

Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías Industriales

Técnicas de ajuste de los métodos Kernel para la regresión.

Autor: Jorge Amor Perea

Tutor: Teodoro Álamo Cantarero

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería

Sevilla, 2019



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Técnicas de ajuste de los métodos Kernel para la regresión.

Autor:
Jorge Amor Perea

Tutor:
Teodoro Álamo Cantarero
Catedrático de Universidad

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2018

Trabajo Fin de Grado: Técnicas de ajuste de los métodos Kernel para la regresión.

Autor: Jorge Amor Perea

Tutor: Teodoro Álamo Cantarero

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi madre
¿A quién si no?

Agradecimientos

A toda la buena gente que me ha acompañado en estos años, por tanto y tantos cafés.

A Teo, que me descubrió que la estadística era un mundo interesante y que no podemos dejar de aprender.

A todos los que cuando fallé, me levantaron.

A todos ellos, mil millones de gracias, pero a mi madre...

A mi madre; que todas estas páginas no sirvan de perdón ni orgullo, que sirvan de agradecimiento a quién me lo ha dado todo sin recibir nada.

Resumen

Este trabajo constituye un acercamiento a los métodos *kernel* para el aprendizaje automático en problemas de regresión. Se ha realizado un estudio sobre el ajuste de dichos métodos para minimizar los errores cometidos en la predicción de secuencias temporales.

Para la realización del estudio, se ha trabajado con tres problemas diferentes; predicción de una señal modulada en amplitud con ruido aleatorio, predicción de la media mensual del número de manchas solares de Wolf y predicción de la demanda de energía eléctrica.

Se encuentran en este documento todos los algoritmos y programas necesarios para la realización del estudio, aplicando técnicas de validación cruzada para la elección de la mejor solución propuesta posible.

El objetivo de este estudio es determinar unas pautas para la elección de los parámetros variables en los métodos *kernel* y comprobar la mejoría en los resultados al aplicar técnicas de validación cruzada en el área del aprendizaje automático.

Abstract

This paper constitutes an approach to kernel methods for machine learning in regression problems. The document includes a study on the tuning of these methods to minimize errors made in the prediction of temporal sequences.

To carry out the study, we have worked with three different problems; prediction of an amplitude-modulated signal with random noise, prediction of Wolf's monthly average sunspot number and prediction of electrical energy demand.

All the algorithms and programs necessary to carry out the study are found in this document, applying cross-validation techniques for the selection of the best purposed solution possible.

The objective of this study is to determine some guidelines for the election of the variable parameters in the kernel methods and to verify the improvement in the results when applying cross validation techniques in machine learning area.

Índice

Agradecimientos	vii
Resumen	viii
Abstract	ix
Índice	x
Índice de Tablas	xv
Índice de Figuras	xvii
Notación	xix
1 Introducción	1
1.1 <i>Objetivos</i>	1
1.2 <i>Organización de la memoria</i>	2
2 Conceptos. Desarrollo Teórico	3
2.1 <i>Aprendizaje Automático</i>	3
2.1.1 <i>Aprendizaje supervisado</i>	4
2.1.2 <i>Aprendizaje no supervisado</i>	5
2.2 <i>Métodos históricos para la regresión. Mínimos cuadrados</i>	5
2.2.1 <i>Formulación matemática</i>	6
2.3 <i>Métodos kernel. Máquinas de soporte vectorial</i>	7
2.3.1 <i>Problemas de clasificación (SVM)</i>	7
2.3.2 <i>Problemas de regresión (SVR)</i>	8
2.3.3 <i>Formulación matemática</i>	9
2.4 <i>Técnicas para la elección y ajuste de regresores e hiperparámetros</i>	11
2.4.1 <i>Validación cruzada</i>	11
3 Presentación De Los Problemas	13
3.1 <i>Predicción de una señal modulada en amplitud. Problema sintético</i>	13
3.2 <i>Predicción del número de manchas solares</i>	16
3.3 <i>Predicción de la demanda de energía eléctrica</i>	17
4 Descripción del Trabajo Desarrollado	21
4.1 <i>Herramientas empleadas</i>	21
4.2 <i>Scripts</i>	21
4.2.1 <i>Generación de datos del problema sintético</i>	21

4.2.2	Predicción de onda modulada en amplitud	22
4.2.3	Predicción del número de Wolf	22
4.2.4	Predicción de la demanda de energía eléctrica	22
4.2.5	Núcleo para ensayar SMA con validación cruzada	22
4.2.6	Núcleo para ensayar SSN con validación cruzada	23
4.2.7	Núcleo para ensayar DEL con validación cruzada	23
4.3	<i>Funciones</i>	24
4.3.1	Preparación de datos	24
4.3.2	Generación de los conjuntos para validación cruzada	24
4.3.3	Entrenamiento y validación de conjuntos	25
4.3.4	Cálculo de <i>kernels</i>	25
4.3.5	Cálculo de predicciones	25
4.3.6	Cálculo de errores	25
4.3.7	Representación de errores	25
4.4	<i>Obtención de datos y tratamiento</i>	26
4.4.1	Señal modulada en amplitud	26
4.4.2	Número de manchas solares	26
4.4.3	Demanda de energía eléctrica	27
5	Algoritmos para la selección de hiperparámetros	29
5.1	<i>Señal modulada en amplitud</i>	31
5.2	<i>Número de manchas solares</i>	32
5.3	<i>Demanda de energía eléctrica</i>	32
6	Presentación de resultados	35
6.1	<i>Resultados SMA</i>	35
6.1.1	Caso 1. Primer regresor y primer conjunto de parámetros	35
6.1.2	Caso 2. Primer regresor y segundo conjunto de parámetros	36
6.1.3	Caso 3. Primer regresor y tercer conjunto de parámetros	38
6.1.4	Caso 4. Segundo regresor y primer conjunto de parámetros	39
6.1.5	Caso 5. Segundo regresor y segundo conjunto de parámetros	41
6.1.6	Caso 6. Segundo regresor y tercer conjunto de parámetros	42
6.1.7	Caso 7. Tercer regresor y primer conjunto de parámetros	44
6.1.8	Caso 8. Tercer regresor y segundo conjunto de parámetros	45
6.1.9	Caso 9. Tercer regresor y tercer conjunto de parámetros	47
6.1.10	Elección del regresor	48
6.1.11	Predicción	49
6.2	<i>Resultados SSN</i>	52

6.2.1	Caso 1. Primer regresor y primer conjunto de parámetros	52
6.2.2	Caso 2. Primer regresor y segundo conjunto de parámetros	54
6.2.3	Caso 3. Primer regresor y tercer conjunto de parámetros	55
6.2.4	Caso 4. Segundo regresor y primer conjunto de parámetros	56
6.2.5	Caso 5. Segundo regresor y segundo conjunto de parámetros	58
6.2.6	Caso 6. Segundo regresor y tercer conjunto de parámetros	59
6.2.7	Caso 7. Tercer regresor y primer conjunto de parámetros	61
6.2.8	Caso 8. Tercer regresor y segundo conjunto de parámetros	62
6.2.9	Caso 9. Tercer regresor y tercer conjunto de parámetros	64
6.2.10	Elección del regresor	65
6.2.11	Predicción	66
6.3	Resultados SMA	69
6.3.1	Caso 1. Primer regresor y primer conjunto de parámetros	69
6.3.2	Caso 2. Primer regresor y segundo conjunto de parámetros	71
6.3.3	Caso 3. Primer regresor y tercer conjunto de parámetros	72
6.3.4	Caso 4. Segundo regresor y primer conjunto de parámetros	73
6.3.5	Caso 5. Segundo regresor y segundo conjunto de parámetros	75
6.3.6	Caso 6. Segundo regresor y tercer conjunto de parámetros	76
6.3.7	Caso 7. Tercer regresor y primer conjunto de parámetros	78
6.3.8	Caso 8. Tercer regresor y segundo conjunto de parámetros	79
6.3.9	Caso 9. Tercer regresor y tercer conjunto de parámetros	81
6.3.10	Elección del regresor	82
6.3.11	Predicción	83
7	Conclusiones	87
7.1	<i>Comparación entre métodos kernel y mínimos cuadrados</i>	87
7.1.1	Señal modulada en amplitud	87
7.1.2	Número de manchas solares	88
7.1.3	Demanda de energía eléctrica	89
7.2	<i>Impacto en los resultados al usar validación cruzada</i>	90
7.3	<i>Valoración de los resultados obtenidos</i>	91
7.4	<i>Líneas de trabajo futuro</i>	91
8	Código desarrollado.	93
8.1	<i>Predicción de señal modulada en amplitud</i>	93
8.1.1	svr_core_SMA.m	93
8.1.2	genera_onda_modulada.m	96
8.1.3	pred_SMA.m	97

8.2	<i>Predicción del número de manchas solares</i>	99
8.2.1	svr_core_SSN.m	99
8.2.2	SSN_pred.m	102
8.3	<i>Predicción de demanda de energía eléctrica</i>	104
8.3.1	svr_core_DEL.m	104
8.3.2	pred_DEL.m	106
8.4	<i>Funciones comunes</i>	108
8.4.1	preparadatos.m	108
8.4.2	kfoldCV_mark.m	110
8.4.3	kernels.m	110
8.4.4	c_krnl.m	114
8.4.5	v_krnl.m	115
8.4.6	calculaerrores.m	115
8.4.7	pintaerrores.m	119
	Referencias	125

Índice de Tablas

Tabla 5-1. Distintas medias y desviaciones estándar en regresión.	31
Tabla 5-2. Conjuntos de parámetros a ensayar en el problema SMA.	32
Tabla 5-3. Conjuntos de parámetros a ensayar en el problema SSN.	32
Tabla 5-4. Conjuntos de parámetros a ensayar en el problema DEL.	33
Tabla 6-1. Parámetros del caso 1 en SMA.	35
Tabla 6-2. Errores cometidos en el caso 1 de SMA.	36
Tabla 6-3. Parámetros del caso 2 en SMA.	37
Tabla 6-4. Errores cometidos en el caso 2 de SMA.	38
Tabla 6-5. Parámetros del caso 3 en SMA	38
Tabla 6-6. Errores cometidos en el caso 3 de SMA.	39
Tabla 6-7. Parámetros del caso 4 en SMA.	40
Tabla 6-8. Errores cometidos en el caso 4 de SMA.	41
Tabla 6-9. Parámetros del caso 5 en SMA	41
Tabla 6-10. Errores cometidos en el caso 5 de SMA.	42
Tabla 6-11. Parámetros del caso 6 en SMA.	43
Tabla 6-12. Errores cometidos en el caso 6 de SMA.	43
Tabla 6-13. Parámetros del caso 7 en SMA.	44
Tabla 6-14. Errores cometidos en el caso 7 de SMA.	45
Tabla 6-15. Parámetros del caso 8 en SMA.	46
Tabla 6-16. Errores cometidos en el caso 8 de SMA.	47
Tabla 6-17. Parámetros del caso 9 en SMA.	47
Tabla 6-18. Errores cometidos en el caso 9 de SMA.	48
Tabla 6-19. Resumen de errores por caso en SMA.	49
Tabla 6-20. Parámetros del regresor elegido en SMA.	49
Tabla 6-21. Parámetros del caso 1 en SSN.	52
Tabla 6-22. Errores cometidos en el caso 1 de SSN.	53
Tabla 6-23. Parámetros del caso 2 en SSN.	54
Tabla 6-24. Errores cometidos en el caso 2 de SSN.	55
Tabla 6-25. Parámetros del caso 3 en SSN.	55
Tabla 6-26. Errores cometidos en el caso 3 de SSN.	56
Tabla 6-27. Parámetros del caso 4 en SSN.	57
Tabla 6-28. Errores cometidos en el caso 4 de SSN.	58

Tabla 6-29. Parámetros del caso 5 en SSN.	58
Tabla 6-30. Errores cometidos en el caso 5 de SSN.	59
Tabla 6-31. Parámetros del caso 6 en SSN.	60
Tabla 6-32. Errores cometidos en el caso 6 de SSN.	60
Tabla 6-33. Parámetros del caso 7 en SSN.	61
Tabla 6-34. Errores cometidos en el caso 7 de SSN.	62
Tabla 6-35. Parámetros del caso 8 en SSN.	63
Tabla 6-36. Errores cometidos en el caso 8 de SSN.	64
Tabla 6-37. Parámetros del caso 9 en SSN.	64
Tabla 6-38. Errores cometidos en el caso 9 de SSN.	65
Tabla 6-39. Resumen de errores por caso en SSN.	66
Tabla 6-40. Parámetros del regresor elegido en SSN.	66
Tabla 6-41. Parámetros del caso 1 en DEL.	69
Tabla 6-42. Errores cometidos en el caso 1 de DEL.	70
Tabla 6-43. Parámetros del caso 2 en DEL.	71
Tabla 6-44. Errores cometidos en el caso 2 de DEL.	72
Tabla 6-45. Parámetros del caso 3 en DEL.	72
Tabla 6-46. Errores cometidos en el caso 3 de DEL.	73
Tabla 6-47. Parámetros del caso 4 en DEL.	74
Tabla 6-48. Errores cometidos en el caso 4 de DEL.	75
Tabla 6-49. Parámetros del caso 5 en SSN.	75
Tabla 6-50. Errores cometidos en el caso 5 de DEL.	76
Tabla 6-51. Parámetros del caso 6 en SSN.	77
Tabla 6-52. Errores cometidos en el caso 6 de DEL.	77
Tabla 6-53. Parámetros del caso 7 en DEL.	78
Tabla 6-54. Errores cometidos en el caso 7 de DEL.	79
Tabla 6-55. Parámetros del caso 8 en DEL.	80
Tabla 6-56. Errores cometidos en el caso 8 de DEL.	81
Tabla 6-57. Parámetros del caso 9 en DEL.	81
Tabla 6-58. Errores cometidos en el caso 9 de DEL.	82
Tabla 6-59. Resumen errores por caso en DEL	83
Tabla 6-60. Parámetros del regresor elegido en DEL.	83
Tabla 7-1. Comparativa LS vs SVR SMA.	88
Tabla 7-2. Comparativa LS vs SVR SSN.	89
Tabla 7-3. Comparativa LS vs SVR SSN.	90

Índice de Figuras

Figura 2-1. Fases generales del aprendizaje automático.	3
Figura 2-2. Técnicas de aprendizaje automático.	4
Figura 2-3. Fases aprendizaje supervisado.	4
Figura 2-4. Fases aprendizaje no supervisado.	5
Figura 2-5. Ejemplo de correlación espúrea.	5
Figura 2-6. Retrato de Johan Carl Friedrich Gauss.	6
Figura 2-7. Ejemplo de SVM para clasificación.	8
Figura 2-8. Ejemplo de resultados de SVR.	9
Figura 2-9. Esquema validación cruzada de k-iteraciones con $k = 4$.	12
Figura 3-1. Señales del problema sintético.	14
Figura 3-2. Detalle de señal modulada.	15
Figura 3-3. Detalles de señales portadora y moduladora.	15
Figura 3-4. Manchas solares vistas desde el centro de vuelo espacial de la NASA Goddard.	16
Figura 3-5. Serie histórica de medias anuales del SSN.	17
Figura 3-6. Recorte de la DEL de esios.ree.es.	18
Figura 3-7. Subestación eléctrica de la compañía Krostec, México.	19
Figura 4-1. Algoritmo script principal.	23
Figura 4-2. Algoritmo «preparadatos.m».	24
Figura 4-3. Recorte de la web www.sidc.be con SSN.	26
Figura 4-4. Recorte de un indicador en la web esios.ree.es .	28
Figura 5-1. Estrategia de ajuste genérica.	30
Figura 6-1. Histogramas de error para el caso 1 en SMA.	36
Figura 6-2. Histogramas de error para el caso 2 en SMA.	37
Figura 6-3. Histogramas de error para el caso 3 en SMA.	39
Figura 6-4. Histogramas de error para el caso 4 en SMA.	40
Figura 6-5. Histogramas de error para el caso 5 en SMA.	42
Figura 6-6. Histogramas de error para el caso 6 en SMA.	43
Figura 6-7. Histogramas de error para el caso 7 en SMA.	45
Figura 6-8. Histogramas de error para el caso 8 en SMA.	46
Figura 6-9. Histogramas de error para el caso 9 en SMA.	48
Figura 6-10. Predicción en SMA.	50

Figura 6-11. Detalle de predicción en SMA.	51
Figura 6-12. Histograma del error en SMA.	52
Figura 6-13. Histogramas de error para el caso 1 en SSN.	53
Figura 6-14. Histogramas de error para el caso 2 en SSN.	54
Figura 6-15. Histogramas de error para el caso 3 en SSN.	56
Figura 6-16. Histogramas de error para el caso 4 en SSN.	57
Figura 6-17. Histogramas de error para el caso 5 en SSN.	59
Figura 6-18. Histogramas de error para el caso 6 en SSN.	60
Figura 6-19. Histogramas de error para el caso 7 en SSN.	62
Figura 6-20. Histogramas de error para el caso 8 en SSN.	63
Figura 6-21. Histogramas de error para el caso 9 en SSN.	65
Figura 6-22. Predicción en SSN.	67
Figura 6-23. Detalle de predicción en SSN.	68
Figura 6-24. Histograma y distribución normal del error en SSN.	69
Figura 6-25. Histogramas de error para el caso 1 en DEL.	70
Figura 6-26. Histogramas de error para el caso 2 en DEL.	71
Figura 6-27. Histogramas de error para el caso 3 en DEL.	73
Figura 6-28. Histogramas de error para el caso 4 en DEL.	74
Figura 6-29. Histogramas de error para el caso 5 en DEL.	76
Figura 6-30. Histogramas de error para el caso 6 en DEL.	77
Figura 6-31. Histogramas de error para el caso 7 en DEL.	79
Figura 6-32. Histogramas de error para el caso 8 en DEL.	80
Figura 6-33. Histogramas de error para el caso 9 en DEL.	82
Figura 6-34. Predicción en DEL.	84
Figura 6-35. Detalle de la predicción en DEL.	85
Figura 6-36. Histograma del error en DEL.	86
Figura 7-1. Histograma error mínimos cuadrados.	87
Figura 7-2. Histograma error métodos kernel.	88
Figura 7-3. Histograma de error LS en SSN.	88
Figura 7-4. Histograma de error SVR en SSN.	89
Figura 7-5. Histograma de error LS DEL.	89
Figura 7-6. Histograma de error SVR DEL.	90
Figura 7-7. Influencia validación cruzada.	90

Notación

SSN	Número de manchas solares.
f.a.c.	Fracción del año en curso.
std	Desviación estándar
DEL	Demanda de energía eléctrica
SMA	Señal modulada en amplitud
MSE	Mean square error
SME	Standard mean error
ECM	Error cuadrático medio
EEM	Error estándar de la media
sios	Sistema de información del operador del sistema
min	Mínimo
max	Máximo
s.a.	Sujeto a
CV	Cross validation
LS-SVM	Least square support vector machines
LS-SVR	Least square support vector machines for regresión.

1 INTRODUCCIÓN

El sistema «se ha ido a Brenes».

- Teodoro Álamo Cantarero -

Los grandes y rápidos progresos de la sociedad hacen posible hoy día logros que eran impensables hace un siglo. Esto es así en prácticamente en todos los ámbitos de conocimiento que existen, pero especialmente en lo referente a tecnología, que está experimentando un crecimiento exponencial en las últimas décadas que puede llegar a ser abrumador.

Gran parte de responsabilidad en los avances que se están produciendo en los últimos años, es de la inteligencia artificial y el aprendizaje automático de las computadoras en general, que son capaces de procesar una cantidad ingente de información y mejorar en su procesamiento a medida que se va añadiendo ésta.

Es por eso que he elegido realizar este trabajo sobre técnicas de aprendizaje automático, ya que me parecen tan útiles como interesantes además de ofrecer una cantidad de posibilidades casi ilimitadas en cuanto a su aplicación.

Aunque de facto este campo no ha formado parte del plan de estudios del Grado en si mismo, sí ha habido contenidos a lo largo del Grado que han hecho despertar mi curiosidad al respecto del mismo, siendo entonces la posibilidad de aprender cosas nuevas y diferentes, otra de las razones que me ha llevado a elegir una ínfima parte de todo lo relacionado con el mundo del aprendizaje automático como tema de mi Trabajo de Fin de Grado.

En concreto, este trabajo se centra en los métodos *kernel* empleados en problemas de regresión y predicción de secuencias temporales y en el estudio de estrategias de ajuste de todos aquellos aspectos variables de éstos que influyen en el buen rendimiento de los mismos.

1.1 Objetivos

En cuanto a los objetivos que se pretenden alcanzar en el desarrollo de esta trabajo son, a grandes rasgos, los que se enumeran a continuación:

- Obtención de los conocimientos teóricos necesarios en el ámbito de aprendizaje automático y en concreto de métodos *kernel* para la clasificación y, más importante, para la regresión.
- Desarrollo de los algoritmos oportunos en MatLab de forma que se puedan realizar ensayos y simulaciones que permitan la obtención de resultados para su posterior análisis y valoración.
- Obtención y preparación de los datos necesarios para la realización de los ensayos.
- Diseño de estrategias para el correcto ajuste de todos aquellos aspectos de los métodos *kernel* que sean variables y que, en gran parte, hacen que los resultados dependan de las bondades del programador.

- Investigación de la influencia de las estrategias mencionadas en los métodos *kernel* y validación de las técnicas previamente propuestas.
- Presentación de resultados consecuencia de los ensayos y el trabajo realizado, de forma objetiva.
- Valoración de la influencia de las técnicas probadas en los resultados obtenidos y posibles mejoras así como establecer futuras líneas de trabajo.

1.2 Organización de la memoria

La organización de este documento pretende ser un viaje desde las bases teóricas de los temas estudiados hasta ejemplos y aplicaciones prácticas que se resuelven en el mundo real de forma que se puedan obtener unas conclusiones aceptables sobre el resultado del trabajo realizado.

En el segundo capítulo podremos encontrar los conceptos y desarrollos teóricos necesarios para la comprensión del trabajo realizado, así como pequeños ejemplos que los ilustren.

Cuando llegamos al tercer capítulo de este documento, se presentarán los problemas sobre los que vamos a centrarnos y los objetivos que nos proponemos sobre ellos de forma general para, en el cuarto capítulo, presentar las herramientas que posibilitaran la consecución de las soluciones que se proponen; además, podremos encontrar una descripción detallada del procedimiento realizado para la obtención y tratamiento de los datos necesarios para el desarrollo de esta memoria de forma detallada.

En el quinto capítulo, encontraremos de forma agrupada y resumida los resultados del trabajo realizado de forma objetiva, sin entrar en valoraciones de ninguna índole. Se establecerán distintos indicadores y marcadores que permitirán más adelante hacer las valoraciones oportunas.

Para las conclusiones arrojadas por los ensayos realizados, tendremos que esperar hasta el sexto y último capítulo, en el que se enfrentarán de forma justa los ensayos realizados, no sólo de los resultados sino de todo el proceso de realización de este trabajo. Para ello contaremos con los resultados e indicadores obtenidos en el capítulo cinco.

Una vez descrito el núcleo del trabajo en los capítulos mencionados en párrafos anteriores, en el anexo «A» se podrá encontrar el código desarrollado para poder concluir este trabajo.

2 CONCEPTOS. DESARROLLO TEÓRICO

Se puede entender este capítulo como una introducción teórica a las bases sobre las que se sustenta el trabajo realizado. Antes de definir concretamente los conceptos propios de los métodos *kernel*, es necesario introducir ciertas ideas y definiciones del ámbito del aprendizaje automático en general –tanto en problemas de clasificación como en problemas de regresión–, así como una pequeña pincelada al contexto histórico del área de conocimiento que nos ocupa.

2.1. Aprendizaje Automático

El aprendizaje automático es una rama de las ciencias de la computación y de la inteligencia artificial en general. Nace con la intención de otorgar a máquinas la capacidad de tomar decisiones y obtener resultados basados en la experiencia; es por ello que se basa en el análisis de datos obtenidos previamente a la ejecución de los métodos de cálculo oportunos y, en un principio, a medida que aumenta el número de muestras, mejoran los resultados obtenidos con dichos métodos de cálculo.

Sus aplicaciones son prácticamente ilimitadas, ya que en la mayoría de cosas que nos rodean, se producen patrones que estos algoritmos y métodos de cálculo son capaces de encontrar para «aprender» y así tomar mejores decisiones y/o generar mejores resultados.

Aún así, el aprendizaje automático –a día de hoy– no está al nivel de superar la intuición de los seres humanos para tomar decisiones.

Algunos de los campos que aplican técnicas de aprendizaje automático, entre muchos otros, son:

- Economía
- Medicina
- Ingeniería
- Automoción
- Marketing

En general, el aprendizaje automático, consta de tres fases. Es necesario entrenar los algoritmos y validar la calidad de los resultados antes de entrar estos en producción, es decir, que se apliquen de forma real y se tengan en cuenta los resultados obtenidos.



Figura 2-1. Fases generales del aprendizaje automático.

Existen muchas técnicas de aprendizaje automático para cada uno de los dos tipos de aprendizaje que existen (aprendizaje automático supervisado y no supervisado). A lo largo de este capítulo se desarrollan algunas de ellas.

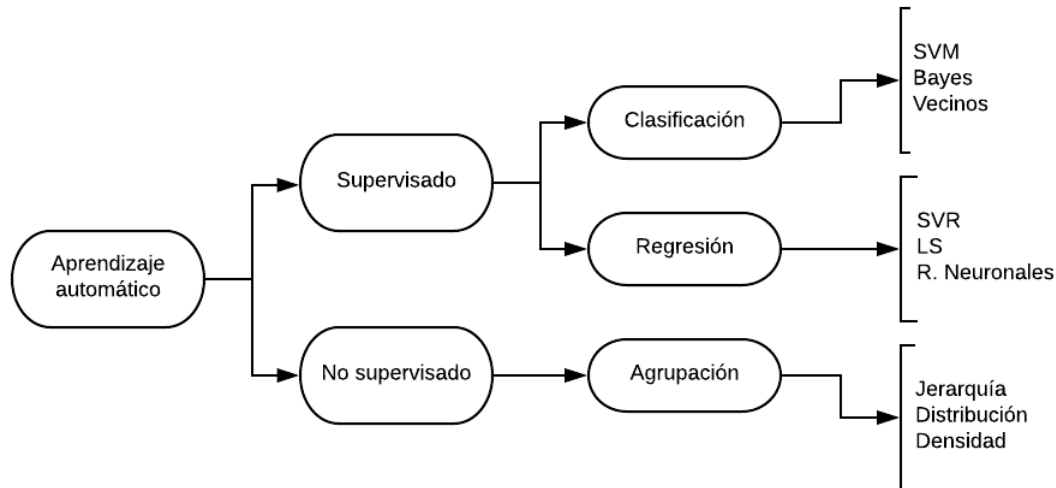


Figura 2-2. Técnicas de aprendizaje automático.

2.1.1 Aprendizaje supervisado

El aprendizaje autónomo supervisado se distingue del no supervisado en que se «entrena» el algoritmo con un histórico de datos con un determinado objetivo. Es decir, se conocen a priori el valor de la salida buscada para los datos de entrenamiento. Estos datos son conocidos como «labeled data» o datos etiquetados.

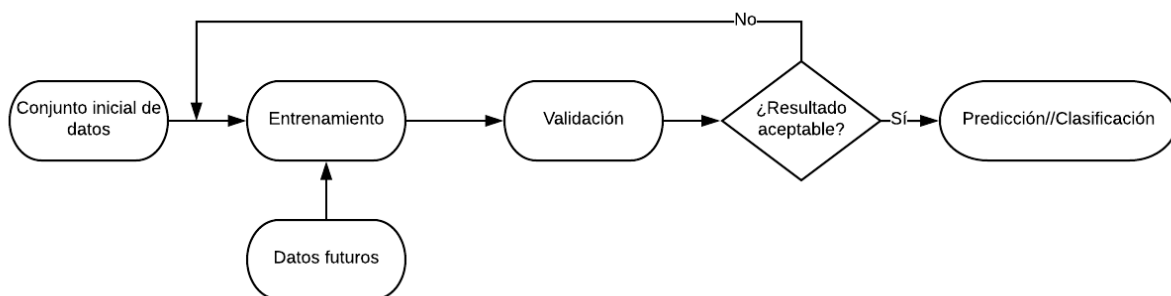


Figura 2-3. Fases aprendizaje supervisado.

Este tipo de aprendizaje puede ser empleado tanto en problemas de clasificación como de regresión. La diferencia entre ellos se encuentra en el tipo de objetivo que se persigue. En el primero, el objetivo es binario, o al menos, en cualquier caso, valores discretos que determinen la pertenencia de una muestra a un conjunto definido con una probabilidad determinada. En el segundo se desea un valor numérico que determine el valor de una variable cuantitativamente.

Los ejemplos típicos para el aprendizaje supervisado son, entre otros:

- Predicción de secuencias temporales como la demanda o el precio de la energía eléctrica.
- Procesamiento de imágenes o visión artificial.
- Clasificador de correo electrónico para detección de spam.
- Detección de la posibilidad de sufrir ciertas enfermedades

2.1.2 Aprendizaje no supervisado

En problemas de aprendizaje no supervisado, al contrario que en los de aprendizaje supervisado, no se dispone del conocimiento de la «salida» buscada de las muestras. Este clase de aprendizaje busca patrones y estructuras «escondidas» en los datos para agrupar las muestras en función de algunas características que siguen una tendencia entre las muestras ensayadas y que, a priori, puede ser desconocida. Sólo permite problemas de clasificación. La técnica más extendida es conocida como agrupación –«clustering en inglés»– .

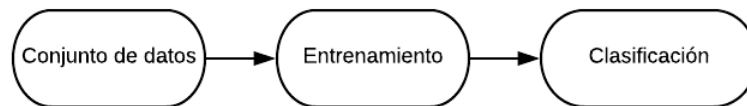


Figura 2-4. Fases aprendizaje no supervisado.

Entre las aplicaciones más destacadas del aprendizaje no supervisado se encuentran:

- Segmentación de consumidores en un mercado.
- Reconocimiento e identificación de imágenes.
- Detección de anomalías.

El mayor problema que presentan las técnicas basadas en este tipo de aprendizaje, es la aparición de correlaciones espúreas que no tienen ningún sentido. Podemos ver muchos ejemplos de esto en [1]. A continuación se muestra una gráfica obtenida de [1] que evidencia la problemática.

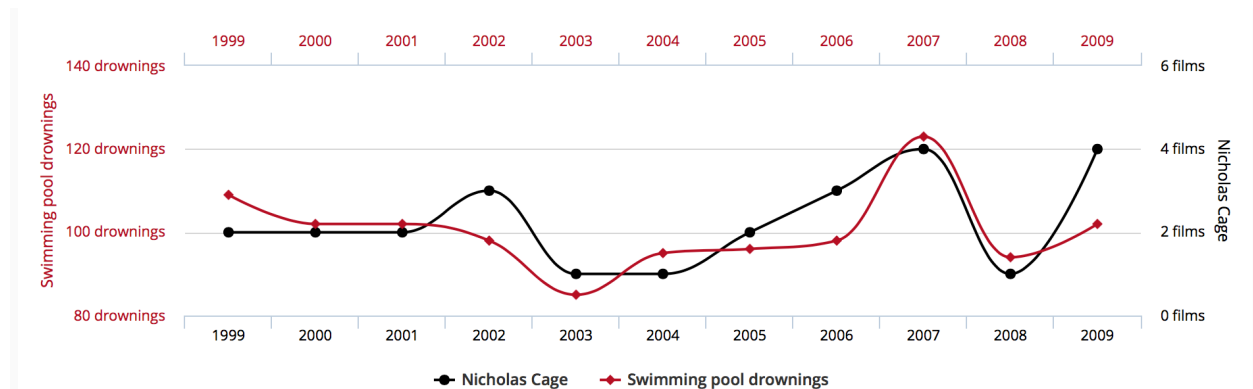


Figura 2-5. Ejemplo de correlación espúrea.

En la imagen vemos una evidente correlación absurda entre las apariciones de Nicholas Cage en películas en cada año y los ahogamientos en piscinas entre 1999 y 2009.

2.2 Métodos históricos para la regresión. Mínimos cuadrados

En 1801, Giuseppe Pazzi –astrónomo italiano– observó durante cuarenta días un nuevo planeta, hasta entonces por descubrir, llamado Ceres [2]. Muchos intentaron entonces, aunque no con mucho acierto, predecir la trayectoria de su órbita para encontrarlo de nuevo en el firmamento.

Ese mismo año, el astrónomo alemán, Franz Xaver von Zach (que ya llevaba tiempo buscando a Ceres, puesto que Johan Elert Bode sugirió su existencia en 1768 en base a la ley de Titius-Bode [3]) fue el único capaz de realizar los cálculos lo suficientemente precisos como para volver a «encontrar» a Ceres y lo hizo con el propuesto por Johan Carl Friedrich Gauss. La metodología la publicó finalmente bajo el título «Theoria motus corporum coelestium in sectionibus conicus solem ambientium» en 1809 y es lo que hoy conocemos como el método de los mínimos cuadrados.



Figura 2-6. Retrato de Johan Carl Friedrich Gauss.

2.2.1 Formulación matemática

El método de los mínimos cuadrados identifica modelos lineales como:

$$a_0 \cdot y_k + a_1 \cdot y_{k-1} + a_2 \cdot y_{k-2} + \dots + a_n \cdot y_{k-n} = 0$$

que es lo mismo que

$$y_k = m_k \cdot \theta$$

con :

$$m_k = [-y_{k-1} \quad -y_{k-2} \quad \dots \quad -y_{k-n}] - \text{Regresor.}$$

$$\theta = [a_1 \quad a_2 \quad \dots \quad a_n]^T - \text{Vector de parámetros.}$$

Podemos definir entonces el error de predicción como

$$e(k, \hat{\theta}) = y_k - \hat{y}_k = y_k - m_k \cdot \hat{\theta}$$

siendo $\hat{\theta}$ e \hat{y}_k los valores predichos del vector de parámetros y la muestra y_k .

El objetivo en todo momento es encontrar el vector de parámetros que minimice el error de predicción de la muestra y_k .

Si tenemos N pares de muestras $\{y_k, m_k\}$,

$$E(N, \theta) = Y_N - M_N \cdot \theta$$

con:

$$M_N = [m_n \quad \dots \quad m_N]^T$$

$$Y_N = [y_n \quad \dots \quad y_N]^T$$

$$E(N, \theta) = [e(n, \theta) \quad \dots \quad e(N, \theta)]^T$$

Definimos la función de coste objetivo del:

$$J(\theta) = \|E(N, \theta)\|^2 = \sum_{k=n}^N e^2(N, \theta) = (Y_N - M_N \theta)^T (Y_N - M_N \theta)$$

Encontramos la solución óptima en el sentido de los mínimos cuadrados (θ^*), minimizando la función de coste $J(\theta)$, por lo que haciendo la derivada igual a cero, obtenemos [11]:

$$\frac{dJ(\theta)}{d\theta} = 0 \rightarrow \theta^* = [M_N^T M_N]^{-1} \cdot M_N^T Y_N$$

Lo que constituye el estimador de mínimos cuadrados según lo descrito anteriormente.

Existen ciertas condiciones que deben cumplirse como que exista la matriz inversa de $M_N^T M_N$ además de que el número de muestras N , debe ser lo suficientemente grande.

2.3 Métodos *kernel*. Máquinas de soporte vectorial

Llamamos métodos *kernel* a aquellas técnicas de aprendizaje automático que emplean funciones *kernel* en sus algoritmos. Una función *kernel* nos permiten convertir un problema de clasificación no lineal en uno que sí lo es convirtiendo el espacio vectorial original en uno de dimensión superior. Ésta es la mayor bondad de estas funciones, por motivos evidentes, en problemas de aprendizaje automático supervisado.

Para que una función sea considerada función *kernel*, deben cumplir la condición Mercer [4], y, por lo tanto, deben ser funciones continuas, simétricas y definidas positivas.

Empleamos estas funciones en las máquinas de soporte vectorial (SVM, por sus siglas en inglés). Algunas de las más empleadas en este campo son:

- Lineal: $K(x_a, x_b) = x_a^T \cdot x_b$
- Polinomial: $K(x_a, x_b) = (c + x_a^T \cdot x_b)^d$
- Gaussiana: $K(x_a, x_b) = e^{-\frac{\|x_a - x_b\|^2}{\sigma^2}}$
- Sigmoidal: $K(x_a, x_b) = \tanh(c \cdot x_a^T \cdot x_b + d)$

Podemos ver que existen ciertos parámetros que requieren de sintonización. En este mismo capítulo, en la sección 2.4 se explica algún método por el que se pueden obtener.

Las máquinas de soporte vectorial son algoritmos que se sirven de estas funciones para resolver problemas de aprendizaje automático supervisado. Son un conjunto de algoritmos que se encargan de resolver estos problemas desarrollados por Vladimir Vapnik en los laboratorios AT&T.

2.3.1 Problemas de clasificación (SVM)

Los problemas de clasificación son aquellos en los que conociendo la clase a la que pertenecen cierto número de muestras, podemos entrenar un algoritmo de forma que en función de la clase y características de las muestras anteriores, clasifique en alguna de las existentes las muestras objetivo. Por tanto, la salida es discreta.

La idea es encontrar un hiperplano que sea capaz de separar los datos en dos clases.

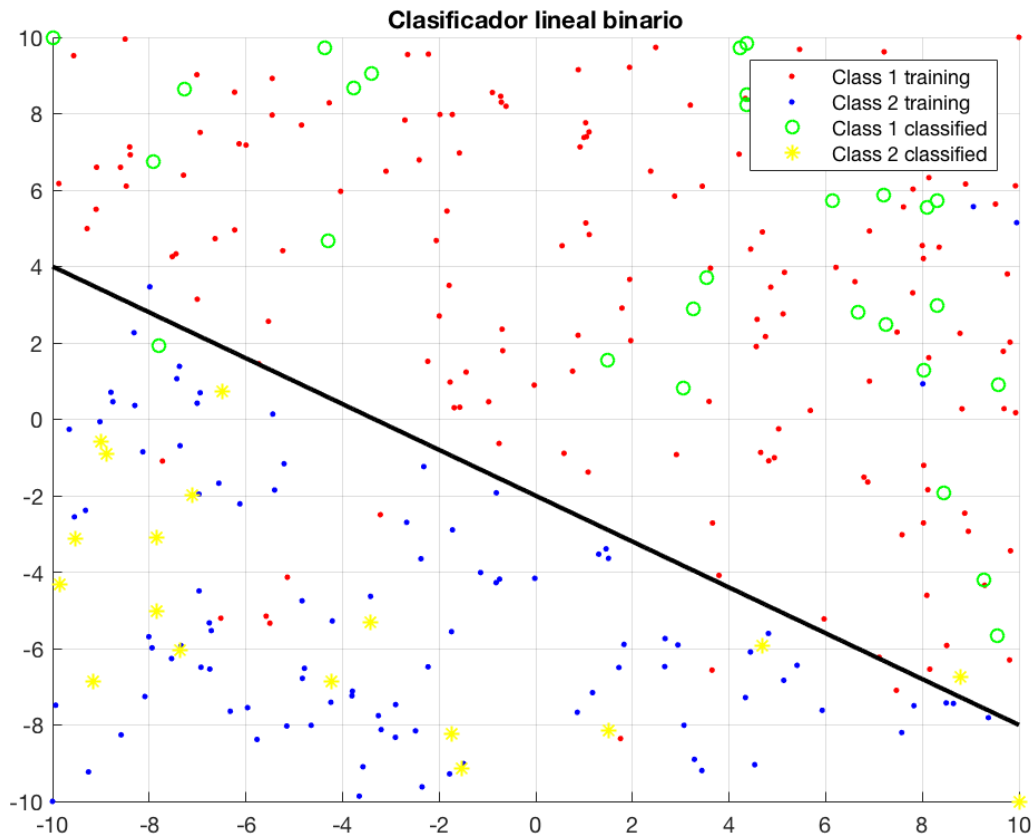


Figura 2-7. Ejemplo de SVM para clasificación.

En el ejemplo vemos cómo partiendo de una serie de muestras, tras entrenar el clasificador, podemos predecir a qué clase va a pertenecer una nueva muestra con gran acierto.

Gracias al empleo de funciones *kernel*, podemos convertir prácticamente cualquier clasificador no lineal en el espacio original en uno lineal en un espacio de dimensión superior.

2.3.2 Problemas de regresión (SVR)

Esta clase de problemas pertenece a la categoría de aprendizaje automático supervisado. Al contrario que en los problemas de clasificación, la salida no es discreta y se pretende predecir el valor numérico de una secuencia, póngase por ejemplo la estimación en un instante determinado de la temperatura que alcanzará un sistema.

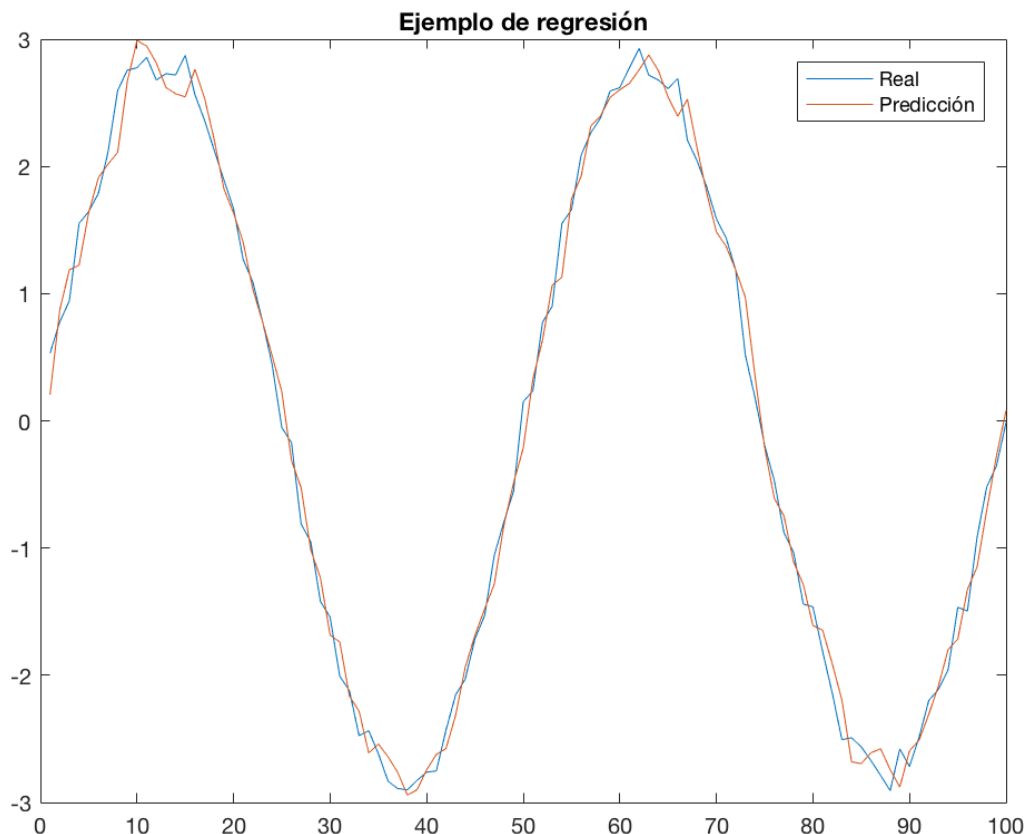


Figura 2-8. Ejemplo de resultados de SVR.

Son una extensión de los SVM en el sentido en que usan los mismos principios que dicha técnica con pequeñas variaciones. Entre las más destacadas se encuentra la ya mencionada consistente en la diferencia en el tipo de salida con respecto al problema de clasificación.

A lo largo de este trabajo se desarrollan múltiples experimentos de regresión con un método particular de éstos: los «Least Square Support Vector Machines» o LS-SVR. La principal diferencia entre este método y las SVR en general es que se pretende resolver el problema de regresión con ecuaciones lineales en lugar de con métodos de programación cuadrática, lo cual presenta una serie de ventajas indiscutibles para su empleo como puede ser la reducción en el gasto computacional.

2.3.3 Formulación matemática

Supongamos que tenemos N pares de muestras $\{x_i, y_i\}$ con $x_i \in \mathbb{R}^{n_x}$ e y_i puede ser una salida binaria o numérica en función del problema que estemos resolviendo.

Dada la función $\varphi: \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_\varphi}$ el objetivo es encontrar $\omega \in \mathbb{R}^{n_\varphi}$ y $b \in \mathbb{R}$ tal que la predicción de la muestra se pueda expresar cómo:

$$\hat{y} = \omega^T \varphi(x_i) + b$$

Podemos definir entonces los siguientes términos de error:

$$e_i = y_i - \hat{y}_i = y_i - (\omega^T \varphi(x_i) + b)$$

Las máquinas de soporte vectorial para la regresión formuladas en el sentido de mínimos cuadrados ponderados, consiguen encontrar ω y b de resolver el siguiente problema de optimización [6]:

$$\min_{\omega, b, e_1, e_2, e_3, \dots, e_N} \frac{1}{2} \left(\omega^T \omega + \gamma \sum_{i=1}^N \tau_i \cdot e_i^2 \right) \quad \text{s. a.} \quad y_i - (\omega^T \varphi(x_i) + b) - e_i = 0$$

Con:

γ – Parámetro que permite balancear el error cometido con el ajuste necesario.

τ_i – Ponderación del error cometido en cada muestra.

Se aprecia en el problema de regresión que se pretende minimizar dos términos:

- $\omega^T \omega$ – minimizar este término permite maximizar el margen de clasificación [6].
- $\sum_{i=1}^N \tau_i \cdot e_i^2$ – este término penaliza el error cometido de forma ponderada.

Si planteamos el problema de optimización dual [8], nos queda que:

$$\max_{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_N} \min_{\omega, b, e_1, e_2, e_3, \dots, e_N} \frac{1}{2} \left(\omega^T \omega + \gamma \sum_{i=1}^N \tau_i \cdot e_i^2 \right) + \sum_{i=1}^N \alpha_i \sqrt{\tau_i} \cdot (y_i - (\omega^T \varphi(x_i) + b) - e_i)$$

Dados los valores de $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_N$ óptimos para ω y todos los e_i , forzando que la derivada con respecto a éstos sea cero, tenemos;

$$\omega = \sum_{i=1}^N \alpha_i \sqrt{\tau_i} \cdot \varphi(x_i) \quad (1)$$

$$e_i = \frac{\alpha_i}{\gamma \sqrt{\tau_i}}$$

Sustituyendo esos valores en la función a optimizar y reordenando:

$$\Gamma = -\frac{1}{2} \left\| \sum_{i=1}^N \alpha_i \sqrt{\tau_i} \cdot \varphi(x_i) \right\|_2^2 + \frac{1}{2\gamma} \sum_{i=1}^N \alpha_i \sqrt{\tau_i} (y_i - b)$$

Introducimos las variables

$$s = \begin{bmatrix} \sqrt{\tau_1} \\ \vdots \\ \sqrt{\tau_N} \end{bmatrix}; \quad D = \text{diag}(s); \quad M = \begin{bmatrix} \varphi^T(x_1) \\ \vdots \\ \varphi^T(x_N) \end{bmatrix}; \quad \vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}; \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix};$$

Sustituyendo estas variables en Γ tenemos que

$$\Gamma = \frac{-1}{2} \vec{\alpha}^T \left(D M M^T D + \frac{1}{\gamma} I \right) \vec{\alpha} + \vec{\alpha}^T (D Y - b s)$$

Igualando a cero la anterior ecuación y reescribiendo de forma matricial

$$\begin{bmatrix} 0 & s^T \\ s & D M M^T D + \frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ D Y \end{bmatrix}$$

y de la ecuación (1)

$$\omega = M^T D \vec{\alpha}$$

Podemos ver que

$$MM^T = \begin{bmatrix} \varphi^T(x_1)\varphi(x_1) & \varphi^T(x_1)\varphi(x_2) & \dots & \varphi^T(x_1)\varphi(x_N) \\ \varphi^T(x_2)\varphi(x_1) & \varphi^T(x_2)\varphi(x_2) & \dots & \varphi^T(x_2)\varphi(x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi^T(x_N)\varphi(x_1) & \varphi^T(x_N)\varphi(x_2) & \dots & \varphi^T(x_N)\varphi(x_N) \end{bmatrix}$$

Definiendo la función *kernel* $K(\cdot, \cdot) : \mathbb{R}^{2n_x} \rightarrow \mathbb{R}$ como

$$K(x_a, x_b) = \varphi^T(x_a)\varphi(x_b)$$

por lo que

$$[M^T M]_{i,j} = K(x_i, x_j)$$

y tenemos la solución óptima como:

$$\begin{bmatrix} 0 & s^T \\ s & DKD + \frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ DY \end{bmatrix}$$

Para el cómputo de la salida, tenemos que:

$$\hat{y} = b + (D\vec{\alpha})^T \cdot \begin{bmatrix} K(x_1, x) \\ \vdots \\ K(x_N, x) \end{bmatrix}$$

En problemas de clasificación binaria, el signo del valor de \hat{y} determinará la pertenencia a una u otra clase. En problemas de regresión, \hat{y} proporciona el valor estimado de la muestra a predecir.

2.4 Técnicas para la elección y ajuste de regresores e hiperparámetros

Como se ha visto, tanto en las funciones *kernel* como en la formulación matemática del problema, existen parámetros que es necesario sintonizar y de los que depende el resultado del problema. Para evitar escoger de forma arbitraria dichos parámetros y aumentar la eficiencia de la predicción, existen diversas técnicas que ofrecen métodos de ajuste, entre ellas, la más sencilla, probablemente sea seleccionar un conjunto de parámetros y comparar estos resultados entre sí. Para ello es necesario un procedimiento fiable, objetivo y estándar de obtener el error de predicción.

2.4.1 Validación cruzada

Probablemente, una de las técnicas más extendidas en su empleo y a la vez más sencilla para la elección y ajuste de regresores mediante la comparación del error cometido en la predicción con distintos parámetros sea la validación cruzada.

Se emplea principalmente para estimar errores y variables estadísticas de la forma menos sesgada posible.

La idea es obtener una generalización del error a partir de un número de muestras finitas. Existen varios tipos de validación cruzada distintos. En este trabajo se emplea el método de validación cruzada de k-iteraciones.

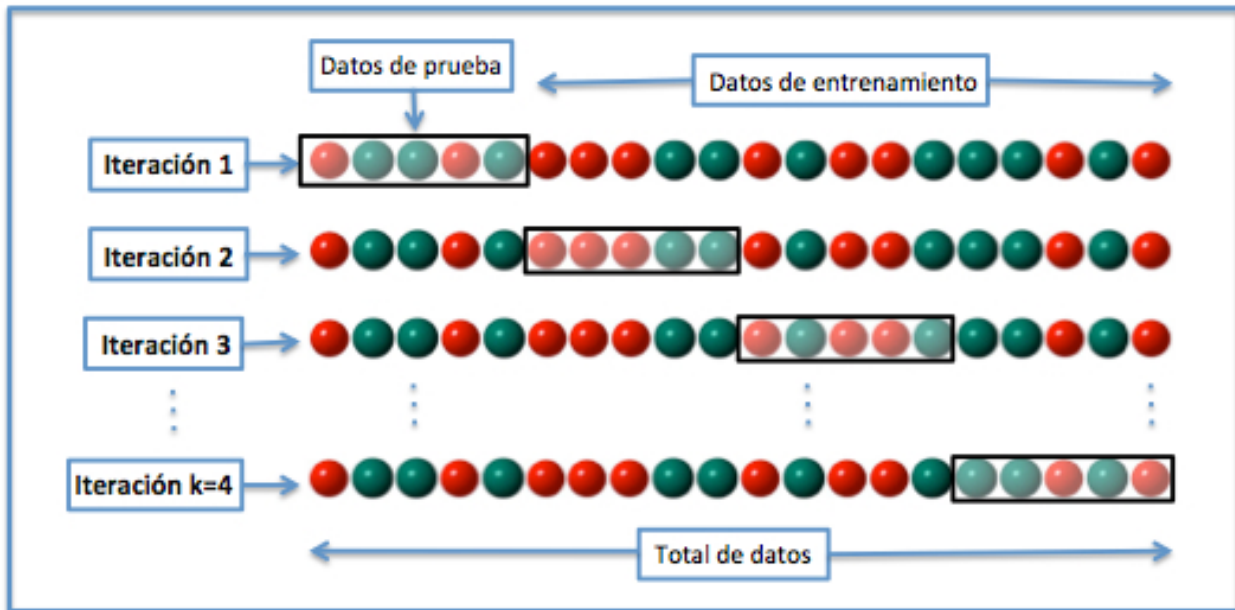


Figura 2-9. Esquema validación cruzada de k-iteraciones con $k = 4$.

El método consiste en dividir el conjunto de datos disponible en k «pliegues» o conjuntos (del nombre del método en inglés, « k -fold cross validation») y emplear en cada una de las k iteraciones un set de validación distinto. El resto de datos que no van a ser empleados en la validación, son utilizados para el entrenamiento del algoritmo.

Por cada una de las iteraciones (k), obtenemos un error (e_i). El error generalizado (E_G) del regresor y parámetros empleados será la media del error de cada iteración. Esto es:

$$E_G = \frac{1}{k} \sum_{i=1}^k e_i$$

Debido al gran gasto computacional que puede llegar a suponer este tipo de validación cruzada, se suelen emplear valores de k entre 5 y 10 [9].

En el caso que nos ocupa, en el proceso de validación cruzada vamos a determinar tanto el error cuadrático medio (ECM o MSE, siglas en inglés) como el error estándar de la media (EEM o SEM, siglas en inglés).

El MSE nos proporciona información sobre el error que podemos esperar del algoritmo en futuras predicciones [10] y se calcula como la media del cuadrado de los errores de cada predicción en el número de muestras N :

$$MSE = \frac{1}{N} \sum_{i=1}^N e_i^2 = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

El SEM nos da una idea de cuál va a ser la desviación típica del MSE en futuras predicciones [10]. Para un conjunto de validación de N muestras y desviación estándar del error cuadrático σ , tenemos que:

$$SEM = \frac{\sigma}{\sqrt{N}}$$

3 PRESENTACIÓN DE LOS PROBLEMAS

En este capítulo, se presentan los diversos problemas de regresión que se han resuelto. Se han elegido estos y no otros debido a la cantidad de datos disponibles principalmente. Además, uno de los tres problemas propuestos es un ejercicio sintético, de escasa utilidad práctica más allá de comprobar el buen funcionamiento de los algoritmos desarrollados además de poder emplearse como ejemplo en los conceptos teóricos que aparecen en los capítulos previos. Por el contrario, los otros dos problemas que se resuelven en el capítulo cinco, sí constituyen ejemplos reales, con abundantes datos obtenidos de páginas web y fuentes confiables en ambos casos.

El objetivo del trabajo para todos los problemas que se proponen a continuación, no es otro que predecir la evolución futura de cada una de las secuencias temporales propuestas en cada caso además de, puntualmente para el ejercicio sintético, comprobar el buen funcionamiento y robustez de los algoritmos desarrollados.

3.1 Predicción de una señal modulada en amplitud. Problema sintético

La modulación de señales es una técnica que se emplea en telecomunicaciones para transmitir información a través de una onda portadora. Consiste en la superposición de señales que contienen dicha información sobre una señal portadora con una longitud de onda mucho menor que la que presenta la que deseamos transmitir.

Estas técnicas son interesantes debido a las ventajas que ofrece su empleo en temas relacionados con la inmunidad frente al ruido e interferencias, tamaño de los equipos emisores y receptores y otros tantos, frente a la transmisión únicamente de la señal que contiene la información.

De las muchas técnicas que existen para la modulación de señales, se ha elegido la modulación en amplitud, aunque hubiera sido igualmente válida cualquier otro tipo.

Como se ha dicho en la introducción del capítulo, este no pretende ser un estudio de aplicación real sino, más bien, un ejercicio teórico del que conocemos todos los parámetros para así poder comprobar el buen funcionamiento de los algoritmos desarrollados además de servir como ejemplo ilustrador de los conceptos teóricos desarrollados a lo largo de todo el trabajo.

Se ha elegido este problema y no otro por no ser demasiado simple en aspectos como por ejemplo, presentar cambios medianamente abruptos que permitan comprobar la robustez y bondad del regresor además de los parámetros de diseño. Se le ha añadido también ruido aleatorio para complicar un poco la predicción, ya que al ser puramente periódica, una vez ensayado un ciclo, tan solo con el punto del periodo en el que nos encontremos es posible predecir con exactitud, eligiendo un buen regresor, el valor que la señal tomará a lo largo del set de validación.

Con todo esto, la señal modulada en amplitud objeto de estudio de este trabajo viene descrita por la función:

$$y(t) = \left[1 + \frac{A_m}{A_p} y_m(t) \right] \cdot A_p \cdot y_c(t) + r(t)$$

dónde:

A_m – Amplitud de la onda moduladora.

A_p – Amplitud de la onda portadora.

$r(t)$ – Ruido aleatorio introducido en la señal a predecir.

$y_m(t)$ – Señal moduladora.

$y_c(t)$ – Señal portadora.

Las señales moduladora y portadora, son las siguientes:

$$y_m(t) = \sin(2\pi\phi_m t)$$

$$y_c(t) = \sin(2\pi\phi_c t)$$

Con ϕ_m y ϕ_c las frecuencias de las ondas de modulación y portadora respectivamente siempre sujeto a la condición $\phi_m < \phi_c$.

Gráficamente, la señal a predecir, así como la portadora y la señal moduladora, presentan la siguiente forma:

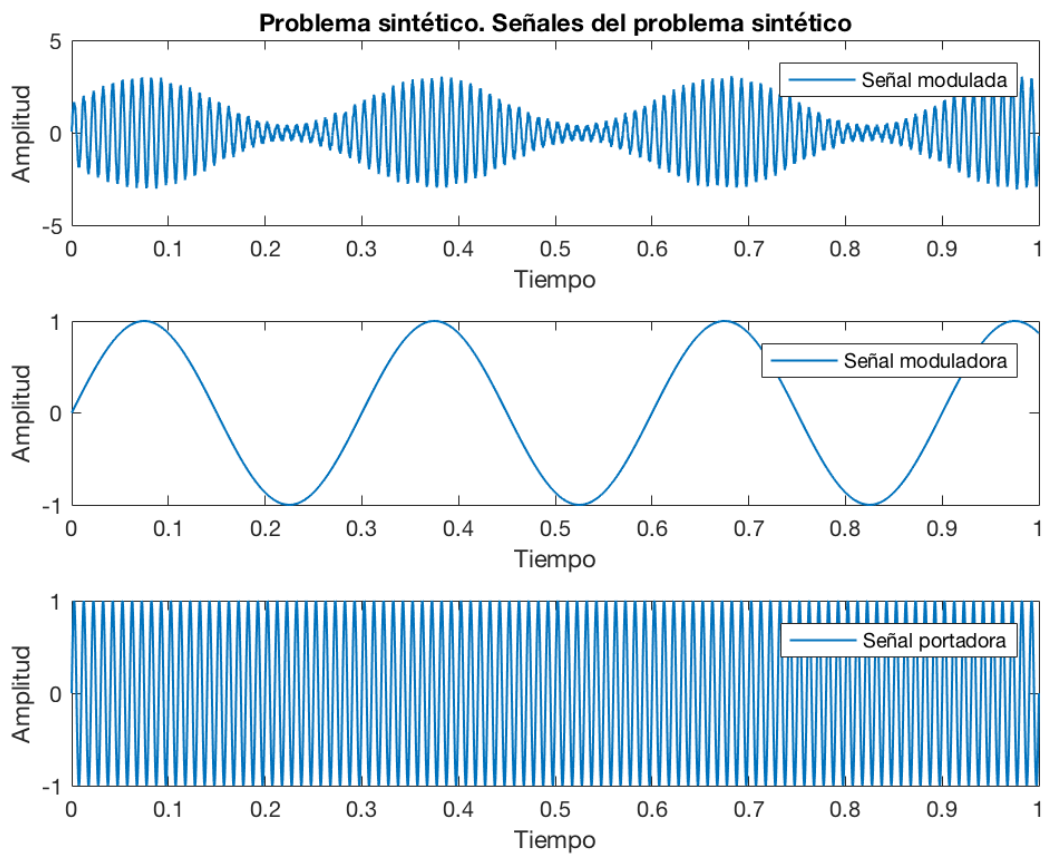


Figura 3-1. Señales del problema sintético.

De forma desglosada y en detalle:

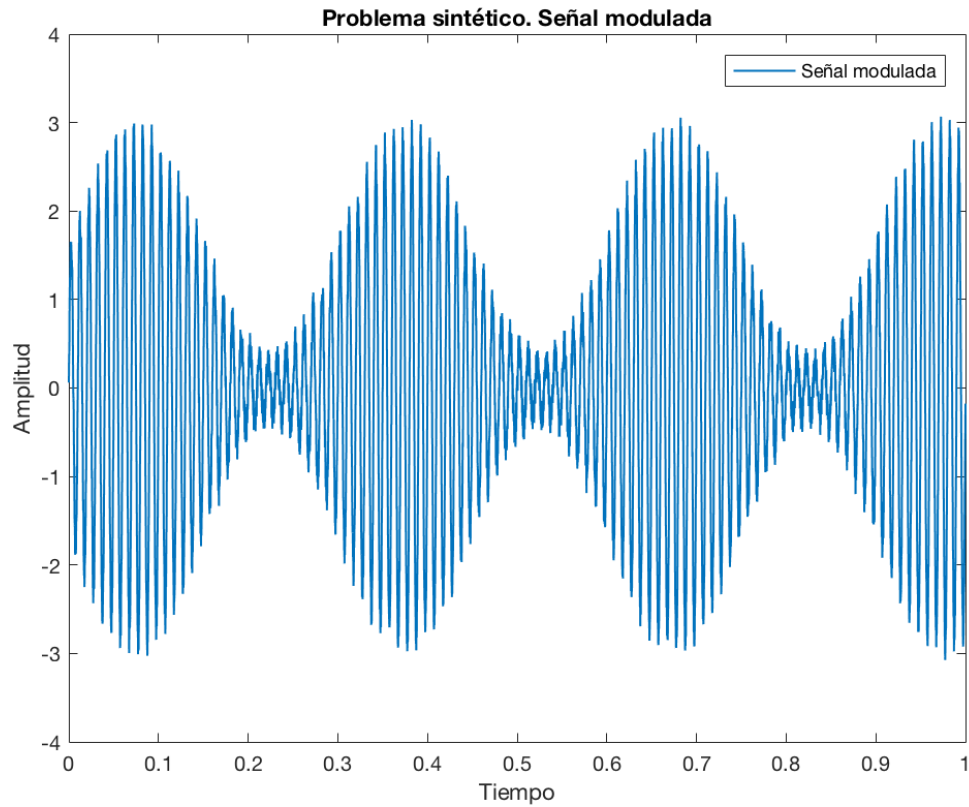


Figura 3-2. Detalle de señal modulada.

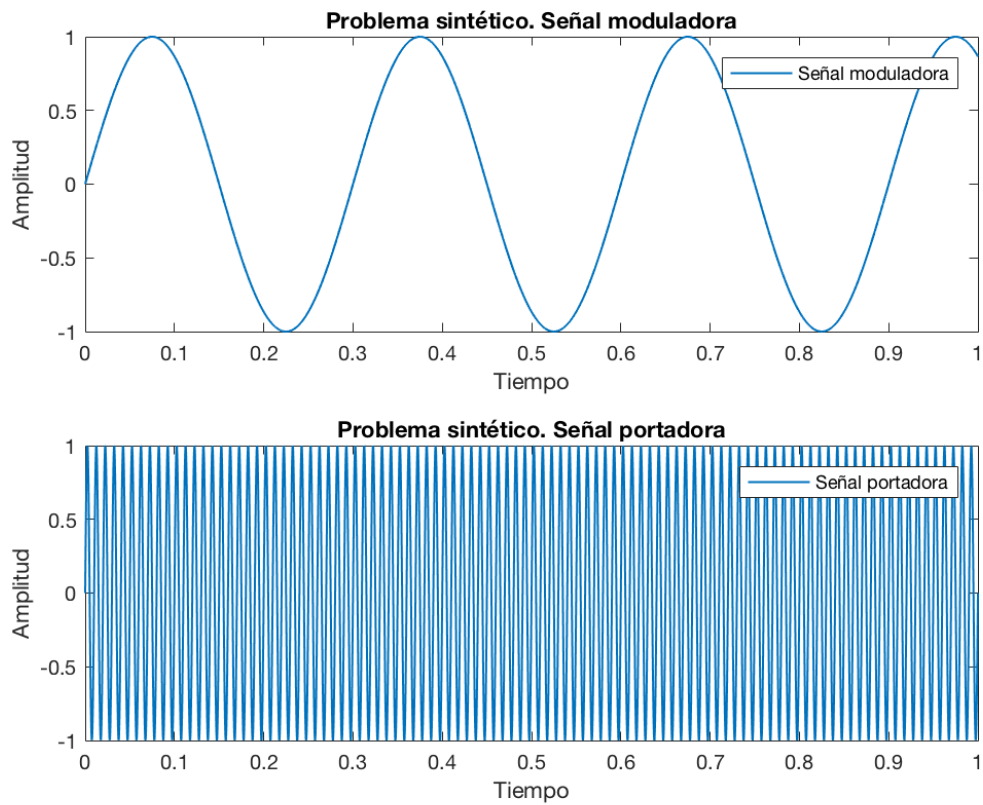


Figura 3-3. Detalles de señales portadora y moduladora.

3.2 Predicción del número de manchas solares

Las manchas solares son producidas por perturbaciones en el campo magnético de la estrella que emana a la fotosfera. Este suceso se puede observar como regiones oscuras en la superficie solar de un tamaño que podría ser equivalente al de un planeta. El hecho de que se vean más oscuras que el resto de la superficie es debido al contraste que se genera entre la temperatura de la mancha y la superficie.

La aparición de estos fuertes campos magnéticos entorno a las manchas solares producen regiones activas en la estrella que pueden provocar el fenómeno conocido como «tormentas solares». Este incremento en la actividad solar está relacionado con la actividad geomagnética en La Tierra, que presenta un ciclo idéntico al solar.

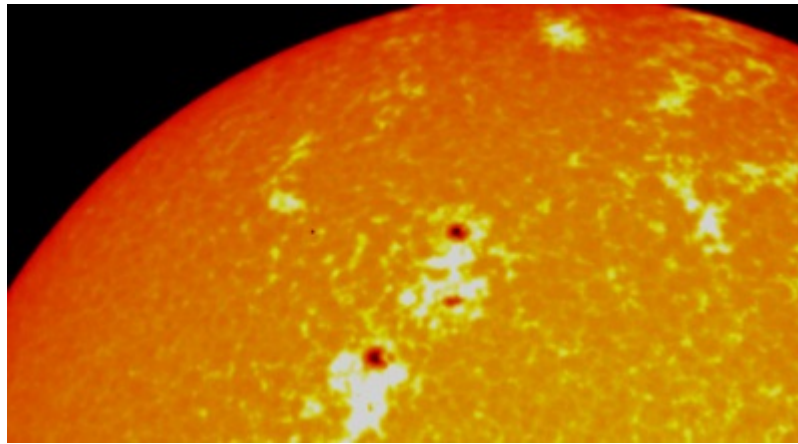


Figura 3-4. Manchas solares vistas desde el centro de vuelo espacial de la NASA Goddard.

Desde alrededor del año 800 a.C., existen casos documentados de manchas solares por astrónomos chinos de forma ocasional ya que las creían augurios de eventos importantes. Con la aparición del telescopio entorno al 1600, en occidente, se empezó a estudiar este fenómeno aunque existían algunas discrepancias entre los astrónomos de la época entorno a qué era este suceso. No obstante, permitió realizar las primeras estimaciones del periodo rotacional del sol debido al movimiento de las manchas solares observadas.

En 1843, el astrónomo aficionado alemán Samuel Heinrich, descubrió que éste fenómeno presentaba un cierto carácter cíclico que estimó en unos diez años (en la actualidad se sabe que son 11 años de media lo que dura un ciclo solar) en el que el número de manchas solares presentaba un máximo y un mínimo bien diferenciado.

No fue hasta 1948 cuando el astrónomo y matemático suizo Johan Rudolf Wolf, propuso la mejor forma de contar el número de manchas, problema que es complejo debido a la variabilidad en el tamaño, agrupación y posición de éstas. Su idea para computar el número de manchas solares diarias consiste en sumar el número observado ese día más diez veces la cantidad de agrupaciones de éstas que existan.

Se define entonces el número de manchas solares –SSN, por sus siglas en inglés–, también llamado número de Wolf, fórmula de Wolf o número de manchas solares de Wolf– como:

$$SSN = k(10a + i)$$

dónde:

k – Factor que varía en función de aspectos como el instrumental empleado en la medida.

a – Número de agrupaciones de manchas solares.

i – Número de manchas solares independientes observadas.

En la actualidad, el Observatorio Real de Bélgica se encarga de mantener una base de datos actualizada de forma diaria y ha revisado las series históricas añadiendo información necesaria para el correcto procesamiento de los datos.

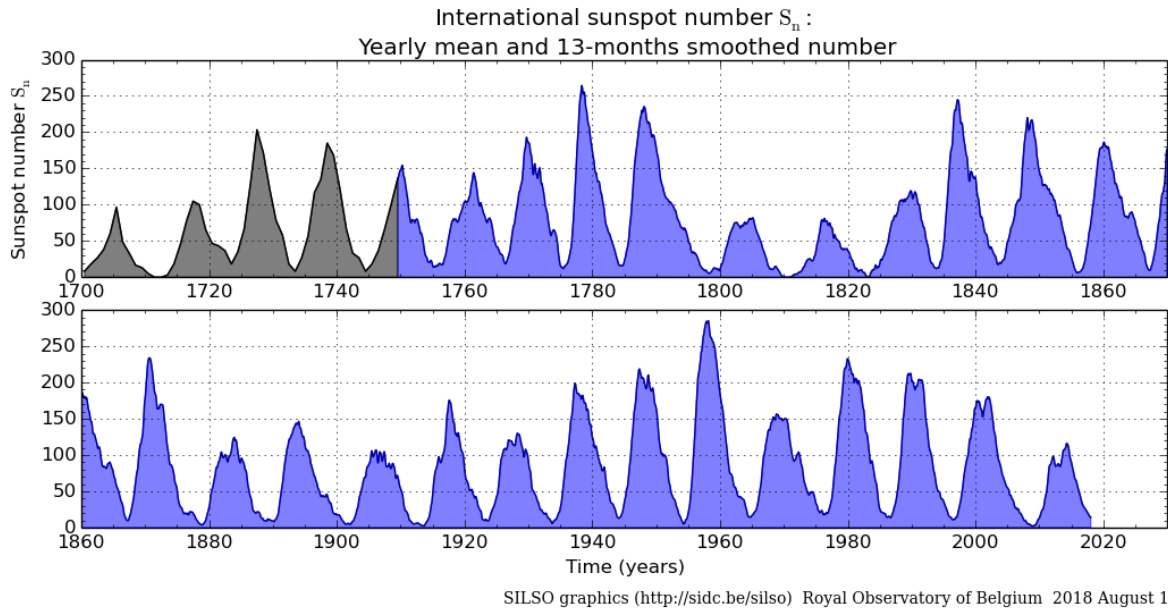


Figura 3-5. Serie histórica de medias anuales del SSN.

Pese a la sencillez de este número, se sigue empleando hoy en día aunque haya otras formas de medir la cantidad de manchas solares más sofisticadas debido a la gran cantidad de datos que existen (unos 400 años) y que permiten resolver problemas como el que se plantea en este documento, consistente en predecir las secuencias del SSN futuras debido al impacto que tiene éste tanto en el clima terrestre como en telecomunicaciones espaciales.

3.3 Predicción de la demanda de energía eléctrica

La predicción de la demanda de energía eléctrica, puesto que ésta no se puede almacenar de forma eficiente, es un problema realmente importante para programar la producción eficazmente. Además, el precio de compra del KW/h fijado por subasta pública también guarda una estrecha relación con la demanda por lo que la predicción de energía es un asunto que involucra a diversos agentes y una buena solución, puede proporcionar grandes beneficios económicos.

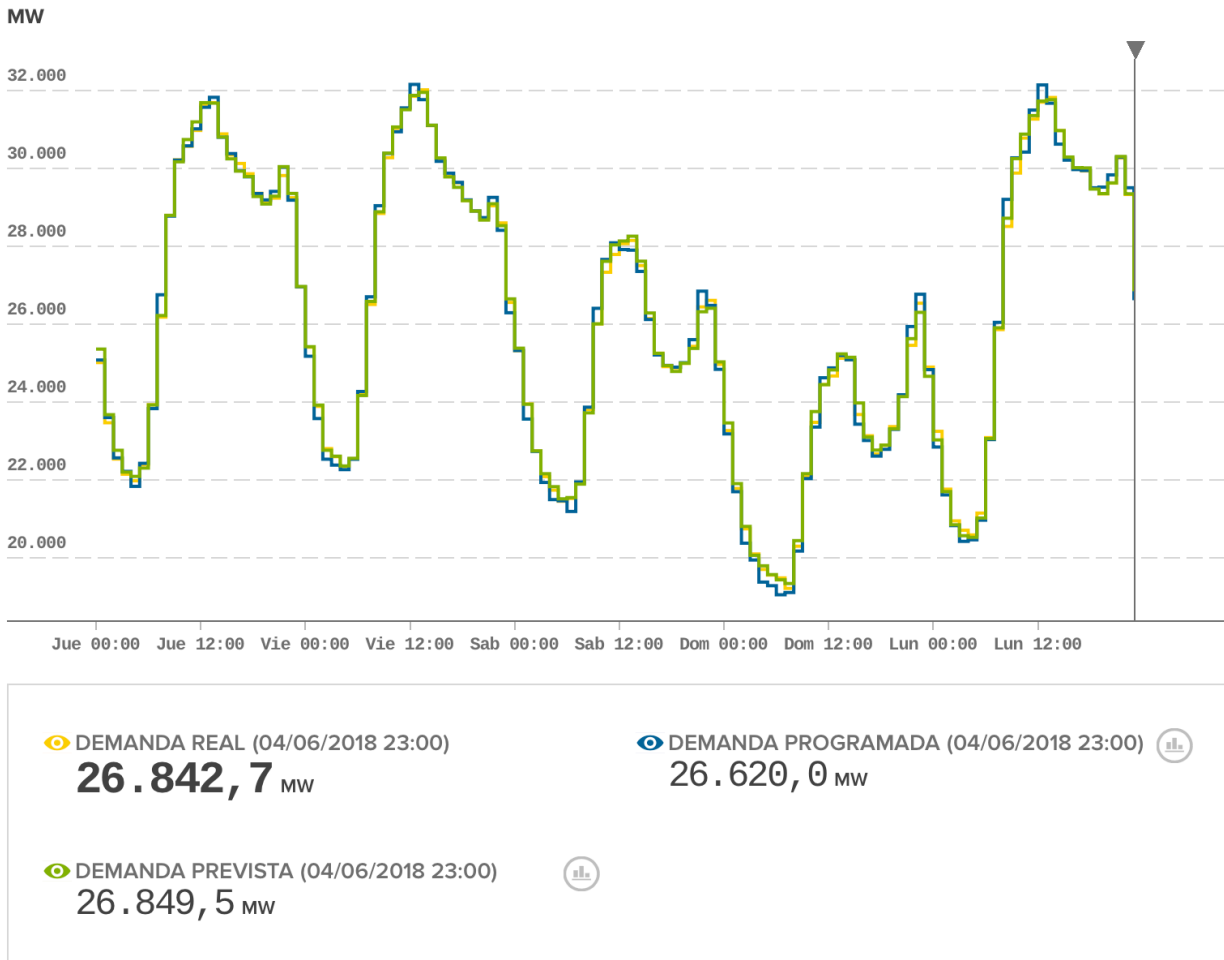


Figura 3-6. Recorte de la DEL de esios.ree.es.

Solucionar este problema no es trivial. Existen muchos factores que influyen en el consumo de energía, algunos de ellos, se listan a continuación:

- Factores ambientales. Por ejemplo, pueden influir en la demanda de energía eléctrica temperatura, humedad o altitud respecto al nivel del mar.
- Estacionalidad. No se consume la misma cantidad de energía en marzo que en enero.
- Bonanza económica. La demanda y producción de energía eléctrica guarda una importante relación con la economía del país. Una economía más «fuerte» implica una mayor producción y adquisición de bienes de consumo que provocan un aumento en las necesidades energéticas de la nación. Por el contrario, cuando un país se encuentra en periodos de crisis económica, las necesidades energéticas se disminuyen debido a la caída de la producción, el consumo y la dificultad de algunos agentes sociales para hacer frente a esos gastos.
- Desarrollo de la región. No se consume, y por tanto no se demanda, tanta energía eléctrica en una región industrializada como en una agrícola, por ejemplo.

En resumen, podríamos concluir que existen tres grupos, que guardan correlación entre sí, de factores que afectan a la demanda energética; factores climáticos, factores geográficos y factores demográficos.



Figura 3-7. Subestación eléctrica de la compañía Krostec, México.

Los resultados arrojados por el estudio, serán únicamente consecuencia de la secuencia histórica de consumo de energía eléctrica. No se incluirán, por tanto, factores ambientales, demográficos o geográficos para las predicciones de la serie. Esto es debido al tiempo en el que se hace la predicción. Es decir, como el estudio se va a realizar haciendo predicciones a una hora vista, estos factores no son excesivamente determinantes ya que las variaciones entre una hora y otra, por ejemplo en cuanto a la temperatura y humedad, no son demasiado relevantes para el estudio.

El escenario en el que se va a desarrollar este estudio no es otro que el territorio nacional Español, excluyendo Islas Canarias, Islas Baleares, Ceuta y Melilla. Se van a analizar series históricas de demanda de energía eléctrica global, es decir, sin diferenciar entre las fuentes que producen esta energía ni el tipo de consumo (industrial, residencial y servicios son las distinciones más habituales que se suelen emplear).

4 DESCRIPCIÓN DEL TRABAJO DESARROLLADO

En este capítulo de la memoria, se van a encontrar todas las herramientas, material, etc. Que ha sido necesario para el desarrollo del trabajo. Lo primero que se va a describir son las herramientas de hardware y software empleadas.

En la tercera sección vamos a encontrar una serie de algoritmos, y explicaciones de los «scripts» que se han desarrollado con el objetivo de concluir esta memoria. La tercera sección incluye lo mismo que la segunda, pero en este caso, con las funciones desarrolladas.

La última y más importante sección de este capítulo, consta de toda la información necesaria para la obtención de los datos empleados en el estudio así como el tratamiento previo que se ha realizado a dichos datos para poder ensayarlos adecuadamente.

4.1 Herramientas empleadas

El «hardware» empleado para la realización de este estudio ha sido el siguiente:

- Equipo: MacBook Pro (Retina 13 pulgadas, principios de 2015)
- Procesador: 2,7 GHz Intel Core i5
- Memoria RAM: 8 GB 1867 MHz DDR3
- Sistema operativo: macOS Sierra Versión 10.12.6 (16G1314)

En cuanto al software empleado, se ha optado por elegir MatLab (MAtrix LABoratory) por ser esta una herramienta eficiente para resolver problemas matemáticos además de haber sido un programa bastante trabajado a lo largo del grado.

MatLab cuenta con un lenguaje (lenguaje M, interpretado) y un entorno de programación propietarios desarrollados por MathWorks y Cleve Moler.

La versión empleada en este trabajo ha sido la v.R2017a (9.2.0.538062)

4.2 Scripts

En esta sección se describen los programas desarrollados junto a sus algoritmos si son necesarios para la correcta comprensión del script realizado. Todos los códigos empleados en este trabajo se han desarrollado para tal propósito. En ningún caso se han empleado programas de terceros.

4.2.1 Generación de datos del problema sintético

Se ha desarrollado un programa que genera la señal objeto de estudio del problema «Predicción de una señal modulada en amplitud. Problema sintético».

En este script simplemente se definen los parámetros característicos de la señal a generar y la crea añadiendo un error aleatorio de ± 0.2 puntos sobre la señal creada.

Además, genera las figuras necesarias para incluir en esta memoria y guarda los valores de la señal en un archivo «.mat» para su posterior tratamiento».

Este script está implementado en el archivo «genera_onda_modulada.m».

4.2.2 Predicción de onda modulada en amplitud

Se ha implementado este programa, con el propósito de entrenar y validar un regresor predeterminado para el problema SMA. Además, calcula el error cometido y pinta las figuras necesarias.

En este caso, el entrenamiento y la predicción lo realiza sobre las muestras disponibles sin escalar.

Además, guarda el valor de todas las variables en un archivo para su posterior tratamiento sin tener que repetir la simulación.

Se puede encontrar el código desarrollado en «pred_SMA.m».

4.2.3 Predicción del número de Wolf

Se ha creado un script para que realice la misma tarea que «pred_SMA.m» pero, en este caso, para el problema SSN.

Podemos encontrar su implementación en «SSN_pred.m».

4.2.4 Predicción de la demanda de energía eléctrica

Igualmente, para el problema de la demanda de energía eléctrica se ha desarrollado un script que realiza la misma función que «pred_SMA.m» y «SSN_pred.m».

Podemos encontrarlo en «pred_DEL.m».

4.2.5 Núcleo para ensayar SMA con validación cruzada

Se ha desarrollado un script para obtener todos los valores necesarios para hacer el análisis de todos los regresores y conjuntos de parámetros propuestos en el problema «Predicción de una señal modulada en amplitud. Problema sintético».

Para ello hace las llamadas a funciones oportunas y las declaraciones necesarias. En él, se definen los regresores y parámetros a emplear en los ensayos.

Al terminar, guarda el valor de todas las variables necesarias para poder trabajar con ellas sin tener que realizar de nuevo la simulación, ya que es bastante costosa en tiempo de computo.

Podemos ver en la siguiente figura de forma esquemática el comportamiento del «script».

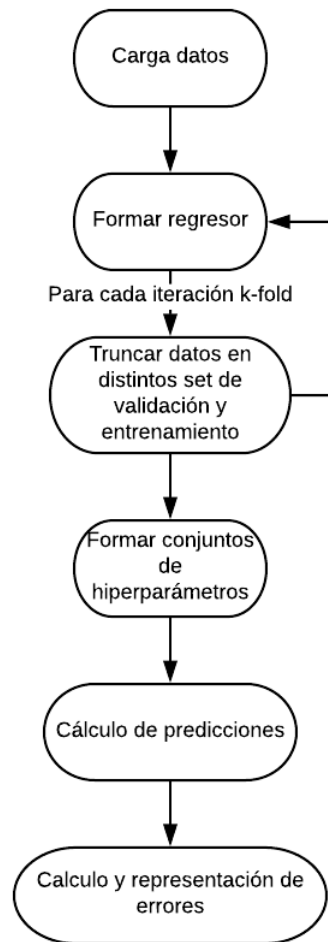


Figura 4-1. Algoritmo script principal.

La implementación de este algoritmo se puede encontrar en «svr_core_SMA.m»

4.2.6 Núcleo para ensayar SSN con validación cruzada

Este programa se ha realizado con la misma intención que «svr_core_SMA.m» definido en la sub-sección 4.2.5 de este mismo capítulo. La diferencia se encuentra en que en este caso, el programa está enfocado a resolver el problema «Predicción del número de manchas solares».

Se encuentra su desarrollo en «svr_core_SSN.m».

4.2.7 Núcleo para ensayar DEL con validación cruzada

Este script ha sido realizado para la misma labor que «svr_core_SMA.m» y «svr_core_SSN.m» definidos en las sub-secciones 4.2.5 y 4.2.6 de este mismo capítulo respectivamente. La diferencia se encuentra en que en este caso, el programa está enfocado a resolver el problema «Predicción de la demanda de energía eléctrica».

El código desarrollado puede encontrarse en «svr_core_DEL.m».

4.3 Funciones

Salvo las funciones básicas de MatLab como «mean» o «load», no se ha empleado ninguna función desarrollada por terceros en este trabajo. A continuación se muestra una breve explicación del funcionamiento de las desarrolladas.

4.3.1 Preparación de datos

Esta función se ha desarrollado con el objetivo de, como su propio nombre indica, preparar los datos para su posterior tratamiento. Para ello, al llamarla debe introducirse un argumento numérico que corresponde a cada uno de los tres problemas según la siguiente equivalencia:

- 1 – «Predicción de una señal modulada en amplitud. Problema sintético»
- 2 – «Predicción del número de manchas solares».
- 3 – «Predicción de la demanda de energía eléctrica ».

Una vez la función reconoce el argumento de entrada y sabe qué datos se desean tratar, carga el archivo correspondiente.

De entre todos los datos que pueda contener dicho archivo, almacena en una variable los correspondientes a la secuencia temporal y, los escala entre dos valores a determinar.

Por último, elimina las muestras más tempranas de forma que la longitud de la serie sea el primer múltiplo de 5 posible ya que esto es necesario para dividir toda la secuencia en 5 partes iguales.

La función devuelve dos variables que almacenan tanto la secuencia temporal como los datos originales, truncados.

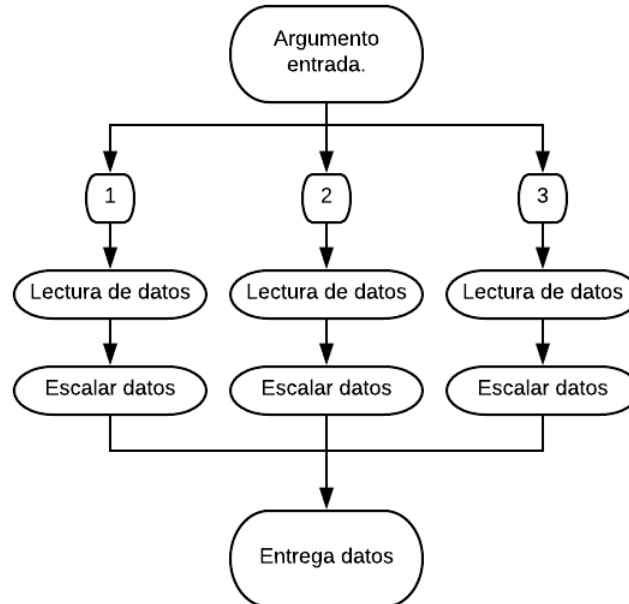


Figura 4-2. Algoritmo «preparadatos.m».

La implementación se puede encontrar en «preparadatos.m».

4.3.2 Generación de los conjuntos para validación cruzada

Se ha desarrollado esta función que crea 5 máscaras lógicas del tamaño de la secuencia temporal que se esté tratando para poder establecer los distintos sets de validación y entrenamiento en las iteraciones del

algoritmo de validación cruzada. Necesita como argumento de entrada la secuencia temporal que se desee tratar y devuelve una matriz de cinco columnas y tantas filas como muestras tenga la secuencia. Cada columna contiene una de las máscaras necesarias para la partición de los datos.

Se puede encontrar su código en «kfoldCV_mask.m».

4.3.3 Entrenamiento y validación de conjuntos

Esta función se ha desarrollado para evitar código repetido en la medida de lo posible. A través de las funciones `c_krn1.m` y `v_krn1.m`, calcula los valores predichos de la secuencia temporal que se esté tratando. Necesita como entrada los distintos sets de validación y entrenamiento así como los parámetros necesarios para la regresión.

Se puede comprobar su implementación en el archivo «kernels.m».

4.3.4 Cálculo de *kernels*

Se ha desarrollado una función que calcula la matriz *kernel* para poder obtener b y α y devolverlo. También devuelve la matriz K dentro de una estructura que contiene todos los parámetros necesarios para poder validar el resultado.

Necesita como argumento de entrada los hiperparámetros que definen el modelo, el regresor montado para el set de entrenamiento y el valor de la serie a lo largo del set.

Su desarrollo se encuentra en el documento «c_krn1.m».

4.3.5 Cálculo de predicciones

Esta función se ha implementado con el objetivo calcular el valor estimado de la serie temporal para un determinado set de validación.

Sus argumentos de entrada son la estructura k obtenida tras ejecutarse `c_krn1.m`, b , α , y los parámetros del modelo. Además, es necesario introducirle el regresor montado para la validación y el montado para el entrenamiento.

Devuelve una secuencia con los valores predichos en el set de validación.

Puede encontrarse desarrollada en el fichero «v_krn1.m».

4.3.6 Cálculo de errores

Se ha desarrollado una función que necesita como argumentos de entrada todos y cada uno de los errores absolutos cometidos en cada uno de los set de validación. Con esa información, calcula los MES y SEM de cada iteración para la validación cruzada, después realiza la media de estos y los imprime por pantalla. También calcula los mínimos valores arrojados por la propia función.

La implementación de esta función se encuentra desarrollada en «calculaerrores.m»

4.3.7 Representación de errores

Como su propio nombre indica, el objetivo de esta función que se ha implementado es, una vez se le han pasado los errores absolutos calculados como la medida real menos la predicción de ésta para cada muestra genera nueve figuras correspondientes a los histogramas de error de cada combinación de regresor-parámetros. En cada figura hay seis «subplots»; uno para cada iteración del proceso de validación cruzada y otro para el valor medio de éstos.

El desarrollo de esta función puede encontrarse en «pintaerrores.m».

4.4 Obtención de datos y tratamiento

En esta sección se detallará para cada problema de dónde y cómo se han obtenido los datos y el tratamiento que han recibido para poder procesarse adecuadamente.

4.4.1 Señal modulada en amplitud

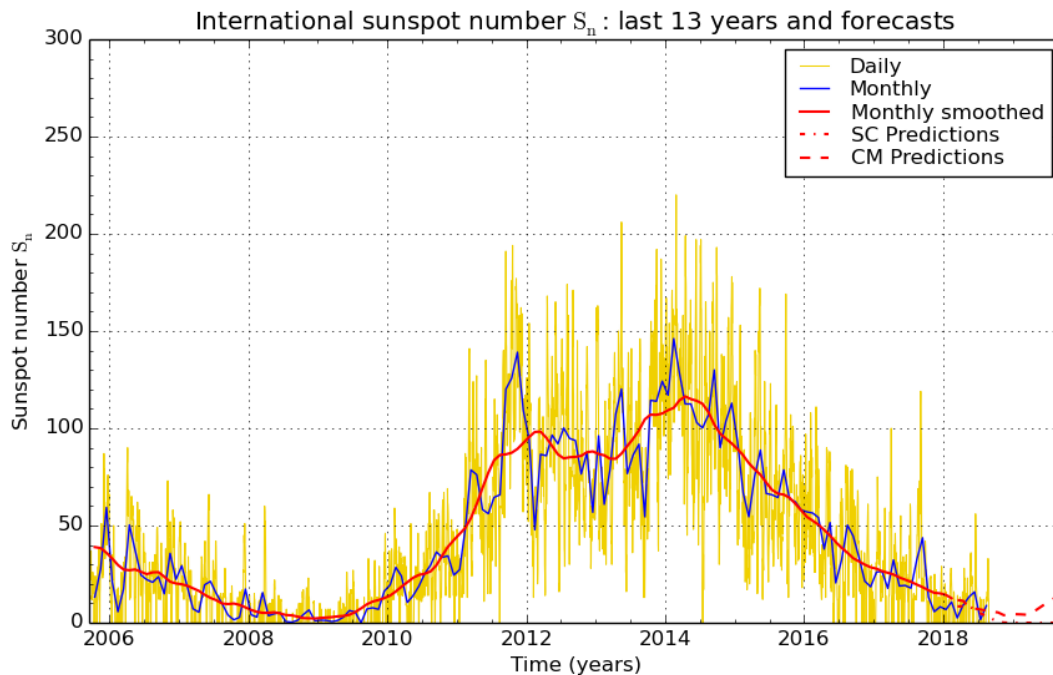
Los datos necesarios para generar esta secuencia temporal no se han obtenido de forma externa. Al tratarse de un problema sintético, se han generado con el script `genera_onda_modulada.m`

El archivo que contiene esta secuencia no es más que un vector correspondiente al valor de la amplitud de la señal en cada una de las cinco mil muestras que se generan.

4.4.2 Número de manchas solares

Las series históricas del número de manchas solares están disponibles en la web www.sidc.be y son públicas. Únicamente se exige por parte del que las use, citar la fuente de las mismas.

Esta página web, pertenece al Centro de Análisis de Datos de la Influencia Solar, que es un centro de investigación del observatorio Real de Bélgica.



SILSO graphics (<http://sidc.be/silso>) Royal Observatory of Belgium 2018 September 1

Figura 4-3. Recorte de la web www.sidc.be con SSN.

El formato en el que nos encontramos el fichero al descargarlo es un «.csv» que podemos abrir con LibreOffice o Excel, por lo que será necesario cambiar el separador decimal (Tanto Excel como LibreOffice, configurados en español, entienden el separador decimal como una coma) para que MatLab lo reconozca adecuadamente.

Se cambia el nombre y la extensión al archivo para poder trabajar cómodamente con él y ya tenemos listas las muestras teniendo en cuenta, que al ejecutar la función `xlsread()`, nos devolverá una matriz con las muestras por filas y en las columnas:

- Columna 1: año de la muestra.
- Columna 2: mes de la muestra.

- Columna 3: Fracción del año en curso.
- Columna 4: Media mensual del índice SSN.
- Columna 5: Desviación típica del valor del índice SSN.

La matriz contiene dos columnas más pero no nos van a ser de utilidad.

4.4.3 Demanda de energía eléctrica

Para este problema, se puede encontrar toda la información empleada en el portal de transparencia de red eléctrica de España <https://esios.ree.es>.

En este portal, se puede encontrar toda la información relativa a la red eléctrica como por ejemplo, demanda agregada, consumo, precios del kWh, PVPC, consumo o demanda programada. También se puede encontrar información relativa a la generación de energías renovables y mucho más.

Para descargar la información, debemos entrar en el indicador que nos interese y «pinchar» en «analizar indicador». Una vez hemos hecho esto, se nos abre una página en la que se muestra información acerca del indicador seleccionado. En nuestro caso, el indicador es la energía demandada por lo que realizamos este proceso para este indicador y en la página de análisis, vemos los gráficos correspondientes.

Se nos permite descargar la información agrupada por horas, cada minutos, días o meses. Se ha optado por descargarla agrupada por horas para tener una buena cantidad de datos disponibles.

Una vez descargamos el archivo en formato «.csv» nos encontramos las siguientes columnas:

- Columna 1: identificador del marcador al que pertenece la fila
- Columna 2: nombre del marcador al que pertenece la fila.
- Columna 3: vacío para el caso que nos ocupa.
- Columna 4: vacío para el caso que nos ocupa.
- Columna 5: valor del marcador (en MW en nuestro caso).
- Columna 6: fecha en formato «AAAA-MM-DDTHH:MM:SS+GMT».

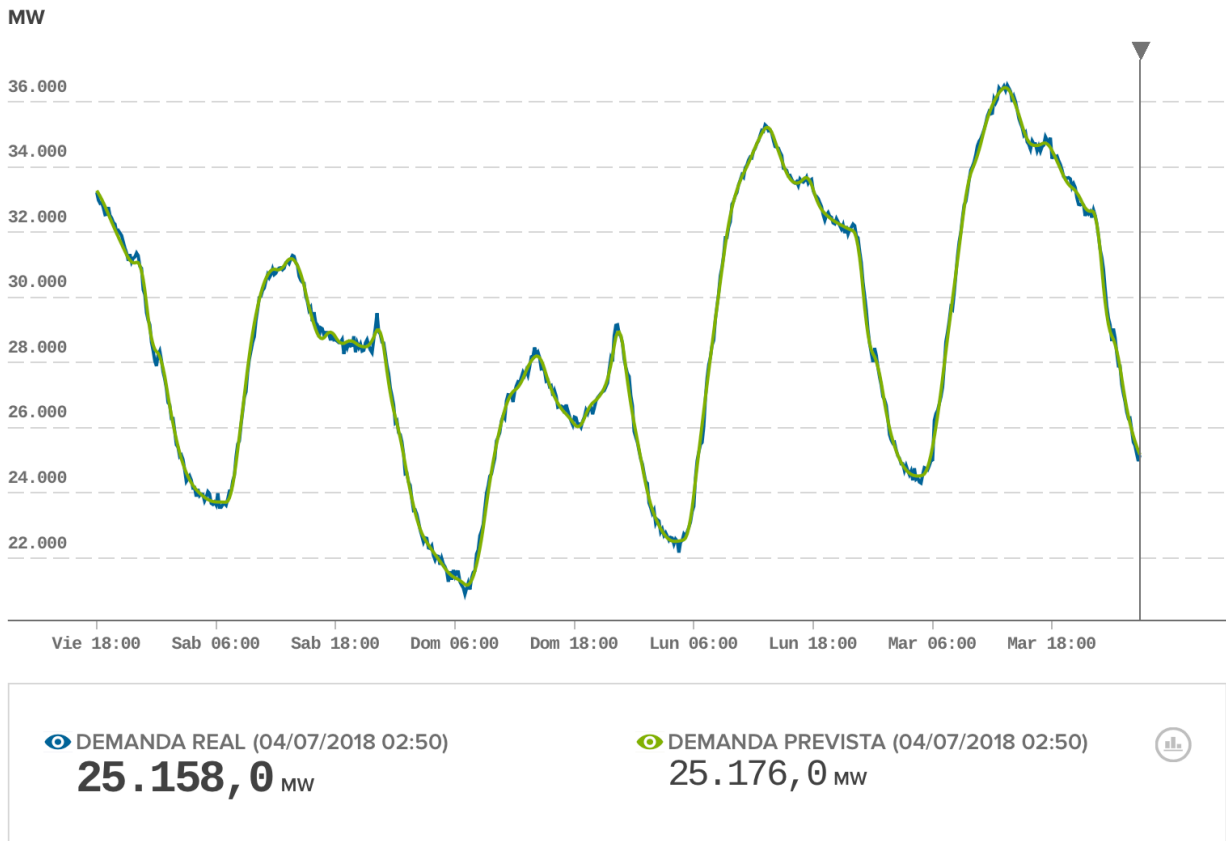


Figura 4-4. Recorte de un indicador en la web esios.ree.es.

Esa información, la procesamos para obtener otro archivo «.csv» con la siguiente distribución:

- Columna 1: año de la muestra.
- Columna 2: día de la muestra.
- Columna 3: mes de la muestra.
- Columna 4: hora de la muestra.
- Columna 5: valor de la muestra (en MW).

Guardamos el archivo en formato «.xlsx» para poder abrirlo con `xlsread()` de MatLab y ya se dispone de la información necesaria para resolver el problema.

5 ALGORITMOS PARA LA SELECCIÓN DE HIPERPARÁMETROS

A lo largo de este capítulo, encontraremos los resultados que son consecuencia de la aplicación de todo el trabajo descrito en el apartado anterior. Para garantizar toda la objetividad que sea posible en dichos resultados, en cada problema se aplicarán de forma sistemática los algoritmos desarrollados, manteniendo orden y forma.

El proceso que se va a seguir en los ensayos, constituye la técnica desarrollada para el ajuste de los métodos *kernel* al que hace referencia el título de este trabajo. El objetivo principal es comprobar el aumento de la eficiencia en la predicción de secuencias futuras (esto es, disminución del error cometido) al aplicar dichas técnicas. Por ello, los resultados presentados en este capítulo serán los indicadores de bondad necesarios para tal fin además de la predicción de las secuencias temporales de los set de validación con los regresores que se han seleccionado, en última instancia, consecuencia de aplicar estas técnicas.

En forma todos los problemas se resuelven de la misma manera, no así en fondo; es decir, aunque el procedimiento sistemático mencionado anteriormente sea el mismo para todos ellos, existen diferencias entre problemas debidos a la idiosincrasia de cada uno, como por ejemplo, el valor de los hiperparámetros o la dimensión y forma de los regresores.

Con todo lo mencionado anteriormente, antes de entrar en ningún problema concreto, se presenta la técnica desarrollada para la consecución de los resultados que se mostrarán a lo largo de este capítulo.

El proceso de ajuste de los parámetros y regresores necesarios es un procedimiento iterativo en el que se comparan los resultados obtenidos entre unas y otras de las soluciones propuestas. Para el fin que nos ocupa, vamos a partir de tres propuestas de regresores y otras tres para los conjuntos de hiperparámetros. Esto hace un total de nueve regresores distintos candidatos a resolver el problema de regresión. Evidentemente, cuantas más posibles regresores y conjuntos de parámetros enfrentemos entre sí, más nos acercaremos a la solución más eficiente posible para el método empleado en la regresión.

De forma esquemática, el algoritmo para cada problema es:

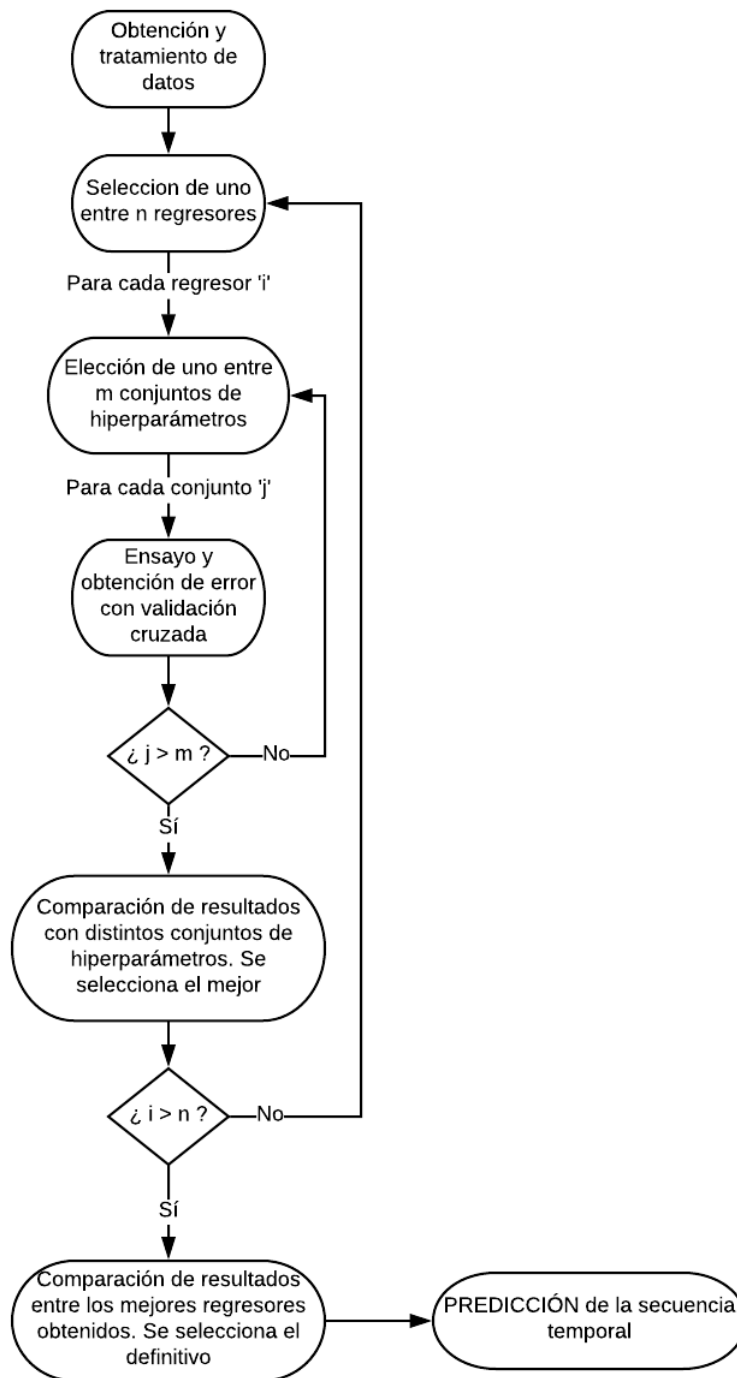


Figura 5-1. Estrategia de ajuste genérica.

Los indicadores de bondad que vamos a obtener mediante validación cruzada para elegir los mejores regresores serán:

- Error cuadrático medio (ECM o MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2$$

- Error estándar de la media (EEM o SEM)

$$SEM = \frac{std(se)}{\sqrt{N}}$$

dónde,

- $std(se)$ – desviación estándar los errores cuadráticos en la predicción
- N – Número de muestras del set de validación
- Y_i – Valor real de la muestra «i» de la secuencia temporal
- \widehat{Y}_i – Valor predicho de la muestra «i» de la secuencia temporal

Es necesario aclarar en el caso de los indicadores de bondad, que se excluirán del cómputo los valores no finitos de los sets de datos existentes, es decir, tanto de entrenamiento como de validación. Además, como se ha dicho anteriormente, estos indicadores se calcularán con técnicas de validación cruzada detalladas en el segundo capítulo de este documento.

En regresión, hay que tener cuidado con las desviaciones estándar y las medias ya se pueden encontrar varias con distintos significados de éstas en función de cómo se calculen. A continuación se explican las diferencias entre ellas.

Tabla 5-1. Distintas medias y desviaciones estándar en regresión.

Forma de cálculo	Significado
Media del MSE de cada iteraciones	Error medio esperado en futuras predicciones
Media de los SEM de cada iteración	Error estándar esperado en futuras predicciones
Desviación típica de los MSE de cada iteración	Varianza de la estimación de los futuros MSE con respecto a los predichos
Desviación típica de los SEM de cada iteración	Varianza de la estimación de los futuros SE con respecto a los predichos

Como se pretende determinar cuál es el regresor que mejor predicción de muestras futuras va a realizar y el objetivo principal de la validación cruzada es determinar cómo de bien se va a comportar dicho regresor, vamos a emplear las dos primeras entradas de la Tabla 5-1.

Además, en todos los problemas, el primer conjunto regresor-parámetros que se va a emplear consistirá en un regresor formado por los últimos cuatro valores disponibles de la salida, con γ y τ iguales a uno, ya que con esta configuración sumado a una función *kernel* tipo lineal, el resultado obtenido es equivalente al obtenido con un regresor de dimensión cuatro por mínimos cuadrados y podemos comprobar cómo los métodos *kernel* mejoran el resultado de los métodos históricos si es que lo hacen.

5.1 Señal modulada en amplitud

Se muestran en esta sección los resultados obtenidos en este problema. El resultado final será la elección justificada de un conjunto de regresor y parámetros. Para ello se proponen los siguientes:

- Regresores:

- Primer regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$

- Segundo regresor: $x_i = [y_{i-1}, y_{i-2}, \frac{y_{i-3}+y_{i-4}+y_{i-5}}{3}]$

- Tercer regresor: $x_i = [y_{i-1}, \sqrt{y_{i-2}}, 0.4 \cdot y_{i-3}, 0.6 \cdot \frac{y_{i-4}+y_{i-3}+y_{i-2}}{3}]$

- Parámetros:

Tabla 5-2. Conjuntos de parámetros a ensayar en el problema SMA.

Parámetro	Primer conjunto	Segundo conjunto	Tercer conjunto
Función kernel	Linear	Radial Basis	Sigmoidal
Tau (τ)	1	0.95	0.9
Gamma (γ)	1	550	70
Param. 1 de la f. <i>kernel</i>	34	300	3
Param. 2 de la f. <i>kernel</i>	0.5	50	15

5.2 Número de manchas solares

Se muestran en esta sección los resultados obtenidos en este problema. El resultado final será la elección justificada de un conjunto de regresor y parámetros. Para ello se proponen los siguientes:

- Regresores:
 - Primer regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
 - Segundo regresor: $x_i = [y_{i-1}, y_{i-2}, f.a.c._{i-1}]$
 - Tercer regresor: $x_i = [y_{i-1}, \sqrt{y_{i-2}}, año_i, 0.6 \cdot \frac{std_{i-1} + std_{i-2} + std_{i-3}}{3}]$
- Parámetros:

Tabla 5-3. Conjuntos de parámetros a ensayar en el problema SSN.

Parámetro	Primer conjunto	Segundo conjunto	Tercer conjunto
Función kernel	Linear	Radial Basis	Sigmoidal
Tau (τ)	1	0.99	0.92
Gamma (γ)	1	470	265
Param. 1 de la f. <i>kernel</i>	0.5	360	78
Param. 2 de la f. <i>kernel</i>	164	54	15

5.3 Demanda de energía eléctrica

Se muestran en esta sección los resultados obtenidos en este problema. El resultado final será la elección justificada de un conjunto de regresor y parámetros. Para ello se proponen los siguientes:

- Regresores:
 - Primer regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
 - Segundo regresor: $x_i = [y_{i-1}, y_{i-2}, mes_{i-1}, día_{i-1}, hora_{i-1}]$
 - Tercer regresor:

$$x_i = [y_{i-1}, 0.4 \cdot y_{i-2} + 0.6 \cdot \frac{y_{i-4} + y_{i-3} + y_{i-2}}{3}, mes_{i-1}, día_{i-1}, hora_{i-1}]$$
- Parámetros:

Tabla 5-4. Conjuntos de parámetros a ensayar en el problema DEL.

Parámetro	Primer conjunto	Segundo conjunto	Tercer conjunto
Función kernel	Linear	Radial Basis	Sigmoidal
Tau (τ)	1	0.97	0.995
Gamma (γ)	1	175	38
Param. 1 de la f. <i>kernel</i>	3	615	34
Param. 2 de la f. <i>kernel</i>	28	167	615

6 PRESENTACIÓN DE RESULTADOS

6.1 Resultados SMA

6.1.1 Caso 1. Primer regresor y primer conjunto de parámetros

Para este caso, se elige el primer regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-1. Parámetros del caso 1 en SMA.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	34
Param. 2 de la f. <i>kernel</i>	0.5

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado. Este caso es idéntico a emplear mínimos cuadrados en la predicción de la secuencia temporal.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

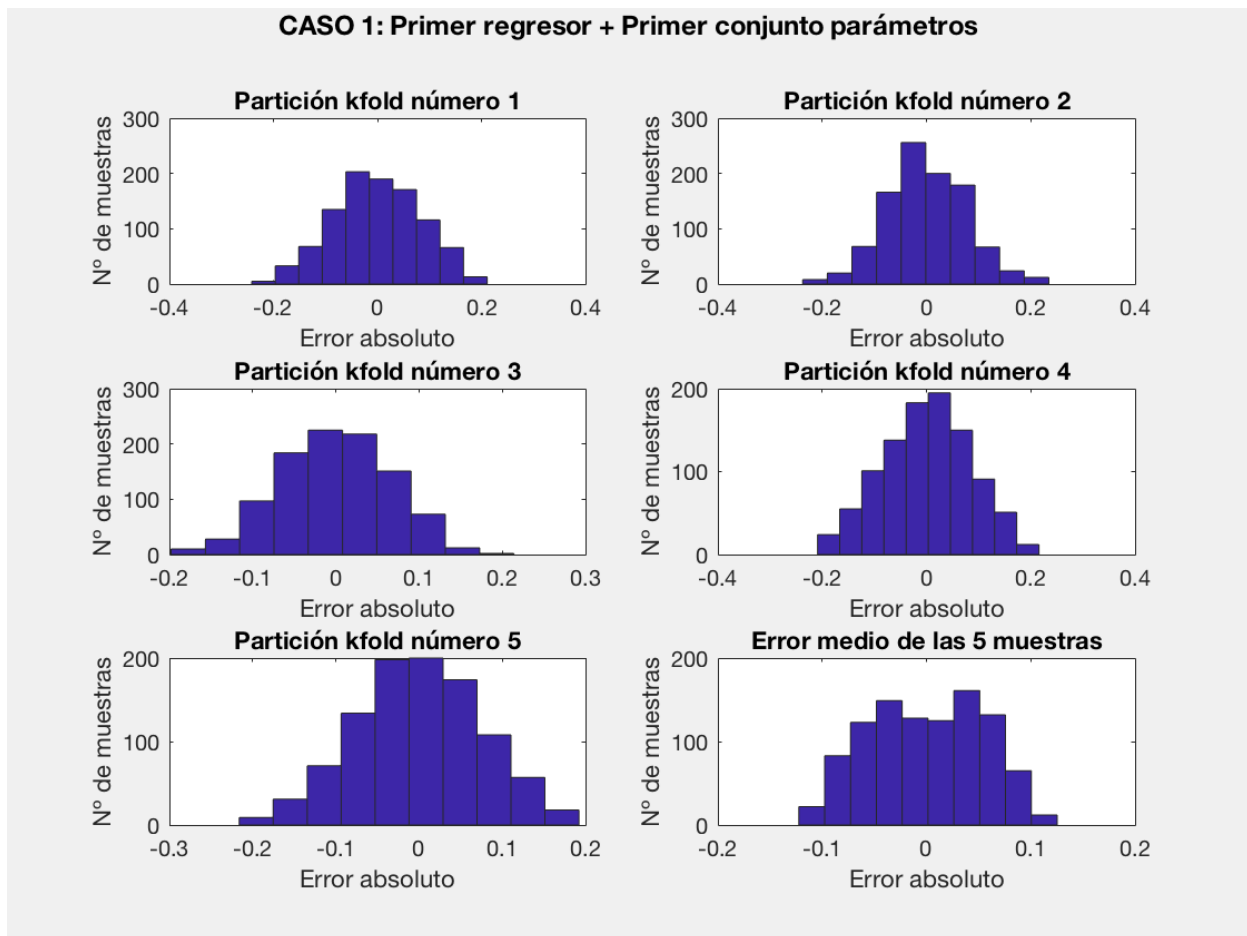


Figura 6-1. Histogramas de error para el caso 1 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-2. Errores cometidos en el caso 1 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.006878	0.005787	0.004315	0.006956	0.005651	0.005917
SEM	0.000268	0.000266	0.000182	0.000273	0.000228	0.000243

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.2 Caso 2. Primer regresor y segundo conjunto de parámetros

Para este caso, se elige el primer regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-3. Parámetros del caso 2 en SMA.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.95
Gamma (γ)	550
Param. 1 de la f. kernel	300
Param. 2 de la f. kernel	50

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

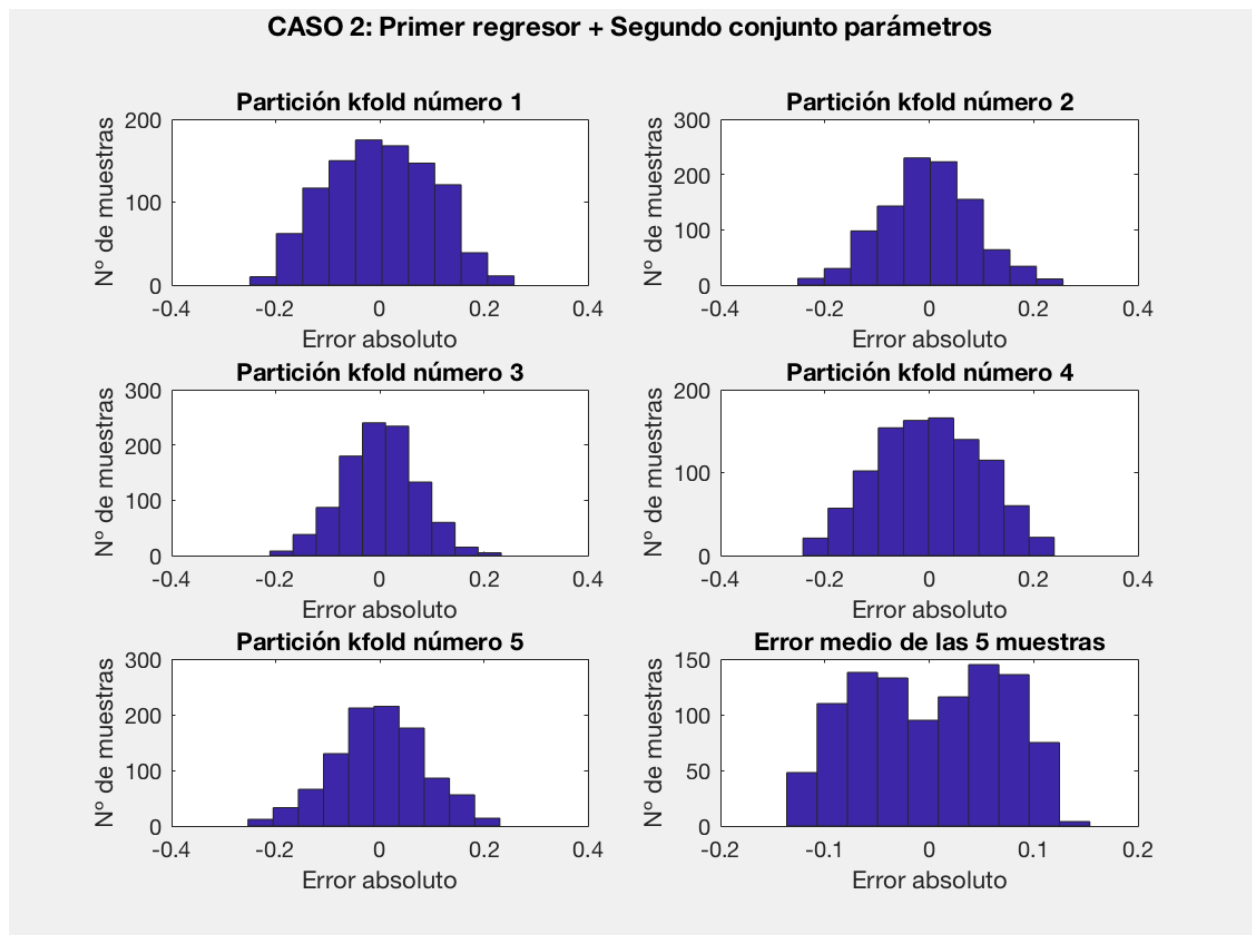


Figura 6-2. Histogramas de error para el caso 2 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-4. Errores cometidos en el caso 2 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.009926	0.007671	0.004982	0.010092	0.007549	0.008049
SEM	0.000363	0.000336	0.000220	0.000372	0.000326	0.000324

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.3 Caso 3. Primer regresor y tercer conjunto de parámetros

Para este caso, se elige el primer regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-5. Parámetros del caso 3 en SMA

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.9
Gamma (γ)	70
Param. 1 de la f. <i>kernel</i>	3
Param. 2 de la f. <i>kernel</i>	15

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

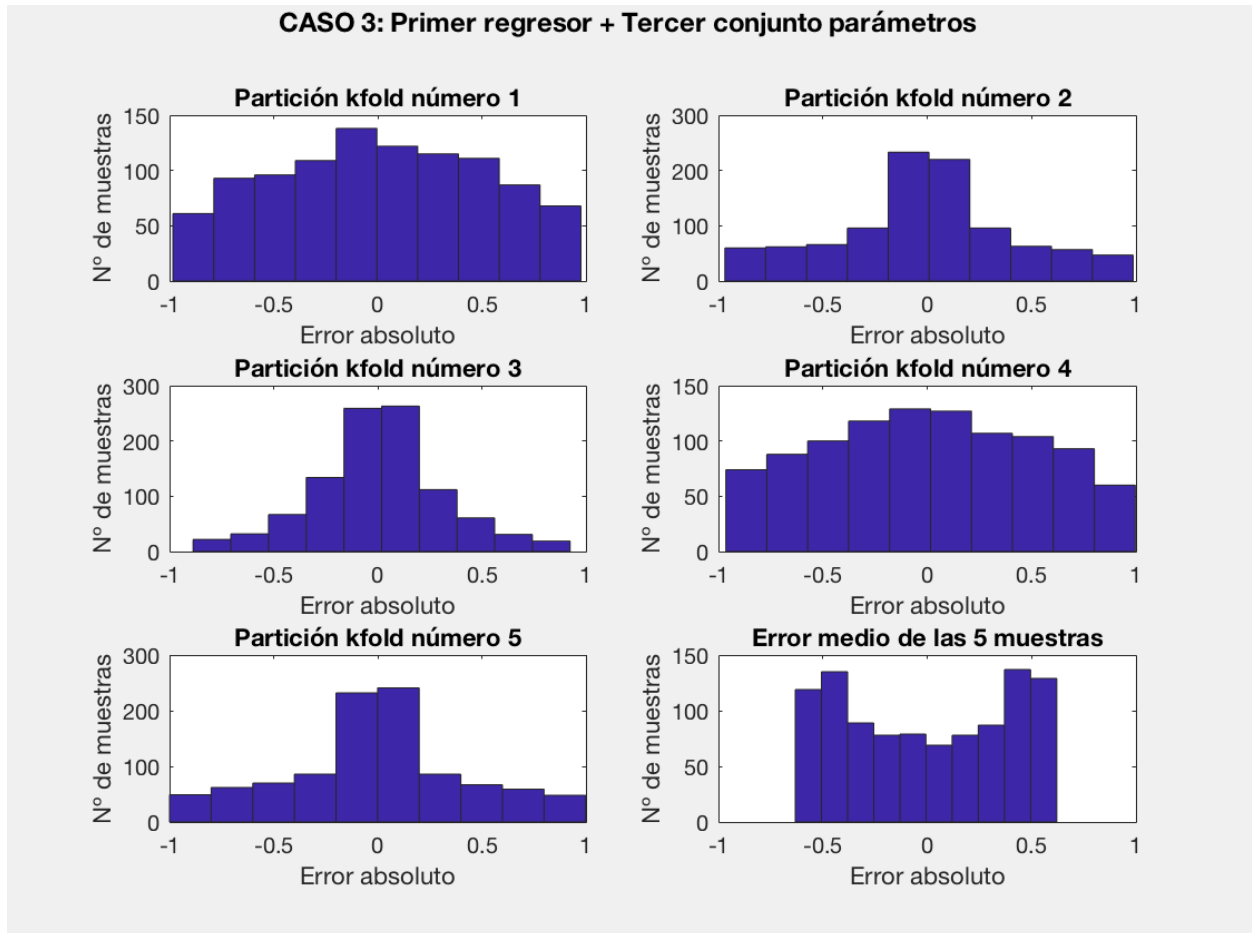


Figura 6-3. Histogramas de error para el caso 3 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-6. Errores cometidos en el caso 3 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.256314	0.190522	0.098324	0.259142	0.190352	0.198931
SEM	0.008086	0.007966	0.004916	0.008134	0.008038	0.007428

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.4 Caso 4. Segundo regresor y primer conjunto de parámetros

Para este caso, se elige el segundo regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor:
$$x_i = \left[y_{i-1}, y_{i-2}, \frac{y_{i-3} + y_{i-4} + y_{i-5}}{3} \right]$$
- Parámetros:

Tabla 6-7. Parámetros del caso 4 en SMA.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	34
Param. 2 de la f. <i>kernel</i>	0.5

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

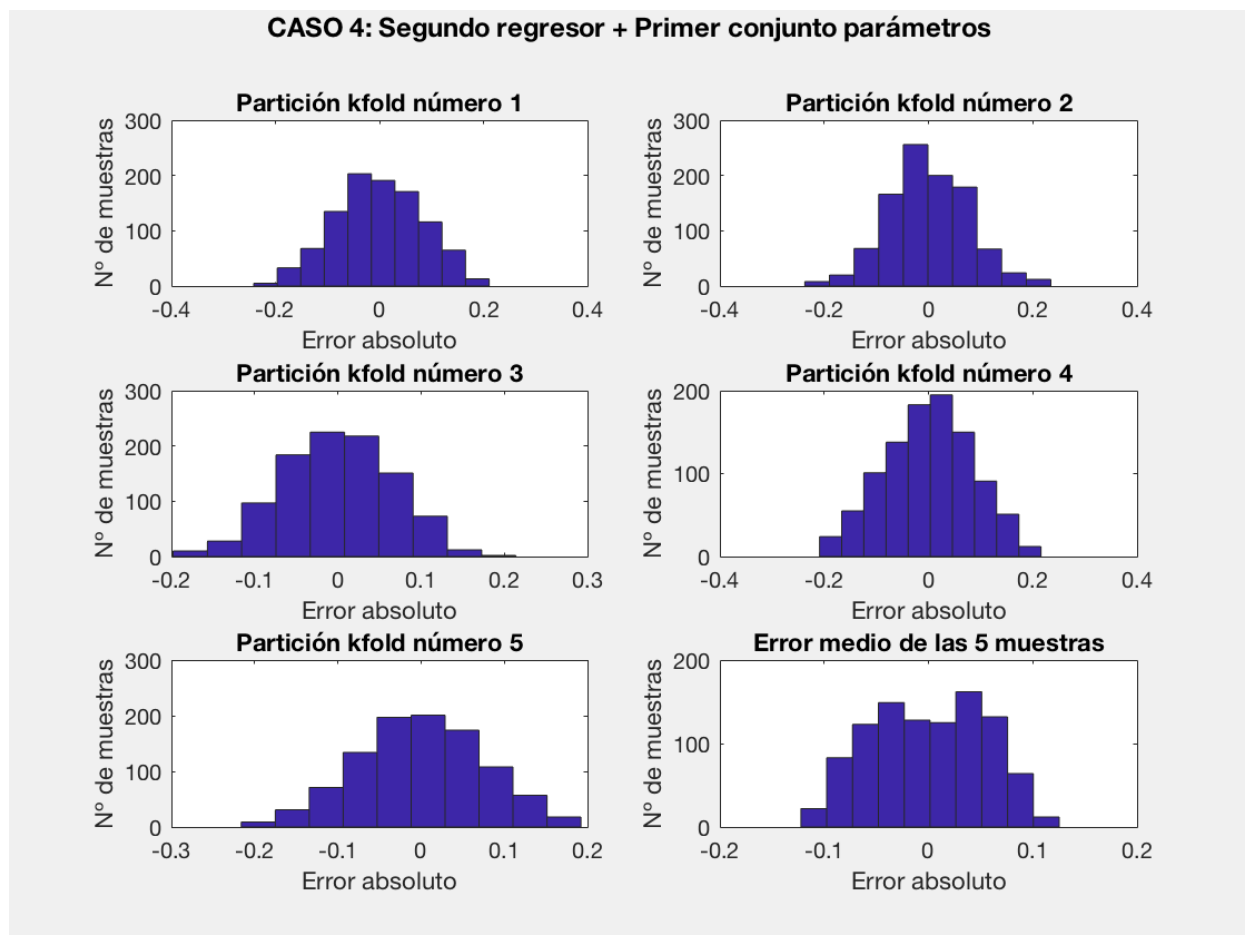


Figura 6-4. Histogramas de error para el caso 4 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-8. Errores cometidos en el caso 4 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.006878	0.005787	0.004315	0.006956	0.005651	0.005912
SEM	0.000268	0.000266	0.000182	0.000273	0.000228	0.000243

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.5 Caso 5. Segundo regresor y segundo conjunto de parámetros

Para este caso, se elige el segundo regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor:
$$x_i = \left[y_{i-1}, y_{i-2}, \frac{y_{i-3} + y_{i-4} + y_{i-5}}{3} \right]$$
- Parámetros:

Tabla 6-9. Parámetros del caso 5 en SMA

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.95
Gamma (γ)	550
Param. 1 de la f. <i>kernel</i>	300
Param. 2 de la f. <i>kernel</i>	50

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

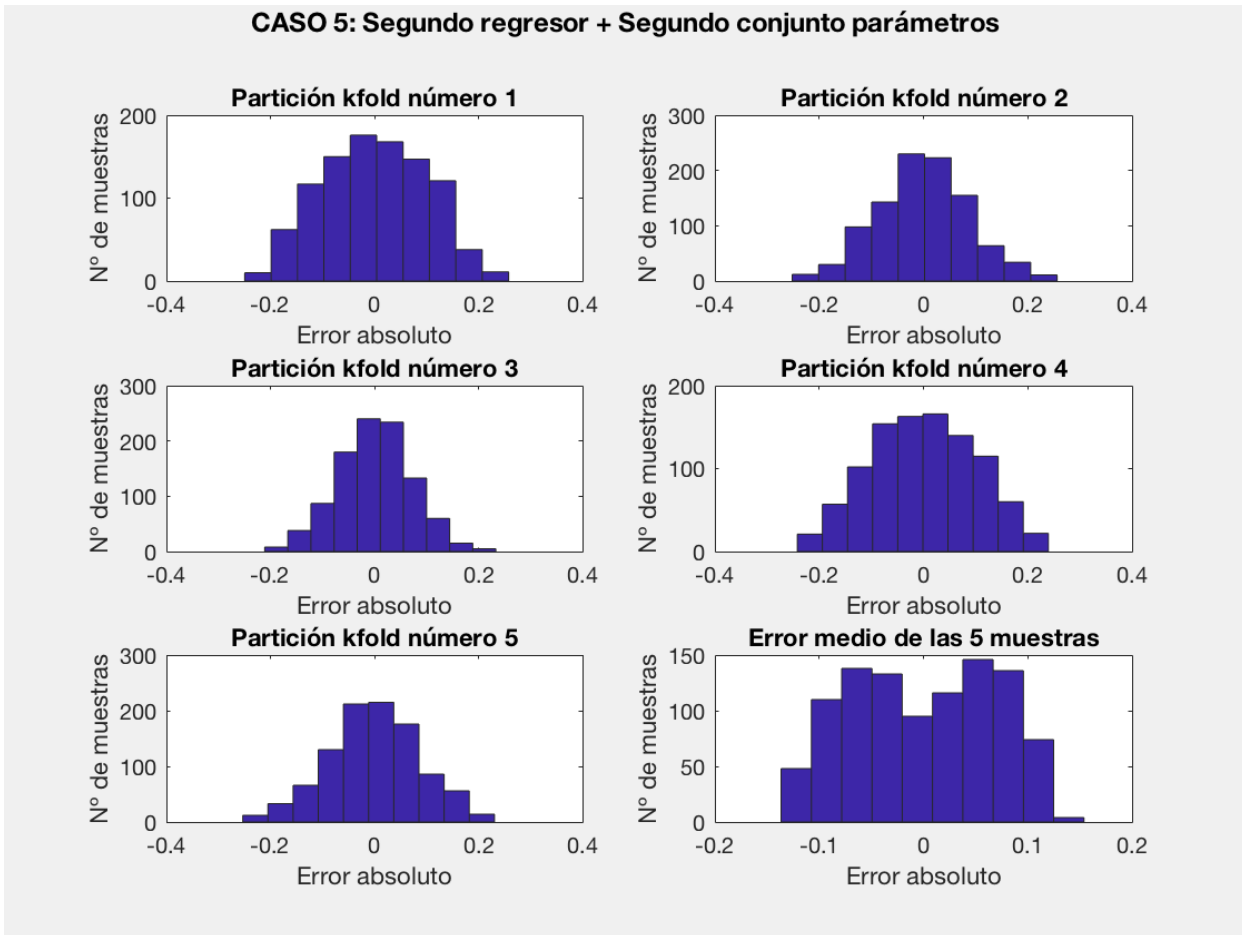


Figura 6-5. Histogramas de error para el caso 5 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-10. Errores cometidos en el caso 5 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.009926	0.007671	0.004982	0.010092	0.007549	0.008044
SEM	0.000363	0.000336	0.000220	0.000372	0.000326	0.000324

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.6 Caso 6. Segundo regresor y tercer conjunto de parámetros

Para este caso, se elige el segundo regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor:
$$x_i = \left[y_{i-1}, y_{i-2}, \frac{y_{i-3} + y_{i-4} + y_{i-5}}{3} \right]$$
- Parámetros:

Tabla 6-11. Parámetros del caso 6 en SMA.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.9
Gamma (γ)	70
Param. 1 de la f. <i>kernel</i>	3
Param. 2 de la f. <i>kernel</i>	15

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

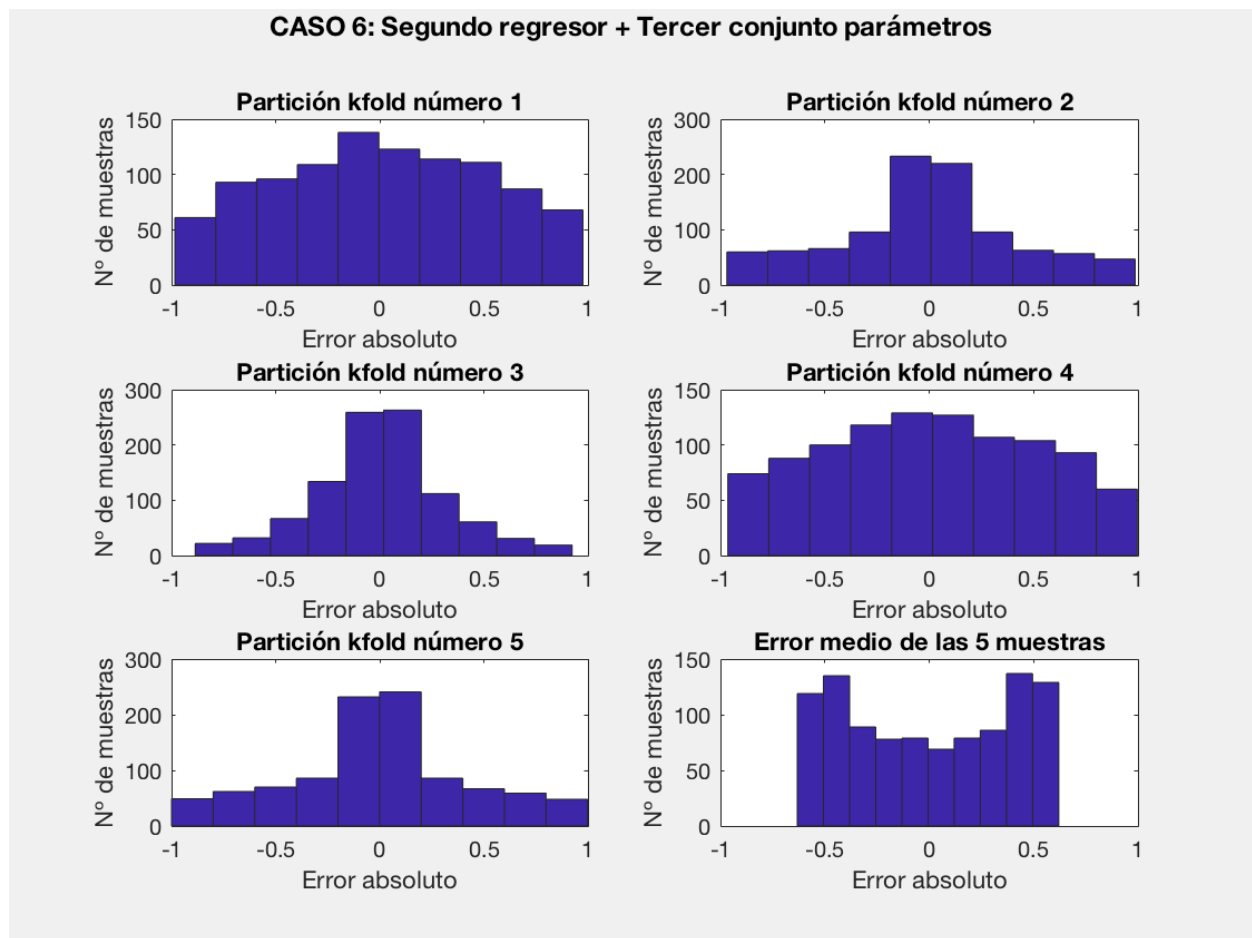


Figura 6-6. Histogramas de error para el caso 6 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-12. Errores cometidos en el caso 6 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.256314	0.190522	0.098324	0.259142	0.190352	0.198914
SEM	0.008086	0.007966	0.004916	0.008134	0.008038	0.007428

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.7 Caso 7. Tercer regresor y primer conjunto de parámetros

Para este caso, se elige el tercer regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor:
$$x_i = \left[y_{i-1}, \sqrt{y_{i-2}}, 0.4 \cdot y_{i-3}, 0.6 \cdot \frac{y_{i-4} + y_{i-3} + y_{i-2}}{3} \right]$$
- Parámetros:

Tabla 6-13. Parámetros del caso 7 en SMA.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	34
Param. 2 de la f. <i>kernel</i>	0.5

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

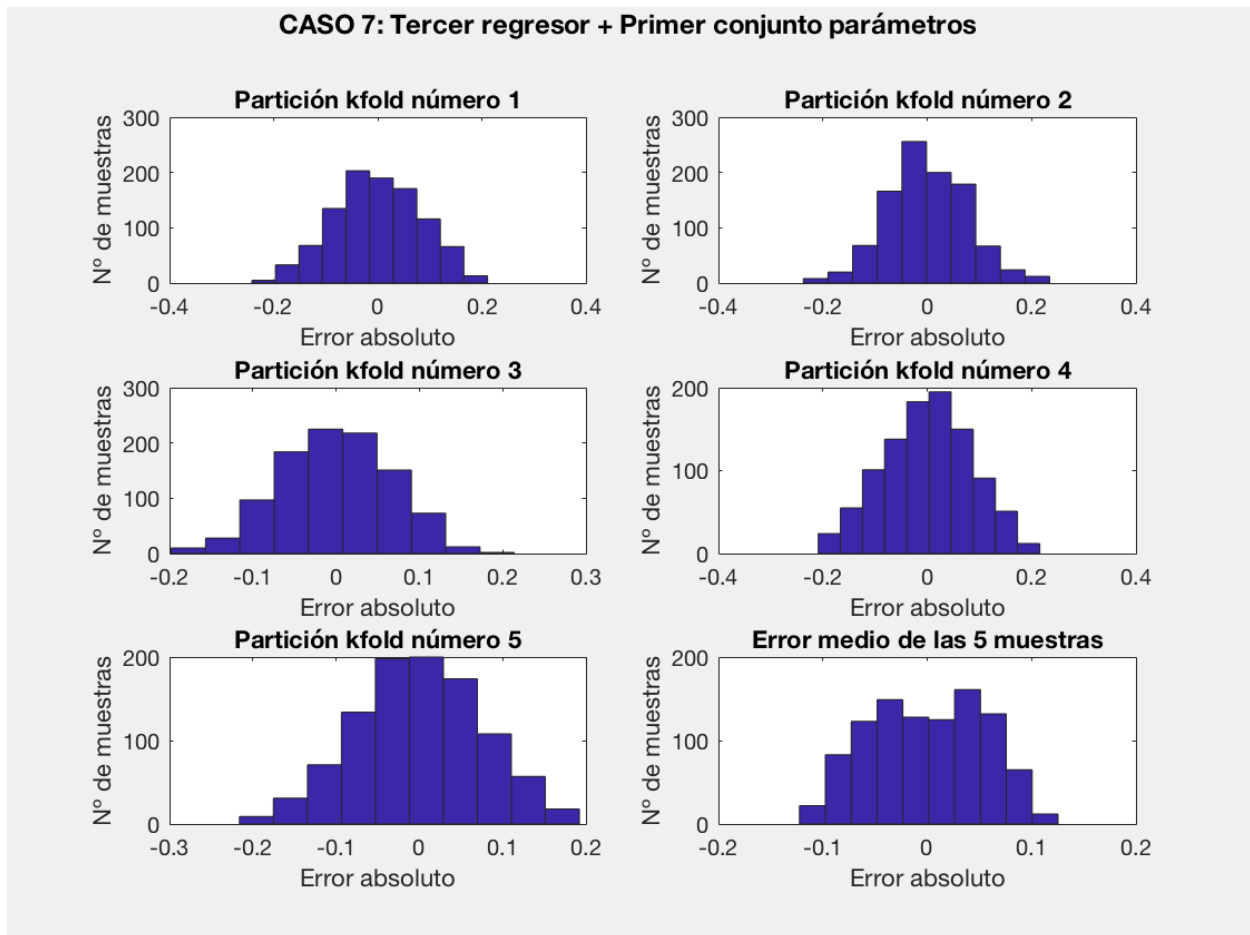


Figura 6-7. Histogramas de error para el caso 7 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-14. Errores cometidos en el caso 7 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.006878	0.005787	0.004315	0.006956	0.005651	0.005917
SEM	0.000268	0.000266	0.000182	0.000273	0.000228	0.000243

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.8 Caso 8. Tercer regresor y segundo conjunto de parámetros

Para este caso, se elige el tercer regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor:
$$x_i = \left[y_{i-1}, \sqrt{y_{i-2}}, 0.4 \cdot y_{i-3}, 0.6 \cdot \frac{y_{i-4} + y_{i-3} + y_{i-2}}{3} \right]$$
- Parámetros:

Tabla 6-15. Parámetros del caso 8 en SMA.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.95
Gamma (γ)	550
Param. 1 de la f. <i>kernel</i>	300
Param. 2 de la f. <i>kernel</i>	50

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

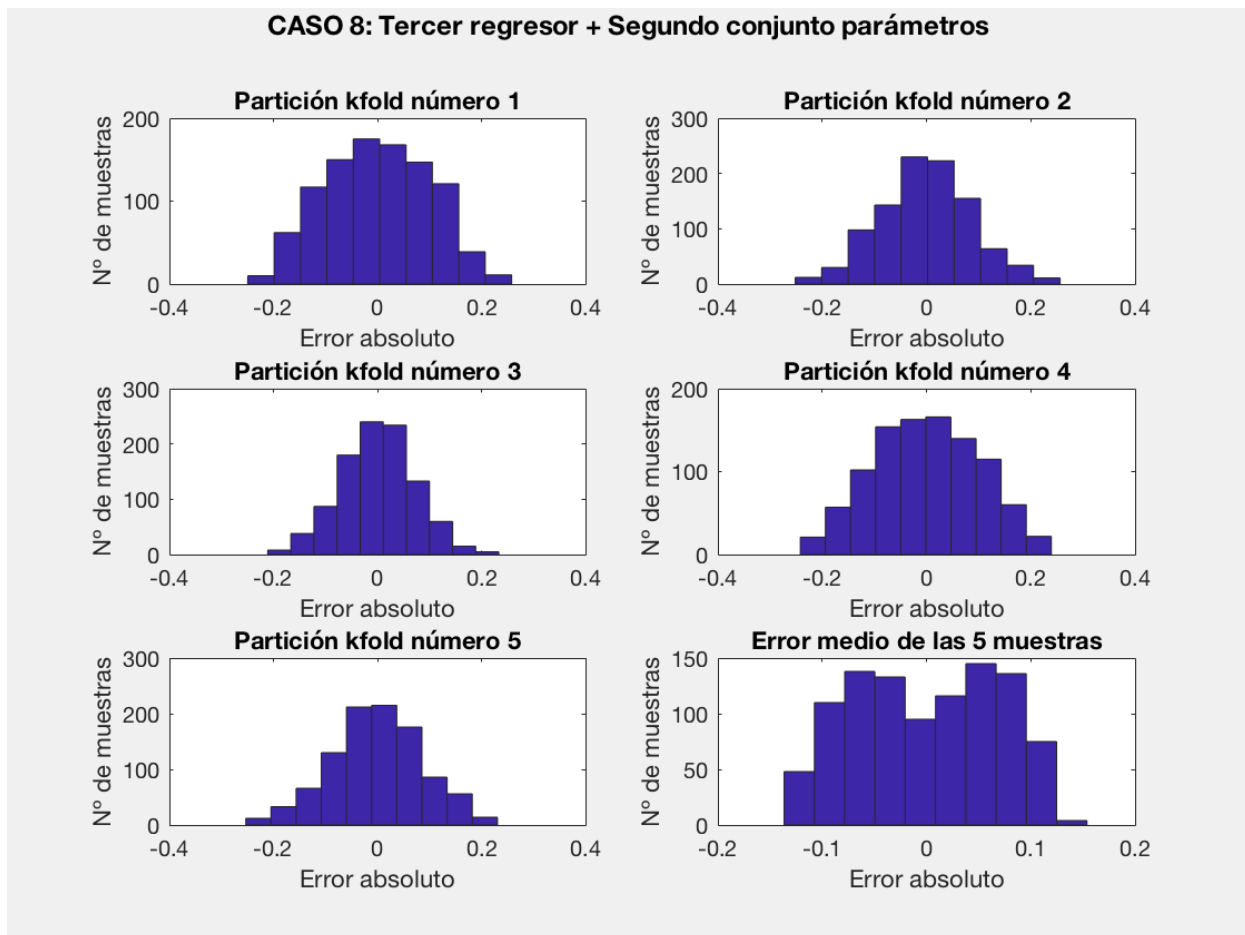


Figura 6-8. Histogramas de error para el caso 8 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-16. Errores cometidos en el caso 8 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.009926	0.007671	0.004982	0.010092	0.007549	0.008049
SEM	0.000363	0.000336	0.000220	0.000372	0.000326	0.000324

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.9 Caso 9. Tercer regresor y tercer conjunto de parámetros

Para este caso, se elige el tercer regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor:
$$x_i = \left[y_{i-1}, \sqrt{y_{i-2}}, 0.4 \cdot y_{i-3}, 0.6 \cdot \frac{y_{i-4} + y_{i-3} + y_{i-2}}{3} \right]$$
- Parámetros:

Tabla 6-17. Parámetros del caso 9 en SMA.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.9
Gamma (γ)	70
Param. 1 de la f. <i>kernel</i>	3
Param. 2 de la f. <i>kernel</i>	15

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

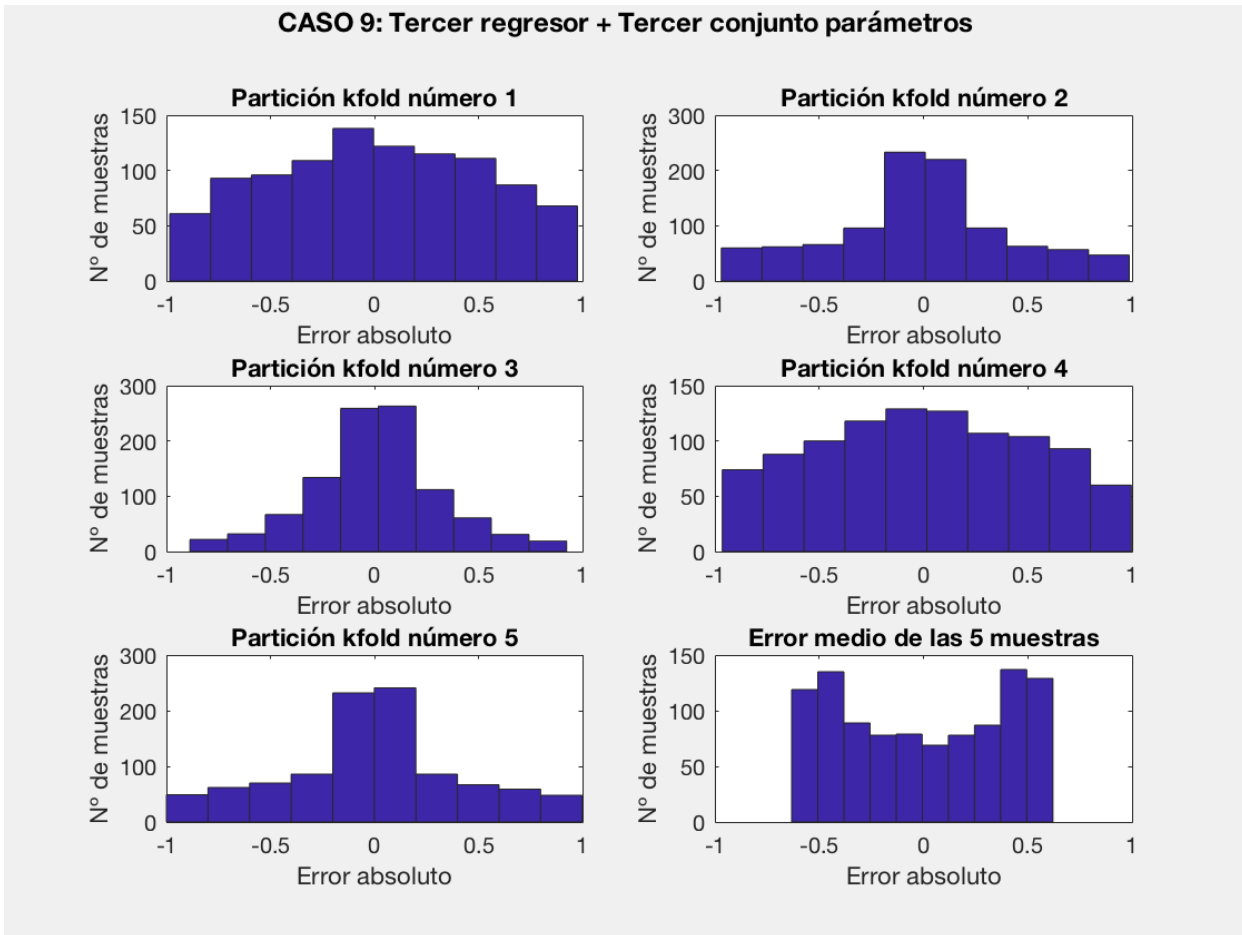


Figura 6-9. Histogramas de error para el caso 9 en SMA.

Los errores que se cometen en este caso, son:

Tabla 6-18. Errores cometidos en el caso 9 de SMA.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	0.256314	0.190522	0.098324	0.259142	0.190352	0.198931
SEM	0.008086	0.007966	0.004916	0.008134	0.008038	0.007428

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.1.10 Elección del regresor

Una vez se han ensayado y validado todos los casos, se dispone de la información necesaria para hacer la elección final, de forma resumida, en la siguiente tabla se pueden comparar los errores medios cometidos en cada caso. Éstos se han calculado como la media de los errores existentes en cada iteración del proceso de validación cruzada.

Tabla 6-19. Resumen de errores por caso en SMA.

CASO	MSE	SEM
Caso 1	0.005917971384299	0.000243998976084
Caso 2	0.008049804893941	0.000324154397818
Caso 3	0.198931243489346	0.007428415690445
Caso 4	0.005912966322028	0.000243890126305
Caso 5	0.008044616243683	0.000324102907981
Caso 6	0.198914195555970	0.007428863015880
Caso 7	0.005917971384299	0.000243998976084
Caso 8	0.008049804893941	0.000324154397818
Caso 9	0.198931243489346	0.007428415690445

De esta forma, el regresor elegido para la regresión, por tener los mejores indicadores de bondad, es el correspondiente al caso 4, ya que tanto el MSE como el SEM son los menores de todos los ensayados:

- Regresor: $x_i = \left[y_{i-1}, y_{i-2}, \frac{y_{i-3} + y_{i-4} + y_{i-5}}{3} \right]$
- Parámetros:

Tabla 6-20. Parámetros del regresor elegido en SMA.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	34
Param. 2 de la f. <i>kernel</i>	0.5

6.1.11 Predicción

Tras aplicar el conjunto regresor-parámetros al problema, separando los datos entre set de entrenamiento y set de validación destinando un 75% de los disponibles para el primero y el resto para el segundo, obtenemos un error medio en valor absoluto de 0.1420 con una desviación estándar de 0.1751. A continuación se muestra una gráfica con el valor real de las muestras y su predicción.

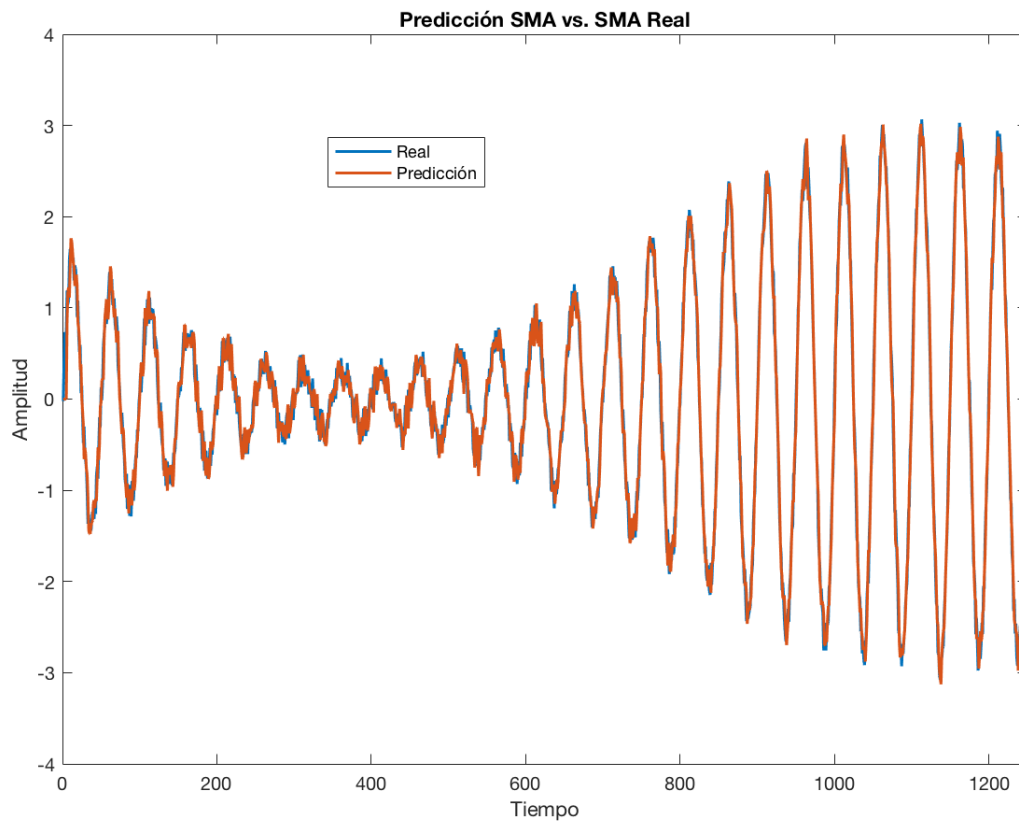


Figura 6-10. Predicción en SMA.

En detalle:

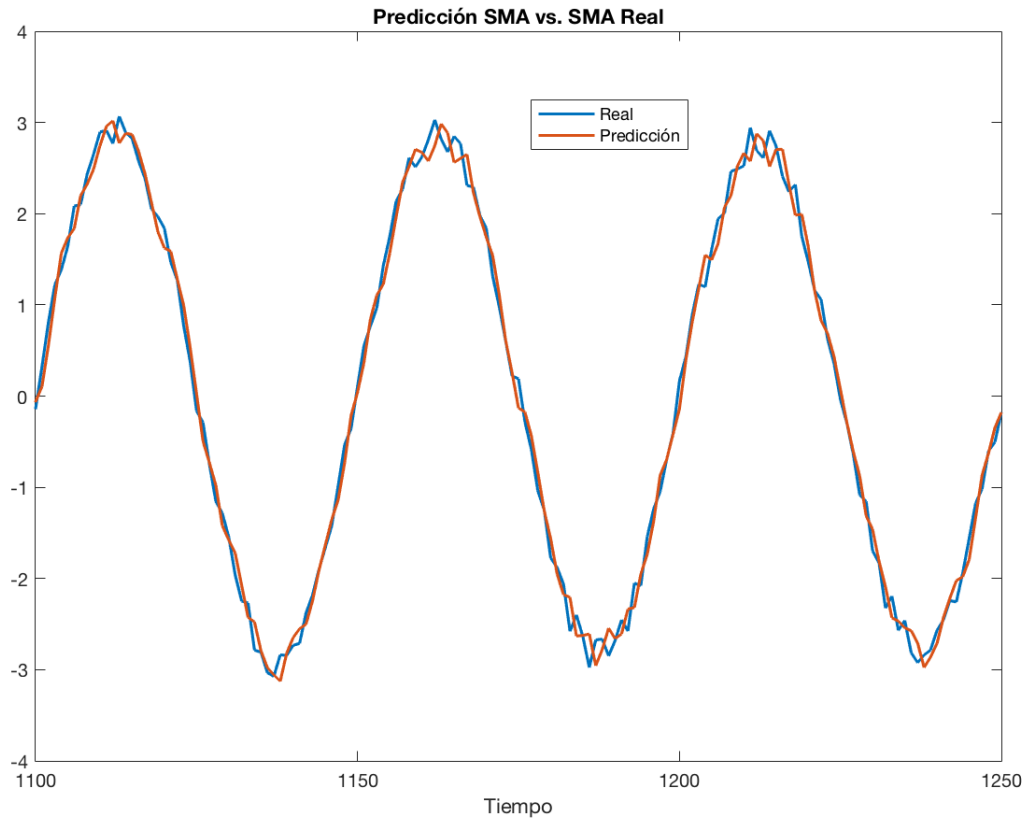


Figura 6-11. Detalle de predicción en SMA.

A continuación, se puede observar un histograma de los errores cometidos y dibujada encima, la curva de la distribución normal asociada a dicho histograma.

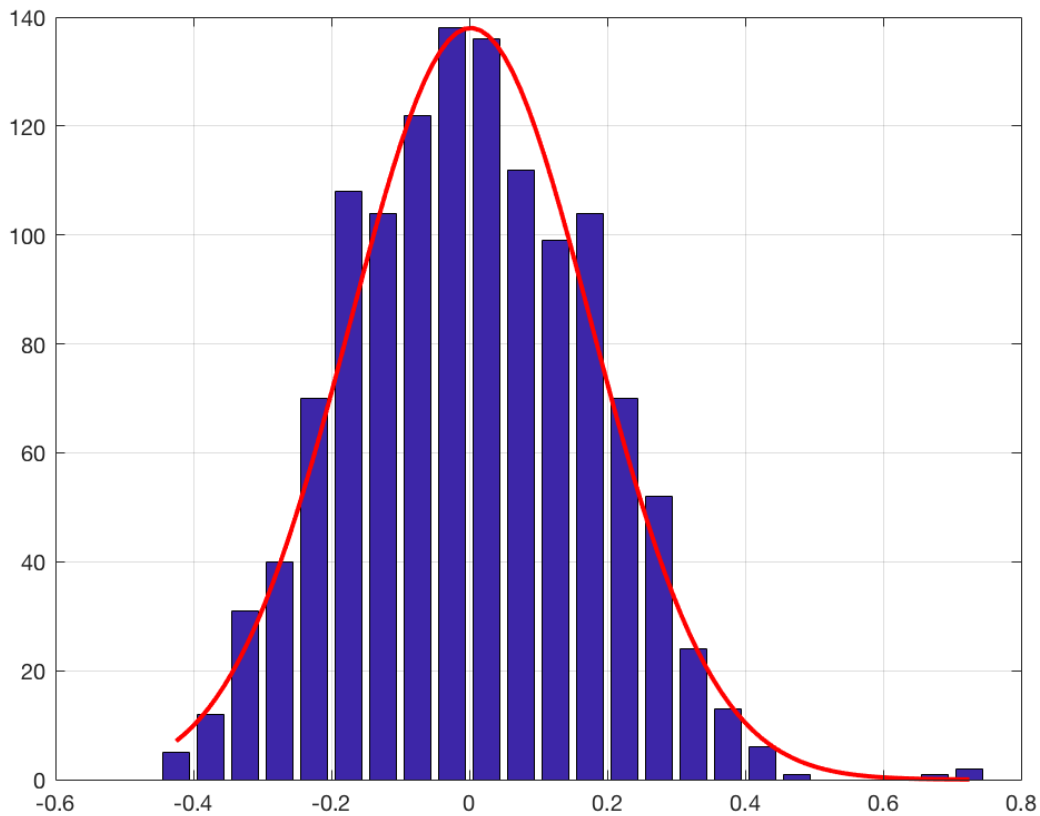


Figura 6-12. Histograma del error en SMA.

6.2 Resultados SSN

6.2.1 Caso 1. Primer regresor y primer conjunto de parámetros

Para este caso, se elige el primer regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-21. Parámetros del caso 1 en SSN.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	0.5
Param. 2 de la f. <i>kernel</i>	164

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado. Este caso es idéntico a emplear mínimos cuadrados en la predicción de la secuencia temporal.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

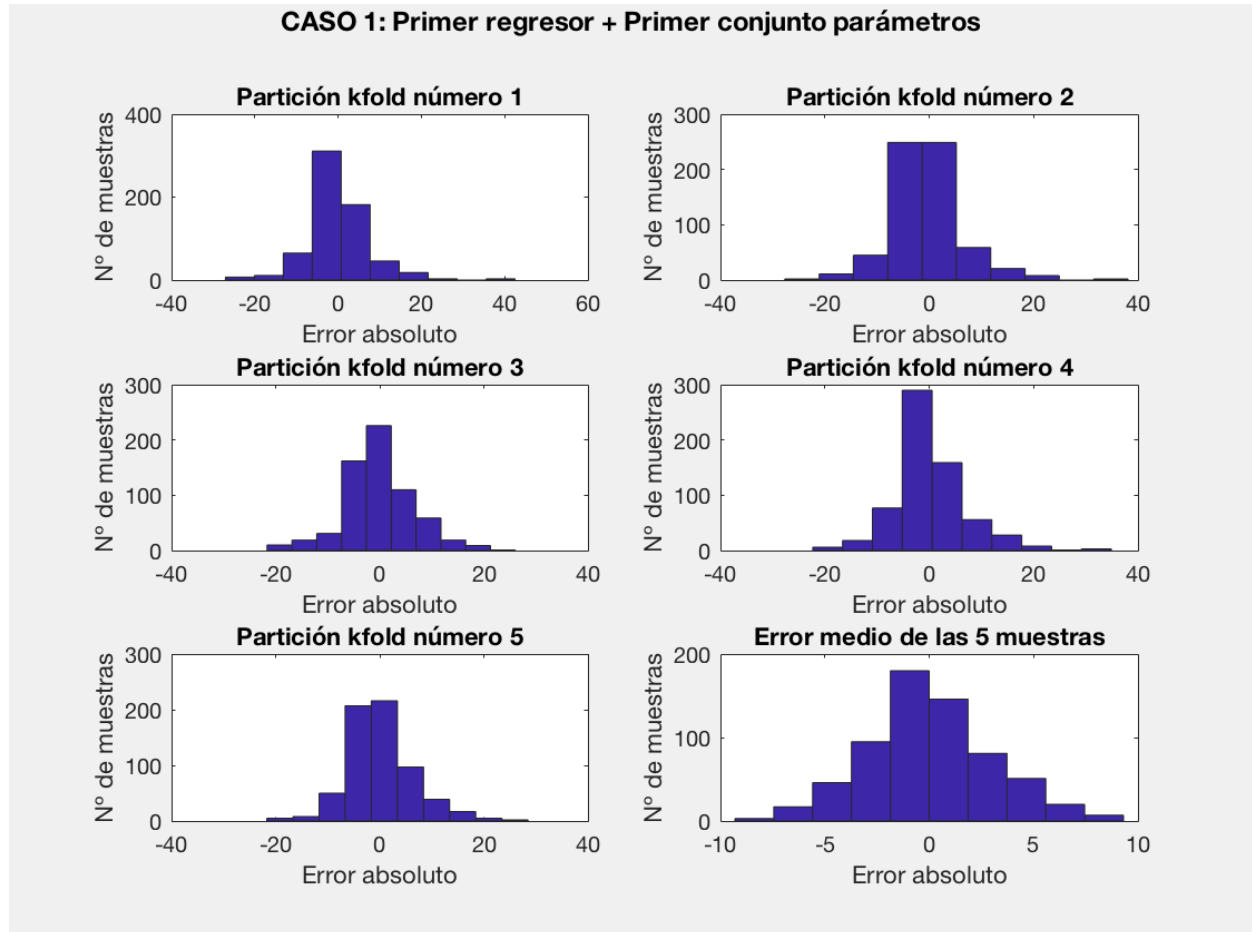


Figura 6-13. Histogramas de error para el caso 1 en SNN.

Los errores que se cometen en este caso, son:

Tabla 6-22. Errores cometidos en el caso 1 de SNN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	54.292448	45.372305	43.667700	47.563140	39.380136	46.055146
SEM	5.393751	4.405221	3.066412	4.094940	3.042716	4.000608

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.2 Caso 2. Primer regresor y segundo conjunto de parámetros

Para este caso, se elige el primer regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-23. Parámetros del caso 2 en SSN.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.99
Gamma (γ)	470
Param. 1 de la f. <i>kernel</i>	360
Param. 2 de la f. <i>kernel</i>	54

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

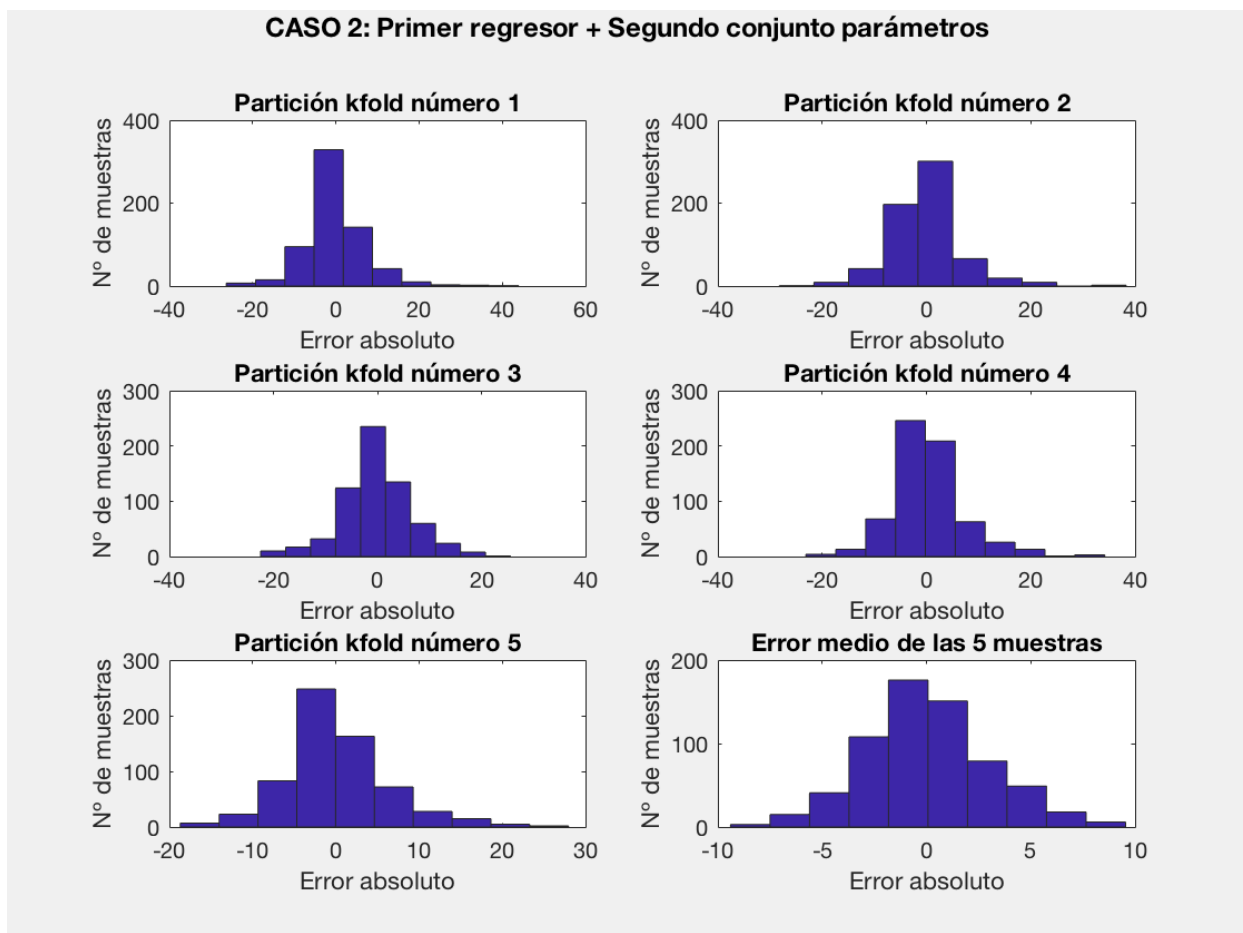


Figura 6-14. Histogramas de error para el caso 2 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-24. Errores cometidos en el caso 2 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	53.133801	44.824724	43.793680	47.248870	38.312076	45.451856
SEM	5.311127	4.341405	3.067860	4.103989	2.891650	3.942963

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.3 Caso 3. Primer regresor y tercer conjunto de parámetros

Para este caso, se elige el primer regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-25. Parámetros del caso 3 en SSN.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.92
Gamma (γ)	265
Param. 1 de la f. <i>kernel</i>	78
Param. 2 de la f. <i>kernel</i>	15

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

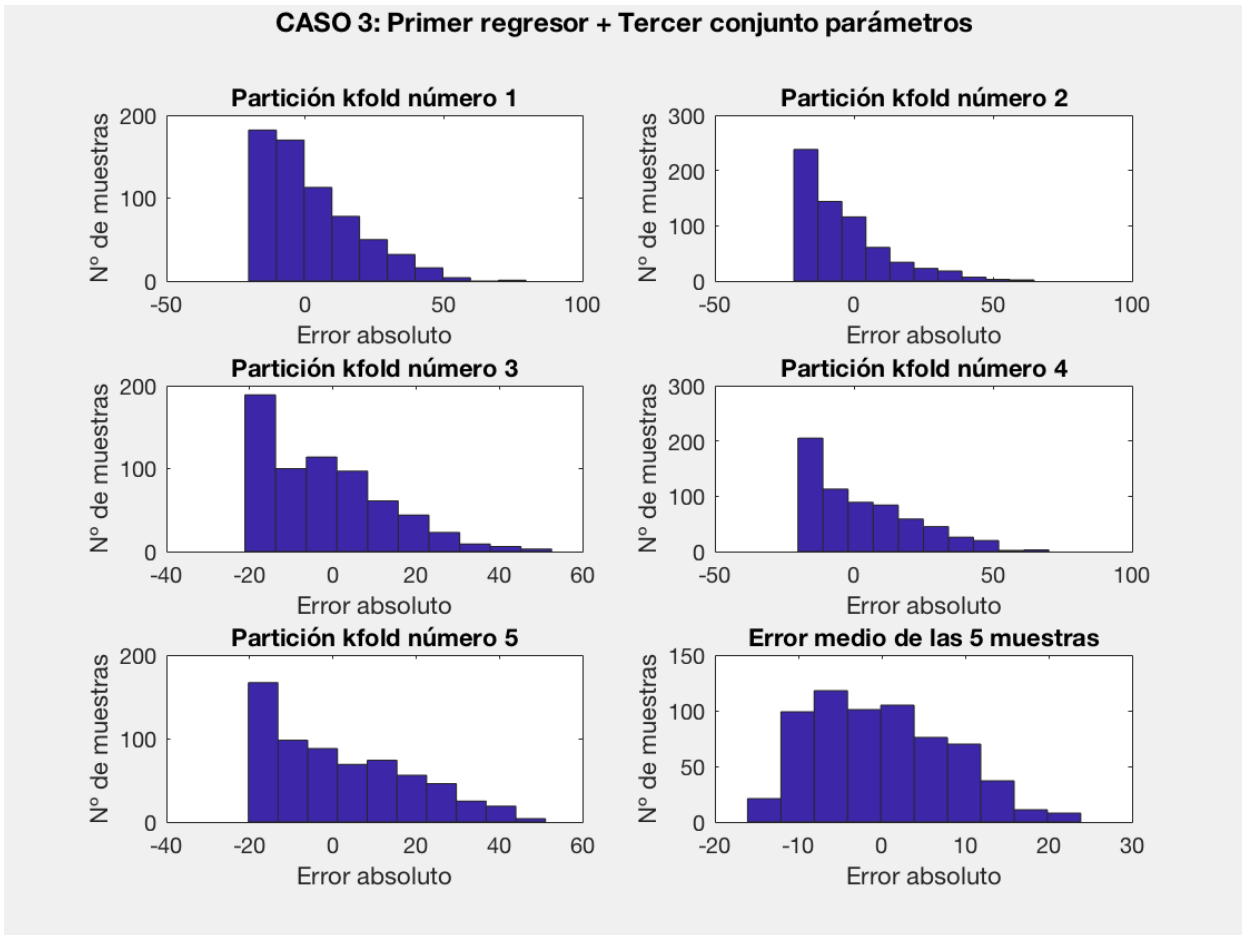


Figura 6-15. Histogramas de error para el caso 3 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-26. Errores cometidos en el caso 3 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	296.970036	273.507816	221.796179	375.414362	297.336629	293.005004
SEM	19.706176	15.704225	11.538894	23.106226	15.037640	17.018632

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.4 Caso 4. Segundo regresor y primer conjunto de parámetros

Para este caso, se elige el segundo regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, f.a.c.i-1]$
- Parámetros:

Tabla 6-27. Parámetros del caso 4 en SSN.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. kernel	0.5
Param. 2 de la f. kernel	164

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

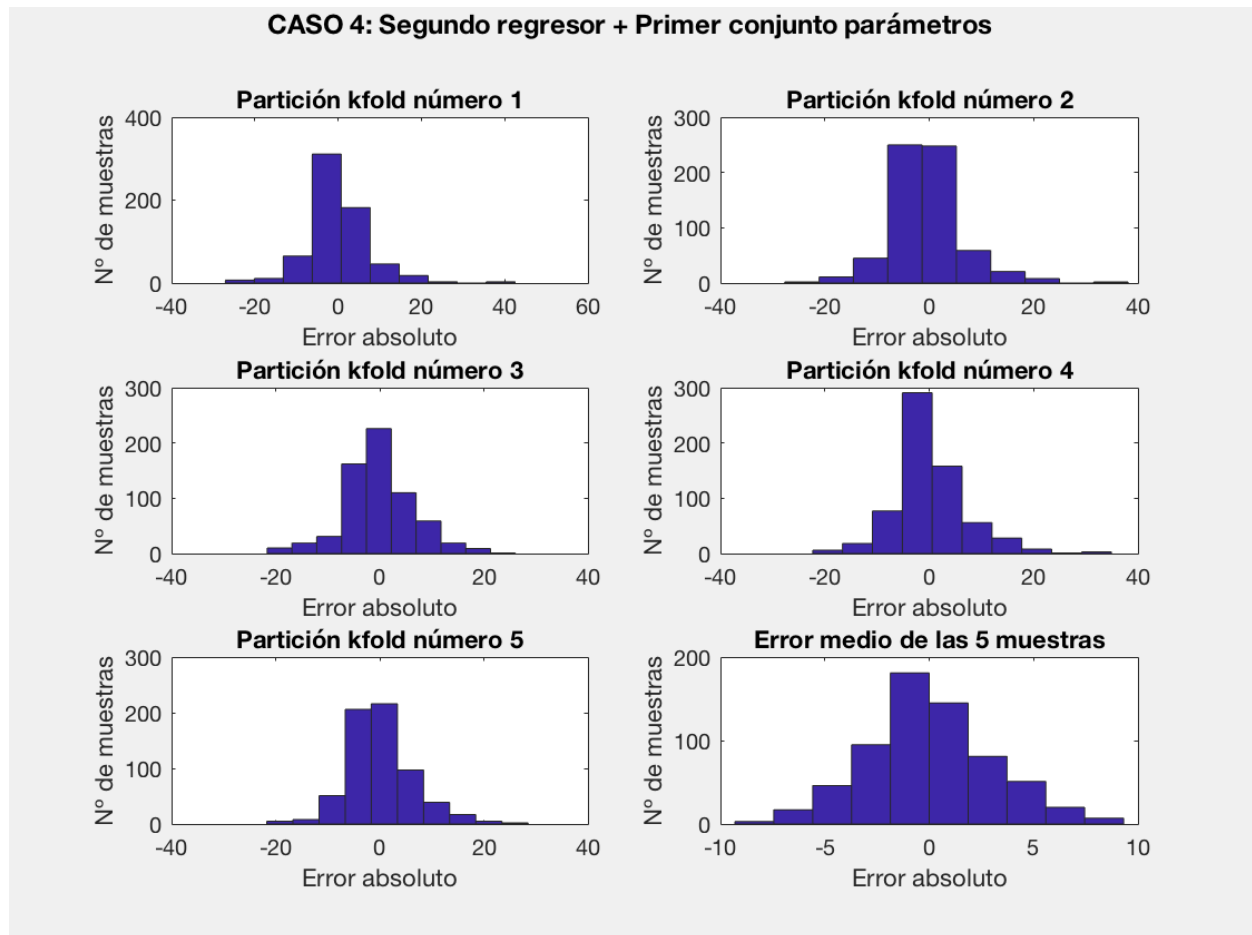


Figura 6-16. Histogramas de error para el caso 4 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-28. Errores cometidos en el caso 4 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	54.292448	45.372305	43.667700	47.563140	39.380136	46.063328
SEM	5.393751	4.405221	3.066412	4.094940	3.042716	4.000728

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.5 Caso 5. Segundo regresor y segundo conjunto de parámetros

Para este caso, se elige el segundo regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, f.a.c_{i-1}]$
- Parámetros:

Tabla 6-29. Parámetros del caso 5 en SSN.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.99
Gamma (γ)	470
Param. 1 de la f. <i>kernel</i>	360
Param. 2 de la f. <i>kernel</i>	54

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

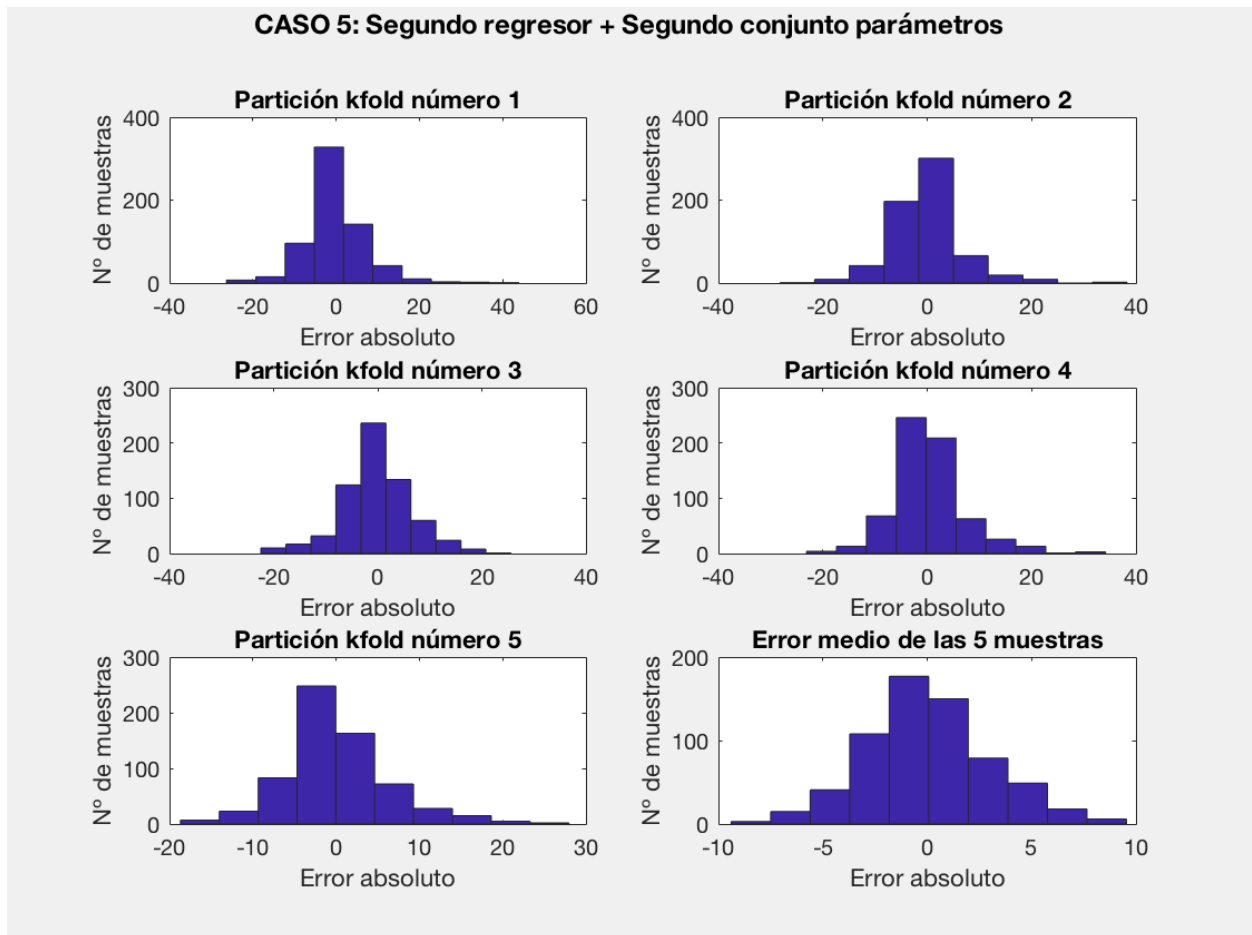


Figura 6-17. Histogramas de error para el caso 5 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-30. Errores cometidos en el caso 5 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	53.133801	44.824724	43.793680	47.248870	38.312076	45.462630
SEM	5.311127	4.341405	3.067860	4.103989	2.891650	3.943206

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.6 Caso 6. Segundo regresor y tercer conjunto de parámetros

Para este caso, se elige el segundo regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, f.a.c._{i-1}]$
- Parámetros:

Tabla 6-31. Parámetros del caso 6 en SSN.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.92
Gamma (γ)	265
Param. 1 de la f. <i>kernel</i>	78
Param. 2 de la f. <i>kernel</i>	15

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

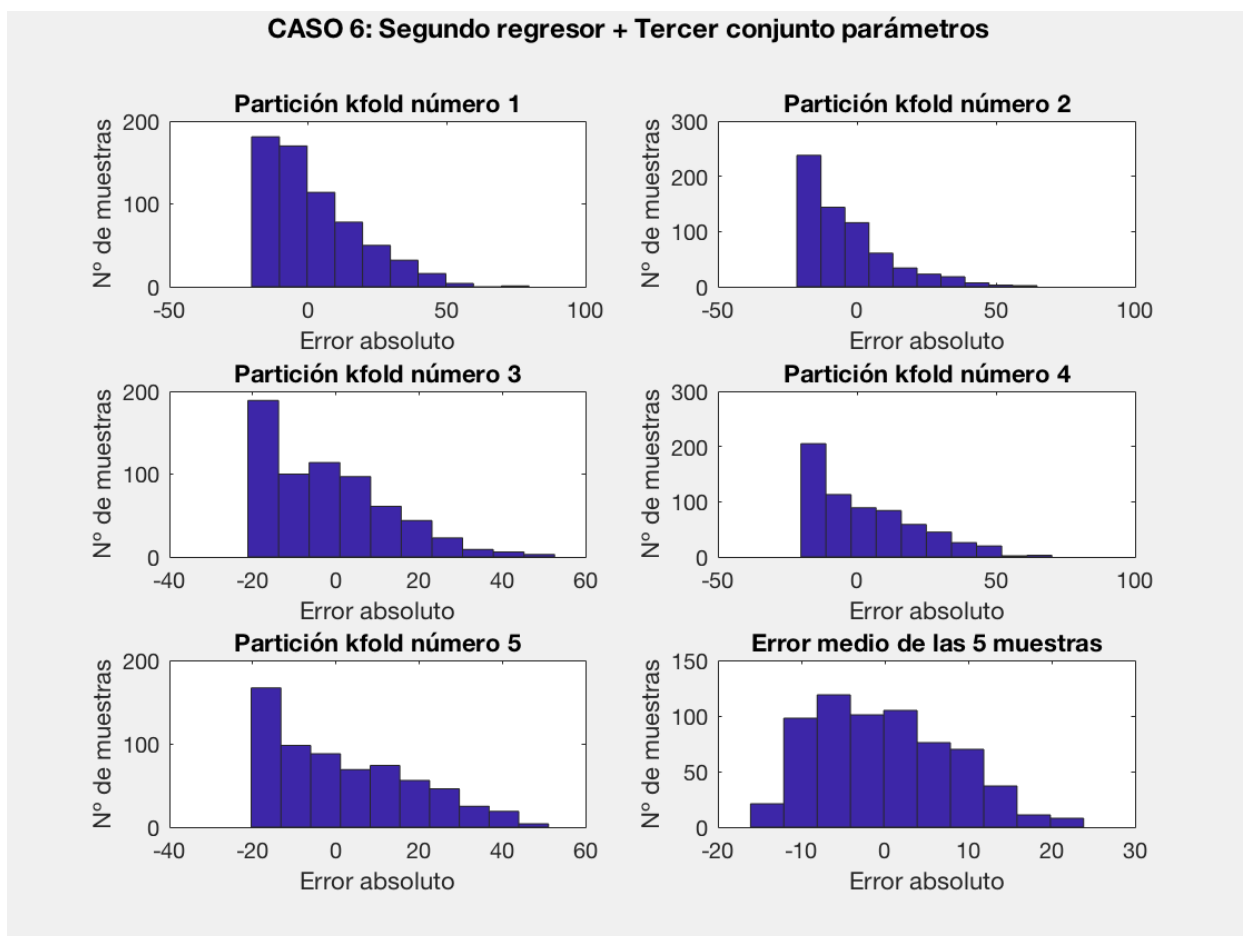


Figura 6-18. Histogramas de error para el caso 6 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-32. Errores cometidos en el caso 6 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	296.970036	273.507816	221.796179	375.414362	297.336629	292.885794
SEM	19.706176	15.704225	11.538894	23.106226	15.037640	17.014109

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.7 Caso 7. Tercer regresor y primer conjunto de parámetros

Para este caso, se elige el tercer regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = \left[y_{i-1}, \sqrt{y_{i-2}}, \text{año}_i, 0.6 \cdot \frac{\text{std}_{i-1} + \text{std}_{i-2} + \text{std}_{i-3}}{3} \right]$
- Parámetros:

Tabla 6-33. Parámetros del caso 7 en SSN.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	0.5
Param. 2 de la f. <i>kernel</i>	164

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

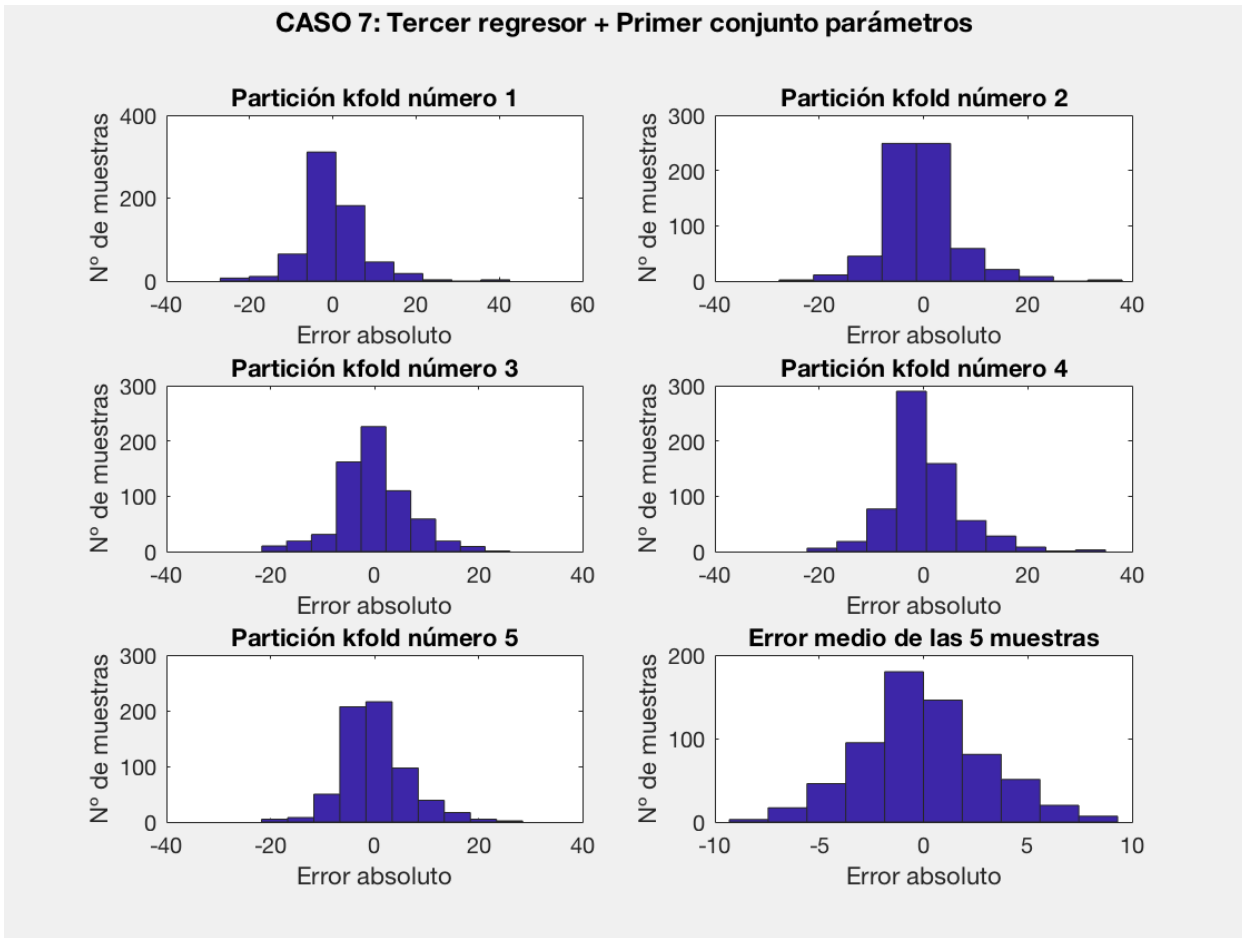


Figura 6-19. Histogramas de error para el caso 7 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-34. Errores cometidos en el caso 7 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	54.292448	45.372305	43.667700	47.563140	39.380136	46.055146
SEM	5.393751	4.405221	3.066412	4.094940	3.042716	4.000608

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.8 Caso 8. Tercer regresor y segundo conjunto de parámetros

Para este caso, se elige el tercer regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor:
$$x_i = \left[y_{i-1}, \sqrt{y_{i-2}}, año_i, 0.6 \cdot \frac{std_{i-1} + std_{i-2} + std_{i-3}}{3} \right]$$
- Parámetros:

Tabla 6-35. Parámetros del caso 8 en SSN.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.99
Gamma (γ)	470
Param. 1 de la f. kernel	360
Param. 2 de la f. kernel	54

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

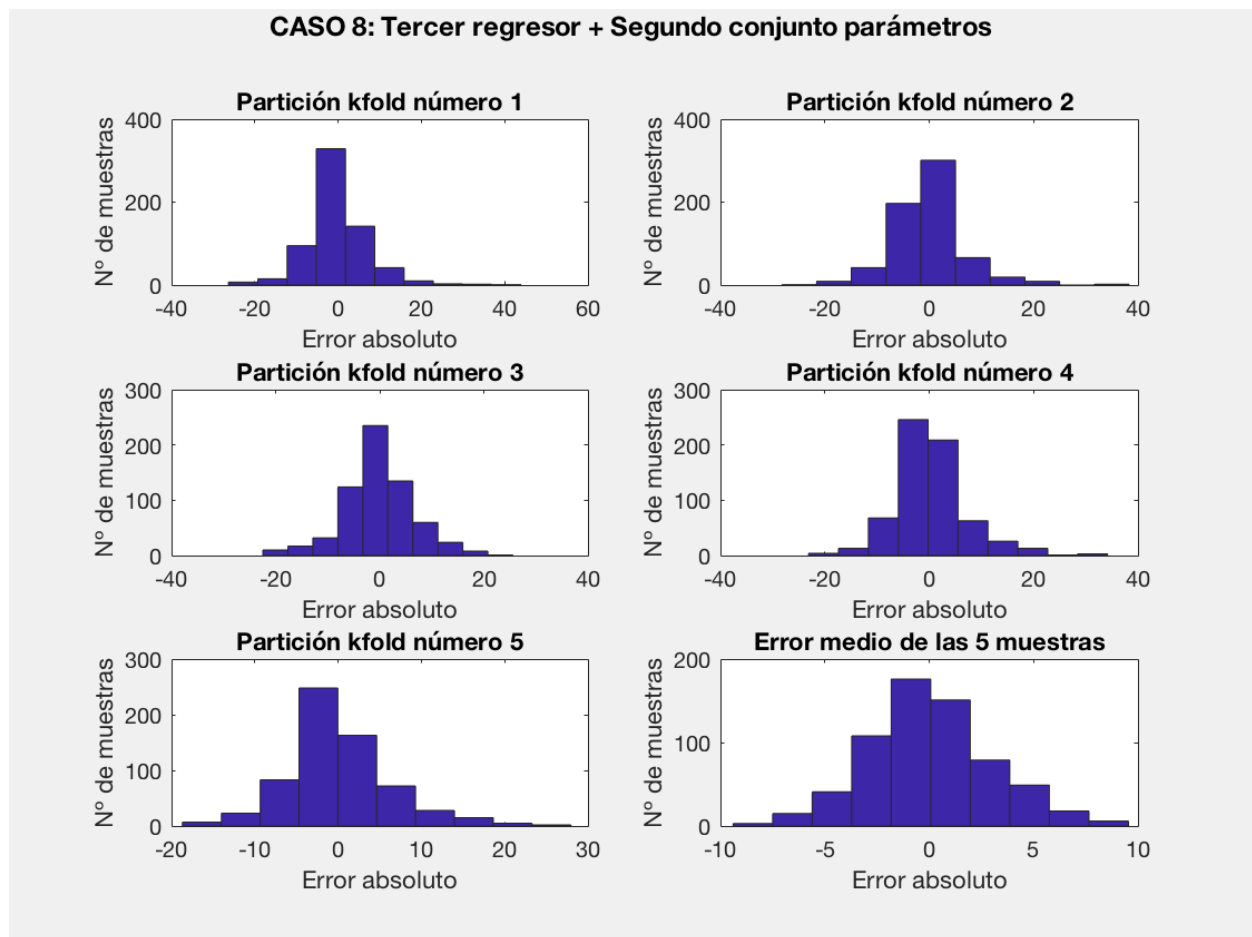


Figura 6-20. Histogramas de error para el caso 8 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-36. Errores cometidos en el caso 8 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	53.133801	44.824724	43.793680	47.248870	38.312076	45.451856
SEM	5.311127	4.341405	3.067860	4.103989	2.891650	3.942963

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.9 Caso 9. Tercer regresor y tercer conjunto de parámetros

Para este caso, se elige el tercer regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = \left[y_{i-1}, \sqrt{y_{i-2}}, \text{año}_i, 0.6 \cdot \frac{\text{std}_{i-1} + \text{std}_{i-2} + \text{std}_{i-3}}{3} \right]$
- Parámetros:

Tabla 6-37. Parámetros del caso 9 en SSN.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.92
Gamma (γ)	265
Param. 1 de la f. <i>kernel</i>	78
Param. 2 de la f. <i>kernel</i>	15

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

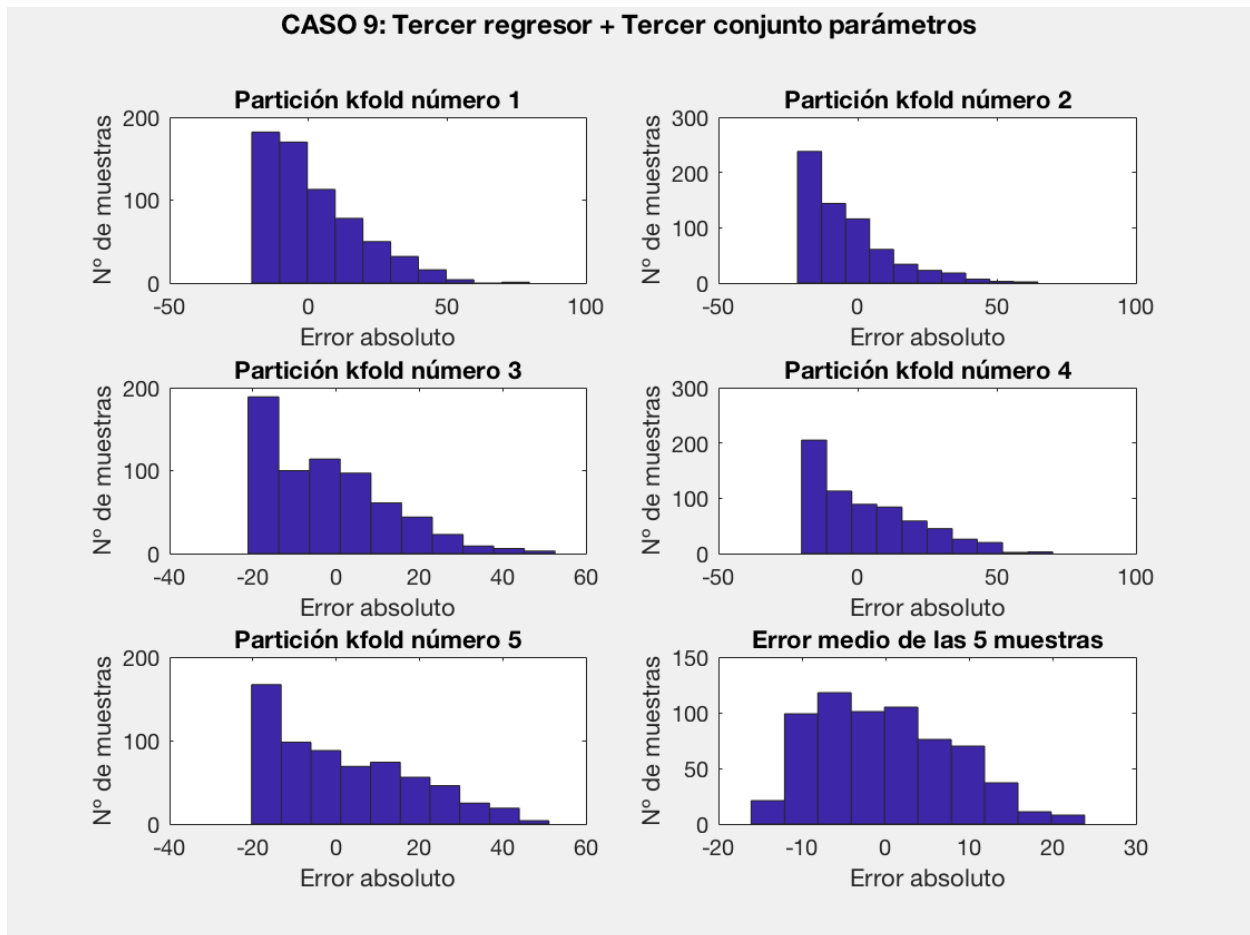


Figura 6-21. Histogramas de error para el caso 9 en SSN.

Los errores que se cometen en este caso, son:

Tabla 6-38. Errores cometidos en el caso 9 de SSN.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	296.970036	273.507816	221.796179	375.414362	297.336629	293.005004
SEM	19.706176	15.704225	11.538894	23.106226	15.037640	17.018632

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.2.10 Elección del regresor

Una vez se han ensayado y validado todos los casos, se dispone de la información necesaria para hacer la elección final, de forma resumida, en la siguiente tabla se pueden comparar los errores medios cometidos en cada caso. Éstos se han calculado como la media de cada iteración del proceso de validación cruzada.

Tabla 6-39. Resumen de errores por caso en SSN.

CASO	MSE	SEM
Caso 1	46.055146097663524	4.000608448727943
Caso 2	45.451856807489655	3.942963419438617
Caso 3	293.005004880137903	17.018632652508835
Caso 4	46.063328080804318	4.000728583469870
Caso 5	45.462630801626808	3.943206677417650
Caso 6	292.885794894466812	17.014109643246698
Caso 7	46.055146097663524	4.000608448727943
Caso 8	45.451856807489655	3.942963419438617
Caso 9	293.005004880137903	17.018632652508835

El regresor elegido para la regresión, por tener los mejores indicadores de bondad, es el correspondiente al caso 8. Tenemos entonces que:

- Regresor: $x_i = \left[y_{i-1}, \sqrt{y_{i-2}}, \text{año}_i, 0.6 \cdot \frac{\text{std}_{i-1} + \text{std}_{i-2} + \text{std}_{i-3}}{3} \right]$
- Parámetros:

Tabla 6-40. Parámetros del regresor elegido en SSN.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.99
Gamma (γ)	470
Param. 1 de la f. <i>kernel</i>	360
Param. 2 de la f. <i>kernel</i>	54

6.2.11 Predicción

Tras aplicar el conjunto regresor-parámetros al problema, separando los datos entre set de entrenamiento y set de validación destinando un 75% de los disponibles para el primero y el resto para el segundo, obtenemos un error medio en valor absoluto de 18.8542 con una desviación estándar de 25.9777 . A continuación se muestra una gráfica con el valor real de las muestras y su predicción.

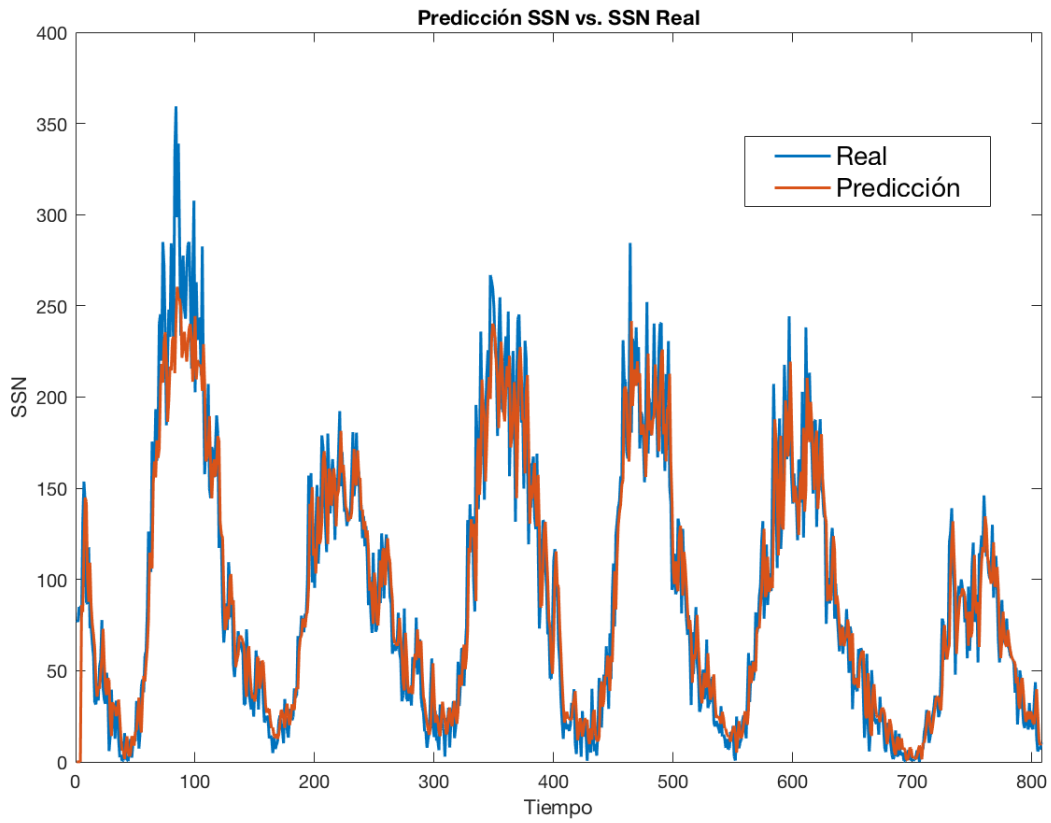


Figura 6-22. Predicción en SSN.

En detalle:

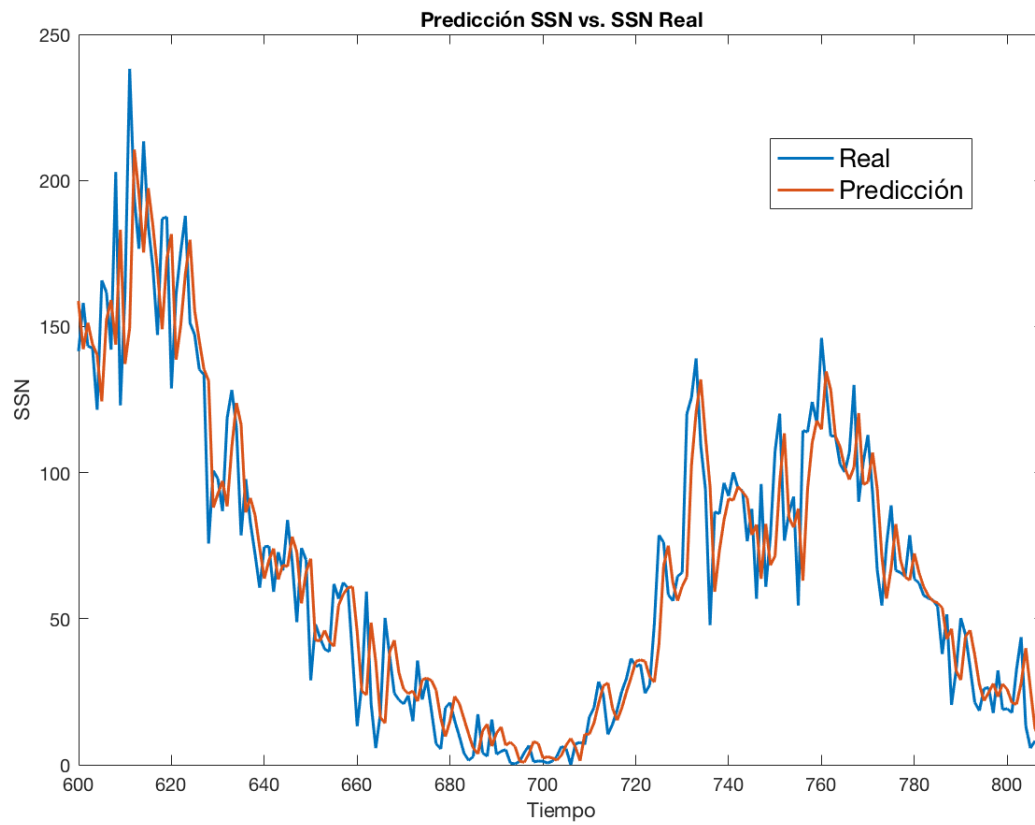


Figura 6-23. Detalle de predicción en SSN.

A continuación, se puede observar un histograma de los errores cometidos y dibujada encima, la curva de la distribución normal asociada a dicho histograma.

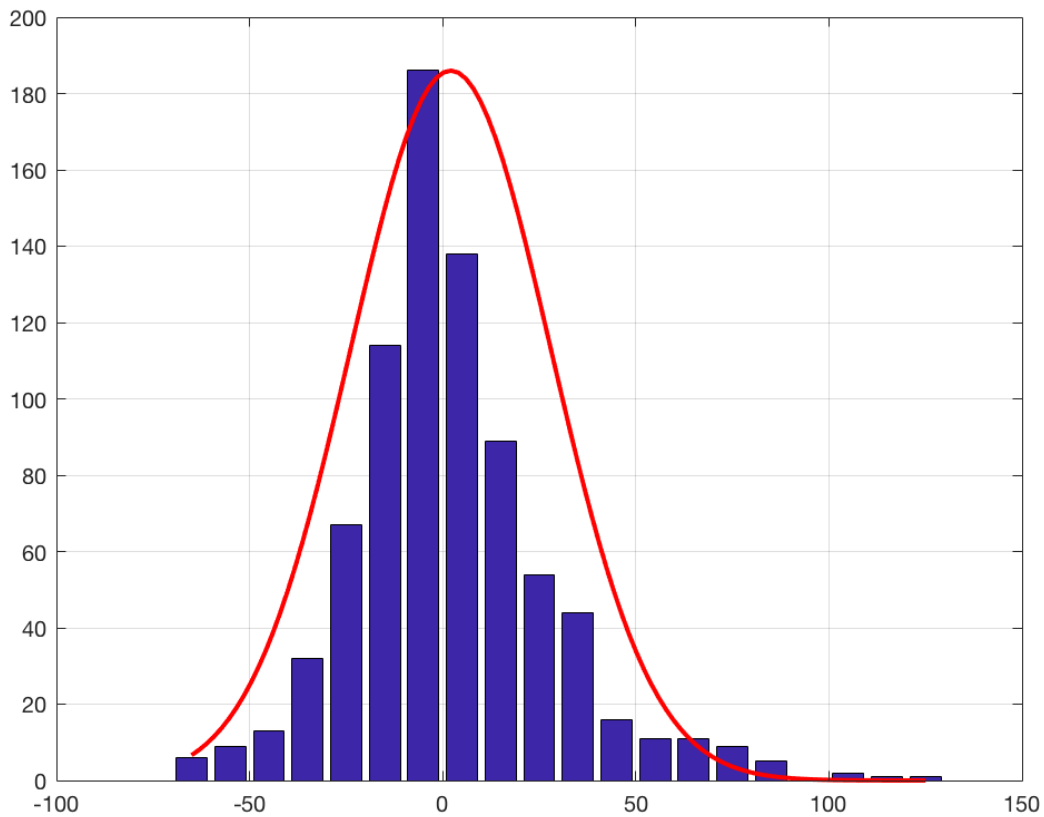


Figura 6-24. Histograma y distribución normal del error en SSN.

6.3 Resultados SMA

6.3.1 Caso 1. Primer regresor y primer conjunto de parámetros

Para este caso, se elige el primer regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-41. Parámetros del caso 1 en DEL.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	3
Param. 2 de la f. <i>kernel</i>	28

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado. Este caso es idéntico a emplear mínimos cuadrados en la predicción de la secuencia temporal.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

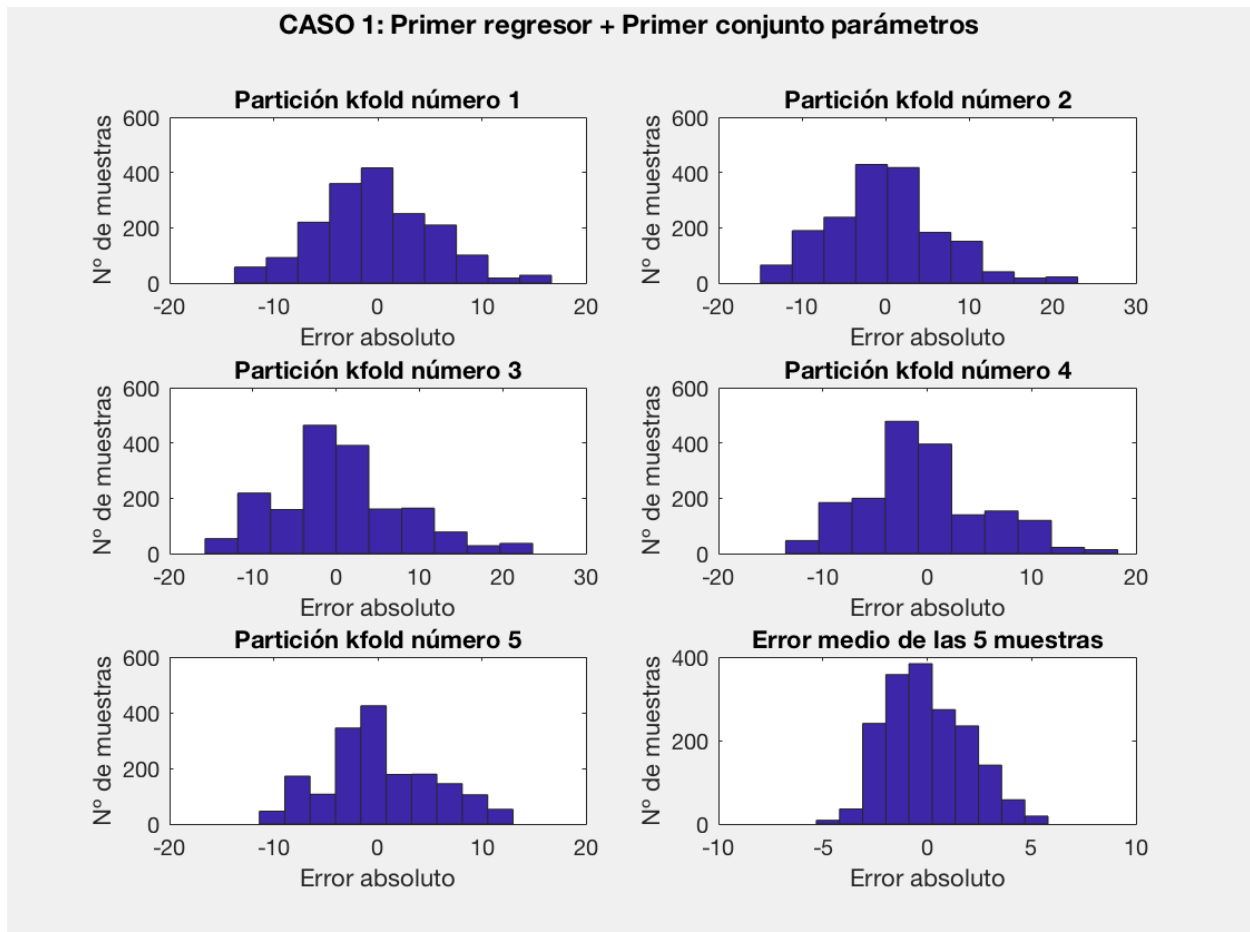


Figura 6-25. Histogramas de error para el caso 1 en DEL

Los errores que se cometen en este caso, son:

Tabla 6-42. Errores cometidos en el caso 1 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	31.356810	45.827506	57.792855	33.584705	27.265117	39.165399
SEM	1.043963	1.699295	2.084170	1.078677	0.804561	1.342133

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.2 Caso 2. Primer regresor y segundo conjunto de parámetros

Para este caso, se elige el primer regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-43. Parámetros del caso 2 en DEL.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.97
Gamma (γ)	175
Param. 1 de la f. <i>kernel</i>	615
Param. 2 de la f. <i>kernel</i>	167

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

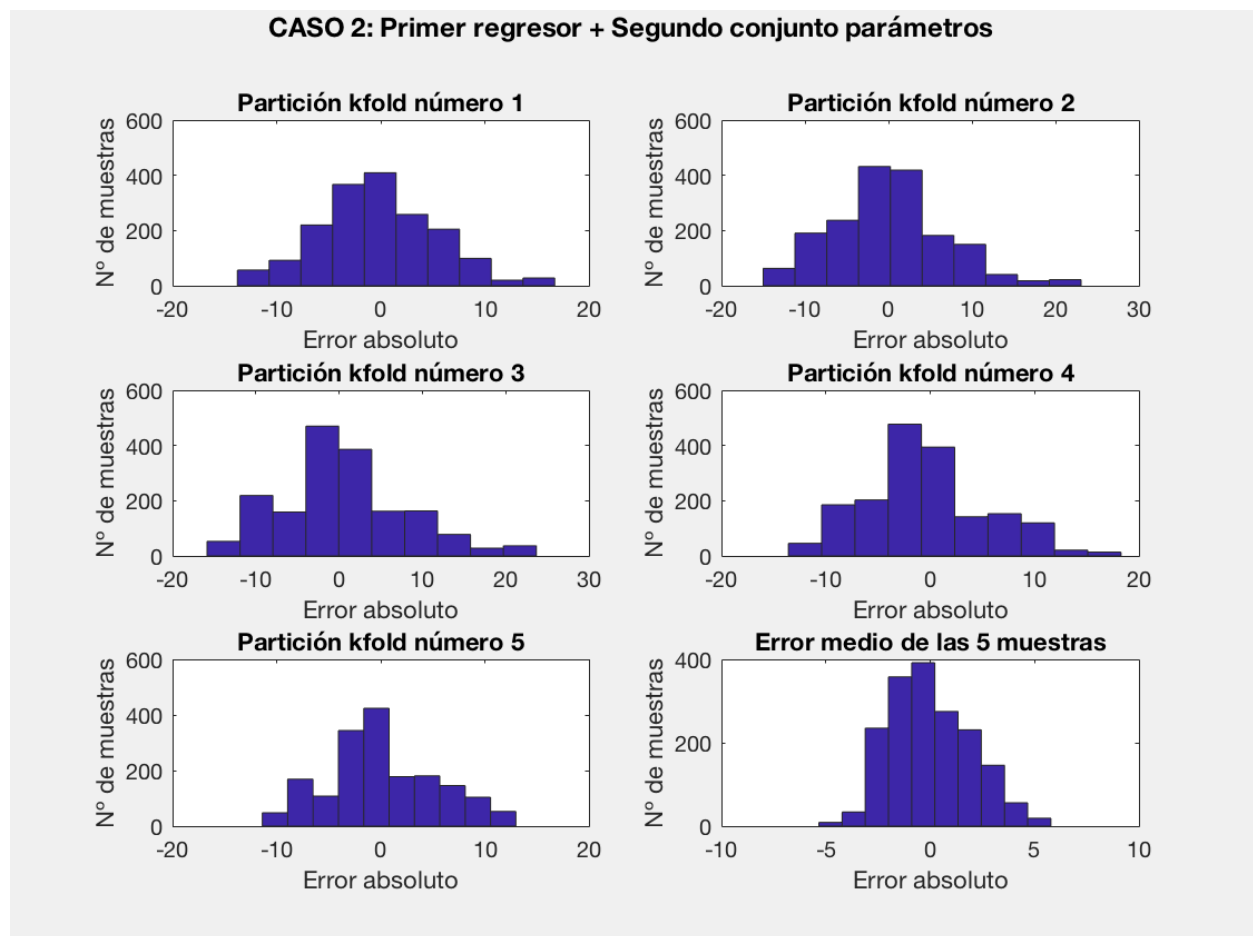


Figura 6-26. Histogramas de error para el caso 2 en DEL.

Los errores que se cometen en este caso, son:

Tabla 6-44. Errores cometidos en el caso 2 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	31.359716	45.843230	57.823545	33.589266	27.274251	39.178632
SEM	1.044412	1.703398	2.096177	1.078995	0.804997	1.345611

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.3 Caso 3. Primer regresor y tercer conjunto de parámetros

Para este caso, se elige el primer regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, y_{i-3}, y_{i-4}]$
- Parámetros:

Tabla 6-45. Parámetros del caso 3 en DEL.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.995
Gamma (γ)	38
Param. 1 de la f. <i>kernel</i>	34
Param. 2 de la f. <i>kernel</i>	615

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

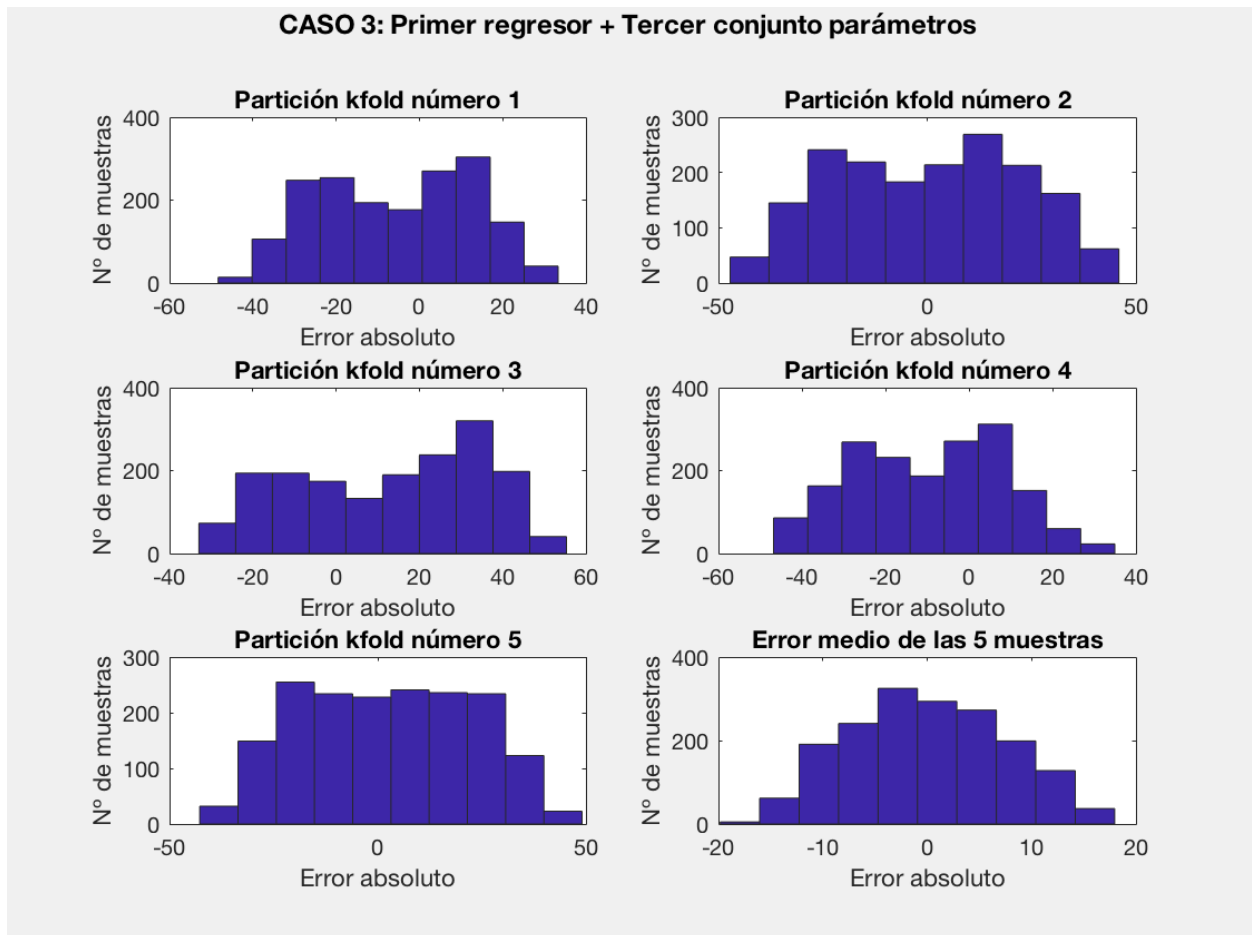


Figura 6-27. Histogramas de error para el caso 3 en DEL.

Los errores que se cometen en este caso, son:

Tabla 6-46. Errores cometidos en el caso 3 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	361.649245	492.156574	652.621462	413.071220	409.842113	465.868123
SEM	9.104952	11.301062	14.666535	11.320950	9.839658	11.246631

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.4 Caso 4. Segundo regresor y primer conjunto de parámetros

Para este caso, se elige el segundo regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, mes_{i-1}, día_{i-1}, hora_{i-1}]$
- Parámetros:

Tabla 6-47. Parámetros del caso 4 en DEL.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	3
Param. 2 de la f. <i>kernel</i>	28

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

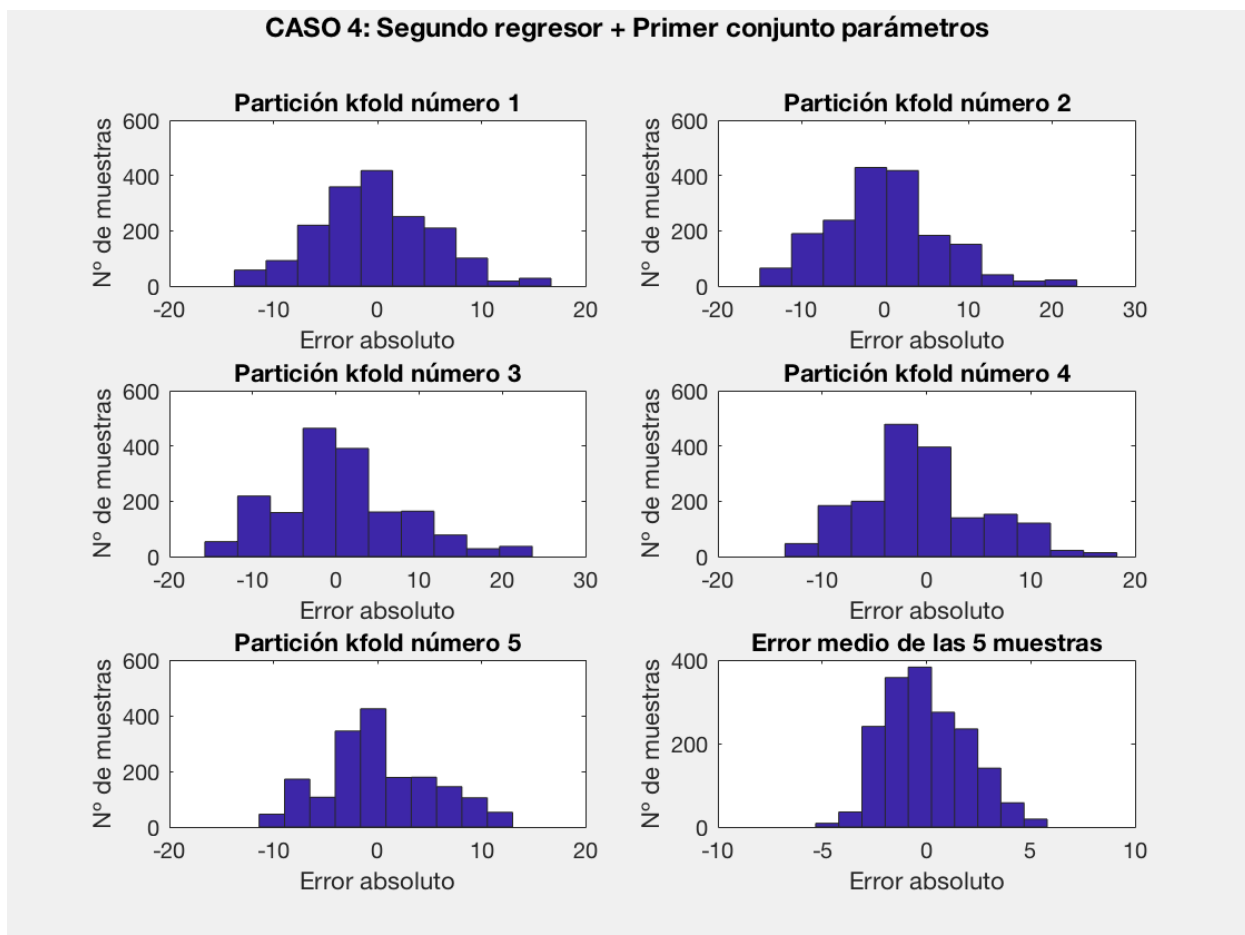


Figura 6-28. Histogramas de error para el caso 4 en DEL.

Los errores que se cometen en este caso, son:

Tabla 6-48. Errores cometidos en el caso 4 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	31.356810	45.827506	57.792855	33.584705	27.265117	39.164996
SEM	1.043963	1.699295	2.084170	1.078677	0.804561	1.342066

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.5 Caso 5. Segundo regresor y segundo conjunto de parámetros

Para este caso, se elige el segundo regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, mes_{i-1}, día_{i-1}, hora_{i-1}]$
- Parámetros:

Tabla 6-49. Parámetros del caso 5 en SSN.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.97
Gamma (γ)	175
Param. 1 de la f. <i>kernel</i>	615
Param. 2 de la f. <i>kernel</i>	167

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

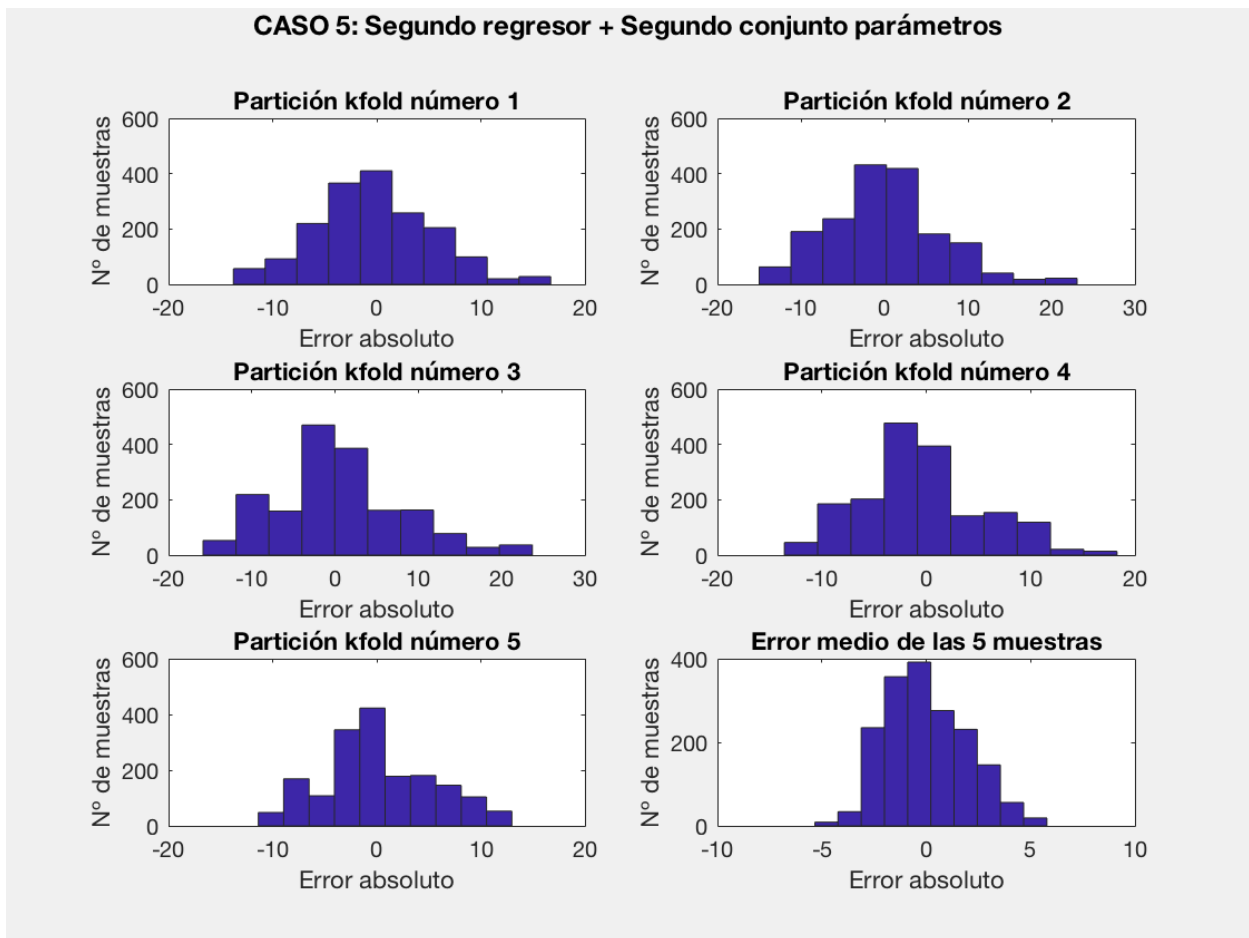


Figura 6-29. Histogramas de error para el caso 5 en DEL

Los errores que se cometen en este caso, son:

Tabla 6-50. Errores cometidos en el caso 5 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	31.359716	45.843230	57.823545	33.589266	27.274251	39.178002
SEM	1.044412	1.703398	2.096177	1.078995	0.804997	1.345596

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.6 Caso 6. Segundo regresor y tercer conjunto de parámetros

Para este caso, se elige el segundo regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, mes_{i-1}, día_{i-1}, hora_{i-1}]$
- Parámetros:

Tabla 6-51. Parámetros del caso 6 en SSN.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.995
Gamma (γ)	38
Param. 1 de la f. kernel	34
Param. 2 de la f. kernel	615

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

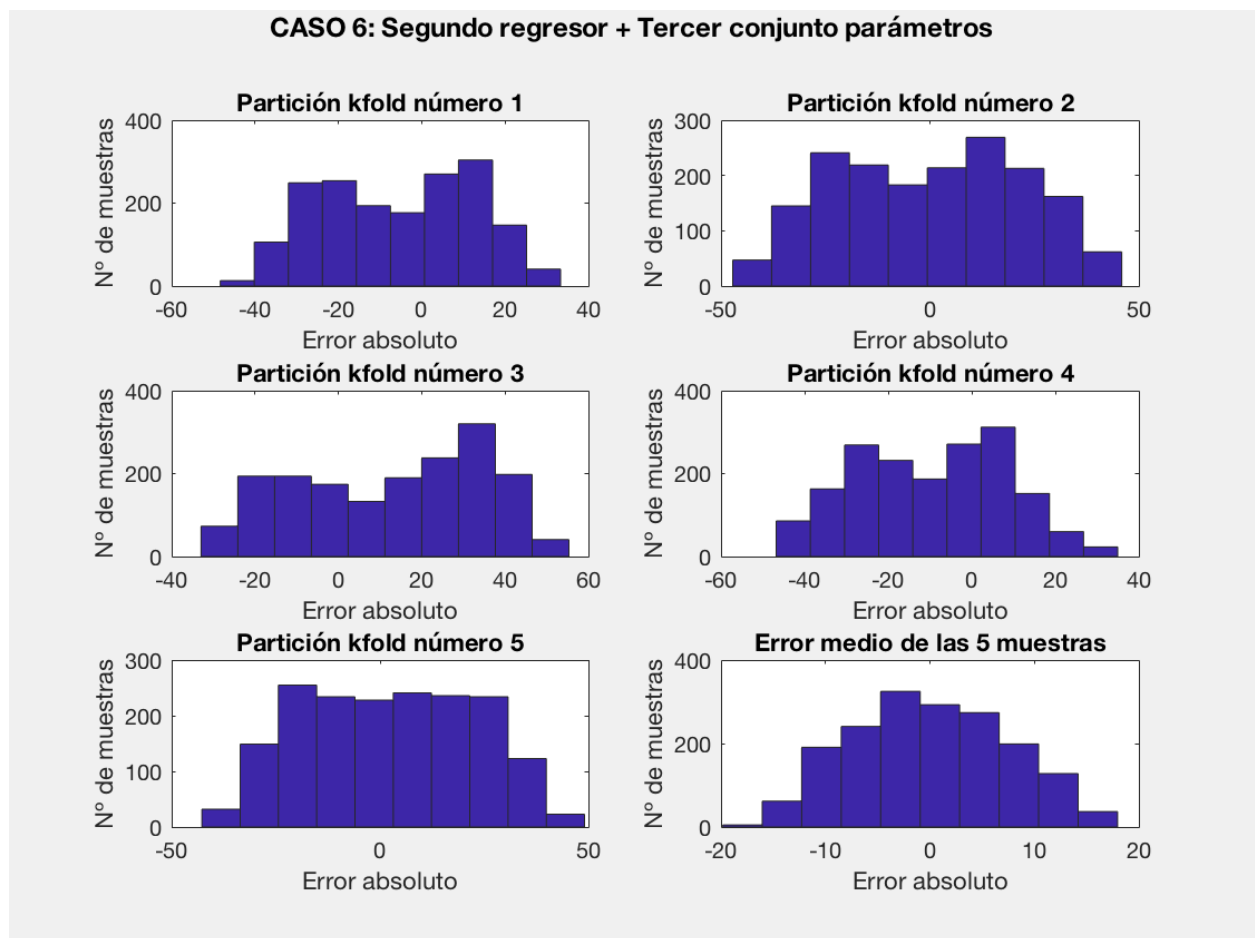


Figura 6-30. Histogramas de error para el caso 6 en DEL.

Los errores que se cometen en este caso, son:

Tabla 6-52. Errores cometidos en el caso 6 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	361.649245	492.156574	652.621462	413.071220	409.842113	465.690668
SEM	9.104952	11.301062	14.666535	11.320950	9.839658	11.233264

Con k_1, k_2, \dots, k_5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.7 Caso 7. Tercer regresor y primer conjunto de parámetros

Para este caso, se elige el tercer regresor y el primer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = \left[y_{i-1}, 0.4 \cdot y_{i-2} + 0.6 \cdot \frac{y_{i-4} + y_{i-3} + y_{i-2}}{3}, mes_{i-1}, día_{i-1}, hora_{i-1} \right]$
- Parámetros:

Tabla 6-53. Parámetros del caso 7 en DEL.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	3
Param. 2 de la f. <i>kernel</i>	28

En este caso, al ser la función *kernel* lineal, los parámetros «Param. 1» y «Param. 2» no tendrán efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

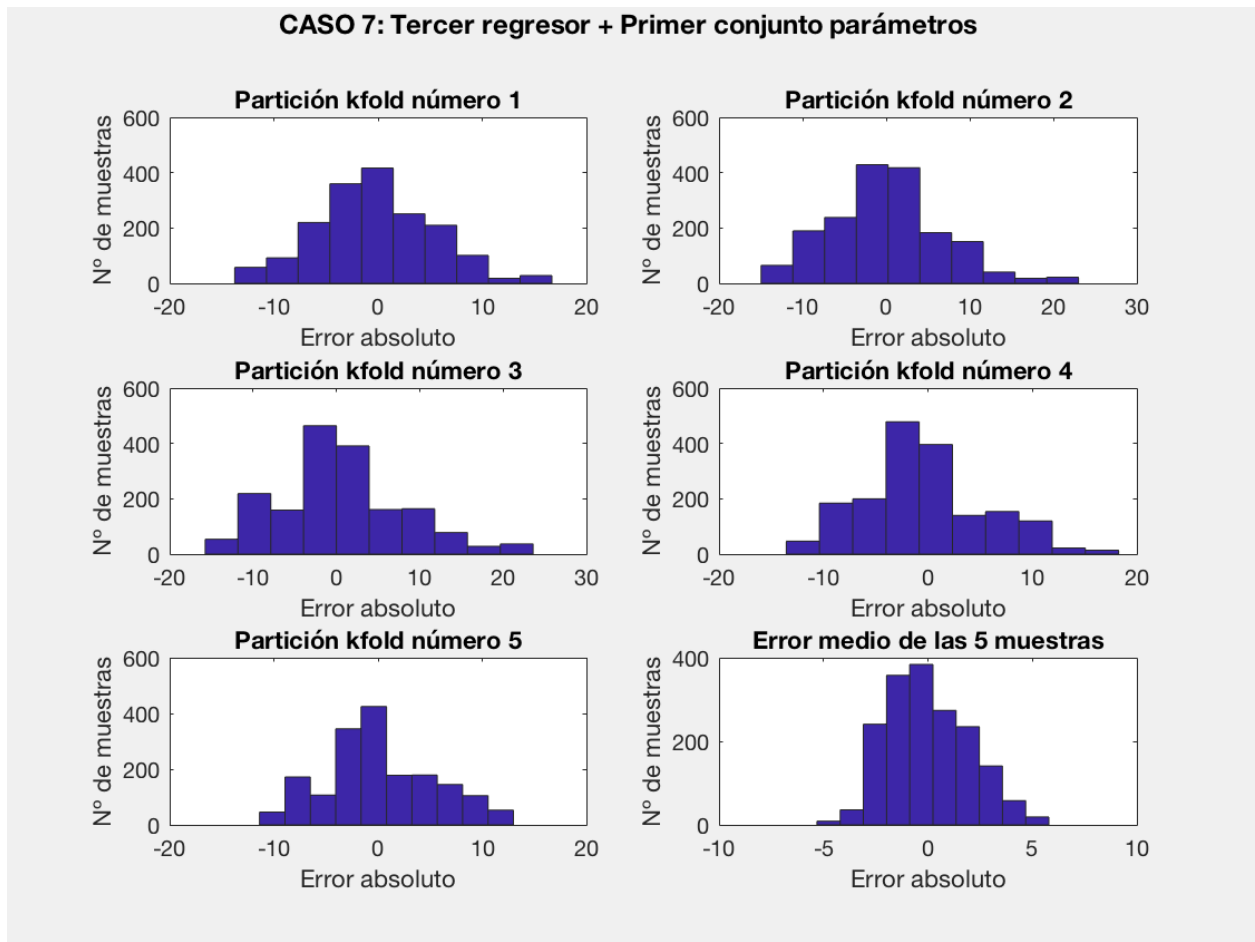


Figura 6-31. Histogramas de error para el caso 7 en DEL.

Los errores que se cometen en este caso, son:

Tabla 6-54. Errores cometidos en el caso 7 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	31.356810	45.827506	57.792855	33.584705	27.265117	39.165399
SEM	1.043963	1.699295	2.084170	1.078677	0.804561	1.342133

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.8 Caso 8. Tercer regresor y segundo conjunto de parámetros

Para este caso, se elige el tercer regresor y el segundo conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = \left[y_{i-1}, 0.4 \cdot y_{i-2} + 0.6 \cdot \frac{y_{i-4} + y_{i-3} + y_{i-2}}{3}, mes_{i-1}, día_{i-1}, hora_{i-1} \right]$
- Parámetros:

Tabla 6-55. Parámetros del caso 8 en DEL.

Parámetro	Valor
Función kernel	Radial Basis
Tau (τ)	0.97
Gamma (γ)	175
Param. 1 de la f. <i>kernel</i>	615
Param. 2 de la f. <i>kernel</i>	167

En este caso, al ser la función *kernel* «Radial Basis», el parámetros «Param. 2» no tendrá efecto alguno en el resultado.

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

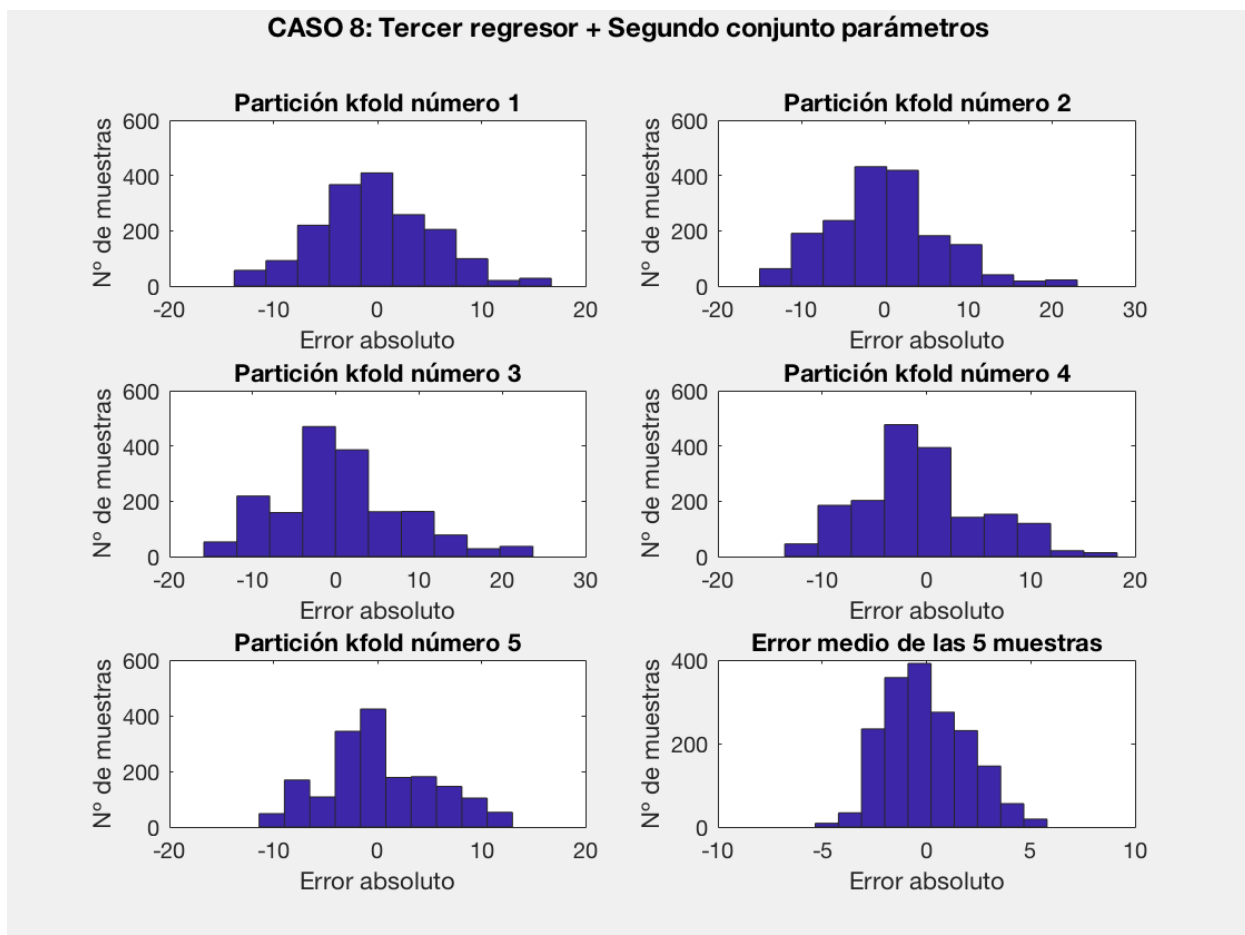


Figura 6-32. Histogramas de error para el caso 8 en DEL.

Los errores que se cometen en este caso, son:

Tabla 6-56. Errores cometidos en el caso 8 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	31.359716	45.843230	57.823545	33.589266	27.274251	39.178632
SEM	1.044412	1.703398	2.096177	1.078995	0.804997	1.345611

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.9 Caso 9. Tercer regresor y tercer conjunto de parámetros

Para este caso, se elige el tercer regresor y el tercer conjunto de parámetros por lo que tenemos que:

- Regresor: $x_i = \left[y_{i-1}, 0.4 \cdot y_{i-2} + 0.6 \cdot \frac{y_{i-4} + y_{i-3} + y_{i-2}}{3}, mes_{i-1}, día_{i-1}, hora_{i-1} \right]$
- Parámetros:

Tabla 6-57. Parámetros del caso 9 en DEL.

Parámetro	Valor
Función kernel	Sigmoidal
Tau (τ)	0.995
Gamma (γ)	38
Param. 1 de la f. <i>kernel</i>	34
Param. 2 de la f. <i>kernel</i>	615

Con estos parámetros y regresor, una vez entrenado y validado el algoritmo, obtenemos los siguientes histogramas de error para cada iteración del proceso de validación cruzada. Se incluye además, el histograma del error medio cometido.

El error representado es el error absoluto, no cuadrático.

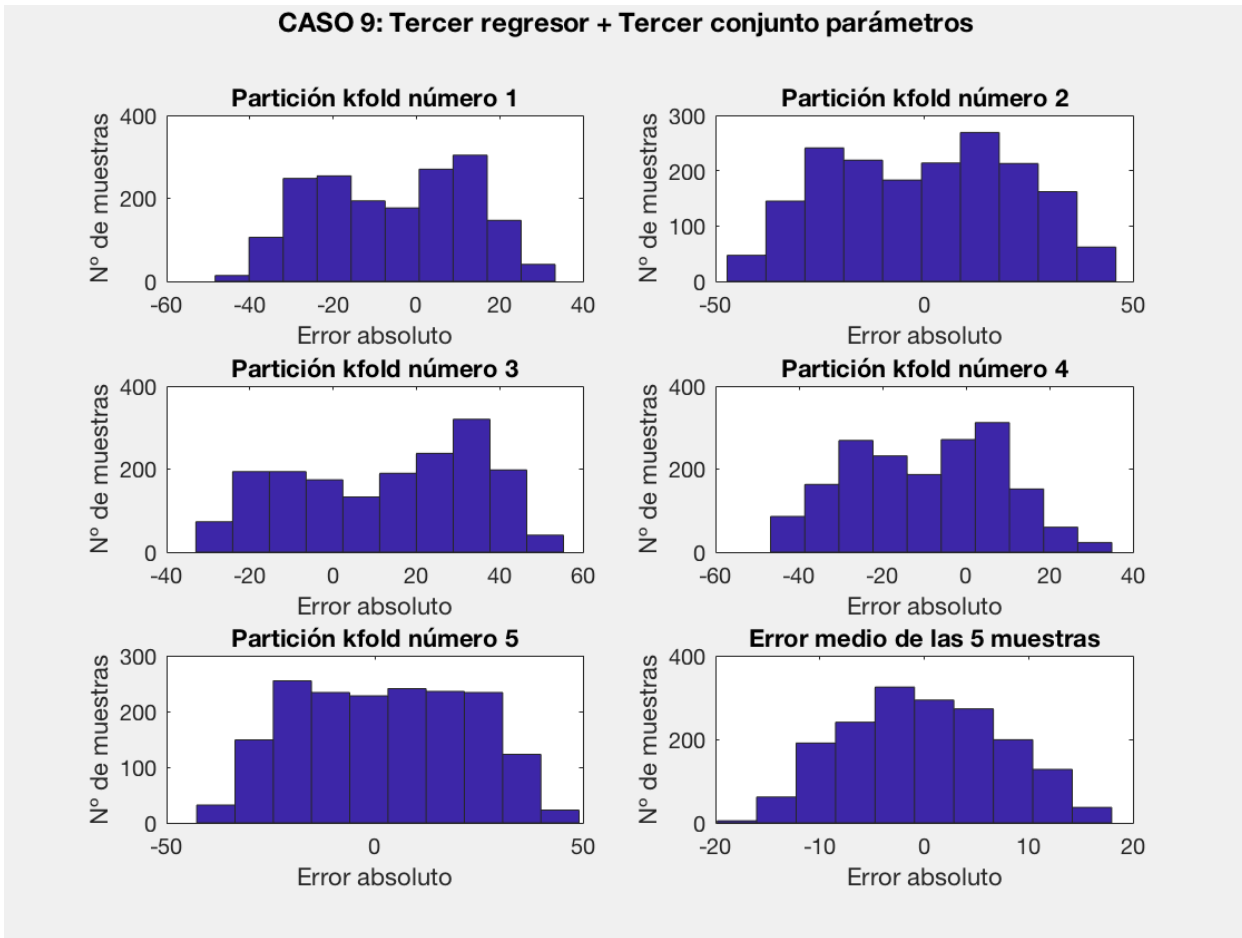


Figura 6-33. Histogramas de error para el caso 9 en DEL.

Los errores que se cometen en este caso, son:

Tabla 6-58. Errores cometidos en el caso 9 de DEL.

Tipo Error	k1	k2	k3	k4	k5	Media
MSE	361.649245	492.156574	652.621462	413.071220	409.842113	465.868123
SEM	9.104952	11.301062	14.666535	11.320950	9.839658	11.246631

Con k1, k2, ..., k5 cada una de las iteraciones del proceso de validación cruzada.

Se ha reducido el número de decimales empleados en el cálculo para su cómoda representación y visualización. En la sub-sección 10 de esta sección, se muestran los resultados de los valores medios (los que vamos a emplear) con todas las cifras significativas.

6.3.10 Elección del regresor

Una vez se han ensayado y validado todos los casos, se dispone de la información necesaria para hacer la elección final, de forma resumida, en la siguiente tabla se pueden comparar los errores medios cometidos en cada caso. Éstos se han calculado como la media de cada iteración del proceso de validación cruzada.

Tabla 6-59. Resumen errores por caso en DEL

CASO	MSE	SEM
Caso 1	39.165399106892565	1.342133636703042
Caso 2	39.178632642847631	1.345611310870960
Caso 3	465.868123390057576	11.246631845909491
Caso 4	39.164996354491194	1.342066822180289
Caso 5	39.178002134574243	1.345596203263492
Caso 6	465.690668755635670	11.233264304007374
Caso 7	39.165399106892565	1.342133636703042
Caso 8	39.178632642847631	1.345611310870960
Caso 9	465.868123390057576	11.246631845909491

De esta forma, el regresor elegido para la regresión, por tener los mejores indicadores de bondad, es el correspondiente al caso 4 por tener tanto el MSE como el SEM más pequeños:

- Regresor: $x_i = [y_{i-1}, y_{i-2}, mes_{i-1}, día_{i-1}, hora_{i-1}]$
- Parámetros:

Tabla 6-60. Parámetros del regresor elegido en DEL.

Parámetro	Valor
Función kernel	Linear
Tau (τ)	1
Gamma (γ)	1
Param. 1 de la f. <i>kernel</i>	3
Param. 2 de la f. <i>kernel</i>	28

6.3.11 Predicción

Tras aplicar el conjunto regresor-parámetros al problema, separando los datos entre set de entrenamiento y set de validación destinando un 75% de los disponibles para el primero y el resto para el segundo, obtenemos un error medio en valor absoluto de 503.3092 MW con una desviación estándar de 1390.2 MW. A continuación se muestra una gráfica con el valor real de las muestras y su predicción.

No se muestran todas las muestras empleadas en el set de validación debido a la gran cantidad de éstas y para su cómoda visualización.

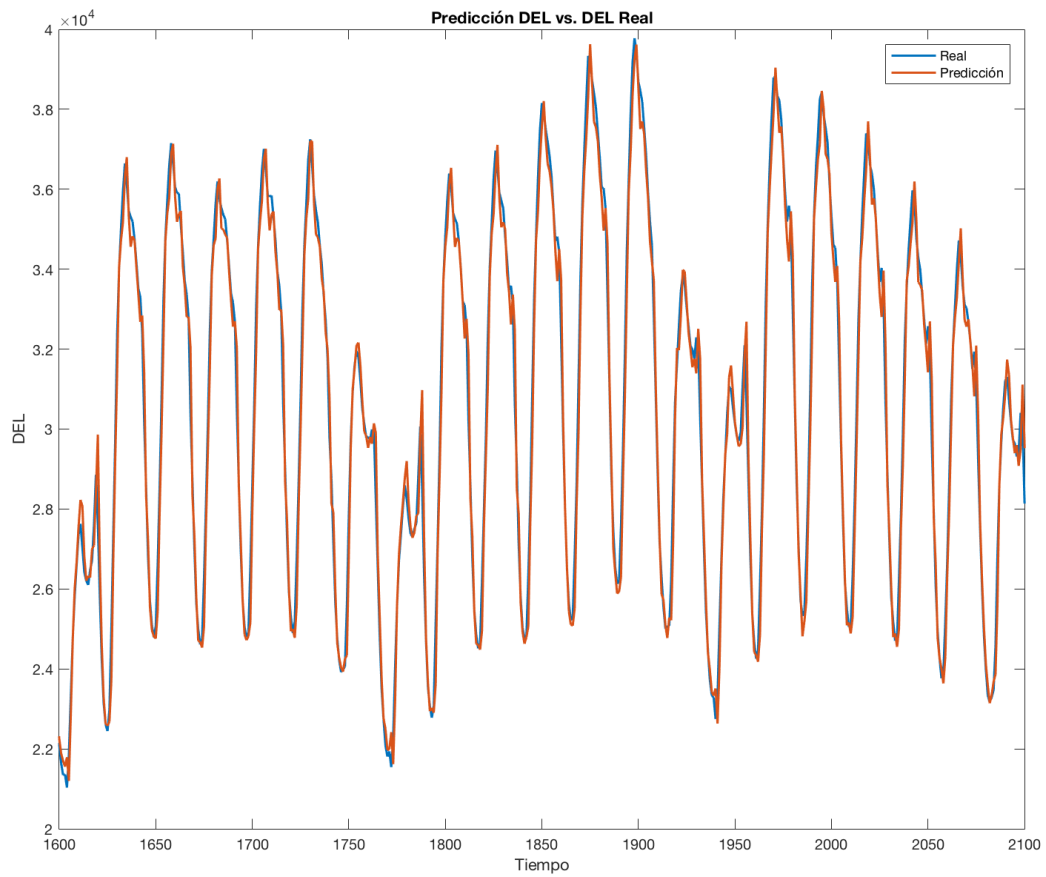


Figura 6-34. Predicción en DEL.

En detalle:

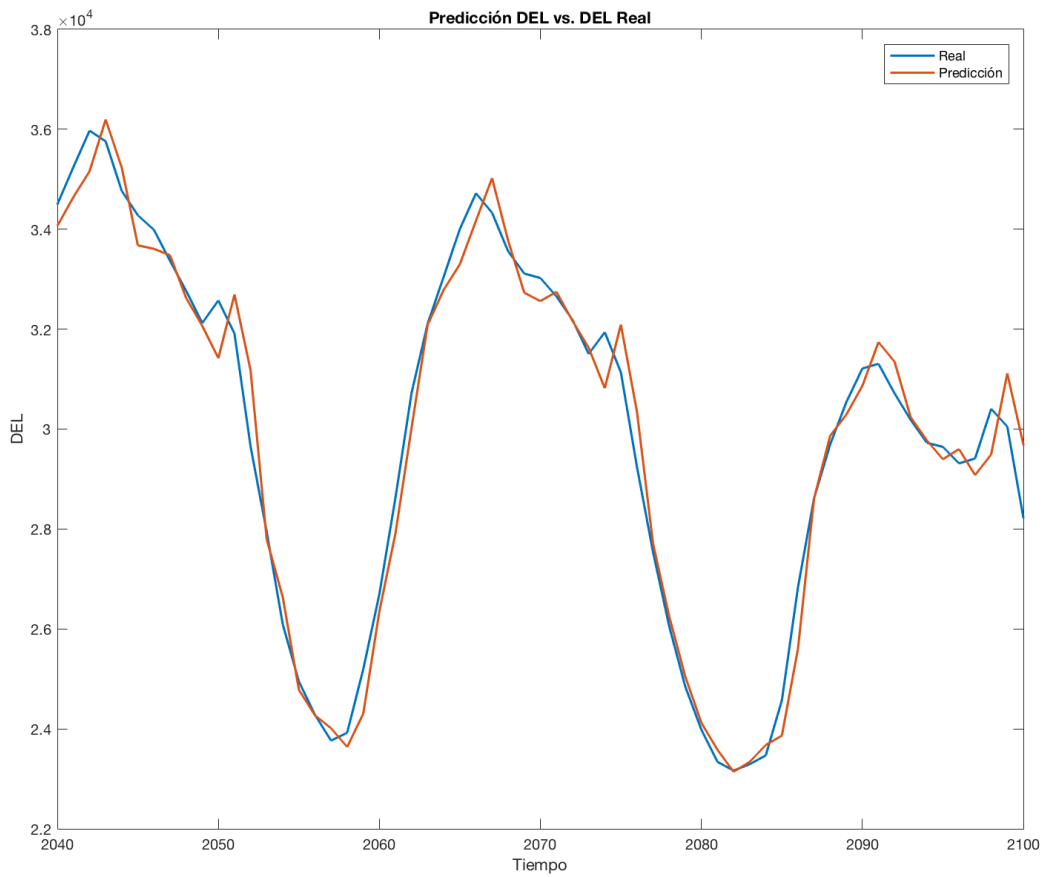


Figura 6-35. Detalle de la predicción en DEL.

A continuación, se puede observar un histograma de los errores cometidos y dibujada encima, la curva de la distribución normal asociada a dicho histograma.

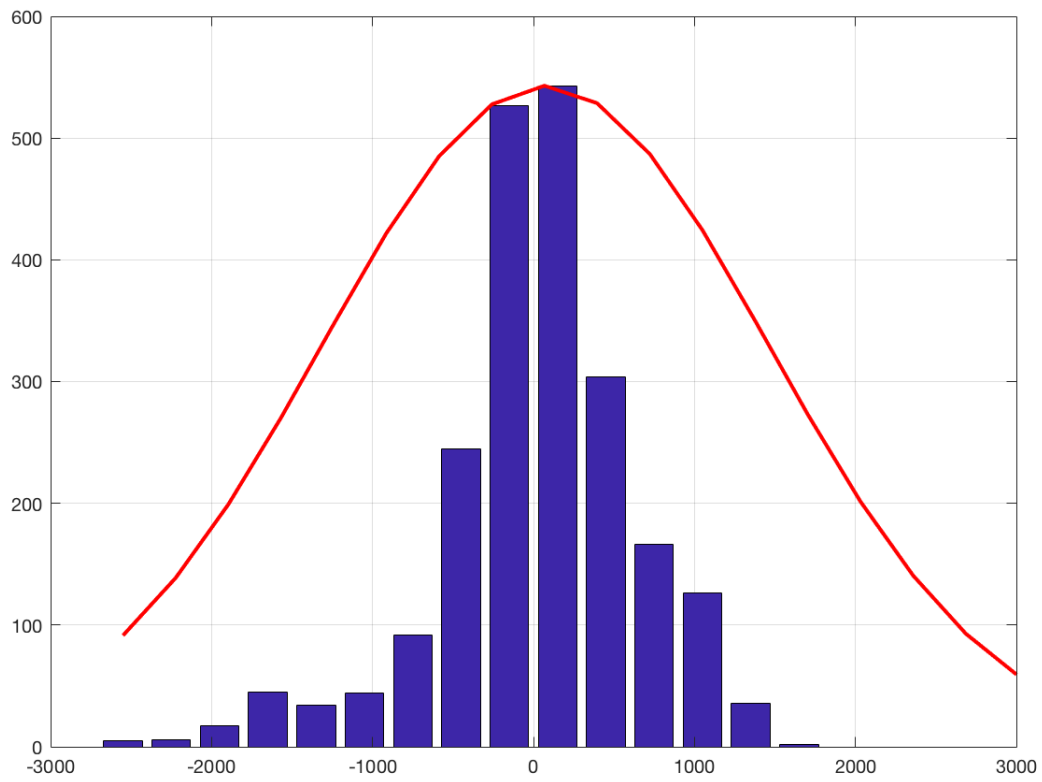


Figura 6-36. Histograma del error en DEL

7 CONCLUSIONES

El séptimo y último capítulo del documento está destinado a esclarecer las conclusiones y valoraciones del trabajo realizado así como definir las líneas de trabajo futuro que se suceden del mismo.

Para ello, se van a realizar una serie de comparativas a partir de los resultados obtenidos en el capítulo anterior. Estas comparativas se van a centrar en dos aspectos fundamentales; primero, se analizarán los resultados obtenidos mediante el método de los mínimos cuadrados y los métodos kernel y, posteriormente se debatirá sobre las bondades e influencia de la aplicación de técnicas de validación cruzada para determinar los parámetros de sintonización de los regresores e incluso los propios regresores.

7.1 Comparación entre métodos *kernel* y mínimos cuadrados

De forma consciente, en todos los ensayos y simulaciones, se ha escogido en el primer caso de todos un regresor lineal de dimensión 4 con todos los hiperparámetros iguales a la unidad. Este singular conjunto regresor-parámetros, nos proporciona el mismo resultado que arroja el método de los mínimos cuadrados ordinarios. Por tanto, podemos comparar los resultados entre los distintos métodos sin desarrollar más scripts y funciones de las que ya hay creadas.

Los regresores relativos a los métodos kernel que vamos a emplear para comparación serán, en cada caso, el que produce el mejor resultado de entre todos los casos exceptuando el primero.

7.1.1 Señal modulada en amplitud

En este problema, encontramos muy poca diferencia entre el resultado obtenido mediante mínimos cuadrados y los métodos *kernel*. No obstante, la mejor solución posible al problema de regresión es la que encontramos en el caso 4.

A continuación, se muestran los histogramas de error medios de cada uno de los dos.

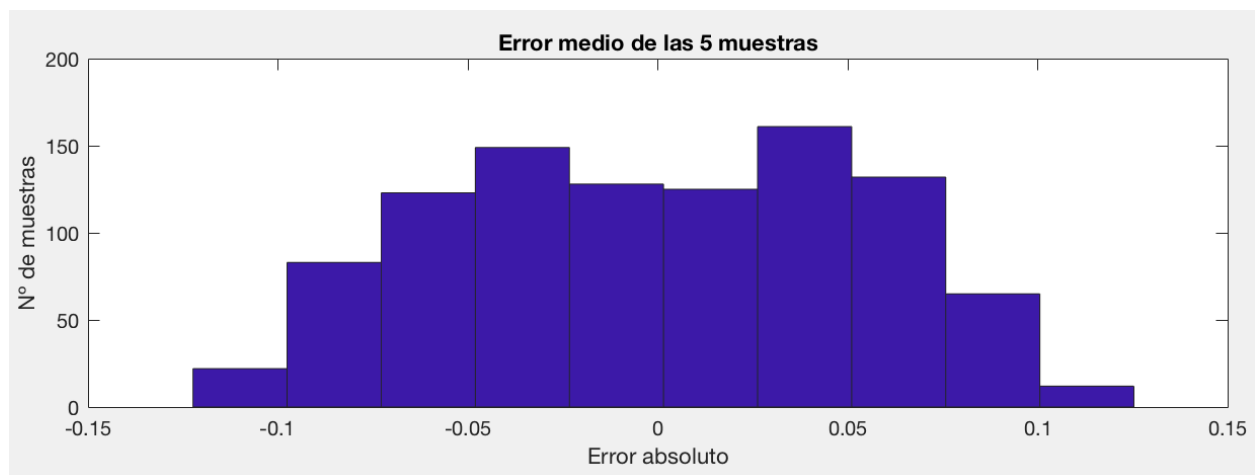


Figura 7-1. Histograma error mínimos cuadrados.

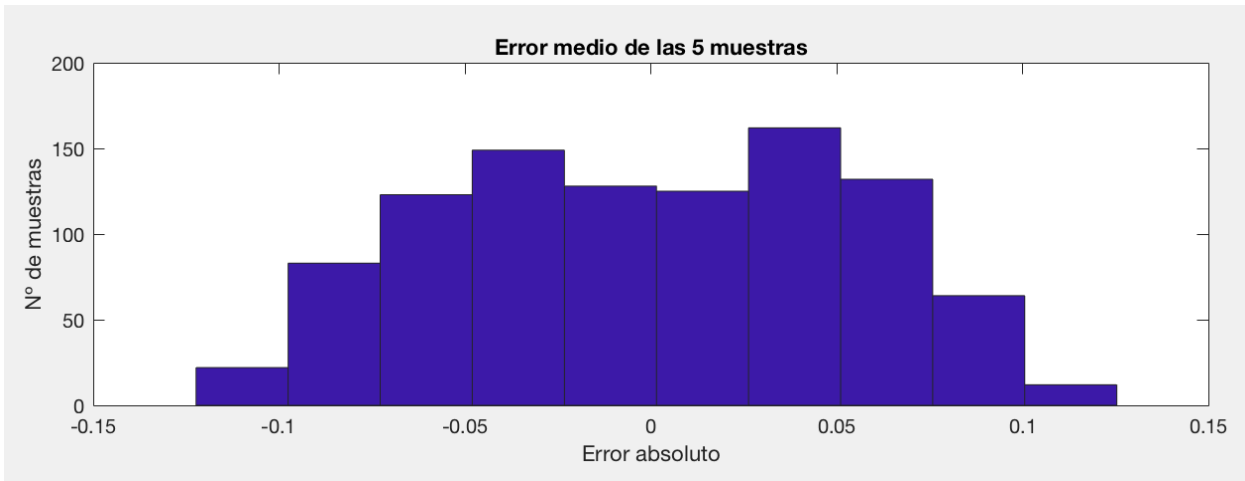


Figura 7-2. Histograma error métodos kernel.

Los errores cometidos en cada caso son:

Tabla 7-1. Comparativa LS vs SVR SMA.

Método	MSE	SEM
Mínimos cuadrados	0.005917	0.0002439
Métodos kernel	0.005912	0.0002438

7.1.2 Número de manchas solares

Para este problema, tampoco se han encontrado diferencias significativas entre los mínimos cuadrados ordinarios y las máquinas de soporte vectorial. Ambas opciones presentan una distribución normal del error y la diferencia entre el valor medio de este es pequeña. No obstante, la mejor opción de las dos es de nuevo, la obtenida con SVR en el caso 8.

Se muestran a continuación los histogramas de error medio de ambos experimentos y los errores medios cometidos.

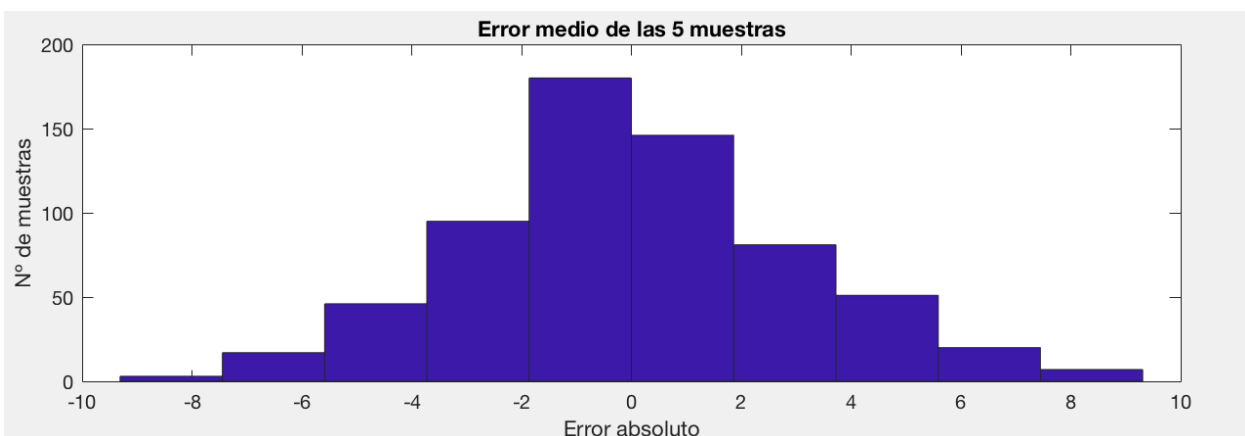


Figura 7-3. Histograma de error LS en SSN.

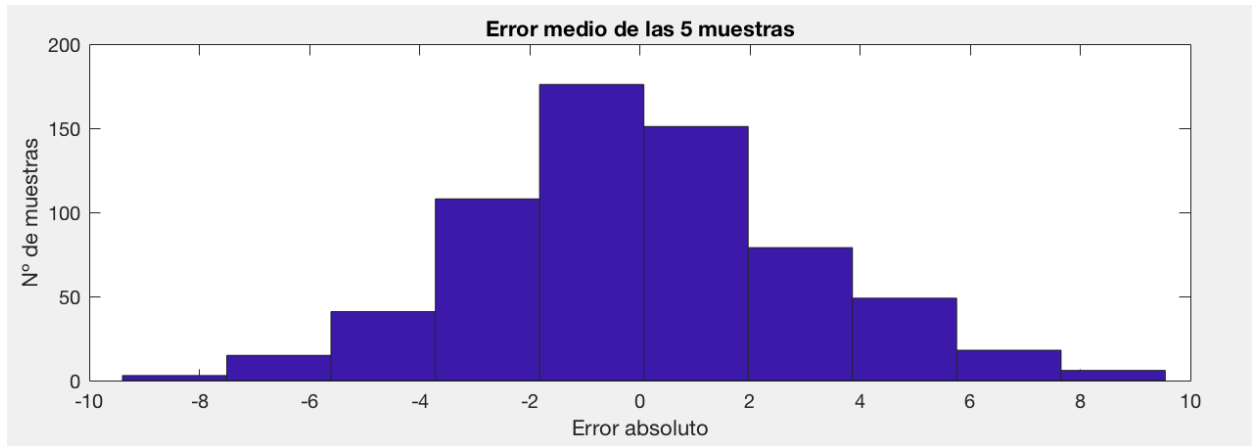


Figura 7-4. Histograma de error SVR en SSN.

Los errores cometidos en cada caso son:

Tabla 7-2. Comparativa LS vs SVR SSN.

Método	MSE	SEM
Mínimos cuadrados	46.055146	4.000608
Métodos kernel	45.451856	3.942963

7.1.3 Demanda de energía eléctrica

Al igual que en el primer caso, el mejor resultado en este problema ha sido el proporcionado por el caso 4. En esta ocasión, de nuevo, la diferencia es pequeña, pero vuelven a ser mejores las máquinas de vectores de soporte para la regresión basadas en los mínimos cuadrados que el propio método original.

Se muestran a continuación los histogramas de error medio de ambos experimentos y los errores medios cometidos por ambos métodos.

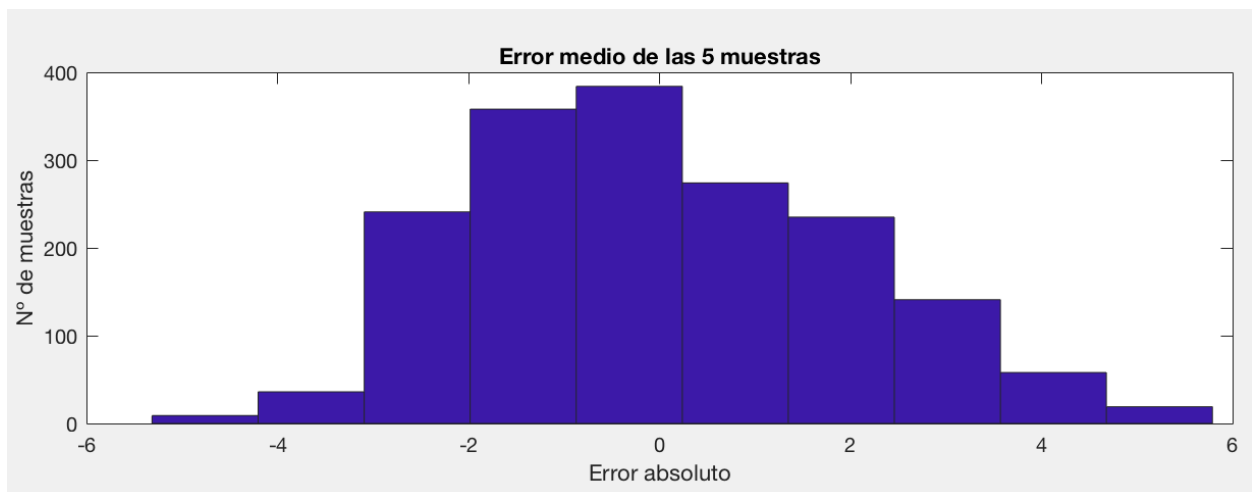


Figura 7-5. Histograma de error LS DEL.

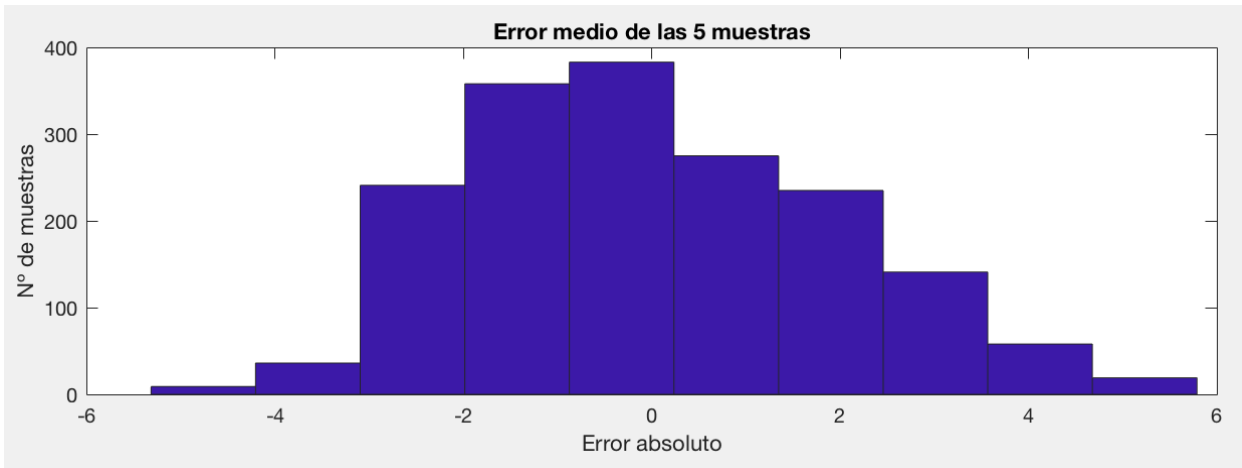


Figura 7-6. Histograma de error SVR DEL.

Los errores cometidos en cada caso son:

Tabla 7-3. Comparativa LS vs SVR SSN.

Método	MSE	SEM
Mínimos cuadrados	39.165399	1.342133
Métodos kernel	39.164996	1.342066

7.2 Impacto en los resultados al usar validación cruzada

Para las conclusiones obtenidas tras haber realizado el proceso de validación cruzada en los ensayos, no se va a individualizar por cada problema, pues el impacto es evidente. Considerando el mejor predictor como aquel que menos error comete en la predicción, es innegable, viendo los resultados del capítulo 5 que es necesario emplear validación cruzada para determinar la generalización del error que nos va a servir como indicador de bondad.

Pongamos por ejemplo, los problemas que se pretenden resolver en este trabajo.

Tenemos 9 posibles soluciones al problema de regresión. Lo que es inmediato pensar, es dividir el conjunto de datos disponibles en dos; validación y entrenamiento. Por ejemplo, el primer 80% de éstos para entrenamiento, y el 20% restante para validación. Si hubiéramos hecho esto, los resultados no serían los adecuados. Vamos a comparar las soluciones que obtendríamos en ese supuesto y las que han resultado del proceso desarrollado en este trabajo.

Solución:	SMA	SSN	DEL
Con validación cruzada	CASO 4	CASO 8	CASO 4
Sin validación cruzada	CASO 1	CASO 2	CASO 7

Figura 7-7. Influencia validación cruzada.

Vemos entonces que, aunque entre resultados y casos muy dispares, la validación cruzada puede no ser tan determinante a la hora de elegir el regresor que previsiblemente cometerá menos errores en ensayos futuros, cuando los problemas arrojan resultados muy similares es necesario aplicarla para evitar la parcialidad de los resultados con respecto a los set de entrenamiento y validación.

7.3 Valoración de los resultados obtenidos

A priori, no es posible saber la mejora que obtendremos al aplicar técnicas de aprendizaje automático a un problema con respecto a los resultados obtenidos con regresión por mínimos cuadrados. La mejora que pueden aportar los métodos kernel es en gran medida debido a la no-linealidad de los problemas.

Los resultados arrojados por los experimentos realizados, no parecen mejorar en gran medida los resultados de aplicar un método más simple como puede ser el de los mínimos cuadrados. No obstante, en todos ellos la predicción es más precisa y el error cometido disminuye.

Como ya se ha dicho, no es sencillo saber en un principio si la mejora al aplicar métodos kernel va a ser considerable o no, por lo que, una vez terminados los ensayos, debemos llegar a una solución de consenso entre la sencillez del método de los mínimos cuadrados y el gasto computacional que supone la introducción de los SVR.

En cuanto al empleo de las técnicas de validación cruzada para la generalización de los errores que nos permitirán seleccionar el mejor regresor posible, es imprescindible su empleo si queremos reducir lo máximo posible la parcialidad de la solución al conjunto de las muestras ensayadas.

7.4 Líneas de trabajo futuro

El gran problema que se ha encontrado en el desarrollo de este trabajo ha sido, sin lugar a dudas, la carga computacional de los experimentos. Para la resolución de los sistemas de ecuaciones necesarios es necesario invertir matrices de gran tamaño además de que, para el cómputo de las matrices kernel, el número de iteraciones necesarias crece con el número de muestras.

Es por ello que considero oportuno mejorar el trabajo haciendo más eficientes los algoritmos desarrollados para la resolución de los problemas propuestos.

Existen dos focos problemáticos en este sentido, el primero es el empleo de iteraciones implementadas con bucles «for», para el que existe solución empleando cálculo matricial en lugar de las iteraciones empleadas. El segundo pasa por reducir el tamaño de la matriz antes de invertirla; igualmente existen diversas técnicas al respecto que podría ser interesante implementar en un futuro trabajo.

En gran medida debido al ya mencionado problema de la carga computacional, los ensayos se han aplicado a un número limitado de posibles casos. Se podría mejorar los resultados obtenidos realizando un barrido de los hiperparámetros que definen la función *kernel* y de los que son propios de la función objetivo de las LS-SVR. La idea es optimizar los parámetros propuestos en cada caso, en lugar de forma singular, haciendo un recorrido de todos ellos en un rango de valores amplio y así, quedarnos con la mejor solución posible empleando, igualmente, técnicas de validación cruzada para la generalización de los errores cometidos y no incurrir en la dependencia de éstos con los conjuntos de datos empleados en la validación y entrenamiento de los regresores.

8 CÓDIGO DESARROLLADO.

En este anexo se encuentra el código desarrollado. Está dividido por capítulos en función de qué problema resuelve cada «script». En la última sección, se ubican todas aquellas funciones y algoritmos comunes para la resolución de los problemas.

8.1 Predicción de señal modulada en amplitud

8.1.1 svr_core_SMA.m

```
% Este script es el core ("núcleo") para resolver el problema de predicción
% de señal modulada en amplitud.
close all; clearvars; clc;

%% Carga de datos
[DATA_V, DATA_O] = preparadatos(1);
mark = kfoldCV_mark(DATA_V);

%% Formamos el set de regresores (3 regresores).

n = length(DATA_V);

%% %% Regresor 1 -> [DATA_V(i-1) DATA_V(i-2) DATA_V(i-3) DATA_V(i-4)]

l = 4; % Dimensión del regresor

x1 = zeros(n,l); % Entradas para entrenamiento
y1 = zeros(n,l); % Entradas para validación

for i = (l+1):n
    x1(i,:) = [DATA_V(i-1) DATA_V(i-2) DATA_V(i-3) DATA_V(i-4)];
    y1(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validación (5 de
% cada).

x1t_1 = x1(mark(:,1)); x1t_2 = x1(mark(:,2)); x1t_3 = x1(mark(:,3));
x1t_4 = x1(mark(:,4)); x1t_5 = x1(mark(:,5));

x1v_1 = x1(~mark(:,1)); x1v_2 = x1(~mark(:,2)); x1v_3 = x1(~mark(:,3));
x1v_4 = x1(~mark(:,4)); x1v_5 = x1(~mark(:,5));

y1t_1 = y1(mark(:,1)); y1t_2 = y1(mark(:,2)); y1t_3 = y1(mark(:,3));
y1t_4 = y1(mark(:,4)); y1t_5 = y1(mark(:,5));

y1v_1 = y1(~mark(:,1)); y1v_2 = y1(~mark(:,2)); y1v_3 = y1(~mark(:,3));
y1v_4 = y1(~mark(:,4)); y1v_5 = y1(~mark(:,5));

%% %% Regresor 2 -> []

l = 3; % Dimensión del regresor
```

```

x2 = zeros(n,1); % Entradas para entrenamiento
y2 = zeros(n,1); % Entradas para validaci n

for i = (1+3):n
    x2(i,:) = [DATA_V(i-1) DATA_V(i-2) mean([DATA_V(i-3) DATA_V(i-4)
DATA_V(i-5)])];
    y2(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validaci n (5 de
cada).

x2t_1 = x2(mark(:,1)); x2t_2 = x2(mark(:,2)); x2t_3 = x2(mark(:,3));
x2t_4 = x2(mark(:,4)); x2t_5 = x2(mark(:,5));

x2v_1 = x2(~mark(:,1)); x2v_2 = x2(~mark(:,2)); x2v_3 = x2(~mark(:,3));
x2v_4 = x2(~mark(:,4)); x2v_5 = x2(~mark(:,5));

y2t_1 = y2(mark(:,1)); y2t_2 = y2(mark(:,2)); y2t_3 = y2(mark(:,3));
y2t_4 = y2(mark(:,4)); y2t_5 = y2(mark(:,5));

y2v_1 = y2(~mark(:,1)); y2v_2 = y2(~mark(:,2)); y2v_3 = y2(~mark(:,3));
y2v_4 = y2(~mark(:,4)); y2v_5 = y2(~mark(:,5));

%% %% _____ Regresor 3 -> []

l = 4; % Dimensi n del regresor

x3 = zeros(n,1); % Entradas para entrenamiento
y3 = zeros(n,1); % Entradas para validaci n

for i = (1+1):n
    x3(i,:) = [DATA_V(i-1) sqrt(DATA_V(i-2)^2+DATA_V(i-3)^2) 0.4*DATA_V(i-3)
0.6*(DATA_V(i-4)+DATA_V(i-3)+DATA_V(i-2))/3];
    y3(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validaci n (5 de
cada).

x3t_1 = x3(mark(:,1)); x3t_2 = x3(mark(:,2)); x3t_3 = x3(mark(:,3));
x3t_4 = x3(mark(:,4)); x3t_5 = x3(mark(:,5));

x3v_1 = x3(~mark(:,1)); x3v_2 = x3(~mark(:,2)); x3v_3 = x3(~mark(:,3));
x3v_4 = x3(~mark(:,4)); x3v_5 = x3(~mark(:,5));

y3t_1 = y3(mark(:,1)); y3t_2 = y3(mark(:,2)); y3t_3 = y3(mark(:,3));
y3t_4 = y3(mark(:,4)); y3t_5 = y3(mark(:,5));

y3v_1 = y3(~mark(:,1)); y3v_2 = y3(~mark(:,2)); y3v_3 = y3(~mark(:,3));
y3v_4 = y3(~mark(:,4)); y3v_5 = y3(~mark(:,5));

%% Formamos 3 conjuntos de hiperpar metros.

%Se definen tantos como el m ximo necesario en funci n del tipo de funci n
%kernel y luego se emplear n los necesarios. Adem s se define el tipo de

```

```

%función kernel que se va a emplear

typ = ["linear", "radial_basis", "sigmoidal"]; % Tipo de función kernel

tau = [1 0.95 0.9]; % Independientes de la función kernel
gamma = [1 550 70]; % Independiente de la función kernel

p1 = [34 300 3]; % Propios de la función kernel
p2 = [0.5 50 15]; % Propios de la función kernel

%% SVR. Aprendizaje y precisión de los distintos regresores con los
distintos set de entrenamiento/validación.
[ y11_1, y12_1, y13_1, y14_1, y15_1, y11_2, y12_2, y13_2, y14_2, y15_2,
y11_3, y12_3, y13_3, y14_3, y15_3, y21_1, y22_1, y23_1, y24_1, y25_1, y21_2,
y22_2, y23_2, y24_2, y25_2, y21_3, y22_3, y23_3, y24_3, y25_3, y31_1, y32_1,
y33_1, y34_1, y35_1, y31_2, y32_2, y33_2, y34_2, y35_2, y31_3, y32_3, y33_3,
y34_3, y35_3] = kernels( tau, gamma, typ, p1, p2, x2t_1, y2t_1, x2t_2, y2t_2,
x2t_3, y2t_3, x2t_4, y2t_4, x2t_5, y2t_5, x3t_1, y3t_1, x3t_2, y3t_2, x3t_3,
y3t_3, x3t_4, y3t_4, x3t_5, y3t_5, x1t_1, y1t_1, x1t_2, y1t_2, x1t_3, y1t_3,
x1t_4, y1t_4, x1t_5, y1t_5, x1v_1, x1v_2, x1v_3, x1v_4, x1v_5, x2v_1, x2v_2,
x2v_3, x2v_4, x2v_5, x3v_1, x3v_2, x3v_3, x3v_4, x3v_5);

%% Cálculo de errores.

% Errores absolutos por cada partición kfoldCV con signo
e11_1a = y1v_1-y11_1;    e21_1a = y2v_1-y21_1;    e31_1a = y3v_1-y31_1;
e12_1a = y1v_2-y12_1;    e22_1a = y2v_2-y22_1;    e32_1a = y3v_2-y32_1;
e13_1a = y1v_3-y13_1;    e23_1a = y2v_3-y23_1;    e33_1a = y3v_3-y33_1;
e14_1a = y1v_4-y14_1;    e24_1a = y2v_4-y24_1;    e34_1a = y3v_4-y34_1;
e15_1a = y1v_5-y15_1;    e25_1a = y2v_5-y25_1;    e35_1a = y3v_5-y35_1;

e11_2a = y1v_1-y11_2;    e21_2a = y2v_1-y21_2;    e31_2a = y3v_1-y31_2;
e12_2a = y1v_2-y12_2;    e22_2a = y2v_2-y22_2;    e32_2a = y3v_2-y32_2;
e13_2a = y1v_3-y13_2;    e23_2a = y2v_3-y23_2;    e33_2a = y3v_3-y33_2;
e14_2a = y1v_4-y14_2;    e24_2a = y2v_4-y24_2;    e34_2a = y3v_4-y34_2;
e15_2a = y1v_5-y15_2;    e25_2a = y2v_5-y25_2;    e35_2a = y3v_5-y35_2;

e11_3a = y1v_1-y11_3;    e21_3a = y2v_1-y21_3;    e31_3a = y3v_1-y31_3;
e12_3a = y1v_2-y12_3;    e22_3a = y2v_2-y22_3;    e32_3a = y3v_2-y32_3;
e13_3a = y1v_3-y13_3;    e23_3a = y2v_3-y23_3;    e33_3a = y3v_3-y33_3;
e14_3a = y1v_4-y14_3;    e24_3a = y2v_4-y24_3;    e34_3a = y3v_4-y34_3;
e15_3a = y1v_5-y15_3;    e25_3a = y2v_5-y25_3;    e35_3a = y3v_5-y35_3;

%% Mostramos los errores por pantalla.

calcula_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a, e11_2a, e12_2a,
e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a, e15_3a, e21_1a,
e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a, e24_2a, e25_2a,
e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a, e33_1a, e34_1a,
e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a, e32_3a, e33_3a,
e34_3a, e35_3a );

%% Pintamos histogramas de errores.

pinta_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a, e11_2a, e12_2a,
e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a, e15_3a, e21_1a,
e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a, e24_2a, e25_2a,
e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a, e33_1a, e34_1a,
e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a, e32_3a, e33_3a,
e34_3a, e35_3a );

```

```
save('SMA_var.mat')
```

8.1.2 genera_onda_modulada.m

```
%% Generar onda modulada en amplitud.

close all; clearvars;

%% PARÁMETROS DE LAS SEÑALES

Ac= 1.6; % Amplitud de la señal portadora
Am= 1.3; % Amplitud de la señal moduladora

fc= 1/0.01; % Frecuencia de la señal portadora
fm= 1/0.3; % Frecuencia de la señal moduladora (fm<fc)

m= Am/Ac; % Índice de modulación

t= 1; % Tiempo en el que queremos obtener datos
t1=linspace(0,t,t*10000/2); % Generamos el número de puntos deseado

%% GENERACIÓN DE LAS SEÑALES

y1=sin(2*pi*fm*t1); % Señal moduladora
y2=sin(2*pi*fc*t1); % Señal portadora

eq=(1+m*y1).*(Ac*y2)+0.2*(2.*rand(1,5000)-1); % Señal modulada

%% REPRESENTACIÓN DE LAS SEÑALES

figure();
%Señal modulada
plot(t1,eq,'LineWidth',1);
title('Problema sintético. Señal modulada');
xlabel('Tiempo');ylabel('Amplitud');
legend('Señal modulada');

figure();
%Señal moduladora
subplot(2,1,1)
plot(t1,y1,'LineWidth',1);
title('Problema sintético. Señal moduladora');
xlabel('Tiempo');ylabel('Amplitud');
legend('Señal moduladora');

%Señal portadora
subplot(2,1,2)
plot(t1,y2,'LineWidth',1);
title('Problema sintético. Señal portadora');
xlabel('Tiempo');ylabel('Amplitud');
legend('Señal portadora');

%% REPRESENTACIÓN DE TODAS JUNTAS

figure();
% Señal modulada
subplot(3, 1, 1);
plot(t1,eq,'LineWidth',1);
xlabel('Tiempo');ylabel('Amplitud');
```

```

legend('Señal modulada');
title('Problema sintético. Señales del problema sintético');

% Señal moduladora
subplot(3, 1, 2);
plot(t1,Am*y1,'LineWidth',1);
xlabel('Tiempo');ylabel('Amplitud');
legend('Señal moduladora');

% Señal portadora
subplot(3, 1, 3);
plot(t1,Ac*y2,'LineWidth',1);
xlabel('Tiempo');ylabel('Amplitud');
legend('Señal portadora');

%% GUARDADO DE DATOS

%Guardamos la señal generada para su posterior tratamiento.
archivo = 'senal_sintetica.mat';
save(archivo, 'eq');
clearvars;

```

8.1.3 pred_SMA.m

```

% Este programa, entrena y valida un regresor para el problema SSN. Calcula
el
% error y pinta las figuras encasarias.
clc; clearvars; close all;

%% Obtención de datos.

datos = load('senal_sintetica.mat');
datos = datos.eq;
sec = datos;

%% Definimos set de entrenamiento y validación.

p = 0.75; % Tanto por ciento que queremos que sea el set de entrenamiento.

DATA = sec; %Desechamos para quedarnos con múltiplo de cinco.
DATA_0 = datos;

n = length(DATA);

DATA_t = sec(1:floor(p*n));
DATA_v = sec(floor(p*n)+1:end);

Nt = length(DATA_t);
Nv = length(DATA_v);

xt = zeros(Nt-5,3);
yt = zeros(Nt-5,1);

xv = zeros(Nv-5,3);

%% Montamos los regresores y definimos los parámetros

%Entrenamiento
for i = 6:Nt

```

```

        xt(i,:) = [DATA_t(i-1) DATA_t(i-2) mean([DATA_t(i-3) DATA_t(i-4)
DATA_t(i-5)])];
        yt(i) = DATA_t(i);
    end
    %Validación
    for i = 6:Nv
        xv(i,:) = [DATA_v(i-1) DATA_v(i-2) mean([DATA_v(i-3) DATA_v(i-4)
DATA_v(i-5)])];
    end

    typ = "linear"; % Tipo de función kernel

    tau = 1; % Independientes de la función kernel
    gamma = 1; % Independiente de la función kernel

    p1 = 34; % Propios de la función kernel
    p2 = 0.5;

    %% Calculamos la matriz kernel y demás parámetros mediante el set de
entrenamiento.

    [ k, b, alpha ] = c_krnl( xt, yt, tau, gamma, typ, p1, p2);

    %% Ensayamos para predicción.

    yp = v_krnl( k, alpha, b, tau, xt, xv);
    yp = yp(:,1)';

    %% Calculamos el error absoluto.

    I_M=isfinite(DATA_v);
    I_LS=isfinite(yp);

    e = (DATA_v(I_M&I_LS)-yp(I_M&I_LS));
    e_a = abs(e);

    disp('Error Medio')
    e_m = mean(e_a);
    disp(e_m);

    %% Representamos

    figure(); % Predicción vs real

    plot(DATA_v, 'LineWidth', 1.5);
    hold on;
    plot(yp,'LineWidth', 1.5);

    title('Predicción SSN vs. SSN Real')
    legend('Real', 'Predicción');
    xlabel('Tiempo');ylabel('SSN')

    %%Histograma del error y distribución normal

    mu = mean(e);
    sd = std(e);
    ndfcn = @(mu,sd,x) exp(-(x-mu).^2 ./ (2*sd^2)) / (sd*sqrt(2*pi));
    % Standard Normal Distribution
    [hc,edges] = histcounts(e);
    ctrs = edges(1:length(edges)-1) + mean(diff(edges))/2; % Calculate Centres

```



```

ctrsx = linspace(min(ctr), max(ctr));           % High-Resolution Vector
sdnd = ndfcn(mu,sd,ctrsx);

figure();
bar(ctr, hc)                                  % Plot Histogram
hold on
plot(ctrsx, sdnd*max(hc)/max(sdnd), '-r', 'LineWidth',2)
% Plot Scaled Standard Normal Distribution
hold off
grid

%% Guardamos los valores para posterior tratamiento.

save('pred_SMA_no_bagging.mat');

```

8.2 Predicción del número de manchas solares

8.2.1 svr_core_SSN.m

```

% Este script es el core ("núcleo") para resolver el problema de predicción
% de número de manchas solares.
close all; clearvars; clc;

%% Carga de datos
[DATA_V, DATA_O] = preparadatos(2);
mark = kfoldCV_mark(DATA_V);

%% Formamos el set de regresores (3 regresores).

n = length(DATA_V);

%% %% ___ Regresor 1 -> [DATA_V(i-1) DATA_V(i-2) DATA_V(i-3) DATA_V(i-4)]

l = 4; % Dimensión del regresor

x1 = zeros(n,l); % Entradas para entrenamiento
y1 = zeros(n,1); % Entradas para validación

for i = (l+1):n
    x1(i,:) = [DATA_V(i-1) DATA_V(i-2) DATA_V(i-3) DATA_V(i-4)];
    y1(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validación (5 de
cada).

x1t_1 = x1(mark(:,1)); x1t_2 = x1(mark(:,2)); x1t_3 = x1(mark(:,3));
x1t_4 = x1(mark(:,4)); x1t_5 = x1(mark(:,5));
x1v_1 = x1(~mark(:,1)); x1v_2 = x1(~mark(:,2)); x1v_3 = x1(~mark(:,3));
x1v_4 = x1(~mark(:,4)); x1v_5 = x1(~mark(:,5));

y1t_1 = y1(mark(:,1)); y1t_2 = y1(mark(:,2)); y1t_3 = y1(mark(:,3));
y1t_4 = y1(mark(:,4)); y1t_5 = y1(mark(:,5));

y1v_1 = y1(~mark(:,1)); y1v_2 = y1(~mark(:,2)); y1v_3 = y1(~mark(:,3));
y1v_4 = y1(~mark(:,4)); y1v_5 = y1(~mark(:,5));

%% %% ___ Regresor 2 -> []

```

```

l = 3; % Dimensi n del regresor

x2 = zeros(n,l); % Entradas para entrenamiento
y2 = zeros(n,l); % Entradas para validaci n

for i = (l+1):n
    x2(i,:) = [DATA_V(i-1) DATA_V(i-2) DATA_O(i-1,3)];
    y2(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validaci n (5 de
cada).

x2t_1 = x2(mark(:,1)); x2t_2 = x2(mark(:,2)); x2t_3 = x2(mark(:,3));
x2t_4 = x2(mark(:,4)); x2t_5 = x2(mark(:,5));

x2v_1 = x2(~mark(:,1)); x2v_2 = x2(~mark(:,2)); x2v_3 = x2(~mark(:,3));
x2v_4 = x2(~mark(:,4)); x2v_5 = x2(~mark(:,5));

y2t_1 = y2(mark(:,1)); y2t_2 = y2(mark(:,2)); y2t_3 = y2(mark(:,3));
y2t_4 = y2(mark(:,4)); y2t_5 = y2(mark(:,5));

y2v_1 = y2(~mark(:,1)); y2v_2 = y2(~mark(:,2)); y2v_3 = y2(~mark(:,3));
y2v_4 = y2(~mark(:,4)); y2v_5 = y2(~mark(:,5));

%% %%%_____ Regresor 3 -> []

l = 4; % Dimensi n del regresor

x3 = zeros(n,l); % Entradas para entrenamiento
y3 = zeros(n,l); % Entradas para validaci n

for i = (l+1):n
    x3(i,:) = [DATA_V(i-1) sqrt(DATA_V(i-2)) DATA_O(i,1) mean([DATA_O(i-1,5)
DATA_O(i-2,5) DATA_O(i-3,5)])];
    y3(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validaci n (5 de
cada).

x3t_1 = x3(mark(:,1)); x3t_2 = x3(mark(:,2)); x3t_3 = x3(mark(:,3)); x3t_4 =
x3(mark(:,4)); x3t_5 = x3(mark(:,5));

x3v_1 = x3(~mark(:,1)); x3v_2 = x3(~mark(:,2)); x3v_3 = x3(~mark(:,3)); x3v_4
= x3(~mark(:,4)); x3v_5 = x3(~mark(:,5));

y3t_1 = y3(mark(:,1)); y3t_2 = y3(mark(:,2)); y3t_3 = y3(mark(:,3));
y3t_4 = y3(mark(:,4)); y3t_5 = y3(mark(:,5));

y3v_1 = y3(~mark(:,1)); y3v_2 = y3(~mark(:,2)); y3v_3 = y3(~mark(:,3));
y3v_4 = y3(~mark(:,4)); y3v_5 = y3(~mark(:,5));

%% Formamos 3 conjuntos de hiperpar metros.

%Se definen t ntos como el m ximo necesario en funci n del tipo de funci n

```

```
%kernel y luego se emplear·n los necesarios. Adem·s se define el tipo de
%funciÛn kernel que se va a emplear
```

```
typ = ["linear", "radial_basis", "sigmoidal"]; % Tipo de funciÛn kernel
```

```
tau = [1 0.99 0.92]; % Independientes de la funciÛn kernel
gamma = [1 470 265]; % Independiente de la funciÛn kernel
```

```
p1 = [0.5 360 78]; % Propios de la funciÛn kernel
p2 = [164 54 15]; % Propios de la funciÛn kernel
```

```
%% SVR. Aprendizaje y precicciÛn de los distintos regresores con los
distintos set de entrenamiento/validaciÛn.
```

```
[ y11_1, y12_1, y13_1, y14_1, y15_1, y11_2, y12_2, y13_2, y14_2, y15_2,
y11_3, y12_3, y13_3, y14_3, y15_3, y21_1, y22_1, y23_1, y24_1, y25_1, y21_2,
y22_2, y23_2, y24_2, y25_2, y21_3, y22_3, y23_3, y24_3, y25_3, y31_1, y32_1,
y33_1, y34_1, y35_1, y31_2, y32_2, y33_2, y34_2, y35_2, y31_3, y32_3, y33_3,
y34_3, y35_3] = kernels( tau, gamma, typ, p1, p2, x2t_1, y2t_1, x2t_2, y2t_2,
x2t_3, y2t_3, x2t_4, y2t_4, x2t_5, y2t_5, x3t_1, y3t_1, x3t_2, y3t_2, x3t_3,
y3t_3, x3t_4, y3t_4, x3t_5, y3t_5, x1t_1, y1t_1, x1t_2, y1t_2, x1t_3, y1t_3,
x1t_4, y1t_4, x1t_5, y1t_5, x1v_1, x1v_2, x1v_3, x1v_4, x1v_5, x2v_1, x2v_2,
x2v_3, x2v_4, x2v_5, x3v_1, x3v_2, x3v_3, x3v_4, x3v_5);
```

```
%% C·lculo de errores.
```

```
% Errores absolutos por cada particiÛn kfoldCV con signo
```

```
e11_1a = y1v_1-y11_1; e21_1a = y2v_1-y21_1; e31_1a = y3v_1-y31_1;
e12_1a = y1v_2-y12_1; e22_1a = y2v_2-y22_1; e32_1a = y3v_2-y32_1;
e13_1a = y1v_3-y13_1; e23_1a = y2v_3-y23_1; e33_1a = y3v_3-y33_1;
e14_1a = y1v_4-y14_1; e24_1a = y2v_4-y24_1; e34_1a = y3v_4-y34_1;
e15_1a = y1v_5-y15_1; e25_1a = y2v_5-y25_1; e35_1a = y3v_5-y35_1;
```

```
e11_2a = y1v_1-y11_2; e21_2a = y2v_1-y21_2; e31_2a = y3v_1-y31_2;
e12_2a = y1v_2-y12_2; e22_2a = y2v_2-y22_2; e32_2a = y3v_2-y32_2;
e13_2a = y1v_3-y13_2; e23_2a = y2v_3-y23_2; e33_2a = y3v_3-y33_2;
e14_2a = y1v_4-y14_2; e24_2a = y2v_4-y24_2; e34_2a = y3v_4-y34_2;
e15_2a = y1v_5-y15_2; e25_2a = y2v_5-y25_2; e35_2a = y3v_5-y35_2;
```

```
e11_3a = y1v_1-y11_3; e21_3a = y2v_1-y21_3; e31_3a = y3v_1-y31_3;
e12_3a = y1v_2-y12_3; e22_3a = y2v_2-y22_3; e32_3a = y3v_2-y32_3;
e13_3a = y1v_3-y13_3; e23_3a = y2v_3-y23_3; e33_3a = y3v_3-y33_3;
e14_3a = y1v_4-y14_3; e24_3a = y2v_4-y24_3; e34_3a = y3v_4-y34_3;
e15_3a = y1v_5-y15_3; e25_3a = y2v_5-y25_3; e35_3a = y3v_5-y35_3;
```

```
%% Mostramos los errores por pantalla.
```

```
calcula_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a, e11_2a, e12_2a,
e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a, e15_3a, e21_1a,
e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a, e24_2a, e25_2a,
e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a, e33_1a, e34_1a,
e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a, e32_3a, e33_3a,
e34_3a, e35_3a );
```

```
%% Pintamos histogramas de errores.
```

```
pinta_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a, e11_2a, e12_2a,
e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a, e15_3a, e21_1a,
e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a, e24_2a, e25_2a,
e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a, e33_1a, e34_1a,
e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a, e32_3a, e33_3a,
```

```
e34_3a, e35_3a );
```

```
save('SSN_var.mat')
```

8.2.2 SSN_pred.m

```
% Este programa, entrena y valida un regresor para el problema SSN. Calcula
%el error y pinta las figuras encesarias.
```

```
clc; clearvars; close all;
```

```
%% ObtenciÛn de datos.
```

```
datos = xlsread('Sun_Spot_Monthly.xlsx');
sec = datos(:,4);
```

```
%% Definimos set de entrenamiento y validaciÛn.
```

```
p = 0.75; % Tanto por ciento que queremos que sea el set de entrenamiento.
```

```
DATA = sec; %Desechamos para quedarnos con m'ltiplo de cinco.
DATA_O = datos;
```

```
n = length(DATA);
```

```
DATA_t = sec(1:floor(p*n));
DATA_v = sec(floor(p*n)+1:end);
```

```
Nt = length(DATA_t);
Nv = length(DATA_v);
```

```
xt = zeros(Nt-4,4);
yt = zeros(Nt-4,1);
```

```
xv = zeros(Nv-4,4);
```

```
%% Montamos los regresores y definimos los par·metros
```

```
    %Entrenamiento
```

```
for i = 5:Nt
    xt(i,:) = [DATA_t(i-1) sqrt(DATA_t(i-2)) DATA_O(i,1) mean([DATA_O(i-1,5)
DATA_O(i-2,5) DATA_O(i-3,5)])];
    yt(i) = DATA_t(i);
end
```

```
    %ValidaciÛn
```

```
for i = 5:Nv
    xv(i,:) = [DATA_v(i-1) sqrt(DATA_v(i-2)) DATA_O(i,1) mean([DATA_O(i-1,5)
DATA_O(i-2,5) DATA_O(i-3,5)])];
end
```

```
typ = "radial_basis"; % Tipo de funciÛn kernel
```

```
tau = 0.99; % Independientes de la funciÛn kernel
gamma = 470; % Independiente de la funciÛn kernel
```

```
p1 = 360; % Propios de la funciÛn kernel
p2 = 54;
```

```
%% Calculamos la matriz kernel y dem·s par·metros mediante el set de
%entrenamiento.
```

```

[ k, b, alpha ] = c_krnl( xt, yt, tau, gamma, typ, p1, p2);

%% Ensayamos para predicci n.

yp = v_krnl( k, alpha, b, tau, xt, xv);
yp = yp(:,1);

%% Calculamos el error absoluto.

I_M=isfinite(DATA_v);
I_LS=isfinite(yp);

e = (DATA_v(I_M&I_LS)-yp(I_M&I_LS));
e_a = abs(e);

disp('Error Medio')
e_m = mean(e_a);
disp(e_m);

%% Representamos

figure(); % Predicci n vs real

plot(DATA_v, 'LineWidth', 1.5);
hold on;
plot(yp, 'LineWidth', 1.5);

title('Predicci n SSN vs. SSN Real')
legend('Real', 'Predicci n');
xlabel('Tiempo');ylabel('SSN')

%Histograma del error y distribuci n normal

mu = mean(e);
sd = std(e);
ndfcn = @(mu,sd,x) exp(-(x-mu).^2 ./ (2*sd^2)) / (sd*sqrt(2*pi));
% Standard Normal Distribution
[hc,edges] = histcounts(e);
ctrs = edges(1:length(edges)-1) + mean(diff(edges))/2;
% Calculate Centres
ctrsx = linspace(min(ctrs), max(ctrs));
% High-Resolution Vector
sdnd = ndfcn(mu,sd,ctrsx);

figure();
bar(ctrs, hc)
% Plot Histogram
hold on
plot(ctrsx, sdnd*max(hc)/max(sdnd), '-r', 'LineWidth',2)
% Plot Scaled Standard Normal Distribution
hold off
grid

%% Guardamos los valores para posterior tratamiento.

save('pred_SSN.mat');

```

8.3 Predicción de demanda de energía eléctrica

8.3.1 svr_core_DEL.m

```

% Este script es el core ("núcleo") para resolver el problema de predicción
% de demanda de energía eléctrica.
close all; clearvars; clc;

%% Carga de datos
[DATA_V, DATA_O] = preparadatos(3);
mark = kfoldCV_mark(DATA_V);

%% Formamos el set de regresores (3 regresores).

n = length(DATA_V);

%% Regresor 1 -> [DATA_V(i-1) DATA_V(i-2) DATA_V(i-3) DATA_V(i-4)]

l = 4; % Dimensión del regresor

x1 = zeros(n,l); % Entradas para entrenamiento
y1 = zeros(n,1); % Entradas para validación

for i = (l+1):n
    x1(i,:) = [DATA_V(i-1) DATA_V(i-2) DATA_V(i-3) DATA_V(i-4)];
    y1(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validación (5 de
% cada).

x1t_1 = x1(mark(:,1)); x1t_2 = x1(mark(:,2)); x1t_3 = x1(mark(:,3));
x1t_4 = x1(mark(:,4)); x1t_5 = x1(mark(:,5));

x1v_1 = x1(~mark(:,1)); x1v_2 = x1(~mark(:,2)); x1v_3 = x1(~mark(:,3));
x1v_4 = x1(~mark(:,4)); x1v_5 = x1(~mark(:,5));

y1t_1 = y1(mark(:,1)); y1t_2 = y1(mark(:,2)); y1t_3 = y1(mark(:,3));
y1t_4 = y1(mark(:,4)); y1t_5 = y1(mark(:,5));

y1v_1 = y1(~mark(:,1)); y1v_2 = y1(~mark(:,2)); y1v_3 = y1(~mark(:,3));
y1v_4 = y1(~mark(:,4)); y1v_5 = y1(~mark(:,5));

%% Regresor 2 -> []

l = 5; % Dimensión del regresor

x2 = zeros(n,l); % Entradas para entrenamiento
y2 = zeros(n,1); % Entradas para validación

for i = (l-1):n
    x2(i,:) = [DATA_V(i-1) DATA_V(i-2) DATA_O(i-1,2) DATA_O(i-1,3) DATA_O(i-
1,4)];
    y2(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validación (5 de
% cada).

```

```

x2t_1 = x2(mark(:,1)); x2t_2 = x2(mark(:,2)); x2t_3 = x2(mark(:,3));
x2t_4 = x2(mark(:,4)); x2t_5 = x2(mark(:,5));

x2v_1 = x2(~mark(:,1)); x2v_2 = x2(~mark(:,2)); x2v_3 = x2(~mark(:,3));
x2v_4 = x2(~mark(:,4)); x2v_5 = x2(~mark(:,5));

y2t_1 = y2(mark(:,1)); y2t_2 = y2(mark(:,2)); y2t_3 = y2(mark(:,3));
y2t_4 = y2(mark(:,4)); y2t_5 = y2(mark(:,5));

y2v_1 = y2(~mark(:,1)); y2v_2 = y2(~mark(:,2)); y2v_3 = y2(~mark(:,3));
y2v_4 = y2(~mark(:,4)); y2v_5 = y2(~mark(:,5));

%% %%% ____ Regresor 3 -> []

l = 5; % Dimensi n del regresor

x3 = zeros(n,l); % Entradas para entrenamiento
y3 = zeros(n,l); % Entradas para validaci n

for i = 1:n
    x3(i,:) = [DATA_V(i-1) 0.4*DATA_V(i-2)+0.6*(DATA_V(i-4)+DATA_V(i-
3)+DATA_V(i-2))/3 DATA_O(i-1,2) DATA_O(i-1,3) DATA_O(i-1,4)];
    y3(i) = DATA_V(i);
end

% Diferenciamos entre los distintos set de entrenamiento y validaci n (5 de
cada).

x3t_1 = x3(mark(:,1)); x3t_2 = x3(mark(:,2)); x3t_3 = x3(mark(:,3));
x3t_4 = x3(mark(:,4)); x3t_5 = x3(mark(:,5));

x3v_1 = x3(~mark(:,1)); x3v_2 = x3(~mark(:,2)); x3v_3 = x3(~mark(:,3));
x3v_4 = x3(~mark(:,4)); x3v_5 = x3(~mark(:,5));

y3t_1 = y3(mark(:,1)); y3t_2 = y3(mark(:,2)); y3t_3 = y3(mark(:,3));
y3t_4 = y3(mark(:,4)); y3t_5 = y3(mark(:,5));

y3v_1 = y3(~mark(:,1)); y3v_2 = y3(~mark(:,2)); y3v_3 = y3(~mark(:,3));
y3v_4 = y3(~mark(:,4)); y3v_5 = y3(~mark(:,5));

%% Formamos 3 conjuntos de hiperpar metros.

%Se definen t ntos como el m ximo necesario en funci n del tipo de funci n
%kernel y luego se emplear n los necesarios. Adem s se define el tipo de
%funci n kernel que se va a emplear

typ = ["linear", "radial_basis", "sigmoidal"]; % Tipo de funci n kernel

tau = [1 0.97 0.995]; % Independientes de la funci n kernel
gamma = [1 175 38]; % Independiente de la funci n kernel

p1 = [3 615 34]; % Propios de la funci n kernel
p2 = [28 167 615]; % Propios de la funci n kernel

%% SVR. Aprendizaje y precisi n de los distintos regresores con los
%distintos set de entrenamiento/validaci n.
[y11_1, y12_1, y13_1, y14_1, y15_1, y11_2, y12_2, y13_2, y14_2, y15_2,

```

```

y11_3, y12_3, y13_3, y14_3, y15_3, y21_1, y22_1, y23_1, y24_1, y25_1, y21_2,
y22_2, y23_2, y24_2, y25_2, y21_3, y22_3, y23_3, y24_3, y25_3, y31_1, y32_1,
y33_1, y34_1, y35_1, y31_2, y32_2, y33_2, y34_2, y35_2, y31_3, y32_3, y33_3,
y34_3, y35_3] = kernels( tau, gamma, typ, p1, p2, x2t_1, y2t_1, x2t_2, y2t_2,
x2t_3, y2t_3, x2t_4, y2t_4, x2t_5, y2t_5, x3t_1, y3t_1, x3t_2, y3t_2, x3t_3,
y3t_3, x3t_4, y3t_4, x3t_5, y3t_5, x1t_1, y1t_1, x1t_2, y1t_2, x1t_3, y1t_3,
x1t_4, y1t_4, x1t_5, y1t_5, x1v_1, x1v_2, x1v_3, x1v_4, x1v_5, x2v_1, x2v_2,
x2v_3, x2v_4, x2v_5, x3v_1, x3v_2, x3v_3, x3v_4, x3v_5);

```

```
%% Cálculo de errores.
```

```
% Errores absolutos por cada particiÙn kfoldCV con signo
```

```

e11_1a = y1v_1-y11_1;    e21_1a = y2v_1-y21_1;    e31_1a = y3v_1-y31_1;
e12_1a = y1v_2-y12_1;    e22_1a = y2v_2-y22_1;    e32_1a = y3v_2-y32_1;
e13_1a = y1v_3-y13_1;    e23_1a = y2v_3-y23_1;    e33_1a = y3v_3-y33_1;
e14_1a = y1v_4-y14_1;    e24_1a = y2v_4-y24_1;    e34_1a = y3v_4-y34_1;
e15_1a = y1v_5-y15_1;    e25_1a = y2v_5-y25_1;    e35_1a = y3v_5-y35_1;

```

```

e11_2a = y1v_1-y11_2;    e21_2a = y2v_1-y21_2;    e31_2a = y3v_1-y31_2;
e12_2a = y1v_2-y12_2;    e22_2a = y2v_2-y22_2;    e32_2a = y3v_2-y32_2;
e13_2a = y1v_3-y13_2;    e23_2a = y2v_3-y23_2;    e33_2a = y3v_3-y33_2;
e14_2a = y1v_4-y14_2;    e24_2a = y2v_4-y24_2;    e34_2a = y3v_4-y34_2;
e15_2a = y1v_5-y15_2;    e25_2a = y2v_5-y25_2;    e35_2a = y3v_5-y35_2;

```

```

e11_3a = y1v_1-y11_3;    e21_3a = y2v_1-y21_3;    e31_3a = y3v_1-y31_3;
e12_3a = y1v_2-y12_3;    e22_3a = y2v_2-y22_3;    e32_3a = y3v_2-y32_3;
e13_3a = y1v_3-y13_3;    e23_3a = y2v_3-y23_3;    e33_3a = y3v_3-y33_3;
e14_3a = y1v_4-y14_3;    e24_3a = y2v_4-y24_3;    e34_3a = y3v_4-y34_3;
e15_3a = y1v_5-y15_3;    e25_3a = y2v_5-y25_3;    e35_3a = y3v_5-y35_3;

```

```
%% Mostramos los errores por pantalla.
```

```

calcula_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a, e11_2a, e12_2a,
e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a, e15_3a, e21_1a,
e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a, e24_2a, e25_2a,
e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a, e33_1a, e34_1a,
e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a, e32_3a, e33_3a,
e34_3a, e35_3a );

```

```
%% Pintamos histogramas de errores.
```

```

pinta_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a, e11_2a, e12_2a,
e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a, e15_3a, e21_1a,
e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a, e24_2a, e25_2a,
e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a, e33_1a, e34_1a,
e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a, e32_3a, e33_3a,
e34_3a, e35_3a );

```

```
save( 'DEL_var.mat' )
```

8.3.2 pred_DEL.m

```
% Este programa, entrena y valida un regresor para el problema DEL. Calcula
%el error y pinta las figuras encesarias.
```

```
clc; clearvars; close all;
```

```
%% ObtenciÙn de datos.
```

```

datos = xlsread( 'esios_31_12_2013_a_15_08_2018_por_horas.xlsx' );
sec = datos(:,5);

```



```

%% Definimos set de entrenamiento y validaci n.

p = 0.75; % Tanto por ciento que queremos que sea el set de entrenamiento.

DATA = sec; %Desechamos para quedarnos con m ltiplo de cinco.
DATA_O = datos;

n = length(DATA);

DATA_t = sec(1:floor(p*n));
DATA_v = sec(floor(p*n)+1:end);

Nt = length(DATA_t);
Nv = length(DATA_v);

xt = zeros(Nt-4,5);
yt = zeros(Nt-4,1);

xv = zeros(Nv-4,5);

%% Montamos los regresores y definimos los par metros

    %Entrenamiento
for i = 5:Nt
    xt(i,:) = [DATA_t(i-1) DATA_t(i-2) DATA_O(i-1,2) DATA_O(i-1,3) DATA_O(i-
1,4)];
    yt(i) = DATA_t(i);
end
    %Validaci n
for i = 5:Nv
    xv(i,:) = [DATA_v(i-1) DATA_v(i-2) DATA_O(i-1,2) DATA_O(i-1,3) DATA_O(i-
1,4)];
end

typ = "linear"; % Tipo de funci n kernel

tau = 1; % Independientes de la funci n kernel
gamma = 1; % Independiente de la funci n kernel

p1 = 3; % Propios de la funci n kernel
p2 = 28;

%% Calculamos la matriz kernel y dem s par metros mediante el set de
entrenamiento.

[ k, b, alpha ] = c_krnl( xt, yt, tau, gamma, typ, p1, p2);

%% Ensayamos para predicci n.

yp = v_krnl( k, alpha, b, tau, xt, xv);
yp = yp(:,1);

%% Calculamos el error absoluto.

I_M=isfinite(DATA_v);
I_LS=isfinite(yp);

e = (DATA_v(I_M&I_LS)-yp(I_M&I_LS));
e_a = abs(e);

```

```

disp('Error Medio')
e_m = mean(e_a);
disp(e_m);

%% Representamos

figure(); % Predicci n vs real

plot(DATA_v, 'LineWidth', 1.5);
hold on;
plot(yp, 'LineWidth', 1.5);

title('Predicci n SSN vs. SSN Real')
legend('Real', 'Predicci n');
xlabel('Tiempo'); ylabel('SSN')

%%Histograma del error y distribuci n normal

mu = mean(e);
sd = std(e);
ndfcn = @(mu,sd,x) exp(-(x-mu).^2 ./ (2*sd^2)) / (sd*sqrt(2*pi));
% Standard Normal Distribution
[hc,edges] = histcounts(e);
ctrs = edges(1:length(edges)-1) + mean(diff(edges))/2;
% Calculate Centres
ctrsx = linspace(min(ctrs), max(ctrs));
% High-Resolution Vector
sdnd = ndfcn(mu,sd,ctrsx);

figure();
bar(ctrs, hc)
% Plot Histogram
hold on
plot(ctrsx, sdnd*max(hc)/max(sdnd), '-r', 'LineWidth',2)
% Plot Scaled Standard Normal Distribution
hold off
grid

%% Guardamos los valores para posterior tratamiento.

save('pred_DEL.mat');

```

8.4 Funciones comunes

8.4.1 preparadatos.m

```

function [ DATA_V, DATA_O ] = preparadatos( c )
% Esta funci n prepara los datos obtenidos de los archivos correspondientes.
% El argumento de entrada es un n mero del 1 al 3 en el que:
%     1 -> Problema se al modulada.
%     2 -> Problema manchas solares.
%     3 -> Problema demanda energ a el ctrica.
% Devuelve las matrices DATA_V y DATA_O, que son los datos de las secuencias
% temporales y las amtrices originales respectivamente. Todos ellos
% debidamente tratados en el sentido de normalizaci n, etc.

switch c
    case 1 % Problema se al modulada

```

```

datos = load('senal_sintetica.mat');
datos = datos.eq;
sec = datos;

% Los datos de la seÑal modulado contienen únicamente la secuencia
% temporal.

%% Normalizamos los valores.
a = -1; b = 1;
% Esto nos servir para escalar los valores entre 0 y 1
x_min = min(sec); % Mìnimo valor de la secuencia temporal
x_max = max(sec); % Maximo valor de la secuencia temporal

sec = a+(sec-x_min).*(b-a)/(x_max-x_min);

%% Definimos los argumentos de salida de la funciÙn.
n = floor(length(sec)/5)*5; %Redondeamos a la baja al 5 m*s cercano
%para CV m*s tarde.
DATA_V = sec(end-n+1:n)'; %Desechamos para quedarnos con mltiplo de
%cinco.
DATA_O = datos(end-n+1:n)';

case 2 % Problema manchas solares
datos = xlsread('Sun_Spot_Monthly.xlsx');
sec = datos(:,4);

% El archivo tiene siete columnas distribuidas de la siguiente forma
% Columna 1: aÑo
% Columna 2: mes
% Columna 3: Fecha expresada como fracciÙn del aÑo en curso
% Columna 4: Media Mensual Òndice SSN
% Columna 5: DesviaciÙn estandard del valor SSN.
% Columna 6: Observaciones usadas para obtenciÙn del valor mensual
%SSN.
% Columna 7: Un '1' indica que el valor es definitivo, un '0' que es
%provisional.

a = 0; b = 100;
% Esto nos servir para escalar los valores entre 0 y 1
x_min = min(sec); % Mìnimo valor de la secuencia temporal
x_max = max(sec); % Maximo valor de la secuencia temporal

sec = a+(sec-x_min).*(b-a)/(x_max-x_min);

%% Definimos los argumentos de salida de la funciÙn.
n = floor(length(sec)/5)*5; %Redondeamos a la baja al 5 m*s cercano
para CV m*s tarde.
DATA_V = sec(end-n+1:n); %Desechamos para quedarnos con mltiplo de
cinco.
DATA_O = datos(end-n+1:n,:);

case 3 % Problema demanda energÌa elÈctrica
datos = xlsread('esios_31_12_2013_a_15_08_2018_por_horas.xlsx');
sec = datos(:,5);

% El archivo contiene cinco columnas distribuidas de la siguiente
%forma:
% Columna 1: aÑo
% Columna 2: mes
% Columna 3: dÌa
% Columna 4: hora
% Columna 5: demanda

```

```

    a = 0; b = 100;
% Esto nos servir para escalar los valores entre 0 y 1
    x_min = min(sec); % Mìnimo valor de la secuencia temporal
    x_max = max(sec); % Maximo valor de la secuencia temporal

    sec = a+(sec-x_min).*(b-a)/(x_max-x_min);

    %% Definimos los argumentos de salida de la funciùn.
    n = floor(length(sec)/5)*5; %Redondeamos a la baja al 5 m*s cercano
para CV m*s tarde.
    DATA_V = sec(end-n+2:n); %Desechamos para quedarnos con m'ltiplo de
cinco.
    DATA_O = datos(end-n+2:n,:);

end
end

```

8.4.2 kfoldCV_mark.m

```

function [ mark ] = kfoldCV_mark( DATA )
%Esta funciùn recoge los datos tratados y le aõade 5 columnas dividiendo
%toda la secuencia temporal en cinco sets de aprendizaje y 5 de validaciùn.
%En dichas columnas un 1 representa que esa muestra es para entrenamiento y
%un 0 que es para validaciùn

n = length(DATA);
m = n/5;
mark = ones(n,5);

%% Primer ensayo
mark(1:m,1) = 0;
%% Segundo ensayo
mark(m+1:2*m,2) = 0;
%% Tercer ensayo
mark(2*m+1:3*m,3) = 0;
%% Cuarto ensayo
mark(3*m+1:4*m,4) = 0;
%% Quinto ensayo
mark(4*m+1:end,5) = 0;

mark = logical(mark);
end

```

8.4.3 kernels.m

```

function [ y11_1, y12_1, y13_1, y14_1, y15_1, y11_2, y12_2, y13_2, y14_2,
y15_2, y11_3, y12_3, y13_3, y14_3, y15_3, y21_1, y22_1, y23_1, y24_1, y25_1,
y21_2, y22_2, y23_2, y24_2, y25_2, y21_3, y22_3, y23_3, y24_3, y25_3, y31_1,
y32_1, y33_1, y34_1, y35_1, y31_2, y32_2, y33_2, y34_2, y35_2, y31_3, y32_3,
y33_3, y34_3, y35_3 ] = kernels( tau, gamma, typ, p1, p2, x2t_1, y2t_1, x2t_2,
y2t_2, x2t_3, y2t_3, x2t_4, y2t_4, x2t_5, y2t_5, x3t_1, y3t_1, x3t_2, y3t_2,
x3t_3, y3t_3, x3t_4, y3t_4, x3t_5, y3t_5, x1t_1, y1t_1, x1t_2, y1t_2, x1t_3,
y1t_3, x1t_4, y1t_4, x1t_5, y1t_5, x1v_1, x1v_2, x1v_3, x1v_4, x1v_5, x2v_1,
x2v_2, x2v_3, x2v_4, x2v_5, x3v_1, x3v_2, x3v_3, x3v_4, x3v_5)

%Esta funciùn entrena los regresores los set de entrenamiento para cada
%iteraciùn de la validaciùn cruzada y obtiene las salidas para validaciùn
%en el correspondiente set.

%% SVR. Aprendizaje de los distintos regresores con los distintos set de

```

entrenamiento/validaci n.

```

    %_Primer regresor con 5 particiones para kfoldCV para el primer
    %conjunto de par metros
    [ k11_1, b11_1, alpha11_1 ] = c_krnl( x1t_1, y1t_1, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k12_1, b12_1, alpha12_1 ] = c_krnl( x1t_2, y1t_2, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k13_1, b13_1, alpha13_1 ] = c_krnl( x1t_3, y1t_3, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k14_1, b14_1, alpha14_1 ] = c_krnl( x1t_4, y1t_4, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k15_1, b15_1, alpha15_1 ] = c_krnl( x1t_5, y1t_5, tau(1), gamma(1),
    typ(1), p1(1), p2(1));

```

```

    %_Segundo regresor con 5 particiones para kfoldCV para el primer
    %conjunto de par metros
    [ k21_1, b21_1, alpha21_1 ] = c_krnl( x2t_1, y2t_1, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k22_1, b22_1, alpha22_1 ] = c_krnl( x2t_2, y2t_2, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k23_1, b23_1, alpha23_1 ] = c_krnl( x2t_3, y2t_3, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k24_1, b24_1, alpha24_1 ] = c_krnl( x2t_4, y2t_4, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k25_1, b25_1, alpha25_1 ] = c_krnl( x2t_5, y2t_5, tau(1), gamma(1),
    typ(1), p1(1), p2(1));

```

```

    %_Tercer regresor con 5 particiones para kfoldCV para el primer
    %conjunto de par metros
    [ k31_1, b31_1, alpha31_1 ] = c_krnl( x3t_1, y3t_1, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k32_1, b32_1, alpha32_1 ] = c_krnl( x3t_2, y3t_2, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k33_1, b33_1, alpha33_1 ] = c_krnl( x3t_3, y3t_3, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k34_1, b34_1, alpha34_1 ] = c_krnl( x3t_4, y3t_4, tau(1), gamma(1),
    typ(1), p1(1), p2(1));
    [ k35_1, b35_1, alpha35_1 ] = c_krnl( x3t_5, y3t_5, tau(1), gamma(1),
    typ(1), p1(1), p2(1));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

    %_Primer regresor con 5 particiones para kfoldCV para el segundo
    %conjunto de par metros
    [ k11_2, b11_2, alpha11_2 ] = c_krnl( x1t_1, y1t_1, tau(2), gamma(2),
    typ(2), p1(2), p2(2));
    [ k12_2, b12_2, alpha12_2 ] = c_krnl( x1t_2, y1t_2, tau(2), gamma(2),
    typ(2), p1(2), p2(2));
    [ k13_2, b13_2, alpha13_2 ] = c_krnl( x1t_3, y1t_3, tau(2), gamma(2),
    typ(2), p1(2), p2(2));
    [ k14_2, b14_2, alpha14_2 ] = c_krnl( x1t_4, y1t_4, tau(2), gamma(2),
    typ(2), p1(2), p2(2));
    [ k15_2, b15_2, alpha15_2 ] = c_krnl( x1t_5, y1t_5, tau(2), gamma(2),
    typ(2), p1(2), p2(2));

```

```

    %_Segundo regresor con 5 particiones para kfoldCV para el segundo
    %conjunto de par metros
    [ k21_2, b21_2, alpha21_2 ] = c_krnl( x2t_1, y2t_1, tau(2), gamma(2),
    typ(2), p1(2), p2(2));
    [ k22_2, b22_2, alpha22_2 ] = c_krnl( x2t_2, y2t_2, tau(2), gamma(2),
    typ(2), p1(2), p2(2));
    [ k23_2, b23_2, alpha23_2 ] = c_krnl( x2t_3, y2t_3, tau(2), gamma(2),

```

```

typ(2), p1(2), p2(2));
[ k24_2, b24_2, alpha24_2 ] = c_krnl( x2t_4, y2t_4, tau(2), gamma(2),
typ(2), p1(2), p2(2));
[ k25_2, b25_2, alpha25_2 ] = c_krnl( x2t_5, y2t_5, tau(2), gamma(2),
typ(2), p1(2), p2(2));

    %_Tercer regresor con 5 particiones para kfoldCV para el segundo
    %conjunto de parámetros
[ k31_2, b31_2, alpha31_2 ] = c_krnl( x3t_1, y3t_1, tau(2), gamma(2),
typ(2), p1(2), p2(2));
[ k32_2, b32_2, alpha32_2 ] = c_krnl( x3t_2, y3t_2, tau(2), gamma(2),
typ(2), p1(2), p2(2));
[ k33_2, b33_2, alpha33_2 ] = c_krnl( x3t_3, y3t_3, tau(2), gamma(2),
typ(2), p1(2), p2(2));
[ k34_2, b34_2, alpha34_2 ] = c_krnl( x3t_4, y3t_4, tau(2), gamma(2),
typ(2), p1(2), p2(2));
[ k35_2, b35_2, alpha35_2 ] = c_krnl( x3t_5, y3t_5, tau(2), gamma(2),
typ(2), p1(2), p2(2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %_Primer regresor con 5 particiones para kfoldCV para el tercer
    %conjunto de parámetros
[ k11_3, b11_3, alpha11_3 ] = c_krnl( x1t_1, y1t_1, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k12_3, b12_3, alpha12_3 ] = c_krnl( x1t_2, y1t_2, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k13_3, b13_3, alpha13_3 ] = c_krnl( x1t_3, y1t_3, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k14_3, b14_3, alpha14_3 ] = c_krnl( x1t_4, y1t_4, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k15_3, b15_3, alpha15_3 ] = c_krnl( x1t_5, y1t_5, tau(3), gamma(3),
typ(3), p1(3), p2(3));

    %_Segundo regresor con 5 particiones para kfoldCV para el tercer
    %conjunto de parámetros
[ k21_3, b21_3, alpha21_3 ] = c_krnl( x2t_1, y2t_1, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k22_3, b22_3, alpha22_3 ] = c_krnl( x2t_2, y2t_2, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k23_3, b23_3, alpha23_3 ] = c_krnl( x2t_3, y2t_3, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k24_3, b24_3, alpha24_3 ] = c_krnl( x2t_4, y2t_4, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k25_3, b25_3, alpha25_3 ] = c_krnl( x2t_5, y2t_5, tau(3), gamma(3),
typ(3), p1(3), p2(3));

    %_Tercer regresor con 5 particiones para kfoldCV para el tercer
    %conjunto de parámetros
[ k31_3, b31_3, alpha31_3 ] = c_krnl( x3t_1, y3t_1, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k32_3, b32_3, alpha32_3 ] = c_krnl( x3t_2, y3t_2, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k33_3, b33_3, alpha33_3 ] = c_krnl( x3t_3, y3t_3, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k34_3, b34_3, alpha34_3 ] = c_krnl( x3t_4, y3t_4, tau(3), gamma(3),
typ(3), p1(3), p2(3));
[ k35_3, b35_3, alpha35_3 ] = c_krnl( x3t_5, y3t_5, tau(3), gamma(3),
typ(3), p1(3), p2(3));

%% SVR. Validación de los distintos ensayos con distintos sets de
entrenamiento y regresores

```

```

%_Primer regresor con 5 particiones para kfoldCV para el primer
%conjunto de parámetros
[ y11_1 ] = v_krnl( k11_1, alpha11_1, b11_1, tau(1), x1t_1, x1v_1);
[ y12_1 ] = v_krnl( k12_1, alpha12_1, b12_1, tau(1), x1t_2, x1v_2);
[ y13_1 ] = v_krnl( k13_1, alpha13_1, b13_1, tau(1), x1t_3, x1v_3);
[ y14_1 ] = v_krnl( k14_1, alpha14_1, b14_1, tau(1), x1t_4, x1v_4);
[ y15_1 ] = v_krnl( k15_1, alpha15_1, b15_1, tau(1), x1t_5, x1v_5);

```

```

%_Primer regresor con 5 particiones para kfoldCV para el segundo
%conjunto de parámetros
[ y11_2 ] = v_krnl( k11_2, alpha11_2, b11_2, tau(2), x1t_1, x1v_1);
[ y12_2 ] = v_krnl( k12_2, alpha12_2, b12_2, tau(2), x1t_2, x1v_2);
[ y13_2 ] = v_krnl( k13_2, alpha13_2, b13_2, tau(2), x1t_3, x1v_3);
[ y14_2 ] = v_krnl( k14_2, alpha14_2, b14_2, tau(2), x1t_4, x1v_4);
[ y15_2 ] = v_krnl( k15_2, alpha15_2, b15_2, tau(2), x1t_5, x1v_5);

```

```

%_Primer regresor con 5 particiones para kfoldCV para el tercer
%conjunto de parámetros
[ y11_3 ] = v_krnl( k11_3, alpha11_3, b11_3, tau(3), x1t_1, x1v_1);
[ y12_3 ] = v_krnl( k12_3, alpha12_3, b12_3, tau(3), x1t_2, x1v_2);
[ y13_3 ] = v_krnl( k13_3, alpha13_3, b13_3, tau(3), x1t_3, x1v_3);
[ y14_3 ] = v_krnl( k14_3, alpha14_3, b14_3, tau(3), x1t_4, x1v_4);
[ y15_3 ] = v_krnl( k15_3, alpha15_3, b15_3, tau(3), x1t_5, x1v_5);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%_Segundo regresor con 5 particiones para kfoldCV para el primer
%conjunto de parámetros
[ y21_1 ] = v_krnl( k21_1, alpha21_1, b21_1, tau(1), x2t_1, x2v_1);
[ y22_1 ] = v_krnl( k22_1, alpha22_1, b22_1, tau(1), x2t_2, x2v_2);
[ y23_1 ] = v_krnl( k23_1, alpha23_1, b23_1, tau(1), x2t_3, x2v_3);
[ y24_1 ] = v_krnl( k24_1, alpha24_1, b24_1, tau(1), x2t_4, x2v_4);
[ y25_1 ] = v_krnl( k25_1, alpha25_1, b25_1, tau(1), x2t_5, x2v_5);

```

```

%_Segundo regresor con 5 particiones para kfoldCV para el segundo
%conjunto de parámetros
[ y21_2 ] = v_krnl( k21_2, alpha21_2, b21_2, tau(2), x2t_1, x2v_1);
[ y22_2 ] = v_krnl( k22_2, alpha22_2, b22_2, tau(2), x2t_2, x2v_2);
[ y23_2 ] = v_krnl( k23_2, alpha23_2, b23_2, tau(2), x2t_3, x2v_3);
[ y24_2 ] = v_krnl( k24_2, alpha24_2, b24_2, tau(2), x2t_4, x2v_4);
[ y25_2 ] = v_krnl( k25_2, alpha25_2, b25_2, tau(2), x2t_5, x2v_5);

```

```

%_Segundo regresor con 5 particiones para kfoldCV para el tercer
%conjunto de parámetros
[ y21_3 ] = v_krnl( k21_3, alpha21_3, b21_3, tau(3), x2t_1, x2v_1);
[ y22_3 ] = v_krnl( k22_3, alpha22_3, b22_3, tau(3), x2t_2, x2v_2);
[ y23_3 ] = v_krnl( k23_3, alpha23_3, b23_3, tau(3), x2t_3, x2v_3);
[ y24_3 ] = v_krnl( k24_3, alpha24_3, b24_3, tau(3), x2t_4, x2v_4);
[ y25_3 ] = v_krnl( k25_3, alpha25_3, b25_3, tau(3), x2t_5, x2v_5);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%_Tercer regresor con 5 particiones para kfoldCV para el primer
%conjunto de parámetros
[ y31_1 ] = v_krnl( k31_1, alpha31_1, b31_1, tau(1), x3t_1, x3v_1);
[ y32_1 ] = v_krnl( k32_1, alpha32_1, b32_1, tau(1), x3t_2, x3v_2);
[ y33_1 ] = v_krnl( k33_1, alpha33_1, b33_1, tau(1), x3t_3, x3v_3);
[ y34_1 ] = v_krnl( k34_1, alpha34_1, b34_1, tau(1), x3t_4, x3v_4);
[ y35_1 ] = v_krnl( k35_1, alpha35_1, b35_1, tau(1), x3t_5, x3v_5);

```

```

%_Tercer regresor con 5 particiones para kfoldCV para el segundo

```

```

    %conjunto de parámetros
    [ y31_2 ] = v_krnl( k31_2, alpha31_2, b31_2, tau(2), x3t_1, x3v_1);
    [ y32_2 ] = v_krnl( k32_2, alpha32_2, b32_2, tau(2), x3t_2, x3v_2);
    [ y33_2 ] = v_krnl( k33_2, alpha33_2, b33_2, tau(2), x3t_3, x3v_3);
    [ y34_2 ] = v_krnl( k34_2, alpha34_2, b34_2, tau(2), x3t_4, x3v_4);
    [ y35_2 ] = v_krnl( k35_2, alpha35_2, b35_2, tau(2), x3t_5, x3v_5);

    %_Tercer regresor con 5 particiones para kfoldCV para el tercer
    %conjunto de parámetros
    [ y31_3 ] = v_krnl( k31_3, alpha31_3, b31_3, tau(3), x3t_1, x3v_1);
    [ y32_3 ] = v_krnl( k32_3, alpha32_3, b32_3, tau(3), x3t_2, x3v_2);
    [ y33_3 ] = v_krnl( k33_3, alpha33_3, b33_3, tau(3), x3t_3, x3v_3);
    [ y34_3 ] = v_krnl( k34_3, alpha34_3, b34_3, tau(3), x3t_4, x3v_4);
    [ y35_3 ] = v_krnl( k35_3, alpha35_3, b35_3, tau(3), x3t_5, x3v_5);

```

```
end
```

8.4.4 c_krnl.m

```

function [ k, b, alpha ] = c_krnl( x, y, tau, gamma, typ, p1, p2 )
%Kernel matrix calculation

%% Defining kernel as a input function.
k = struct;
k.type = typ;
k.param1 = p1;
k.param2 = p2;

%% Obtaining Kernel Matrix
n = length(x);
K = zeros(n);

switch k.type
    case "linear"
        for i = 1:n
            for j = 1:n
                K(i,j) = x(i,:)*x(j,:)' ;
            end
        end
    case "polinomial"
        for i = 1:n
            for j = 1:n
                K(i,j) = (k.param1 + x(i,:)*x(j,:))'^k.param2;
            end
        end
    case "radial_basis"
        for i = 1:n
            for j = 1:n
                K(i,j) = exp(-norm(x(i,:)-x(j,:))^2/k.param1^2);
            end
        end
    case "sigmoidal"
        for i = 1:n
            for j = 1:n
                K(i,j) = tanh(k.param1*x(i,:)*x(j,:)' + k.param2);
            end
        end
end

k.matrix = K;

%% Solving optimitation problem (dual formulation)

```



```

s = tau*ones(n,1);
D = diag(s);
X = [0 s'; s D*K*D+eye(size(D))*1/gamma]\[0; D*y];
alpha = X(2:end);
b = X(1);

```

```
end
```

8.4.5 v_krnl.m

```

function [ y ] = v_krnl( k, alpha, b, tau, x, xv)

n = length(alpha);
y = zeros(size(xv));
v = zeros(n,1);
p = diag(tau*ones(n,1))*alpha;
for l = 1:length(xv(:,1))
    for i = 1 : length(x(:,1))
        switch k.type
            case "linear"
                v(i,1) = x(i,:)*xv(l,:)' ;
            case "polinomial"
                v(i,1) = (k.param1 + x(i,:)'*xv(l,:))^k.param2;
            case "radial_basis"
                v(i,1) = exp(-norm(x(i,:)-xv(l,:))^2/k.param1^2);
            case "sigmoidal"
                v(i,1) = tanh(k.param1*x(i,:)'*x(l,:)+k.param2);
        end
    end
    y(l,1) = b + p'*v;
end
end

```

8.4.6 calculaerrores.m

```

function [ ] = calcula_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a,
e11_2a, e12_2a, e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a,
e15_3a, e21_1a, e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a,
e24_2a, e25_2a, e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a,
e33_1a, e34_1a, e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a,
e32_3a, e33_3a, e34_3a, e35_3a )
%Esta función calcula todos los errores necesarios a partir de los errores
%en valor absoluto calculados como e = y_real-y_predicha;

n = sqrt(length(e11_1a));

%% Errores cuadráticos por cada particiÖn kfoldCV
se11_1 = (e11_1a).^2;    se21_1 = (e21_1a).^2;    se31_1 = (e31_1a).^2;
se12_1 = (e12_1a).^2;    se22_1 = (e22_1a).^2;    se32_1 = (e32_1a).^2;
se13_1 = (e13_1a).^2;    se23_1 = (e23_1a).^2;    se33_1 = (e33_1a).^2;
se14_1 = (e14_1a).^2;    se24_1 = (e24_1a).^2;    se34_1 = (e34_1a).^2;
se15_1 = (e15_1a).^2;    se25_1 = (e25_1a).^2;    se35_1 = (e35_1a).^2;

se11_2 = (e11_2a).^2;    se21_2 = (e21_2a).^2;    se31_2 = (e31_2a).^2;
se12_2 = (e12_2a).^2;    se22_2 = (e22_2a).^2;    se32_2 = (e32_2a).^2;
se13_2 = (e13_2a).^2;    se23_2 = (e23_2a).^2;    se33_2 = (e33_2a).^2;
se14_2 = (e14_2a).^2;    se24_2 = (e24_2a).^2;    se34_2 = (e34_2a).^2;
se15_2 = (e15_2a).^2;    se25_2 = (e25_2a).^2;    se35_2 = (e35_2a).^2;

se11_3 = (e11_3a).^2;    se21_3 = (e21_3a).^2;    se31_3 = (e31_3a).^2;
se12_3 = (e12_3a).^2;    se22_3 = (e22_3a).^2;    se32_3 = (e32_3a).^2;

```

```

se13_3 = (e13_3a).^2;    se23_3 = (e23_3a).^2;    se33_3 = (e33_3a).^2;
se14_3 = (e14_3a).^2;    se24_3 = (e24_3a).^2;    se34_3 = (e34_3a).^2;
se15_3 = (e15_3a).^2;    se25_3 = (e25_3a).^2;    se35_3 = (e35_3a).^2;

%% Error cuadrático medios en cada particiûn kfoldCV

% Primer set de par·metros

mse11_1 = mean(se11_1);    mse21_1 = mean(se21_1);    mse31_1 = mean(se31_1);
mse12_1 = mean(se12_1);    mse22_1 = mean(se22_1);    mse32_1 = mean(se32_1);
mse13_1 = mean(se13_1);    mse23_1 = mean(se23_1);    mse33_1 = mean(se33_1);
mse14_1 = mean(se14_1);    mse24_1 = mean(se24_1);    mse34_1 = mean(se34_1);
mse15_1 = mean(se15_1);    mse25_1 = mean(se25_1);    mse35_1 = mean(se35_1);

mse11_1_se = std(se11_1)/n;    mse21_1_se = std(se21_1)/n;    mse31_1_se =
std(se31_1)/n;
mse12_1_se = std(se12_1)/n;    mse22_1_se = std(se22_1)/n;    mse32_1_se =
std(se32_1)/n;
mse13_1_se = std(se13_1)/n;    mse23_1_se = std(se23_1)/n;    mse33_1_se =
std(se33_1)/n;
mse14_1_se = std(se14_1)/n;    mse24_1_se = std(se24_1)/n;    mse34_1_se =
std(se34_1)/n;
mse15_1_se = std(se15_1)/n;    mse25_1_se = std(se25_1)/n;    mse35_1_se =
std(se35_1)/n;

MSE1_1 = [mse11_1 mse12_1 mse13_1 mse14_1 mse15_1]; %Matriz con errores
cuadr·ticos medios del primer regresor con el primer set de parametros]
MSE2_1 = [mse21_1 mse22_1 mse23_1 mse24_1 mse25_1]; %Matriz con errores
cuadr·ticos medios del segundo regresor con el primer set de parametros]
MSE3_1 = [mse31_1 mse32_1 mse33_1 mse34_1 mse35_1]; %Matriz con errores
cuadr·ticos medios del tercer regresor con el primer set de parametros]

MSESE1_1 = [mse11_1_se mse12_1_se mse13_1_se mse14_1_se mse15_1_se]; %Matriz
con error est·ndar del mse del primer regresor con el primer set de
parametros]
MSESE2_1 = [mse21_1_se mse22_1_se mse23_1_se mse24_1_se mse25_1_se]; %Matriz
con error est·ndar del mse del segundo regresor con el primer set de
parametros]
MSESE3_1 = [mse31_1_se mse32_1_se mse33_1_se mse34_1_se mse35_1_se]; %Matriz
con error est·ndar del mse del tercer regresor con el primer set de
parametros]

% Segundo set de par·metros

mse11_2 = mean(se11_2);    mse21_2 = mean(se21_2);    mse31_2 = mean(se31_2);
mse12_2 = mean(se12_2);    mse22_2 = mean(se22_2);    mse32_2 = mean(se32_2);
mse13_2 = mean(se13_2);    mse23_2 = mean(se23_2);    mse33_2 = mean(se33_2);
mse14_2 = mean(se14_2);    mse24_2 = mean(se24_2);    mse34_2 = mean(se34_2);
mse15_2 = mean(se15_2);    mse25_2 = mean(se25_2);    mse35_2 = mean(se35_2);

mse11_2_se = std(se11_2)/n;    mse21_2_se = std(se21_2)/n;    mse31_2_se =
std(se31_2)/n;
mse12_2_se = std(se12_2)/n;    mse22_2_se = std(se22_2)/n;    mse32_2_se =
std(se32_2)/n;
mse13_2_se = std(se13_2)/n;    mse23_2_se = std(se23_2)/n;    mse33_2_se =
std(se33_2)/n;
mse14_2_se = std(se14_2)/n;    mse24_2_se = std(se24_2)/n;    mse34_2_se =
std(se34_2)/n;
mse15_2_se = std(se15_2)/n;    mse25_2_se = std(se25_2)/n;    mse35_2_se =
std(se35_2)/n;

```

```

MSE1_2 = [mse11_2 mse12_2 mse13_2 mse14_2 mse15_2]; %Matriz con errores
cuadráticos medios del primer regresor con el segundo set de parámetros]
MSE2_2 = [mse21_2 mse22_2 mse23_2 mse24_2 mse25_2]; %Matriz con errores
cuadráticos medios del segundo regresor con el segundo set de parámetros]
MSE3_2 = [mse31_2 mse32_2 mse33_2 mse34_2 mse35_2]; %Matriz con errores
cuadráticos medios del tercer regresor con el segundo set de parámetros]

MSESE1_2 = [mse11_2_se mse12_2_se mse13_2_se mse14_2_se mse15_2_se]; %Matriz
con error estándar del primer regresor con el segundo set de parámetros]
MSESE2_2 = [mse21_2_se mse22_2_se mse23_2_se mse24_2_se mse25_2_se]; %Matriz
con error estándar del segundo regresor con el segundo set de parámetros]
MSESE3_2 = [mse31_2_se mse32_2_se mse33_2_se mse34_2_se mse35_2_se]; %Matriz
con error estándar del tercer regresor con el segundo set de parámetros]

    % Tercer set de parámetros

mse11_3 = mean(se11_3);    mse21_3 = mean(se21_3);    mse31_3 = mean(se31_3);
mse12_3 = mean(se12_3);    mse22_3 = mean(se22_3);    mse32_3 = mean(se32_3);
mse13_3 = mean(se13_3);    mse23_3 = mean(se23_3);    mse33_3 = mean(se33_3);
mse14_3 = mean(se14_3);    mse24_3 = mean(se24_3);    mse34_3 = mean(se34_3);
mse15_3 = mean(se15_3);    mse25_3 = mean(se25_3);    mse35_3 = mean(se35_3);

mse11_3_se = std(se11_3)/n;    mse21_3_se = std(se21_3)/n;    mse31_3_se =
std(se31_3)/n;
mse12_3_se = std(se12_3)/n;    mse22_3_se = std(se22_3)/n;    mse32_3_se =
std(se32_3)/n;
mse13_3_se = std(se13_3)/n;    mse23_3_se = std(se23_3)/n;    mse33_3_se =
std(se33_3)/n;
mse14_3_se = std(se14_3)/n;    mse24_3_se = std(se24_3)/n;    mse34_3_se =
std(se34_3)/n;
mse15_3_se = std(se15_3)/n;    mse25_3_se = std(se25_3)/n;    mse35_3_se =
std(se35_3)/n;

MSE1_3 = [mse11_3 mse12_3 mse13_3 mse14_3 mse15_3]; %Matriz con errores
cuadrático medios del primer regresor con el tercer set de parámetros]
MSE2_3 = [mse21_3 mse22_3 mse23_3 mse24_3 mse25_3]; %Matriz con errores
cuadrático medios del segundo regresor con el tercer set de parámetros]
MSE3_3 = [mse31_3 mse32_3 mse33_3 mse34_3 mse35_3]; %Matriz con errores
cuadrático medios del tercer regresor con el tercer set de parámetros]

MSESE1_3 = [mse11_3_se mse12_3_se mse13_3_se mse14_3_se mse15_3_se]; %Matriz
con error estándar del primer regresor con el tercer set de parámetros]
MSESE2_3 = [mse21_3_se mse22_3_se mse23_3_se mse24_3_se mse25_3_se]; %Matriz
con error estándar del segundo regresor con el tercer set de parámetros]
MSESE3_3 = [mse31_3_se mse32_3_se mse33_3_se mse34_3_se mse35_3_se]; %Matriz
con error estándar del tercer regresor con el tercer set de parámetros]

%% Errores medios de cada grupo de particiones

mse11 = mean(MSE1_1); se11 = mean(MSESE1_1);
mse12 = mean(MSE1_2); se12 = mean(MSESE1_2);
mse13 = mean(MSE1_3); se13 = mean(MSESE1_3);

mse21 = mean(MSE2_1); se21 = mean(MSESE2_1);
mse22 = mean(MSE2_2); se22 = mean(MSESE2_2);
mse23 = mean(MSE2_3); se23 = mean(MSESE2_3);

mse31 = mean(MSE3_1); se31 = mean(MSESE3_1);
mse32 = mean(MSE3_2); se32 = mean(MSESE3_2);
mse33 = mean(MSE3_3); se33 = mean(MSESE3_3);

```

```

%% Matriz de errores
MSE = [mse11 mse21 mse31; mse12 mse22 mse32; mse13 mse23 mse33];
SE_MSE = [se11 se21 se31; se12 se22 se32; se13 se23 se33];

%% Mostrar resultados ordenados por pantalla

fprintf("\n\nCASO 1:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse11_1, mse11_1_se, mse12_1, mse12_1_se, mse13_1,
mse13_1_se, mse14_1, mse14_1_se, mse15_1, mse15_1_se, mse11, se11);
fprintf("\n\nCASO 2:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse21_2, mse21_2_se, mse22_2, mse22_2_se, mse23_2,
mse23_2_se, mse24_2, mse24_2_se, mse25_2, mse25_2_se, mse12, se12);
fprintf("\n\nCASO 3:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse31_3, mse31_3_se, mse32_3, mse32_3_se, mse33_3,
mse33_3_se, mse34_3, mse34_3_se, mse35_3, mse35_3_se, mse13, se13);
fprintf("\n\nCASO 4:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse11_1, mse11_1_se, mse12_1, mse12_1_se, mse13_1,
mse13_1_se, mse14_1, mse14_1_se, mse15_1, mse15_1_se, mse21, se21);
fprintf("\n\nCASO 5:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse21_2, mse21_2_se, mse22_2, mse22_2_se, mse23_2,
mse23_2_se, mse24_2, mse24_2_se, mse25_2, mse25_2_se, mse22, se22);
fprintf("\n\nCASO 6:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse31_3, mse31_3_se, mse32_3, mse32_3_se, mse33_3,
mse33_3_se, mse34_3, mse34_3_se, mse35_3, mse35_3_se, mse23, se23);
fprintf("\n\nCASO 7:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse11_1, mse11_1_se, mse12_1, mse12_1_se, mse13_1,
mse13_1_se, mse14_1, mse14_1_se, mse15_1, mse15_1_se, mse31, se31);
fprintf("\n\nCASO 8:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse21_2, mse21_2_se, mse22_2, mse22_2_se, mse23_2,
mse23_2_se, mse24_2, mse24_2_se, mse25_2, mse25_2_se, mse32, se32);
fprintf("\n\nCASO 9:\n\n  kfold-1;\n sem %.15f se %.15f\n  kfold-2;\n sem
%.15f se %.15f\n  kfold-3;\n sem %.15f se %.15f\n  kfold-4;\n sem %.15f se
%.15f\n  kfold-5;\n sem %.15f se %.15f\n  medias;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse31_3, mse31_3_se, mse32_3, mse32_3_se, mse33_3,
mse33_3_se, mse34_3, mse34_3_se, mse35_3, mse35_3_se, mse33, se33);

fprintf("Global:\n\n  caso1;\n sem %.15f se %.15f\n  caso2;\n sem %.15f se
%.15f\n  caso3;\n sem %.15f se %.15f\n  caso4;\n sem %.15f se %.15f\n
caso5;\n sem %.15f se %.15f\n  caso6;\n sem %.15f se %.15f\n  caso7;\n sem
%.15f se %.15f\n  caso8;\n sem %.15f se %.15f\n  caso9;\n sem %.15f se
%.15f\n\n\n\n\n\n\n\n", mse11, se11, mse12, se12, mse13, se13, mse21, se21,
mse22, se22, mse23, se23, mse31, se31, mse32, se32, mse33, se33);

%% Error y Error std mínimo del conjunto de las 9 combinaciones

```

```

par·metros/regresor
mse_min = min(reshape(MSE,1,9));
se_min = min(reshape(SE_MSE,1,9));

fprintf('El mínimo MSE es: %.15f\n\nEl mínimo SE es:
%.15f\n\n\n\n\n\n\n',mse_min,se_min);

end

```

8.4.7 pintaerrores.m

```

function [ ] = pinta_errores( e11_1a, e12_1a, e13_1a, e14_1a, e15_1a,
e11_2a, e12_2a, e13_2a, e14_2a, e15_2a, e11_3a, e12_3a, e13_3a, e14_3a,
e15_3a, e21_1a, e22_1a, e23_1a, e24_1a, e25_1a, e21_2a, e22_2a, e23_2a,
e24_2a, e25_2a, e21_3a, e22_3a, e23_3a, e24_3a, e25_3a, e31_1a, e32_1a,
e33_1a, e34_1a, e35_1a, e31_2a, e32_2a, e33_2a, e34_2a, e35_2a, e31_3a,
e32_3a, e33_3a, e34_3a, e35_3a)
%UNTITLED5 Summary of this function goes here
% Detailed explanation goes here

% Histograma de errores para cada partici n kfold y regresorconjunto de
par·metros

%% CASO 1: Regresor 1, conjunto de par·metros 1.
f1 = figure;

p1 = uipanel('Parent',f1,'BorderType','none');
p1.Title = 'CASO 1: Primer regresor + Primer conjunto par·metros';
p1.TitlePosition = 'centertop';
p1.FontSize = 12;
p1.FontWeight = 'bold';

subplot(3,2,1,'Parent',p1); hist(e11_1a);
title('Partici n kfold n mero 1');xlabel('Error absoluto');ylabel('N  de
muestras');
subplot(3,2,2,'Parent',p1); hist(e12_1a);
title('Partici n kfold n mero 2');xlabel('Error absoluto');ylabel('N  de
muestras');
subplot(3,2,3,'Parent',p1); hist(e13_1a);
title('Partici n kfold n mero 3');xlabel('Error absoluto');ylabel('N  de
muestras');
subplot(3,2,4,'Parent',p1); hist(e14_1a);
title('Partici n kfold n mero 4');xlabel('Error absoluto');ylabel('N  de
muestras');
subplot(3,2,5,'Parent',p1); hist(e15_1a);
title('Partici n kfold n mero 5');xlabel('Error absoluto');ylabel('N  de
muestras');

elm_1a = (e11_1a+e12_1a+e13_1a+e14_1a+e15_1a)./5;
subplot(3,2,6,'Parent',p1); hist(elm_1a);
title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('N  de muestras');

%% CASO 2: Regresor 1, conjunto de par·metros 2.
f2 = figure;

p2 = uipanel('Parent',f2,'BorderType','none');
p2.Title = 'CASO 2: Primer regresor + Segundo conjunto par·metros';
p2.TitlePosition = 'centertop';
p2.FontSize = 12;
p2.FontWeight = 'bold';

```

```

subplot(3,2,1,'Parent',p2); hist(e11_2a);
title('Partición kfold número 1');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,2,'Parent',p2); hist(e12_2a);
title('Partición kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,3,'Parent',p2); hist(e13_2a);
title('Partición kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,4,'Parent',p2); hist(e14_2a);
title('Partición kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,5,'Parent',p2); hist(e15_2a);
title('Partición kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

elm_2a = (e11_2a+e12_2a+e13_2a+e14_2a+e15_2a)./5;
subplot(3,2,6,'Parent',p2); hist(elm_2a);
title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

%% CASO 3: Regresor 1, conjunto de parámetros 3.
f3 = figure;

p3 = uipanel('Parent',f3,'BorderType','none');
p3.Title = 'CASO 3: Primer regresor + Tercer conjunto parámetros';
p3.TitlePosition = 'centertop';
p3.FontSize = 12;
p3.FontWeight = 'bold';

subplot(3,2,1,'Parent',p3); hist(e11_3a);
title('Partición kfold número 1');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,2,'Parent',p3); hist(e12_3a);
title('Partición kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,3,'Parent',p3); hist(e13_3a);
title('Partición kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,4,'Parent',p3); hist(e14_3a);
title('Partición kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,5,'Parent',p3); hist(e15_3a);
title('Partición kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

elm_3a = (e11_3a+e12_3a+e13_3a+e14_3a+e15_3a)./5;
subplot(3,2,6,'Parent',p3); hist(elm_3a);
title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

%% CASO 4: Regresor 2, conjunto de parámetros 1.
f4 = figure;

p4 = uipanel('Parent',f4,'BorderType','none');
p4.Title = 'CASO 4: Segundo regresor + Primer conjunto parámetros';
p4.TitlePosition = 'centertop';
p4.FontSize = 12;
p4.FontWeight = 'bold';

subplot(3,2,1,'Parent',p4); hist(e21_1a);
title('Partición kfold número 1');xlabel('Error absoluto');ylabel('Nº de

```

```

muestras');
    subplot(3,2,2,'Parent',p4); hist(e22_1a);
    title('Partição kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,3,'Parent',p4); hist(e23_1a);
    title('Partição kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,4,'Parent',p4); hist(e24_1a);
    title('Partição kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,5,'Parent',p4); hist(e25_1a);
    title('Partição kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

    e2m_1a = (e21_1a+e22_1a+e23_1a+e24_1a+e25_1a)./5;
    subplot(3,2,6,'Parent',p4); hist(e2m_1a);
    title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

    %% CASO 5: Regresor 2, conjunto de parámetros 2.
    f5 = figure;

    p5 = uipanel('Parent',f5,'BorderType','none');
    p5.Title = 'CASO 5: Segundo regresor + Segundo conjunto parámetros';
    p5.TitlePosition = 'centertop';
    p5.FontSize = 12;
    p5.FontWeight = 'bold';

    subplot(3,2,1,'Parent',p5); hist(e21_2a);
    title('Partição kfold número 1');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,2,'Parent',p5); hist(e22_2a);
    title('Partição kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,3,'Parent',p5); hist(e23_2a);
    title('Partição kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,4,'Parent',p5); hist(e24_2a);
    title('Partição kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,5,'Parent',p5); hist(e25_2a);
    title('Partição kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

    e2m_2a = (e21_2a+e22_2a+e23_2a+e24_2a+e25_2a)./5;
    subplot(3,2,6,'Parent',p5); hist(e2m_2a);
    title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

    %% CASO 6: Regresor 2, conjunto de parámetros 3.
    f6 = figure;

    p6 = uipanel('Parent',f6,'BorderType','none');
    p6.Title = 'CASO 6: Segundo regresor + Tercer conjunto parámetros';
    p6.TitlePosition = 'centertop';
    p6.FontSize = 12;
    p6.FontWeight = 'bold';

    subplot(3,2,1,'Parent',p6); hist(e21_3a);
    title('Partição kfold número 1');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,2,'Parent',p6); hist(e22_3a);

```

```

    title('Partición kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,3,'Parent',p6); hist(e23_3a);
    title('Partición kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,4,'Parent',p6); hist(e24_3a);
    title('Partición kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
    subplot(3,2,5,'Parent',p6); hist(e25_3a);
    title('Partición kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

    e2m_3a = (e21_3a+e22_3a+e23_3a+e24_3a+e25_3a)./5;
    subplot(3,2,6,'Parent',p6); hist(e2m_3a);
    title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

%% CASO 7: Regresor 3, conjunto de parámetros 1.
f7 = figure;

p7 = uipanel('Parent',f7,'BorderType','none');
p7.Title = 'CASO 7: Tercer regresor + Primer conjunto parámetros';
p7.TitlePosition = 'centertop';
p7.FontSize = 12;
p7.FontWeight = 'bold';

subplot(3,2,1,'Parent',p7); hist(e31_1a);
title('Partición kfold número 1');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,2,'Parent',p7); hist(e32_1a);
title('Partición kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,3,'Parent',p7); hist(e33_1a);
title('Partición kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,4,'Parent',p7); hist(e34_1a);
title('Partición kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,5,'Parent',p7); hist(e35_1a);
title('Partición kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

    e3m_1a = (e31_1a+e32_1a+e33_1a+e34_1a+e35_1a)./5;
    subplot(3,2,6,'Parent',p7); hist(e3m_1a);
    title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

%% CASO 8: Regresor 3, conjunto de parámetros 2.
f8 = figure;

p8 = uipanel('Parent',f8,'BorderType','none');
p8.Title = 'CASO 8: Tercer regresor + Segundo conjunto parámetros';
p8.TitlePosition = 'centertop';
p8.FontSize = 12;
p8.FontWeight = 'bold';

subplot(3,2,1,'Parent',p8); hist(e31_2a);
title('Partición kfold número 1');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,2,'Parent',p8); hist(e32_2a);
title('Partición kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');

```



```

subplot(3,2,3,'Parent',p8); hist(e33_2a);
title('Partición kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,4,'Parent',p8); hist(e34_2a);
title('Partición kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,5,'Parent',p8); hist(e35_2a);
title('Partición kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

e3m_2a = (e31_2a+e32_2a+e33_2a+e34_2a+e35_2a)./5;
subplot(3,2,6,'Parent',p8); hist(e3m_2a);
title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

%% CASO 9: Regresor 3, conjunto de parámetros 3.
f9 = figure;

p9 = uipanel('Parent',f9,'BorderType','none');
p9.Title = 'CASO 9: Tercer regresor + Tercer conjunto parámetros';
p9.TitlePosition = 'centertop';
p9.FontSize = 12;
p9.FontWeight = 'bold';

subplot(3,2,1,'Parent',p9); hist(e31_3a);
title('Partición kfold número 1');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,2,'Parent',p9); hist(e32_3a);
title('Partición kfold número 2');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,3,'Parent',p9); hist(e33_3a);
title('Partición kfold número 3');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,4,'Parent',p9); hist(e34_3a);
title('Partición kfold número 4');xlabel('Error absoluto');ylabel('Nº de
muestras');
subplot(3,2,5,'Parent',p9); hist(e35_3a);
title('Partición kfold número 5');xlabel('Error absoluto');ylabel('Nº de
muestras');

e3m_3a = (e31_3a+e32_3a+e33_3a+e34_3a+e35_3a)./5;
subplot(3,2,6,'Parent',p9); hist(e3m_3a);
title('Error medio de las 5 muestras');xlabel('Error
absoluto');ylabel('Nº de muestras');

end

```


REFERENCIAS

- [1] Tyler Vigen, Spurious Correlations, <http://tylervigen.com/spurious-correlations>
- [2] Giuseppe Piazzi, (s.f.). En Wikipedia, https://es.wikipedia.org/wiki/Giuseppe_Piazzi
- [3] Nieto, M. M. The Titius-Bode Law of Planetary Distances: Its History and Theory. Oxford: Pergamon Press Ltd., 1972.
- [4] Ruye Wang, Kernels and Mercer's Theorem. En <http://fourier.eng.hmc.edu/e161/lectures/gaussianprocess/node8.html>
- [5] B. Schölkopf and A.J. Smola, Support Vector Machines and Kernel Algorithms,
- [6] Teodoro Álamo y J.M. Bravo, Weighted least squares support vector machines, Tech. report, Universidad de Sevilla, 2015.
- [7] Andrew Ng, Machine Learning, Stanford University, En <https://es.coursera.org/learn/machine-learning>
- [8] Vsevolod I. Ivanov. Duality in nonlinear programming. Optimization Letters, 7(8):1643–1658, 2013
- [9] Trevor J. Hastie, Robert John Tibshirani, and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction. Springer, 2013.
- [10] Waclerly, Dennis; Mendenhall, William; Scheaffer, Richard L. (2008). Mathematical Statistics with Applications (7 ed.). Belmont, CA, USA
- [11] Daniel Rodriguez y Teodoro Álamo, Identificación mediante el método de los mínimos cuadrados, http://control-class.com/Tema_2/Slides/Tema_2_IdentificacionMinimosCuadrados.pdf

