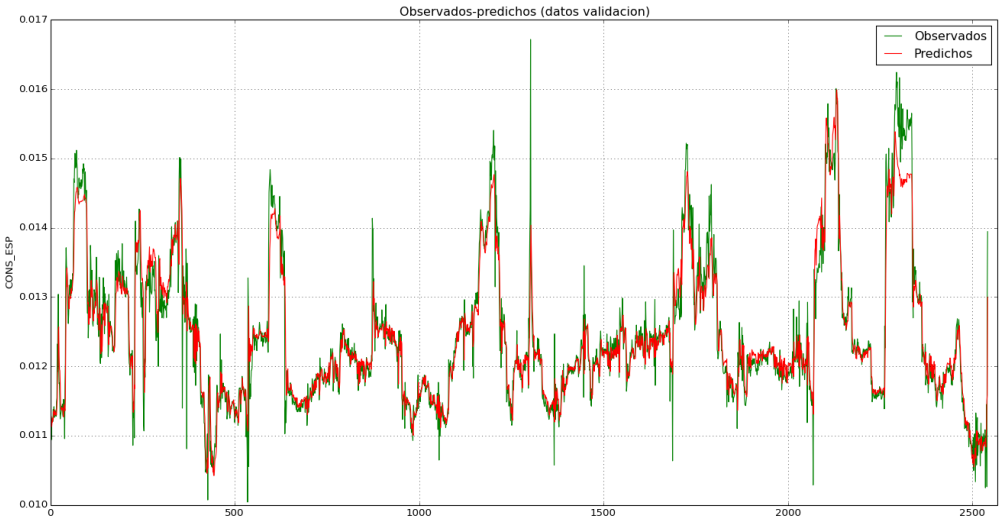


Figura 5.29.- Gráfico observados frente a predichos (datos de validación)

5.8.7 Diagnóstico de fallos

Cabe decir que el RF de clasificación se ha obtenido mediante la librería "randomForestSRC". Sin embargo, el modelo de regresión que se utilizará para el diagnóstico de fallos se ha entrenado con las librerías "sklearn" y "TreeInterpreter" desarrolladas en Python (ver apartado 3.4.3.2), que permiten calcular las contribuciones de cada variable lo que se utilizará para el diagnóstico de fallos.

A continuación se presentarán algunos ejemplos para comprobar la capacidad de diagnóstico de fallos de la metodología propuesta. Para ello se tomarán algunas de las observaciones con mayor probabilidad de ser anómalas (ver Tabla 5.6).

Tabla 5.6.- Observaciones con mayor probabilidad de ser anómalas

Observación	Probabilidad
14183	0.805
13104	0.802
14184	0.760
13754	0.759
14180	0.756
14185	0.748
13755	0.746
13245	0.744
13249	0.742
13244	0.708

Como primer ejemplo se tomará la observación con mayor probabilidad de ser anómala. Se trata de la observación "14183" que posee una probabilidad de anomalía de 0.805 (esta observación está marcada con un círculo rojo en la Figura 5.28).

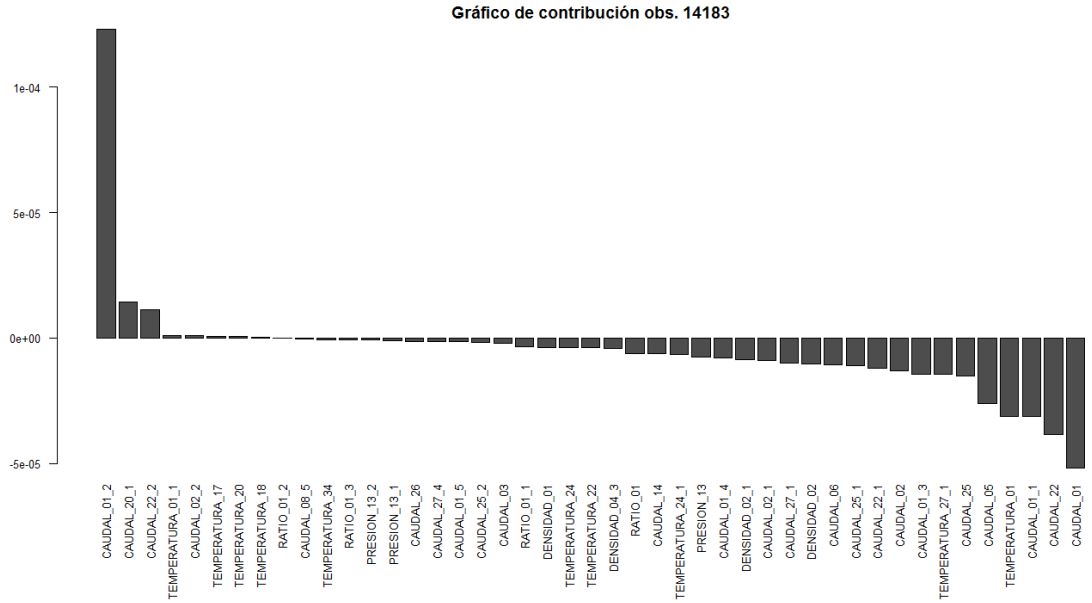


Figura 5.30.- Gráfico de contribución de la observación "14183" (sin la variable respuesta decalada)

En la Figura 5.30 se muestran las contribuciones para la observación "14183". Las contribuciones de la propia variable "CONS_ESP" decalada son altas y enmascaran al resto de las contribuciones, por lo que se han eliminado las contribuciones de estas variables. La mayor contribución corresponde a la variable "CAUDAL_01_2" (es decir, la variable "CAUDAL_01" decalada 2 horas), que está representada en la Figura 5.31. La línea vertical roja indica la posición de la observación "14183". Puede observarse cómo, en este punto, el valor de la variable "CAUDAL_01_2" sufre un gran cambio, responsable de la situación anómala detectada en la variable respuesta.

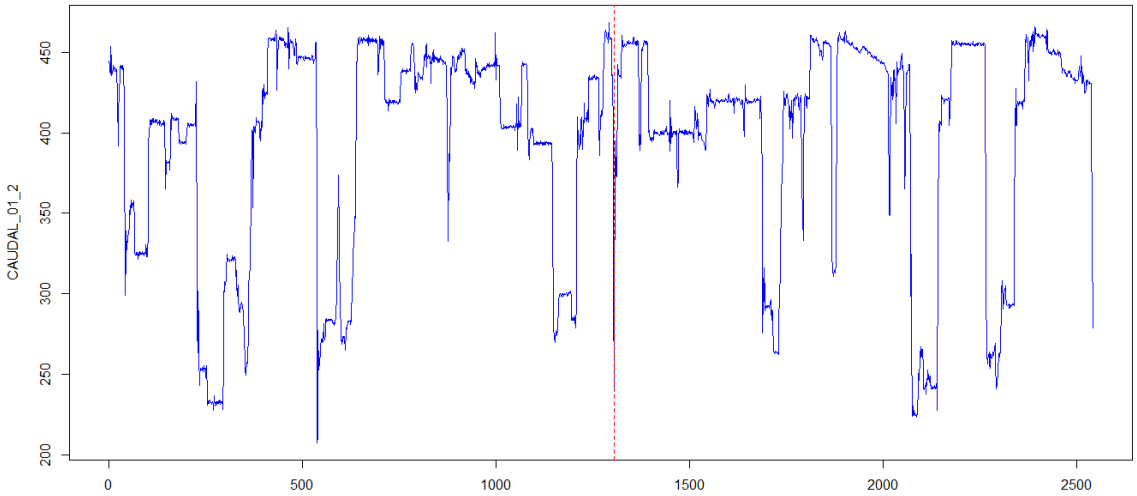


Figura 5.31.- Variable "CAUDAL_01_2" (datos de validación)

El siguiente ejemplo corresponde a la observación "13104" con una probabilidad de anomalía de 0.802 (marcada con un círculo verde en la Figura 5.28).

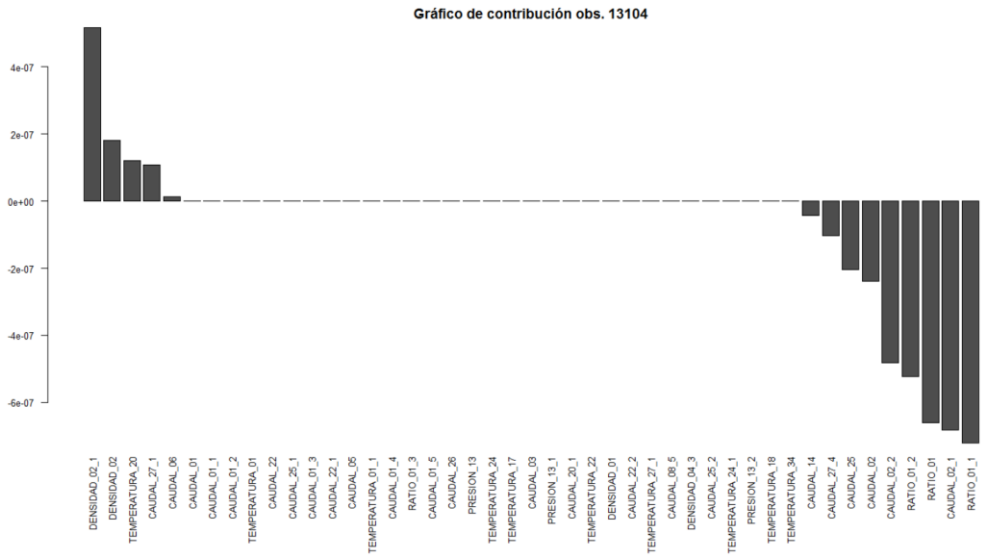


Figura 5.32.- Gráfico de contribución de la observación "13104" (sin la variable respuesta decalada)

En la Figura 5.32 se muestran las contribuciones para la observación "13104". Las mayores contribuciones corresponden a las variables "RATIO_01_1" y "CAUDAL_02_1", que están representadas en la Figura 5.33 y Figura 5.34, respectivamente. La línea vertical roja indica la posición de la observación "13104". Puede observarse cómo en este punto el valor de ambas variables sufre un gran cambio, responsable de la situación anómala.

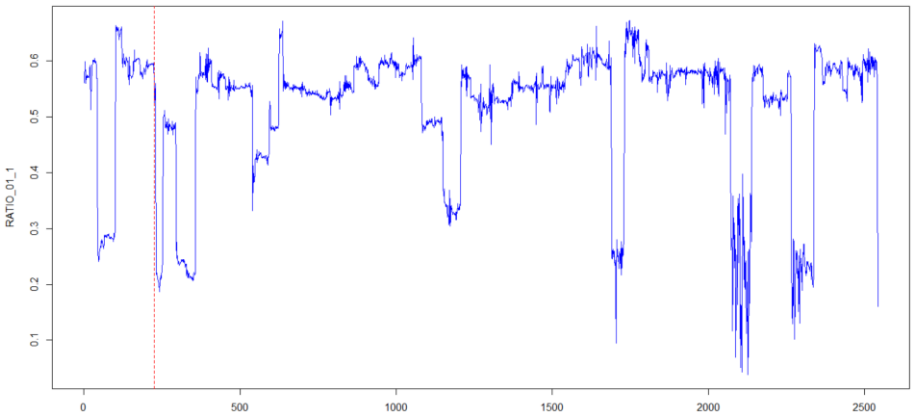


Figura 5.33.- Variable "RATIO_01_1" (datos de validación)

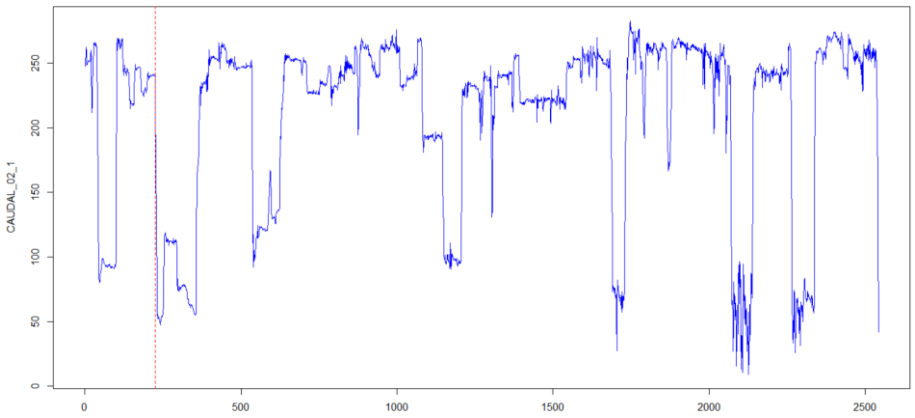


Figura 5.34.- Variable "CAUDAL_02_1" (datos de validación)

El último ejemplo seleccionado corresponde a la observación “13754” con una probabilidad de anomalía de 0.759 (marcada con un círculo naranja en la Figura 5.28).

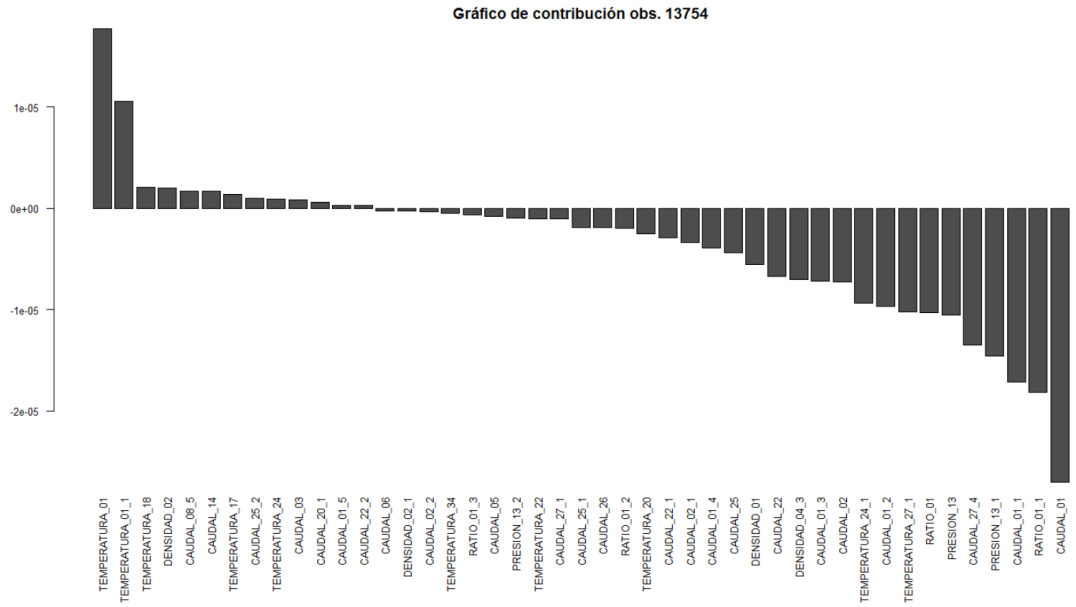


Figura 5.35.- Gráfico de contribución de la observación “13754” (sin la variable respuesta decalada)

En la Figura 5.35 se muestran las contribuciones para la observación “13754”. Las mayores contribuciones corresponden a las variables “CAUDAL_01”, “RATIO_01_1” y “TEMPERATURA_01”, que están representadas en la Figura 5.36, Figura 5.37 y Figura 5.38, respectivamente. La línea vertical roja indica la posición de la observación “13754”.En este caso tanto la variable “CAUDAL_01” como “TEMPERATURA_01” sufren cambios importantes alrededor de la observación a estudio. Sin embargo, en este caso, la otra variable detectada como importante (“RATIO_01_1”) aparentemente no presenta un comportamiento que parezca responsable de la situación anómala detectada en la observación “13754”.

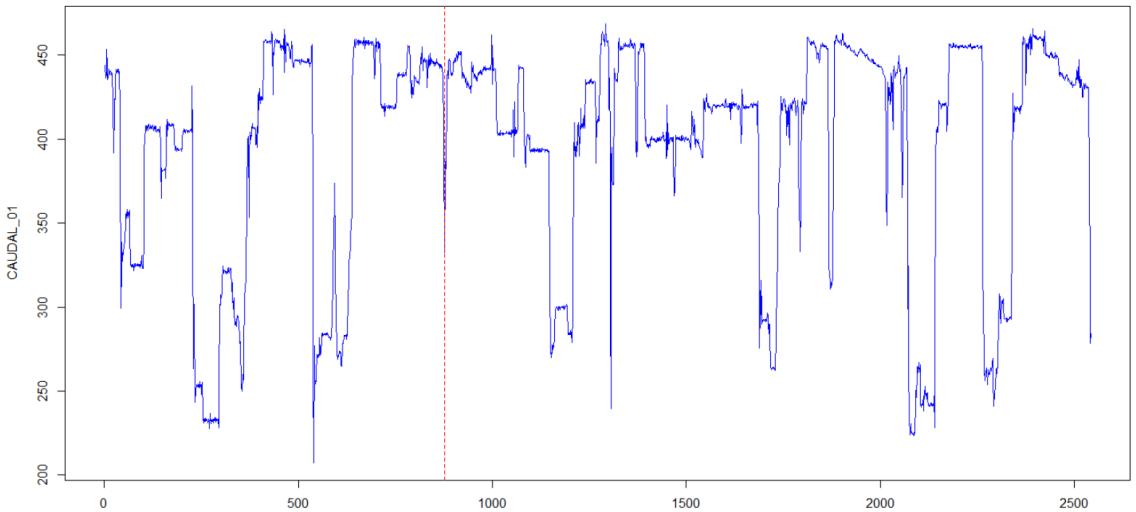


Figura 5.36.- Variable “CAUDAL _01” (datos de validación)

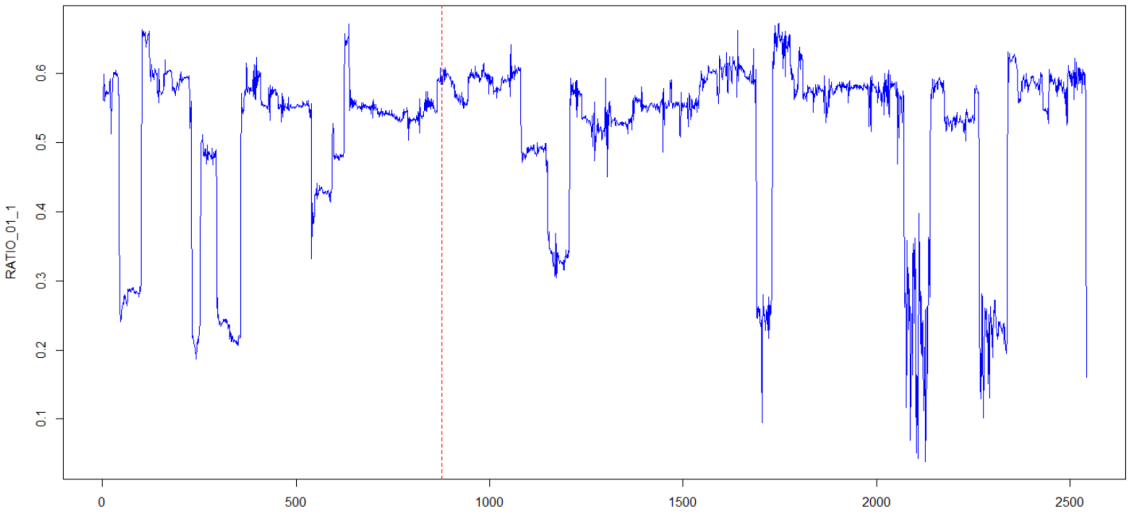


Figura 5.37.- Variable "RATIO _01_1" (datos de validación)

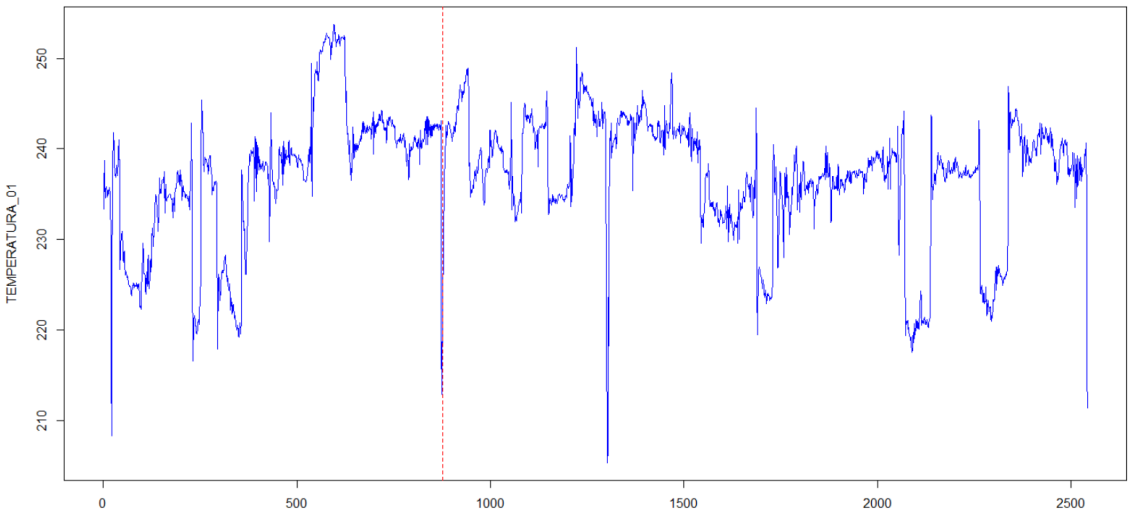


Figura 5.38.- Variable "TEMPERATURA _01" (datos de validación)

5.9 Support Vector Machines

5.9.1 Introducción

A diferencia de las anteriores técnicas (PLS y RF), las SVM no disponen de funcionalidades para el diagnóstico de fallos, por lo que únicamente se utilizarán para modelizar el proceso. Además, debido al alto coste computacional de la búsqueda de los parámetros óptimos, el estudio se aplicará únicamente a un caso, que será el de las variables decaladas hasta 10 retardos, dado que este caso ofrece un mejor ajuste como se ha visto con las otras técnicas.

5.9.2 Segmentación de datos

Se ha ajustado una *one-class* SVM utilizando como parámetro ν = 10/5956 (ver apartado 4.1.6.4). Es decir, se pretende "aislar" las 10 observaciones más anómalas de entre las 5956 observaciones que forman el conjunto de entrenamiento. Se han probado dos tipos de *kernel*: "radial" y "sigmoid". Este último parece tener más capacidad de separación, dado que el *kernel* "radial" devuelve un conjunto de 65 observaciones anómalas, mientras que el "sigmoid" devuelve 11.

En cualquier caso, si se compara este método con los vistos anteriormente, este método no resulta útil, dado que debe prefijarse un porcentaje de observaciones a eliminar mediante el parámetro ν. El principal problema es que este parámetro ν es desconocido, es decir, no existe un criterio preestablecido para determinar el límite a partir del cual una observación puede considerarse anómala. En el caso del PLS, se dispone de los límites de control (ver apartado 4.1.6.2) para determinar qué observaciones son anómalas y cuáles no. En el caso de los RF, se dispone de un límite para el grado de anomalía de la observación (ver apartado 4.1.6.3).

Dado que las otras técnicas no han detectado observaciones anómalas, en el caso de las SVM se considerará, también, que todos los datos son NOC.

5.9.3 Ajuste de parámetros

En el caso de las SVM, los parámetros a ajustar son (ver apartado 3.5):

- *C*: parámetro de penalización (*soft margin*).
- Parámetros propios del *kernel*.

Además, pueden existir parámetros adicionales dependiendo del método seleccionado para resolver el problema. En este trabajo, los problemas de clasificación se resolverán mediante el algoritmo "C-classification", que no utiliza parámetros adicionales. Para los problemas de regresión se utilizará el algoritmo "eps-regression" que utiliza el parámetro ε como parámetro de penalización de la regresión.

Respecto al *kernel* utilizado, en ambos casos se utilizará el *kernel* RBF (*Radial Basis Function*) para el que es necesario ajustar el parámetro γ (ver apartado 3.5.1).

Por otro lado, para afrontar el problema de la dimensionalidad (*curse of dimensionality*) típico de las SVM, se trabajará con las variables más importantes. Esto supondrá añadir un nuevo parámetro más al proceso de optimización, que será el número de variables a tener en cuenta por la SVM. Es decir, durante el proceso de optimización, aparte de los parámetros propios de la SVM, otro parámetro a tener en cuenta será el número de variables a seleccionar del conjunto de variables más importantes. Para realizar dicha selección se utilizará uno método de filtrado (ver apartado 4.1.7.2). Se han realizado pruebas con 3 métodos de filtrado: la ganancia de información, el test chi-cuadrado y el coeficiente de correlación de Pearson. En cada caso se dispone de una lista en la que las variables se encuentran ordenadas en orden descendiente según el valor del filtro en cuestión. De esta manera, cuando se fija un valor para el número de variables predictoras a seleccionar (n_{var}), en cada caso se tomarán las n_{var}

primeras variables de cada lista. Se han comparado los resultados del ajuste de varias SVM con los mismos parámetros utilizando los 3 métodos de filtrado y los mejores resultados se han obtenido siempre para el método de la ganancia de información, por lo que será el método utilizado finalmente para seleccionar las variables durante la optimización.

Respecto al tema de optimización, se utilizará la librería "mco"[16] de R, que resuelve el problema de optimización multiobjetivo mediante algoritmos genéticos. Los algoritmos genéticos están inspirados en la biología y se basan en cruzar, mutar y seleccionar miembros de una población (posibles soluciones) para minimizar una función objetivo a lo largo de diferentes generaciones (iteraciones). Existen numerosas variantes de algoritmos genéticos. En este caso se utilizará el algoritmo *nondominated sorting genetic algorithm II* (NSGA-II) (Deb et al. 2002).

5.9.4 Monitorización de procesos

Por motivos de tiempo y coste computacional, en el caso de las SVM los estudios se han realizado únicamente sobre el conjunto de variables decaladas.

Siguiendo la metodología expuesta en el apartado 4.2.4, una vez obtenidos los datos NOC, para seleccionar las variables más importantes, se ajustará una SVM de regresión (SVR) con los datos de entrenamiento. Para evaluar los modelos durante el proceso de optimización se utilizará una doble validación (ver apartado 4.1.9.1): los datos de entrenamiento se validarán mediante validación *k-fold* con *k*=5 (esto permitirá calcular R^2_{train}) y, por otro lado, se calculará también la R^2_{test} a partir del conjunto de datos externo. El proceso de optimización es multiobjetivo: se buscará el número de variables mínimo que ofrece el mayor ajuste (calculado como la suma de R^2_{train} y R^2_{test}).

Tabla 5.7.- Mejores resultados del proceso de optimización para la SVR

Nº Var	ε	$log(C)$	$log(\gamma)$	R^2_{train}	R^2_{test}
3	0,1660	2,020	-2,333	0,9391	0,9092
4	0,1756	1,465	-2,937	0,9390	0,9091
2	0,0438	-0,6369	-2,067	0,9383	0,9084
6	0,0360	2,010	-4,574	0,9385	0,9064
7	0,05783	0,9231	-4,939	0,9382	0,9064
10	0,07423	3,230	-4,608	0,9378	0,9062
5	0,06039	-0,1145	-4,818	0,9375	0,9052

Lamentablemente los resultados no han permitido continuar con la metodología inicialmente propuesta en el apartado 4.2.4. El problema es que, debido a la elevada autocorrelación de la variable respuesta "CONS_ESP", es suficiente con 2 o 3 variables (la propia variable respuesta "CONS_ESP" decalada en el tiempo) para tener los mejores ajustes (ver Tabla 5.7). Con estas pocas variables no podría obtenerse una buena SVM de clasificación. Para mantener el criterio

[16] The "mco" package, https://cran.r-project.org/web/packages/mco/mco.pdf

de entrenar ambos modelos (SVR y SVC) con el mismo conjunto de variables, se modificará la metodología propuesta en el apartado 4.2.4, de manera que se empezará ajustando la SVC y, con las variables óptimas obtenidas, se entrenará la SVR.

Así pues, se entrena una SVM de clasificación (SVC) a partir del conjunto de datos creados para trabajar en modo no supervisado (ver apartado 3.5.2). Se pretende optimizar la precisión a la hora de predecir si la observación es anómala o no. El número de variables será una de las variables de decisión durante este proceso de optimización, junto con los parámetros de la SVC. Después de 17 horas de optimización, se obtienen los siguientes resultados:

- Nºvariables = 46
- $C = 10^{2.912}$
- $\gamma = 10^{-3.432}$
- Precisión = 100%

En la Figura 5.39 se muestra la probabilidad calculada mediante la SVM de clasificación para cada una de las observaciones del conjunto de validación. La probabilidad dibujada indica si la observación es anómala o no, de manera que los valores superiores a 0.5 (línea horizontal roja) indican observaciones que pueden ser anómalas. Cuanto más cercana a 1 sea la probabilidad, mayor será el grado de anomalía.

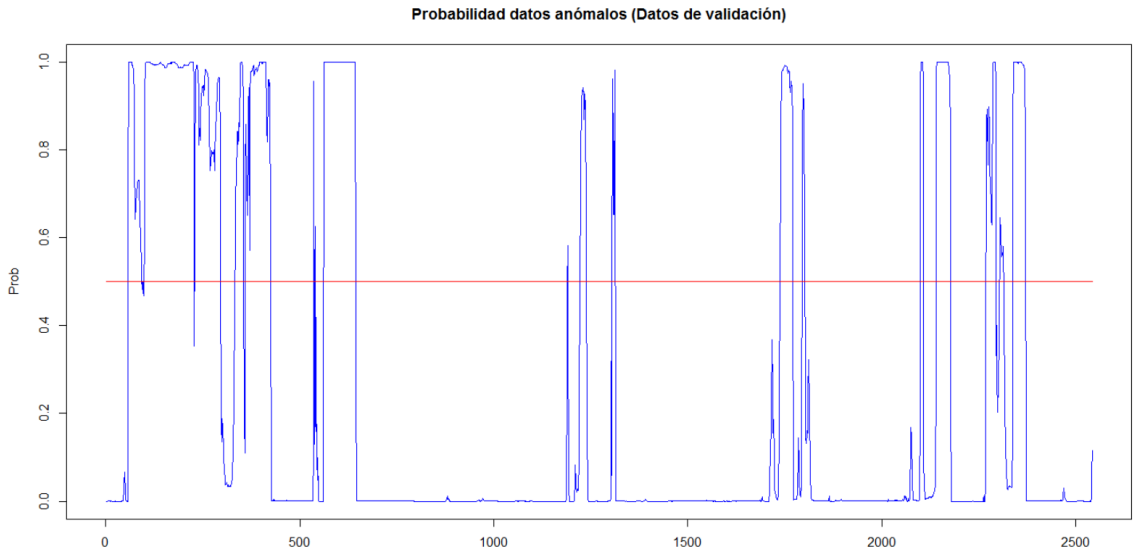


Figura 5.39.- Probabilidad observaciones anómalas (datos de validación)

El siguiente paso es entrenar una SVR con las 46 variables seleccionadas en la etapa anterior para la SVC. En este caso, el proceso de optimización será simple, buscando los parámetros de la SVR que proporcionen el mayor ajuste (calculado como la suma de R^2_{train} y R^2_{test}, como se ha comentado anteriormente). Después de 10 horas de optimización se obtienen los siguientes resultados:

- $\varepsilon = 0.06802$
- $C = 10^{0.2828}$
- $\gamma = 10^{-4.9547}$
- $R^2_{train} = 0.8855$ (validación cruzada)
- $R^2_{test} = 0.8731$

La Figura 5.40 muestra el gráfico de observados frente a predichos para los datos de entrenamiento. Estos datos no se han obtenido mediante validación cruzada por lo que, en este caso, R^2 = 0.9481, lo que confirma el sobreajuste típico de las SVM. La Figura 5.41, por otro lado, muestra el gráfico de observados frente a predichos para los datos de validación. En este caso, R^2_{test} = 0.8731, que es un valor más coherente con el que se ha obtenido mediante validación cruzada (R^2_{train} = 0.8855).

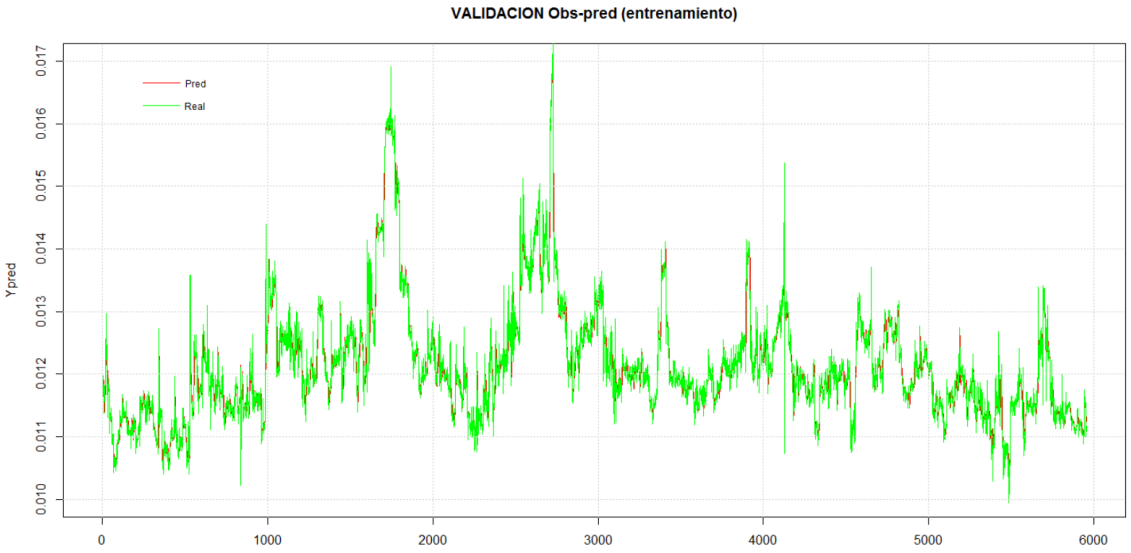


Figura 5.40.- Gráfico de observados frente a predichos para los datos de entrenamiento.

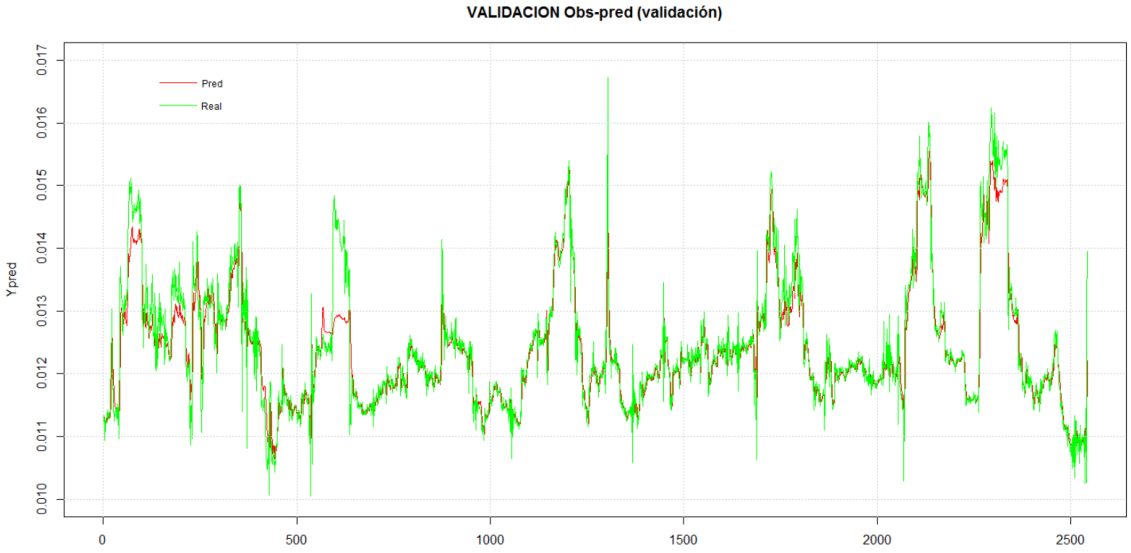


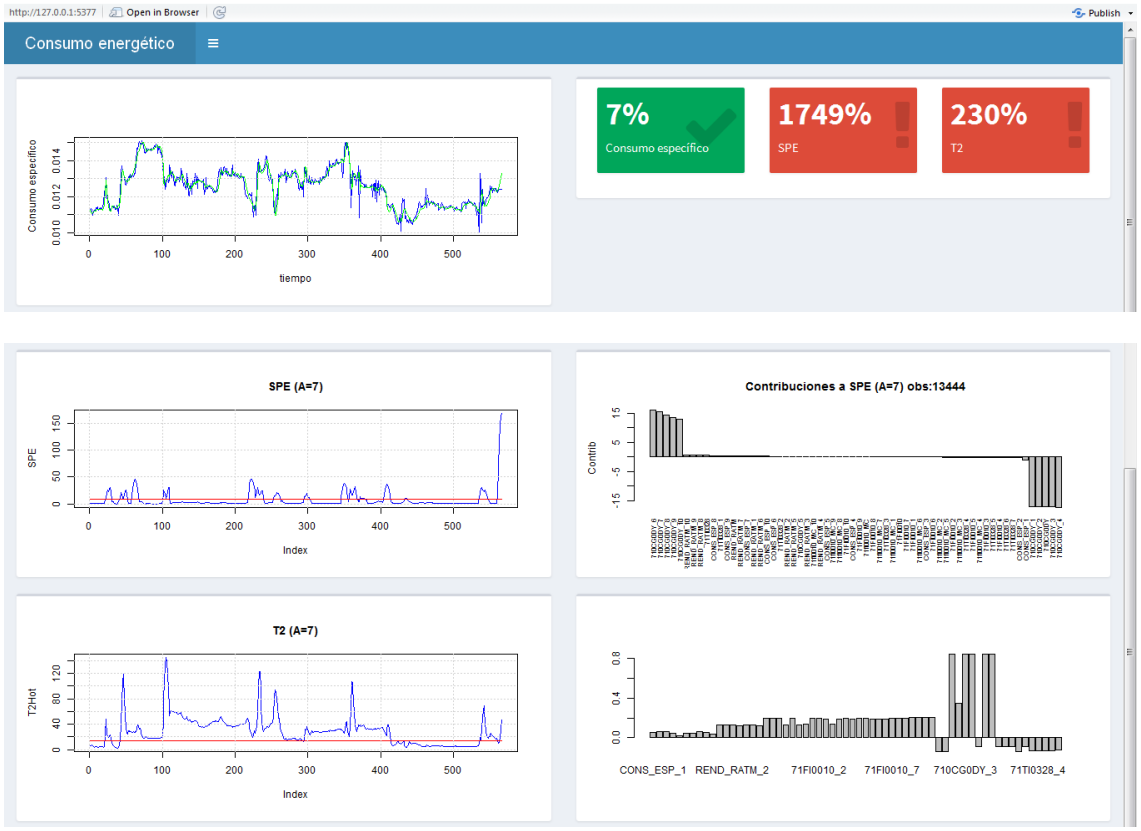
Figura 5.41.- Gráfico de observados frente a predichos para los datos de validación.

6 Cuadro de mando

Se ha creado un cuadro de mando utilizando *Shiny*[17], una librería de R creada en 2012 por Rstudio para desarrollar fácilmente aplicaciones web que permiten a los usuarios interactuar con sus datos sin tener que manipular el código. Para su uso no son necesarios conocimientos de HTML, CSS o JavaScript.

Sobre esta plataforma, se ha desarrollado otra librería denominada "Shiny Dashboard"[18] que está especialmente diseñada para la construcción de cuadros de mando basados en análisis de R de una manera muy sencilla.

Utilizando esta herramienta se ha creado el siguiente cuadro de mando a partir de los análisis obtenidos mediante la técnica de PLS, que es la que ha ofrecido mejores resultados (ver apartado 7 Resultados y conclusiones). El ejemplo que se muestra a continuación corresponde al comentado en el apartado 5.7.5.



La página web está dividida en 3 partes. La parte superior muestra la evolución del consumo específico en las últimas horas. En azul se muestra el valor real del indicador y en verde se muestra el valor predicho. A la derecha se muestran 3 cajas informativas. La primera indica la variación entre el valor real del indicador y el predicho. En este caso, el valor predicho es un

[17] Shiny, http://shiny.rstudio.com/
[18] ShinyDashboard, https://rstudio.github.io/shinydashboard/index.html

7% superior al real, por lo que la unidad está operando de manera mejor a lo esperado. Las siguientes cajas informativas muestran la variación de los estadísticos SPE y T^2 respecto a sus respectivos límites de control. En este caso, ambos estadísticos están por encima de sus respectivos límites, como puede comprobarse en los correspondientes gráficos de control. Primero se muestra el gráfico SPE. A la derecha de este gráfico se encuentran las contribuciones al SPE. En el caso de la T^2, las contribuciones mostradas corresponden al componente con mayor valor.

De esta forma el operador de la planta puede conocer si la eficiencia energética del horno está por encima o por debajo de la esperada en condiciones normales y, en caso de no ser así, saber cuáles son las variables que debería investigarse para conocer las causas de tal situación.

7 Resultados y conclusiones

En la Tabla 7.1 se muestra un resumen de la bondad de ajuste de las tres técnicas utilizadas. El PLS ofrece mejores resultados que el RF utilizando las variables originales. Seguramente esto es debido a la existencia de correlación entre variables, situación ante la cual el PLS proporciona mejores resultados. Estos datos no están disponibles para el SVM, debido al elevado coste computacional.

Al utilizar todas las variables decaladas, los resultados son ligeramente mejores para el caso de RF. Como se ha comentado, cuando se decalan las variables, no todas aportan información, por lo que es necesario realizar un *pruning* o poda de variables para seleccionar las más importantes, descartando aquellas que sólo aportan ruido. Cuando se realiza esta poda de variables, los datos son mejores nuevamente para el PLS.

Tabla 7.1.- Resumen de la bondad de ajuste de las diferentes técnicas

Método	Variables originales (nºvariables=82)		Variables decaladas (nºvariables=912)		Variables decaladas + pruning		
	R^2_{train}	R^2_{test}	R^2_{train}	R^2_{test}	n_{vars}	R^2_{train}	R^2_{test}
PLS	0.8790	0.7000	0.9276	0.8526	62	0.9482	0.9252
RF	0.9912	0.5724	0.9902	0.8787	55	0.9902	0.8944
SVM	-	-	-	-	46	0.9481	0.8731

En la Tabla 7.2 pueden verse los tiempos de entrenamiento para cada una de las técnicas. La técnica que ajusta el modelo más rápidamente es el PLS. El tiempo de entrenamiento de una SVM para 46 variables está alrededor de los 12 s, pero hay que tener en cuenta que es necesario obtener los parámetros mediante optimización, lo cual puede suponer varias horas. Esto es debido a que durante el proceso de optimización se prueban diferentes parámetros y para cada uno de ellos parámetros se evalúa la capacidad predictiva del modelo (esto es lo que puede suponer unos segundos). El inconveniente es que el proceso de optimización puede necesitar la evaluación de miles de conjuntos de parámetros hasta llegar a la solución óptima, por lo que el proceso global es muy costoso y lento.

Tabla 7.2.- Tiempos de entrenamiento para las diferentes técnicas

Método	Variables originales (nºvariables=82)		Variables decaladas (nºvariables=912)		Variables decaladas + pruning		
	t_{train}	t_{pred}	t_{train}	t_{pred}	n_{vars}	t_{train}	t_{pred}
PLS	4 s	< 1s	51 s	< 1s	62	2 s	< 1s
RF	9 min	< 1s	97 min	< 1s	55	8 min	< 1s
SVM	-		-	-	46	12 s	< 1s

La metodología propuesta en este trabajo se basa en dos fases: una fase de construcción del modelo y otra de explotación (*online*). La fase de construcción corresponde al entrenamiento del modelo, que se realiza de manera *offline*, por lo que el tiempo de entrenamiento no es un inconveniente. La fase de explotación corresponde a la predicción de nuevos datos a partir del

modelo obtenido en la etapa anterior. Esta fase se lleva a cabo de manera *online*. En este sentido hay que decir que las predicciones en todos los casos son del orden de décimas de segundo, por lo que cualquiera de las técnicas sería aplicable en modo *online* para el caso que se está estudiando, dado que la dinámica del proceso es muy lenta. Hay que tener en cuenta que, en el caso estudiado, el modo *online* implica predecir un valor de la variable respuesta cada hora.

La unidad de destilación de crudo estudiada trabaja en unas condiciones de operación relativamente poco cambiantes. El modelo obtenido será válido mientras la unidad no sufra cambios. Estos cambios pueden ser debidos a tareas de mantenimiento o limpieza de equipos o, por supuesto, obras de reingeniería que alteren el comportamiento de la unidad (por ejemplo para optimizar el consumo energético). En estos casos, sería necesario reajustar el modelo. Sin embargo, pueden darse casos en los que sea necesario obtener un modelo de manera más frecuente (esto se conoce como *data stream mining*). En estos casos en los que es necesario reajustar el modelo con mayor frecuencia, la técnica de PLS, debido a su bajo coste computacional (tiempos de entrenamiento bajos) sería más apta que el resto de técnicas estudiadas.

Respecto al ajuste de parámetros, la técnica de RF se ha mostrado poco sensible a este tema y los parámetros propuestos por defecto en los algoritmos utilizados han dado buenos resultados. Sin embargo, para las SVM este tema es fundamental y los resultados pueden variar completamente según los parámetros seleccionados, por lo que es necesaria una etapa de optimización previa que seleccione los parámetros que proporcionen un mejor ajuste.

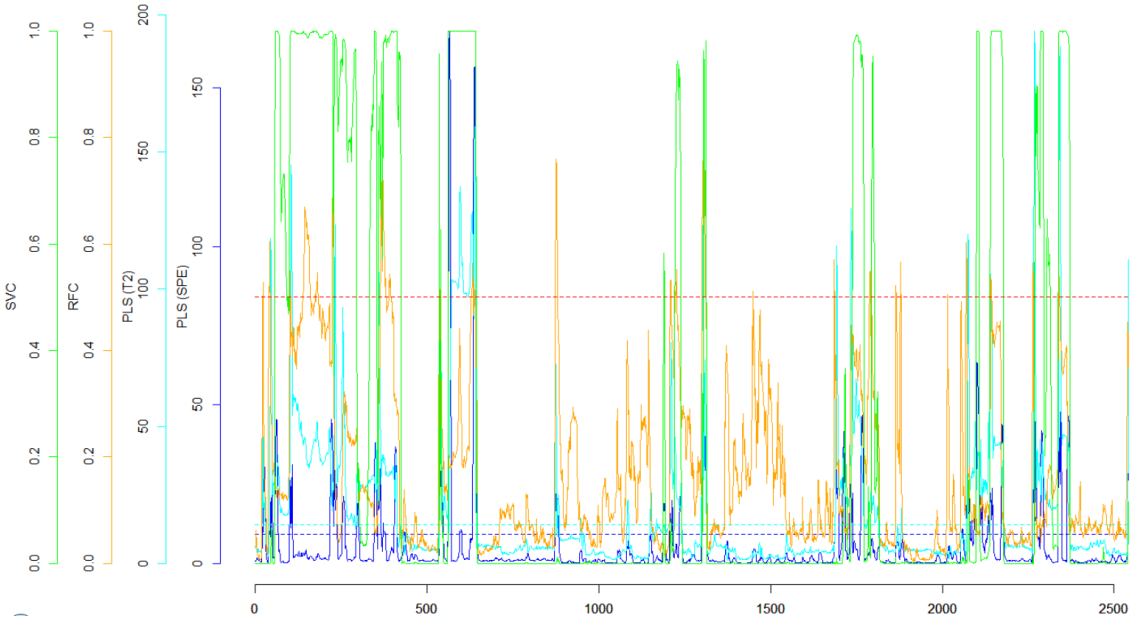


Figura 7.1.- Detección de observaciones anómalas (comparación entre PLS, RFC y SVC)

Otro tema estudiado ha sido la detección de observaciones anómalas. En la Figura 7.1 se han superpuesto los resultados de los estadísticos SPE (color azul oscuro) y T^2 de Hotelling (color azul claro) del PLS (ver apartado 5.7.5), los de la clasificación del RFC (ver apartado 5.8.6) en color naranja, y los del SVC (ver apartado 5.9.4), en color verde, obtenidos todos ellos a partir de los datos de validación. Cada resultado posee su propio color, eje y límite de control

superior (a partir del cual se considera que la observación es anómala). Este límite de control está representado mediante una línea horizontal discontinua del mismo color que el correspondiente resultado, excepto en el caso del RFC y del SVC, para los que el límite se muestra en color rojo en ambos casos. Para mayor legibilidad se han ampliado algunas zonas del gráfico (Figura 7.2 y Figura 7.3). Se tomará como referencia los resultados del PLS, dado que es una técnica ampliamente utilizada y de eficiencia probada para este tipo de casos. Así, como regla general puede concluirse que la clasificación obtenida mediante SVM es capaz de detectar mejor las zonas de observaciones anómalas que el RF.

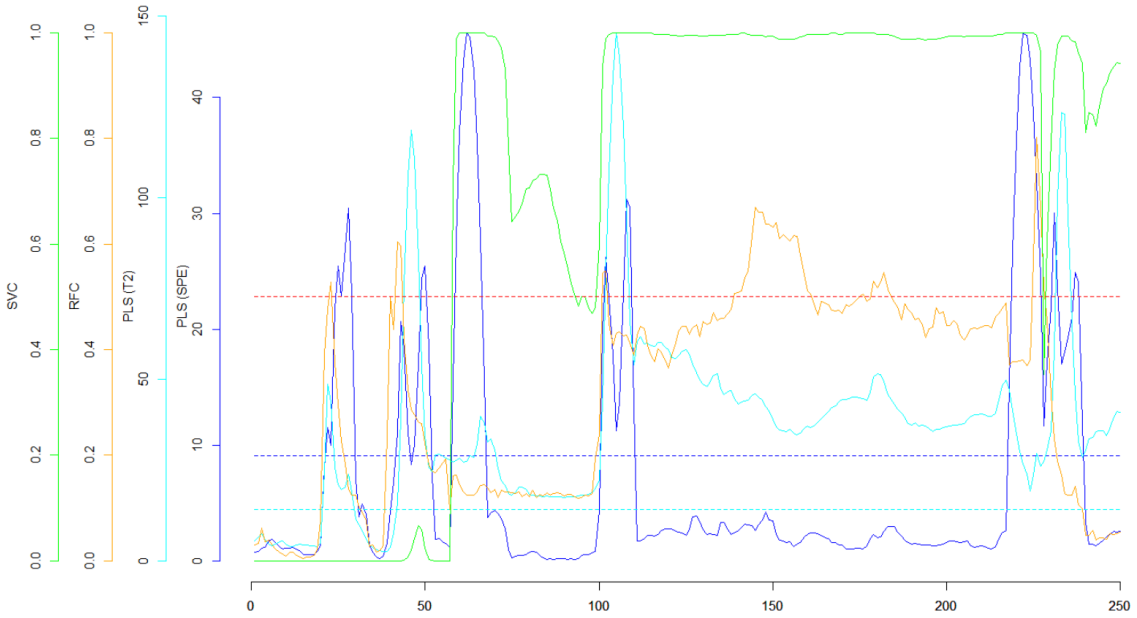


Figura 7.2.- Detección de observaciones anómalas. Observaciones 1-250

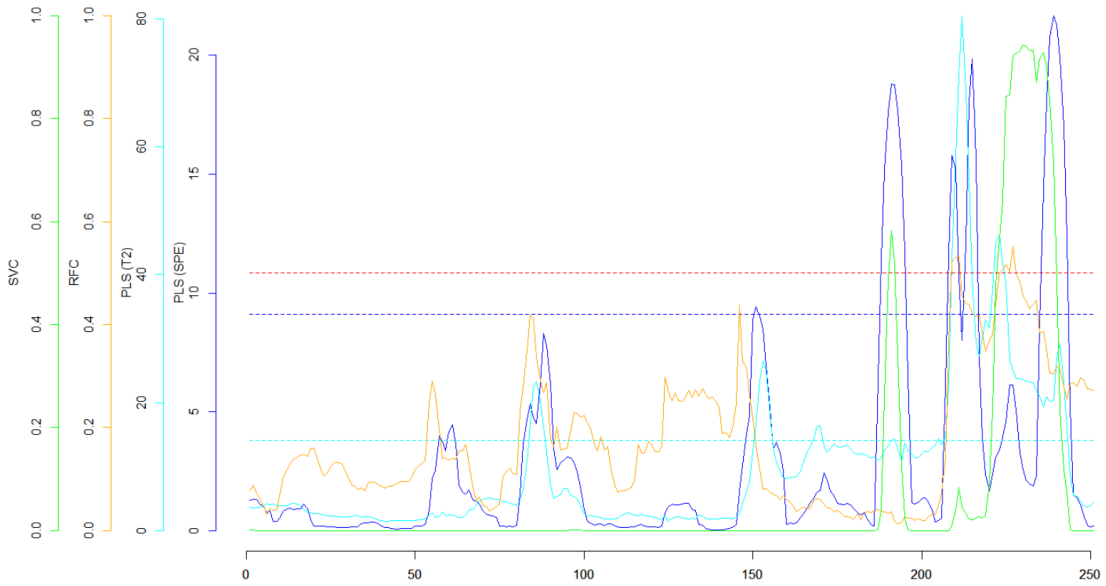


Figura 7.3.- Detección de observaciones anómalas. Observaciones 1000-1250

Finalmente, respecto al tema de diagnóstico de fallos, para poder comparar los resultados del PLS y los del RF, se escogerán los ejemplos de cada una de estas técnicas y se analizarán con la otra técnica.

Primero se tomará la observación “13444” del ejemplo del PLS (ver apartado 5.7.5) y se analizará mediante el RF. La probabilidad calculada para esta observación por el RFC es de 0.179. Este valor es cercano a 0, lo que indica que esta observación sería considerada por el RFC como un dato normal (no anómalo). Sin embargo, la probabilidad calculada por el SVC para esta misma observación sería de 1, es decir, en este caso sí se consideraría anómala. Esto confirma lo comentado anteriormente acerca de que el SVC es capaz de detectar mejor las zonas de observaciones anómalas que el RFC.

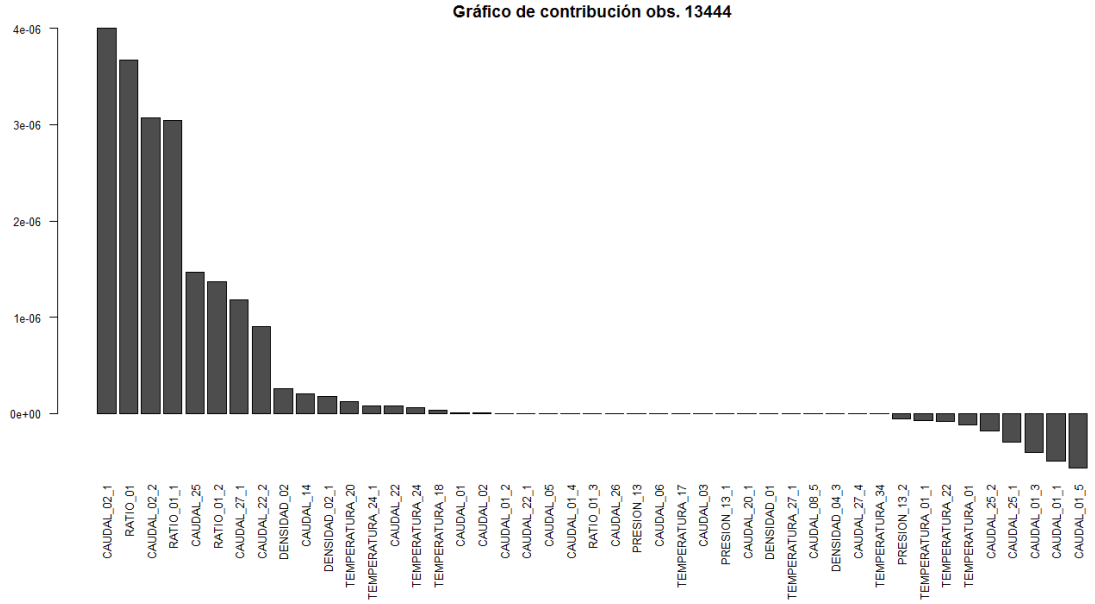


Figura 7.4.- Gráfico de contribución de la observación “13444” (sin la variable respuesta decalada)

Por otro lado, las contribuciones obtenidas para esta observación mediante el RFR se muestran en la Figura 7.4. Las variables con mayor contribución son “CAUDAL_02_1” y “RATIO_01”. En la Figura 7.5 y en la Figura 7.6 pueden verse los gráficos de estas variables. La línea vertical roja indica la posición de la observación “13444”. A diferencia del caso del PLS, en este caso no se aprecian cambios en los valores de estas variables que hagan pensar que son las responsables de la situación anómala en la observación “13444”.

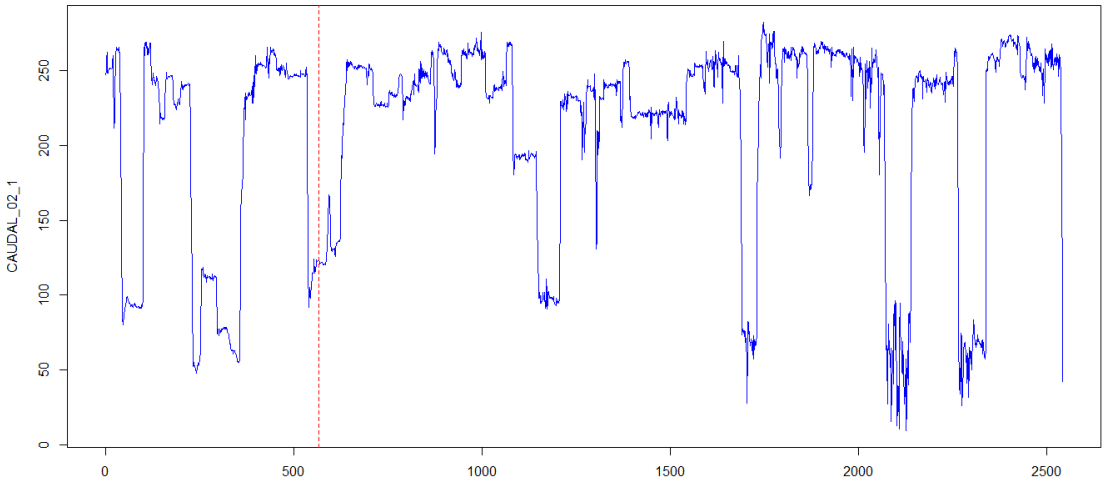


Figura 7.5.- Variable “CAUDAL_02_1” (datos de validación)

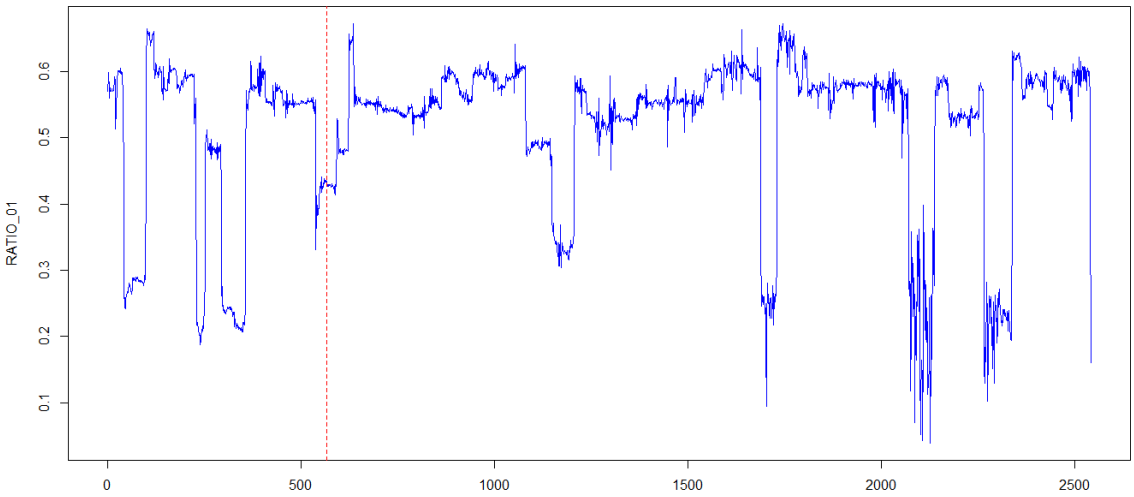


Figura 7.6.- Variable "RATIO_01" (datos de validación)

Ahora se analizará la observación "14183" del ejemplo del RF (ver apartado 5.8.7) mediante el PLS. La Figura 7.7 muestra el gráfico de control del estadístico SPE para los datos de validación. La línea vertical roja discontinua indica la posición de la observación "14183". Como puede comprobarse, el SPE para esta observación es claramente superior al límite de control (línea horizontal roja), por lo que esta observación es una observación anómala. Si se comprueban las contribuciones al SPE para dicha observación (Figura 7.8), sin tener en cuenta las contribuciones de la propia variable respuesta decalada ("CONS_ESP"), la mayor contribución se obtiene para la variable "TEMPERATURA_01_7". Como puede observarse en la Figura 7.9, esta variable sufre un gran cambio en la observación "14183", que es responsable de la situación anómala. Para esta observación, el RFR había detectado como variable con mayor contribución a esta situación la variable "CAUDAL_01_2", que se había comprobado que también sufría un cambio brusco en la observación "14183" (ver apartado 5.8.7, Figura 5.30).

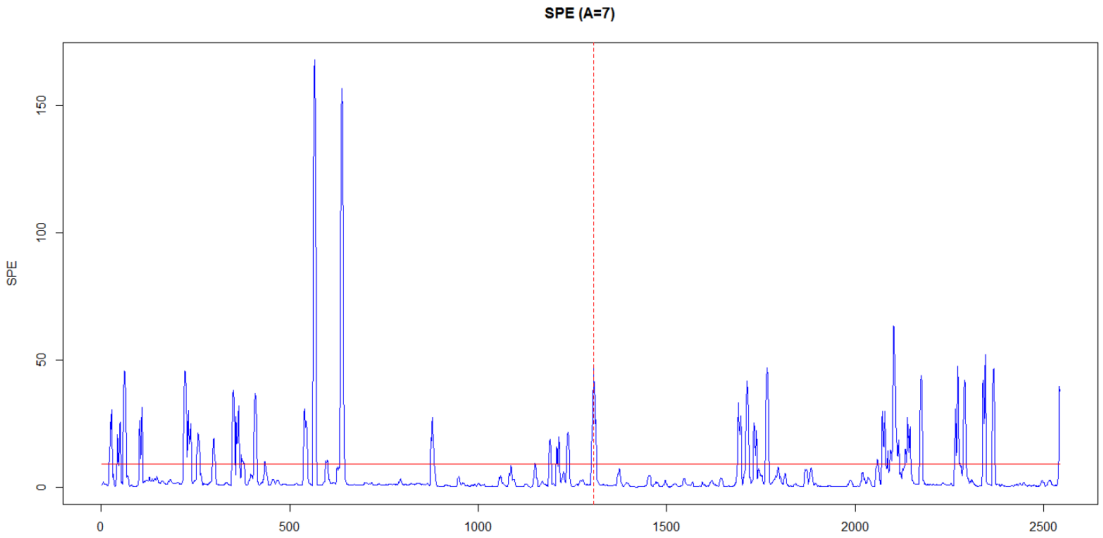


Figura 7.7.- Gráfico de control SPE para los datos de validación.

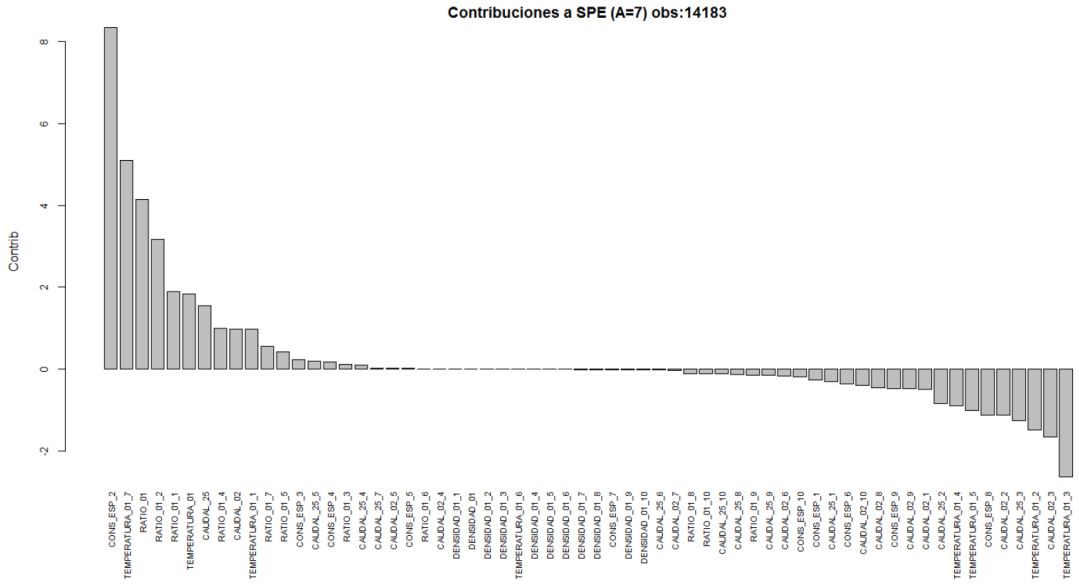


Figura 7.8.- Contribuciones a SPE para la observación "14183"

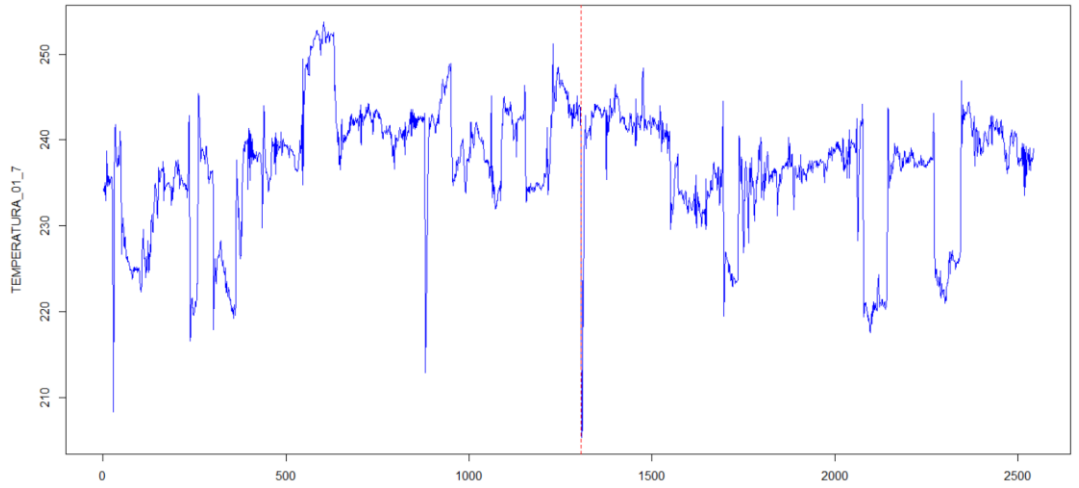


Figura 7.9.- Variable "TEMPERATURA_01_7"

Se ha comprobado que el PLS es capaz de explicar las situaciones halladas con la metodología basada en RF pero no al revés. Es decir, el RF no es capaz de detectar y explicar ciertas situaciones detectadas mediante la técnica del PLS. Además, en uno de los ejemplos del RF, una de las variables con mayor contribución no parecía ser realmente responsable de la situación anómala detectada (ver apartado 5.8.7, Figura 5.37). Esto puede ser debido a que la metodología basada en RF identifica las contribuciones de las variables de la propia variable respuesta (ver apartado 3.4.3.2). Sin embargo, el PLS es capaz de calcular las predicciones tanto de la variable respuesta como de las variables predictoras. Esta capacidad se utiliza para calcular las contribuciones al SPE, que se calculan a partir de los errores de predicción de las variables predictoras (no de la variable respuesta). Esto aporta una mejor estimación de qué variables están más alejadas de las condiciones de operación normales, es decir, qué variables son las responsables de las situaciones anómalas.

8 Apéndice

Tabla 8.1.- Resultados de la bondad de ajuste del RF para diferentes tamaños del horizonte de predicción (variables originales).

N_{train}	P_{pred}	MSE_{train}	MSE_{pred}	MSE_{test}
500	1	4.839E-09	1.733E-08	2.445E-06
500	5		8.707E-08	
500	10		3.559E-07	
500	20		4.356E-07	
500	30		5.262E-07	
500	40		5.421E-07	
1000	1	6.695E-09	5.723E-08	1.715E-06
1000	5		3.305E-07	
1000	10		2.480E-07	
1000	20		2.516E-07	
1000	30		2.482E-07	
1000	40		4.791E-07	
1500	1	6.494E-09	2.423E-07	9.239E-07
1500	5		2.188E-07	
1500	10		2.293E-07	
1500	20		2.399E-06	
1500	30		1.922E-06	
1500	40		1.538E-06	
2000	1	7.131E-09	7.472E-08	5.272E-07
2000	5		9.324E-08	
2000	10		6.726E-08	
2000	20		2.327E-07	
2000	30		3.734E-07	
2000	40		7.694E-07	
2500	1	7.647E-09	1.544E-08	5.426E-07
2500	5		1.165E-06	
2500	10		1.917E-06	
2500	20		1.171E-06	
2500	30		9.332E-07	
2500	40		7.214E-07	
3000	1	8.171E-09	4.365E-07	4.547E-07
3000	5		3.029E-07	
3000	10		2.160E-07	
3000	20		2.001E-07	
3000	30		1.770E-07	
3000	40		2.056E-07	

3500	1	7.763E-09	3.305E-08	4.443E-07
3500	5		1.451E-07	
3500	10		9.272E-08	
3500	20		1.920E-07	
3500	30		1.657E-07	
3500	40		2.126E-07	
4000	1	7.462E-09	9.468E-08	4.366E-07
4000	5		5.032E-07	
4000	10		3.205E-07	
4000	20		3.311E-07	
4000	30		2.740E-07	
4000	40		3.506E-07	
4500	1	7.765E-09	1.676E-07	4.423E-07
4500	5		3.115E-07	
4500	10		3.082E-07	
4500	20		2.752E-07	
4500	30		5.190E-07	
4500	40		5.334E-07	
5000	1	7.376E-09	1.268E-07	4.780E-07
5000	5		1.231E-07	
5000	10		5.159E-07	
5000	20		7.911E-07	
5000	30		6.947E-07	
5000	40		6.509E-07	
5500	1	7.504E-09	1.460E-07	4.201E-07
5500	5		8.495E-07	
5500	10		5.393E-07	
5500	20		5.505E-07	
5500	30		4.279E-07	
5500	40		4.033E-07	
6000	1	7.581E-09	1.009E-06	4.784E-07
6000	5		8.365E-07	
6000	10		7.602E-07	
6000	20		4.950E-07	
6000	30		4.785E-07	
6000	40		4.880E-07	

Tabla 8.2.- Resultados de la bondad de ajuste del RF para diferentes tamaños del horizonte de predicción (variables decaladas).

N_{train}	P_{pred}	MSE_{train}	MSE_{pred}	MSE_{valid}
500	1	5.066E-09	1.382E-08	1.507E-06
500	5		6.914E-08	
500	10		3.169E-07	
500	20		3.372E-07	
500	30		3.617E-07	
500	40		2.858E-07	
1000	1	8.833E-09	4.741E-07	7.387E-07
1000	5		4.366E-07	
1000	10		2.486E-07	
1000	20		1.527E-07	
1000	30		1.155E-07	
1000	40		1.330E-07	
1500	1	8.191E-09	1.861E-08	4.658E-07
1500	5		4.743E-08	
1500	10		1.542E-07	
1500	20		2.607E-06	
1500	30		1.807E-06	
1500	40		1.362E-06	
2000	1	8.372E-09	3.141E-08	1.759E-07
2000	5		2.888E-08	
2000	10		3.429E-08	
2000	20		5.167E-08	
2000	30		1.128E-07	
2000	40		2.908E-07	
2500	1	8.971E-09	6.146E-08	1.454E-07
2500	5		5.373E-07	
2500	10		8.012E-07	
2500	20		4.256E-07	
2500	30		3.013E-07	
2500	40		2.356E-07	
3000	1	9.432E-09	8.657E-08	1.344E-07
3000	5		8.135E-08	
3000	10		4.966E-08	
3000	20		4.598E-08	
3000	30		4.340E-08	
3000	40		6.107E-08	
3500	1	9.066E-09	1.753E-08	1.212E-07
3500	5		2.143E-08	
3500	10		2.429E-08	
3500	20		6.407E-08	

3500	30		5.272E-08	
3500	40		4.753E-08	
4000	1	8.679E-09	4.333E-08	1.206E-07
4000	5		1.219E-07	
4000	10		7.491E-08	
4000	20		5.481E-08	
4000	30		4.346E-08	
4000	40		4.598E-08	
4500	1	8.853E-09	4.285E-08	1.377E-07
4500	5		3.781E-08	
4500	10		3.095E-08	
4500	20		2.913E-08	
4500	30		4.181E-08	
4500	40		7.123E-08	
5000	1	8.468E-09	2.346E-08	1.392E-07
5000	5		2.135E-08	
5000	10		3.996E-08	
5000	20		5.153E-08	
5000	30		1.050E-07	
5000	40		9.829E-08	
5500	1	8.260E-09	2.782E-08	1.352E-07
5500	5		6.367E-08	
5500	10		8.040E-08	
5500	20		1.377E-07	
5500	30		1.081E-07	
5500	40		1.107E-07	
6000	1	8.311E-09	3.148E-07	1.321E-07
6000	5		1.931E-07	
6000	10		2.056E-07	
6000	20		1.364E-07	
6000	30		1.257E-07	
6000	40		1.352E-07	

Tabla 8.3.- Resultados del estudio del ajuste de parámetros para RF.

n_{tree}	m_{try}	node size	R^2	MSE	t (min)
100	304	1	0.8755	1.380E-07	3.4
100	304	2	0.8697	1.450E-07	3.4
100	304	3	0.8707	1.440E-07	3.4
100	304	4	0.8722	1.420E-07	3.3
100	304	5	0.8799	1.330E-07	3.3
100	304	6	0.8627	1.520E-07	3.3
100	183	1	0.8633	1.520E-07	3.0
100	183	2	0.8590	1.560E-07	3.0
100	183	3	0.8517	1.650E-07	2.9
100	183	4	0.8670	1.480E-07	2.9
100	183	5	0.8635	1.520E-07	2.8
100	183	6	0.8542	1.620E-07	2.8
100	91	1	0.8275	1.920E-07	2.6
100	91	2	0.8433	1.740E-07	2.6
100	91	3	0.8322	1.860E-07	2.5
100	91	4	0.8385	1.790E-07	2.5
100	91	5	0.8224	1.970E-07	2.5
100	91	6	0.8378	1.800E-07	2.5
200	304	1	0.8724	1.420E-07	6.8
200	304	2	0.8744	1.390E-07	6.7
200	304	3	0.8665	1.480E-07	6.7
200	304	4	0.8803	1.330E-07	6.6
200	304	5	0.8818	1.310E-07	6.5
200	304	6	0.8744	1.390E-07	6.5
200	183	1	0.8636	1.510E-07	5.8
200	183	2	0.8602	1.550E-07	5.8
200	183	3	0.8607	1.550E-07	5.8
200	183	4	0.8618	1.530E-07	5.7
200	183	5	0.8731	1.410E-07	5.6
200	183	6	0.8675	1.470E-07	5.6
200	91	1	0.8470	1.700E-07	5.2
200	91	2	0.8386	1.790E-07	5.1
200	91	3	0.8451	1.720E-07	5.0
200	91	4	0.8423	1.750E-07	5.2
200	91	5	0.8381	1.800E-07	5.0
200	91	6	0.8368	1.810E-07	5.0
500	304	1	0.8765	1.370E-07	17.1
500	304	2	0.8763	1.370E-07	17.3
500	304	3	0.8730	1.410E-07	17.1

500	304	4	0.8809	1.320E-07	16.4
500	304	5	0.8765	1.370E-07	16.3
500	304	6	0.8815	1.320E-07	16.1
500	183	1	0.8645	1.500E-07	14.7
500	183	2	0.8663	1.480E-07	14.4
500	183	3	0.8676	1.470E-07	14.3
500	183	4	0.8681	1.460E-07	14.1
500	183	5	0.8613	1.540E-07	14.0
500	183	6	0.8655	1.490E-07	13.8
500	91	1	0.8405	1.770E-07	12.8
500	91	2	0.8416	1.760E-07	12.7
500	91	3	0.8402	1.770E-07	12.5
500	91	4	0.8431	1.740E-07	12.3
500	91	5	0.8415	1.760E-07	12.1
500	91	6	0.8357	1.820E-07	12.0
1000	304	1	0.8786	1.350E-07	34.1
1000	304	2	0.8717	1.420E-07	33.9
1000	304	3	0.8799	1.330E-07	33.5
1000	304	4	0.8753	1.380E-07	33.2
1000	304	5	0.8748	1.390E-07	32.7
1000	304	6	0.8786	1.350E-07	32.5
1000	183	1	0.8671	1.480E-07	29.5
1000	183	2	0.8680	1.470E-07	29.4
1000	183	3	0.8596	1.560E-07	28.9
1000	183	4	0.8658	1.490E-07	28.7
1000	183	5	0.8637	1.510E-07	28.4
1000	183	6	0.8674	1.470E-07	28.1
1000	91	1	0.8430	1.740E-07	26.0
1000	91	2	0.8420	1.750E-07	25.7
1000	91	3	0.8411	1.760E-07	25.4
1000	91	4	0.8430	1.740E-07	25.0
1000	91	5	0.8441	1.730E-07	24.6
1000	91	6	0.8424	1.750E-07	24.6

9 Bibliografía

Abonyi, Janos, Tibor Kulcsar, Miklos Balaton, and Laszlo Nagy. "Energy Monitoring of Process Systems: Time-series Segmentation-based Targeting Models." *Clean Technologies and Environmental Policy* 16.7 (2014): 1245-253.

Aggarwal, Charu C. *Outlier Analysis*. New York: Springer, 2013.

Aitani, A. (2004). Oil Refining and Products, In: Encyclopaedia of Energy, Elsevier, NY, v.4, Chapter 4

Aldrich, C., and Lidia Auret. Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods. London: Springer, 2013

Bowman, A. W., and Azzalini, A. *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. Oxford: Clarendon, 1997

Breiman, Leo. "Random forests. *Machine Learning* 45.1 (2001): 5-32

Cortes, Corinna and Vladimir Vapnik. "Support-vector networks". Machine Learning 20.3 (1995): 273

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation IEEE Trans. Evol. Computat. 6.2 (2002): 182-97

Ferrer, Alberto, Daniel Aguado, Santiago Vidal-Puig, José Manuel Prats, and Manuel Zarzo. "PLS: A Versatile Tool for Industrial Process Improvement and Optimization." *Appl. Stochastic Models Bus. Ind. Applied Stochastic Models in Business and Industry* 24.6 (2008): 551-67.

Folch-Fortuny, A., F. Arteaga, and A. Ferrer. "PCA Model Building with Missing Data: New Proposals and a Comparative Study." *Chemometrics and Intelligent Laboratory Systems* 146 (2015): 77-88.

Guyon, I. and Elisseeff, A. “An Introduction to Variable and Feature Selection.” *Journal of Machine Learning Research,* 3 (2003): 1157–82.

Hastie, Trevor, Robert Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2013. 436-55

Ishwaran, H., and U.B. Kogalur. "Random Survival Forests for R." *Rnews* 7.2 (2007): 25-31

Jones, D. S. J., and Peter R. Pujadó. *Handbook of Petroleum Processing*. Dordrecht: Springer, 2006

Kourti, Theodora, and John F. MacGregor. "Process Analysis, Monitoring and Diagnosis, Using Multivariate Projection Methods." *Chemometrics and Intelligent Laboratory Systems* 28.1 (1995): 3-21

Ku, Wenfu, Robert H. Storer, and Christos Georgakis. "Disturbance Detection and Isolation by Dynamic Principal Component Analysis." *Chemometrics and Intelligent Laboratory Systems* 30.1 (1995): 179-96

Kuhn, M. "Variable Selection Using The Caret Package." *Variable Selection Using The Caret Package*. N.p., 30 Jan. 2009. Web. 25 Apr. 2016. https://r-forge.r-project.org/scm/viewvc.php/*checkout*/pkg/caret/inst/doc/caretSelection.pdf?revision=77&root=caret&pathrev=90

Liaw, A. and Matthew Wiener. "Classification and Regression by randomForest". R News 2(3). (2002): 18-22.

Malik, Kamal, H. Sadawarti, and Kalra G. S. "Comparative Analysis of Outlier Detection Techniques." *International Journal of Computer Applications IJCA* 97.8 (2014): 12-21.

Meyer, David. "Support Vector Machines." *Support Vector Machines* (n.d.): n. pag. *Support Vector Machines*. 5 Aug. 2015. Web. 20 Mar. 2016. https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf

Nilsson, R., J. Peña, J. Björkegren and J. Tegnér. "Consistent Feature Selection for Pattern Recognition in Polynomial Time." *The Journal of Machine Learning Research*, 8 (2007): 589-612

Olsen, T. "An oil refinery walk-through", Chemical Engineering Progress, Vol.110, Nº5.2014

Peña, Daniel. "Análisis de series temporales". Madrid: Alianza Editorial, 2005

Roobaert, D., G. Karakoulas, and N. Chawla. "Information Gain, Correlation and Support Vector Machines." Ed. I. Guyon. *Feature Extraction: Foundations and Applications*. Berlin: Springer-Verlag, 2006. 463-70

Saabas, A. "Interpreting Random Forests." *Diving into Data*. N.p., 19 Oct. 2014. Web. 20 June 2016. http://blog.datadive.net/interpreting-random-forests/

Seborg, Dale E., Thomas F. Edgar, and Duncan A. Mellichamp. Process Dynamics and Control. New York: Wiley, 2004, p.1-11

Schölkopf, B., J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. *Estimating the Support of a High-Dimensional Distribution* (n.d.): n. pag. *Estimating the Support of a High-Dimensional Distribution - Microsoft Research*. Microsoft Research, 27 Nov. 1999. Web. 22 Mar. 2016. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-99-87.pdf

Shardt, Yuri, and Sirish Shah. "Segmentation Methods for Model Identification from Historical Process Data." *Proceedings of the 19th IFAC World Congress* (2014)

Sharp Sight Labs. "What's the Difference between Machine Learning, Statistics, and Data Mining?" *Rbloggers*. N.p., 09 May 2016. Web. 12 Aug. 2016. https://www.r-bloggers.com/whats-the-difference-between-machine-learning-statistics-and-data-mining/

Smola, Alex J., and Bernhard Scholkopf. "Support Vector Regression (SVR)." A Tutorial on Support Vector Regression (2011): 97-118. A Tutorial on Support Vector Regression. Oct. 1998. Web. 20 Mar. 2016. http://www.svms.org/regression/SmSc98.pdf

The Economist, "Manufacturers Must Learn to Behave More like Tech Firms." *Machine Learning*. The Economist, 21 Nov. 2015. Web. 12 Aug. 2016. http://www.economist.com/news/leaders/21678786-manufacturers-must-learn-behave-more-tech-firms-machine-learning

Wold, S., Johansson, E., and Cocchi, M., *3D-QSAR in Drug Design, Theory, Methos, and Applications*, Ledien: ESCOM Science, 1993: 523-550

Wold, S., Michael Sjöström, and Lennart Eriksson. "PLS-regression: A Basic Tool of Chemometrics." *Chemometrics and Intelligent Laboratory Systems* 58.2 (2001): 109-30.

Worrell, Ernst, Mariëlle Corsten, and Christina Galitsky. "Energy Efficiency Improvement and Cost Saving Opportunities for Petroleum Refineries: An ENERGY STAR(R) Guide for Energy and Plant Managers." (2015)

Zhu, F., *Energy and Process Optimization for the Process Industries*, John Wiley & Sons, 2014, p.35-36.