UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Modelos de aprendizaje automático en la predicción de viento a corto plazo.

Máster Universitario en Investigación e Innovación en TIC

Autor: Cortés Alonso, María

Tutor: Suárez González, Alberto
Departamento de Ingenería Informática

FECHA: Septiembre, 2020

Agradecimientos

Agradecemos a L. Prieto-Godino y a S. Salcedo-Sanz, por cedernos los datos de viento para la realización de este estudio.

También me gustaría dar las gracias a toda la gente que me ha apoyado durante la realización del Trabajo Fin de Máster y a mi tutor, Alberto Suárez.

Índice general

Índice de figuras

Índice de tablas

Resumen

El modelo energético actual tiene problemas de sostenibilidad a largo plazo debido a la dependencia de los combustibles fósiles. Por ello, poco a poco, se han de ir cambiando las distintas fuentes de energía para llegar a un desarrollo sostenible. En este contexto, la generación de energía eólica es fundamental, ya que es una energía renovable y limpia. Esta energía se obtiene mediante aerogeneradores que convierten la energía del viento en energía eléctrica. Estos aerogeneradores funcionan en un régimen determinado de viento ya que necesitan una velocidad del viento mínima para empezar a moverse y un viento con una velocidad muy alta podría romperlos. Por ello, para la adecuada gestión de un parque eólico, es necesario realizar predicciones precisas de la intensidad del viento en el emplazamiento donde se encuentra el parque eólico.

Para realizar estas predicciones, se pueden utilizar datos de reanálisis. Un reanálisis es un modelo meteorológico que ha pasado por un proceso de asimilación de datos que provienen de satélites, estaciones de medición en tierra y otras fuentes. En concreto, en este trabajo se utilizarán datos generados por el modelo del centro europeo ERA-Interim1. En la práctica, dado que la situación geográfica de un parque eólico puede no coincidir con un punto exacto de la rejilla de predicción del modelo meteorológico, es necesario adaptar estas predicciones.

El objetivo de este Trabajo Fin de Máster es estudiar cómo varía el error de predicción de distintos modelos de aprendizaje automático. Para ello, se han aplicado distintas técnicas de aprendizaje automático. Además se ha realizado un estudio de cuáles son las variables regresoras más importantes a la hora de realizar predicciones.

Capítulo 1

Introducción

Unos de los retos que tenemos como sociedad es encontrar una fuente de energía abundante, barata y sostenible. En concreto, es deseable que la explotación de dicha fuente de energía no genere residuos contaminantes de difícil gestión (por ejemplo, los residuos radiactivos en la energía nuclear) o elevadas emisiones de dióxido de carbono (como, por ejemplo, las derivadas del uso de combustibles fósiles). En este contexto, la energía eólica es una de las opciones más atractivas ya que es renovable y limpia. Esta se obtiene mediante aerogeneradores que convierten la energía cinética del viento en potencia eléctrica: el viento incide en las aspas de un aerogenerador, las cuales, al girar, impulsan un alternador que transforma la energía rotacional en energía eléctrica. Una gran ventaja de la energía eólica es que en su explotación no se genera residuo alguno. El mayor inconveniente que presenta es la variabilidad en la potencia generada, la cual sufre grandes fluctuaciones dependiendo de las condiciones de viento. Por una parte, si la velocidad del viento es muy baja, no es posible hacer girar las aspas de los aerogeneradores y, en consecuencia, la producción de energía es nula. Por otra parte, si dicha velocidad es muy alta, puede que sea necesario bloquear el aerogenerador para evitar daños. Estas rachas de viento intenso son las que mayor perjuicio económico pueden producir debido a la posible rotura del aerogenerador. Por tanto, para la adecuada gestión de un parque eólico, es necesario disponer de predicciones precisas de la intensidad del viento en la localización del parque, especialmente para velocidades elevadas. Otra razón por la cual es importante disponer de estimaciones de la intensidad del viento en el futuro cercano es la necesidad de planificar la ofertas de generación para el mercado eléctrico. En la actualidad apenas existen mecanismos eficientes para almacenar la energía generada en un parque eólico, por lo que es necesario transferirla a la red eléctrica al precio que establezca el mercado. Dada la variabilidad de los precios de la energía eólica en el mercado eléctrico es importante disponer de predicciones fiables de la potencia generada. Esta potencia es función sobre todo de la intensidad del viento, por lo que es útil realizar con suficiente antelación buenas estimaciones de dicha cantidad.

El objetivo concreto de este Trabajo de Fin de Máster es predecir la magnitud del viento en el parque eólico de Peñaparda, Salamanca, utilizando sistemas construidos mediante la aplicac´ón de aprendizaje automático a datos de reanálisis. Un reanálisis es un modelo meteorológico que resulta de la integración de datos que provienen de satélites, estaciones de medición en tierra y otras fuentes. En concreto, en este trabajo se utilizan datos generados por el modelo del centro europeo ERA-Interim [10].

Un conjunto de datos de reanálisis es aquel que ha sido generado por un modelo numérico de predicción meteorológica. Para hacer que este modelo se ajuste a la realidad, se entrena con condiciones iniciales actualizadas y además, pasa por un proceso de asimilación de datos. Este proceso se realiza cada 6 horas.

La asimilación de datos consiste en obligar al modelo a reproducir las medidas tomadas por los diferentes instrumentos de medidas meteorológicas, como por ejemplo termómetros, pluviómetros, repartidas por todo el mundo. Estas localizaciones no se encuentran localizadas regularmente. Esta característica sí que la poseen los datos procedentes de reanálisis. Así mismo, también poseen regularidad temporal. Esto es una gran ventaja respecto a la fuente de datos proporcionada por el parque, ya que son robustos frente a incidencias que pueda sufrir el parque.

En los estudios de reanálisis se proporcionan estimaciones en una rejilla de localizaciones. La situación geográfica de un parque eólico puede no coincidir con uno de los puntos de la rejilla de predicción del modelo meteorológico, por lo que es necesario adaptar estas predicciones a la localización concreta del

parque. Se quiere que este modelo de predicción pueda ser entrenado de manera eficiente, que sea preciso y que utilice la cantidad mínima de datos. Para ello, se ha estudiado la variación del error con distintos tamaños del conjunto de entrenamiento. Los métodos utilizados han sido las redes neuronales, las máquinas de soporte vectorial, los procesos gaussianos, los bosques aleatorios, la potenciación del gradiente y la potenciación extrema del gradiente. El modelo que ha obtenido menor error tanto en validación cruzada como en test ha sido el bosque aleatorio.

Además, también se ha realizado una selección de variables. El criterio utilizado para dicha selección está basado en aproximaciones utilizadas en el campo de la meteorología. Una aproximación utilizada consiste en que la atmósfera está estratificada. Eso quiere decir que se comporta como si tuviera capas de fluido. Esto entre otras cosas indica que la componente vertical del viento es despreciable frente a las componentes horizontales. Si tenemos capas estratificadas separadas entre sí, también se puede considerar que dichas capas son independientes. Una vez realizada la selección de variables, se ha comprobado que la capa más importante para realizar la predicción es el viento a 850 hPa, aproximadamente 1500 metros de altura.

1.1. Modelos meteorológicos

Este Trabajo de Fin de Máster trata sobre la aplicación de métodos de aprendizaje automático al problema de predicción de la velocidad del viento a partir de datos de reanálisis.

Además de las técnicas de aprendizaje automático, se pueden obtener estas predicciones resolviéndolas ecuaciones que describen la evolución de los procesos físicos relevantes. Es concreto, se pueden utilizar los llamados *modelos expertos*. Estos son modelos meteorológicos basados en el planteamiento y resolución de las ecuaciones que describen la evolución temporal de la atmósfera. Dado que se trata de un sistema complejo, en este tipo de modelos se requiere el uso de numerosas variables para caracterizar el estado físico de la atmósfera [17]. Adicionalmente, es necesario realizar una parametrización de los accidentes topográficos y de los diferentes obstáculos, tales como las masas de vegetación o las ciudades.

Existen dos tipos de modelos meteorológicos: los modelos globales y los modelos de área limitada. En los globales se modelizan las distintas componentes que afectan al tiempo meteorológico. Están compuestos por sub-modelos para la atmósfera, los océanos, los casquetes polares, ciclo del carbono e incluso aerosoles. Cuanto más sofisticado sea el modelo global, más sub-modelos acoplados tendrá. Por ejemplo, un modelo puede considerar únicamente la atmósfera. Sin embargo, si a este modelo de atmósfera se le acopla un modelo de océano, es necesario tener en cuenta las interacciones: temperatura de la superficie del océano, flujos de calor, el viento como generador de corrientes, etcétera. Los modelos de área limitada son modelos que no intentan abarcar todo el globo terráqueo, sino un área más pequeña con más resolución. Pero este área no está aislada del resto del globo, así que utilizan la salida del modelo global como condición de contorno.

En cualquier caso, ambos tipos de modelo de predicción númerica suelen ser modelos muy costosos debido a la rejilla utilizada y a la cantidad de variables que involucran. Para resolver estos modelos, hay que conocer el sistema meteorológico y como se describe.

La atmósfera se puede describir como un fluido. Dado que el aire no está a la misma temperatura en todo el planeta, se trata de un sistema que no está en equilibrio termodinámico. Para describir este sistema, además la ecuación de estado para un gas ideal, se requiere conocer la dinámica de un fluido en tres dimensiones y las ecuaciones que regulan el intercambio de calor entre los distintos puntos de la atmósfera.

A continuación se describen las distintas ecuaciones que rigen la dinámica del sistema.

Las ecuaciones de Navier-Stokes (para un estudio detallado de la física de la atmósfera, consultar [1]) son un sistema de ecuaciones diferenciales en derivadas parciales que se utilizan para modelizar el movimiento de un fluido con viscosidad. Dentro de la atmósfera hay capas de aire estratificadas. La viscosidad es el rozamiento existente entre estos estratos de aire que se mueven a distinta velocidad. Las ecuaciones de Navier-Stokes en forma vectorial se pueden escribir como:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{F} - 2\Omega\times\mathbf{u} \tag{1.1}$$

donde $\mathbf{u} = (u, v, w)$ son las distintas componentes del viento, ρ es la densidad del fluido, p es la presión, F representa las distintas fuerzas externas y Ω es el vector de velocidad angular de la tierra. Estas ecuaciones

tienen varias características que la hacen interesante. El término $(\mathbf{u}\cdot\nabla)\mathbf{u}$ hace que el sistema no sea lineal y por tanto pueda presentar caos. En un sistema caótico, pequeñas diferencias en las condiciones iniciales son amplificadas de manera exponencial por la dinámica, dando lugar a trayectorias que pueden ser muy dispares. Por eso, modelos con las mismas ecuaciones diferenciales y condiciones iniciales próximas pueden llegar a tener distintas predicciones.

El término $\frac{1}{\rho}\nabla p$ refleja el hecho de que el viento se produce por diferencias de presión en la atmósfera. El término de la fuerza de Coriolis, $2\Omega\times\mathbf{u}$, el que determina la dirección del viento. Esta dirección es perpendicular al vector ∇p y el sentido depende de si se trata de un anticiclón o una borrasca. En un anticiclón del hemisferio norte, los vientos circulan en la dirección de las agujas del reloj; en una borrasca van en dirección contraria. Eso es porque la dirección de la diferencia de presión siempre apunta al centro del anticiclón o borrasca. La fuerza de Coriolis está relacionada con la rotación terrestre. Por último, el término $\mathbf{F}$ está asociado a las fuerzas externas como la gravedad o el rozamiento.

La siguiente ecuación diferencial es la obtenida debido a la conservación de la masa. En la atmósfera, ni se crea ni se destruye masa. La cantidad de fluido es constante. Esto se modeliza mediante la expresión:

$$\frac{\partial\rho}{\partial t} = -\nabla\cdot(\rho\mathbf{u}). \tag{1.2}$$

Es decir, la masa movida por el viento es igual a la masa desplazada en el espacio. Debido a que el planeta es una esfera, tienen que existir divergencias y convergencias en el sistema. Las divergencias son centros de bajas presiones o borrascas y las convergencias son anticiclones. Gracias a este sistema de convergencias y divergencias, existe una configuración aproximadamente constante de borrascas y anticiclones. De ahí, la borrasca o zona de bajas presiones de Islandia o el anticiclón de las Azores.

La atmósfera es un sistema termodinámico [17]. Por tanto, hay que considerar también los distintos flujos de energía que se producen en su seno (figura 1.1). La energía que llega del sol a la superficie terrestre se transmite mediante dos tipos de flujos de calor: el sensible y el latente. El calor sensible es el que produce una diferencia de temperatura en el fluido sin modificar su estructura molecular (es decir, sin producir un cambio estado). Un ejeplo de este tipo de transferencia es el calentamiento de una masa agua sin que llegue a hervir. El calor latente es el que es invertido en cambios de estado. Un ejemplo de calor latente es la energía que se invierte desde que el agua comienza a hervir hasta que se evapora completamente. En esa fase, la temperatura del agua es constante y la energía se asocia al cambio de estado líquido a gaseoso.

También es necesario tener en cuenta que la superficie terrestre refleja parte de la energía que llega al planeta desde el sol. El porcentaje de energía que una superficie refleja es su albedo. En el caso de la superficie terrestre, la cantidad de energía reflejada se obtiene por mediciones en satélites. El albedo es mayor en zonas cubiertas por hielo o nieve y menor en océanos y zonas cubiertas de vegetación. También hay que tener en cuenta el comportamiento de las nubes ante los flujos de calor. Las nubes evitan que el calor incida sobre la superficie terrestre, pero también impide que sea reflejado. Las nubes se representan en el modelo mediante una parametrización de submalla. Es importante que esta parametrización sea de calidad para obtener un buen diagnóstico del estado de la atmósfera. Por último hay que tener en cuenta la distribución la presencia de aerosoles y gases, como el ozono y los gases de efecto invernadero (*GEI*)ca.

Combinado estos elementos, se puede plantear un sistema de ecuaciones que describe, de manera aproximada, la dinámica atomosférica. Este sistema de ecuaciones se puede resolver mediante métodos numéricos. El más utilizado es el método de elementos finitos. En el m´etodo de elementos finitos, se realiza una discretización del espacio y del tiempo. En este espacio discreto, las derivadas se aproximan mediante diferencias. También es posible resolver estas ecuaciones por un método espectral. El método espectral realiza una transformada de Fourier y resuelve estas ecuaciones en el espacio de frecuencias y momentos.

Todas estas ecuaciones se pueden simplificar mediante métodos de escala. Por ejemplo, se puede tener en cuenta que muy lejos del suelo el rozamiento influye poco. Sin embargo, cerca del suelo lo que no afectaría sería la fuerza de Coriolis. No son las únicas simplificaciones que se pueden realizar. Otra opción para reducir coste es disminuir la resolución. Los fenómenos de escala menor que la resolución del modelo (las nubes o los flujos de calor locales) se parametrizan. Gracias a la potencia computacional actual, la solución de modelos con muy alta resolución. De hecho, los modelos de área limitada tienen una resolución parecida a algunos modelos globales actuales. En la figura 1.2 se pueden observar dos imágenes. La primera representa la mejora de la resolución de la rejilla en los modelos. Esto permite modelar mejor la orografía. Por ejemplo, en la primera imagen, la península es una gran montaña irregular. En la segunda,

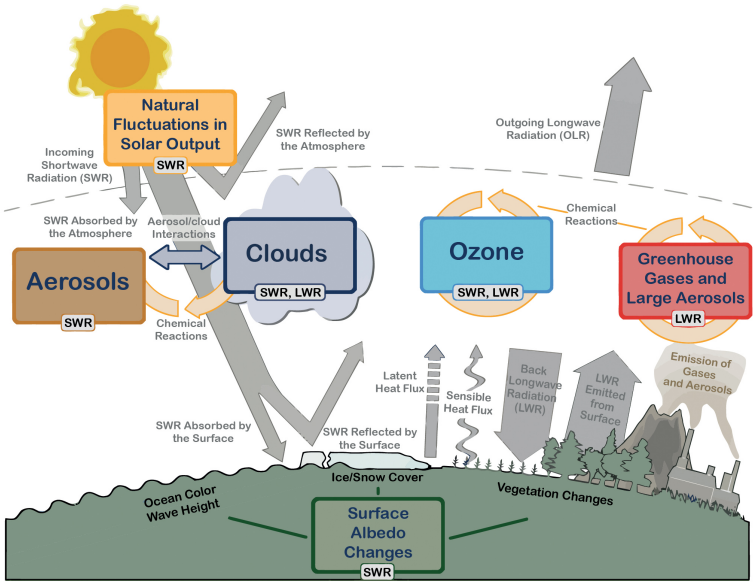
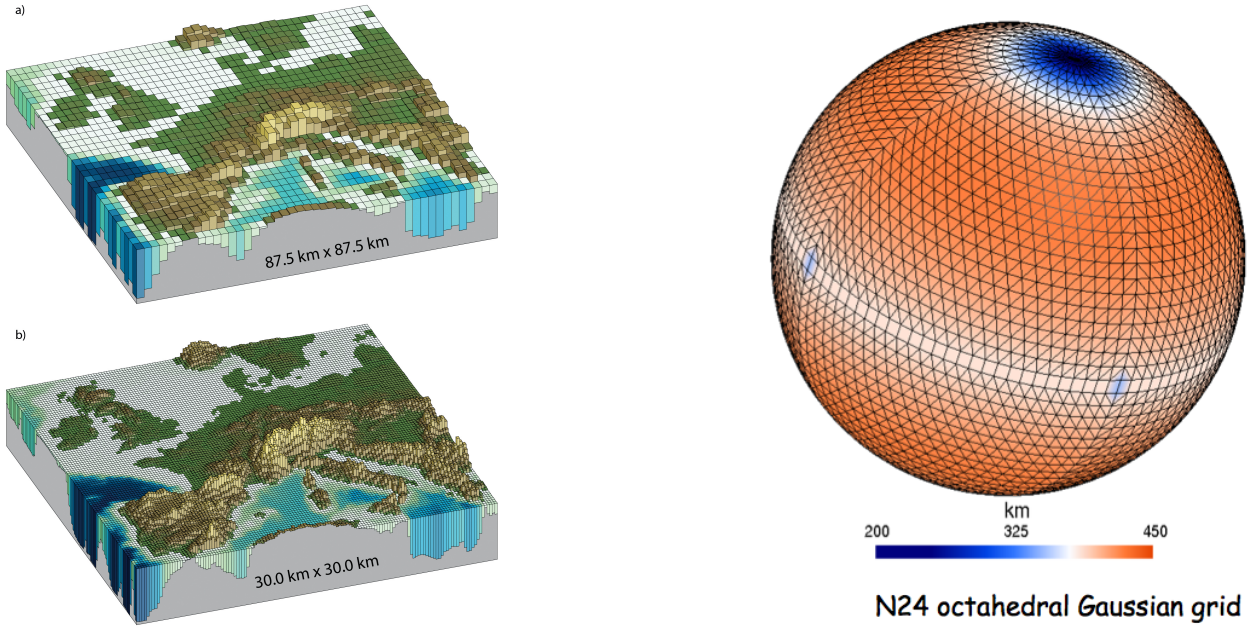


Figura 1.1: Procesos radiativos en la atmósfera [37].

que representa la rejilla horizontal utilizada por el modelo determinista del centro europeo, se distinguen los distintos sistemas montañosos del país.



(a) (b)

Figura 1.2: (a) Comparación de la representación orográfica de Europa mediante un aumento de la resolución [37]. (b) Representación de la rejilla utilizada por el modelo determinista del centro europeo.

En un párrafo anterior se ha indicado que se pueden describir de manera diferente la atmósfera libre, en zonas alejadas de la superficie terrestre, en la que el rozamiento es despreciable, y la atmósfera cerca del suelo, para la cual el rozamiento es significativo. Esta capa de la atmósfera afectada por el rozamiento recibe el nombre de capa límite [2]. Esta capa intercambia calor, masa y momento entre el suelo y la atmósfera. Está fuertemente condicionada por el rozamiento y es donde se produce la turbulencia. Esta turbulencia genera mecanismos de mezcla y transferencia de calor.

Hay múltiples factores que condicionan la capa límite. Estos son el ciclo diurno, la continentalidad, los sistemas sinópticos o a gran escala (del orden de 1000 km o más), la topografía, la rugosidad (vegetación, casas u obstáculos varios) y el uso del terreno. La frontera superior de la capa límite se suele asociar a una

inversión en altura o a cuando un perfil de viento alcanza el 90 % del valor del viento libre (i.e. la velocidad del viento en las capas sin rozamiento). Un ejemplo visible del efecto de una inversión en altura es el límite de la famosa "boina" de contaminación de Madrid. Típicamente tiene 1 kilómetro de altura, por lo que ocupa aproximadamente el 10 % de la troposfera.

En el modelo global se suelen considerar varios niveles en altura. La atmósfera está estratificada de forma que las capas más cercanas al suelo son las más densas. La presión es la fuerza por metro cuadrado que ejerce la atmósfera sobre el suelo. A más cantidad de aire, mayor presión. Por esto, existe una relación monótona entre presión y altura de la atmósfera. En condiciones ideales, esta relación se mantiene siempre. Debido a que la atmósfera no es un sistema ideal y existen masas de aire con distintas propiedades, a veces esta relación varía. Por ello, se suele seleccionar niveles de presión en lugar de niveles de altura. La resolución vertical depende del modelo seleccionado. Aunque se consideran más niveles para la discretización de las ecuaciones del modelo meteorológico, se suelen proporcionar valores finales únicamente para tres niveles: 10 metros de altura, 850 hPa o 1500 metros y 500 hPa o 5500 metros. El primero está dentro de la capa límite. El segundo se sitúa por encima de la capa límite, que llega hasta aproximadamente 1000 m. El tercer nivel representa la atmósfera libre.

Por lo general este tipo de modelos es razonablemente preciso para la predicción de variables meteorológicas como la temperatura, la precipitación, o la intensidad media del viento. Sin embargo, la magnitud del viento en un punto determinado es intermitente y presenta fluctuaciones muy grandes, ya que se ve afectado por factores muy concretos asociados a las características de la orografía, presencia de obstáculos y otros factores. Esto hace que dichos modelos no sean precisos a la hora de predecir el viento en un punto. Una de los procedimientos que se podría utilizar para mejorar estas predicciones es el uno de métodos de aprendizaje automático sobre datos de reanálisis. Es preferible utilizar los datos de reanálisis ya que las predicciones del modelo sin asimilación - es decir, sin ser corregidas por observaciones - suelen ser poco fiables.

1.2. Importancia de le energía eólica.

Como el viento es una fuente de energía, es necesario explicar cual es el impacto de la energía eólica en el sistema energético futuro. El modelo energético actual tiene problemas de sostenibilidad a largo plazo debido a la dependencia de los combustibles fósiles. Por ello, poco a poco, se han de ir cambiando las distintas fuentes de energía para llegar a un desarrollo sostenible. Las directrices propuestas por la Unión Europea en esta área son las que conforman el compromiso 20/20/20. Estos objetivos se muestran en la Fig. 1.3, donde se explica que a largo plazo las fuentes de energía emisoras de gases de efecto invernadero (GEI) serán suprimidas, mientras que el porcentaje de energías renovables para 2030 debería ser el 27 % de las fuentes energéticas de Europa.

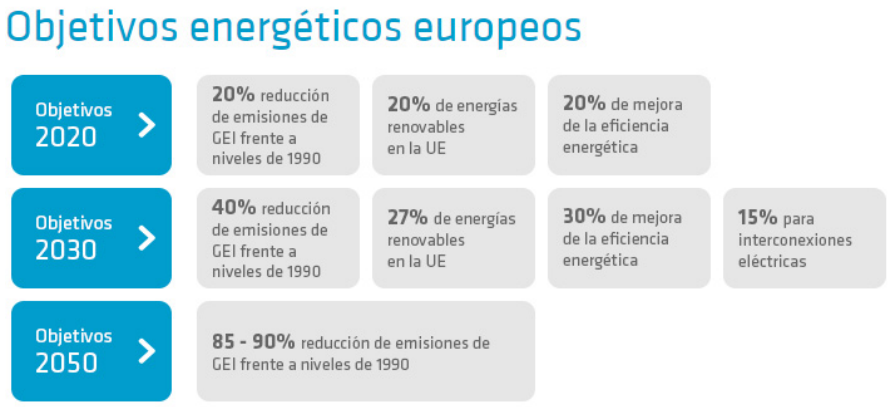


Figura 1.3: Objetivos energéticos europeos para el año 2020, 2030 y 2050 obtenidos de [33] Red Eléctrica de España.

Para ello, la Unión Europea define claramente tres elementos claves en el futuro modelo económico: la electrificación de la economía, la eficiencia energética y una mayor integración de las energías renovables

(Fig. 1.4). La electrificación de la economía implica que ciertos sectores abandonen su dependencia de los combustibles fósiles. Sectores como el transporte o la industria tienen una gran dependencia de este tipo de fuentes de energía. Sin embargo, si no se cambia la fuente de energía, el hecho de utilizar electricidad supone solo una movilización del foco de emisiones contaminantes, sin ventaja neta.

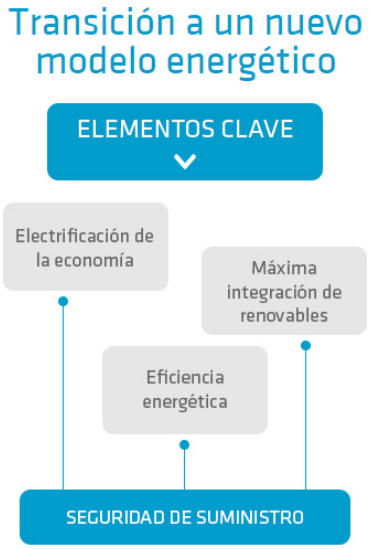


Figura 1.4: Directrices de la transición hacia el nuevo modelo energético europeo obtenidos de [33]

Para reducir las emisiones de gases que producen el efecto invernadero es necesario que las energías renovables tengan un papel más importante. La energía eólica adquirirá una mayor importancia en este sentido. La Red Eléctrica de España (REE) ha propuesto que, frente a la variabilidad de la producción renovable e incertidumbre en la predicción y su comportamiento ante las perturbaciones se mejore la calidad en la predicción.

1.3. Estructura del Trabajo Fin de Máster.

Este Trabajo Fin de Máster se articula en los siguientes capítulos:

- **Modelos de aprendizaje automático.** En este capítulo se explicará la teoría de los modelos de aprendizaje automático utilizados en los experimentos. Los algoritmos utilizados son las redes neuronales, máquinas de soporte vectorial, procesos gaussiano, Árboles Aleatorios, Potenciación del Gradiente y Potenciación Extrema del Gradiente.
- **Evaluación empírica.** En esta sección se explicarán los dos experimentos realizados y los resultados obtenidos. También se describirá el conjunto de datos utilizado durante los experimentos. Se aplicarán los distintos algoritmos explicados en el capítulo anterior para ver cual es el mejor.
- **Conclusiones.** El último capítulo resume las conclusiones obtenidas durante la realización de los experimentos.

Capítulo 2

Modelos de aprendizaje automático

En el capítulo anterior se han presentado algunos modelos numéricos meteorológicos utilizados para la predicción de viento. La mayoría de ellos son modelos deterministas que proporcionan predicciones razonablemente precisas para cantidades cuya variación es suave como la temperatura o la presión. Sin embargo, la naturaleza intermitente del viento hace que estos modelos no sean precisos a la hora de predecir dicha cantidad en un punto. Por ello, para la predicción de viento se utilizan a menudo modelos que incluyen términos estocásticos.

Otra dificultad es que los modelos meteorológicos generalmente proporcionan predicciones en una rejilla. En el caso de que la localización de interés no coincida con alguno de los puntos de esta rejilla, es necesario recurrir a sistemas que tomen como entrada las predicciones del modelo en los puntos de la rejilla y realicen la predicción en el lugar de interés. En este estudio aplicaremos distintos métodos de aprendizaje automático para la predicción de la velocidad del viento a partir de datos de reanálisis. El lugar en el que se realiza la predicción es un parque eólico sito en Peñaparda, Salamanca. Para la predicción se utilizan datos de reanálisis correspondientes a los 4 puntos más próximos en la rejilla del modelo de reanálisis.

En este capítulo se describen los métodos de aprendizaje automático utilizados para predicción. El objetivo es proporcionar una breve descripción de los modelos y de los hiperparámetros asociados. Comprender el papel de dichos hiperparámetros es importante a la hora de seleccionar configuraciones óptimas para realizar el aprendizaje y obtener buenos modelos de predicción.

2.1. Aprendizaje automático en problemas de regresión

En un problema de regresión, la meta es expresar una variable objetivo ($y \in \mathscr{Y}$) en función de otras variables del sistema ($\mathbf{x} \in \mathscr{X} \subset \mathbb{R}^D$), conocidas como variables predictoras. D representa el número total de variables predictoras. Con este fin, disponemos un conjunto de datos $\mathscr{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$, para los cuales son conocidos tanto los valores de las variables predictoras como los de la variable objetivo.

Para formalizar el problema, se realiza la suposición que $\mathscr{D}$ es una muestra de valores independientes e idénticamente distribuidos de las variables aleatorias $(\mathbf{X}, Y)$. Nuestra meta es identificar la hipótesis $h(\mathbf{x}) \in \mathscr{Y}$ que proporcione el mejor ajuste posible a los datos. Para medir la calidad del ajuste es necesario considerar una función de pérdida. Ejemplos de funciones de pérdida son el error absoluto medio

$$EAM = \mathbb{E}\left[|Y - h(\mathbf{X})|\right] \tag{2.1}$$

o el error cuadrático medio

$$ECM = \mathbb{E}\left[|Y - h(\mathbf{X})|^2\right]. \tag{2.2}$$

En ambas expresiones, los promedios son sobre la distribución conjunta de $(\mathbf{X}, Y)$. El problema de regresión se traduce en encontrar la hipótesis $h(\mathbf{x})$ que minimice el valor de la función de pérdida considerada.

Estas métricas se pueden generalizar en normas L^p:

$$L^p = \mathbb{E}\left[|Y - h(\mathbf{X})|^p\right] \text{ con } p \geq 1 \tag{2.3}$$

En este caso el EAM sería el equivalente a $p = 1$, mientras que el ECM sería si $p = 2$. Se puede demostrar que para cada norma L^p existe una constante que minimiza esta métrica. En el caso de $p = 1$, sería la

mediana y en el caso de $p = 2$, sería la esperanza. En el caso de que $p \to \infty$ la función que minimizaría esta norma sería el punto medio entre los valores supremo e ínfimo de la muestra. Las normas más comunes en la literatura son la norma L^1 y L^2, de ahí que en este trabajo sean las dos métricas utlizadas.

Dado que, en general, la distribución conjunta de $(\mathbf{X}, Y)$ no es conocida, se utilizan en el proceso de aprendizaje un estimador muestral de la función de pérdida. Por ejemplo, el estimador muestral del error cuadrático medio es

$$ECM = \frac{1}{N}\sum_{n=1}^{N} \left| y_n - h(\mathbf{x}_n) \right|^2. \tag{2.4}$$

Para que el problema de optimización sea abordable se suele restringir el espacio funcional en el que se realiza la búsqueda. A menudo se realiza la suposición de que la hipótesis se encuentra dentro del espacio funcional correspondiente a la familia $h(\mathbf{x};\theta)$, parametrizada por θ. Partiendo de esta suposición, el problema de aprendizaje se puede resolver minimizando la función de pérdida

$$\theta^* = \arg\min_{\theta} \text{Loss}\left(\{h(\mathbf{x}_n;\theta), y_n\}_{n=1}^{N} \right). \tag{2.5}$$

El espacio funcional considerado debe ser suficientemente rico como para poder capturar la relación existente entre las variables predictoras y la variable objetivo. En caso contrario, se pueden encontrar ajustes de baja calidad, alejados del óptimo.

También pueden surgir problemas en el caso de que el espacio funcional considerado sea muy flexible. En tal situación puede ocurrir que la hipótesis encontrada prediga con gran precisión los valores de la variable objetivo para el conjunto de datos de entrenamiento, y sin embargo presente errores considerables para un conjunto de test, compuesto por datos del mismo tipo, pero distintos de los utilizados para el aprendizaje. Este fenómeno, conocido como sobreajuste, se produce cuando el espacio de funciones en el que en el que se realiza la búsqueda es muy rico. En tal situación el proceso de aprendizaje conduce a la identificación de modelos complejos que reflejan no solo los patrones de regularidad útiles para la predicción, sino también patrones espúrios, a menudo asociados con ruido presente en los datos. Por este motivo, el sistema entrenado no es capaz de generalizar; es decir, predecir con precisión los valores de la variable objetivo en datos distintos a los utilizados para el entrenamiento. La tendencia al sobreajuste se puede reducir o bien aumentando el tamaño del conjunto de datos de entrenamiento, o bien limitando la complejidad del modelo mediante técnicas de regularización o con un enfoque Bayesiano.

Una ilustración gráfica de las situaciones descritas se encuentra en la figura 2.1. Se trata de un problema de regresión en una dimensión. En él se ha generado puntos utilizando la función seno contaminada con ruido aditivo con una distribución de probabilidad $\mathcal{N}(\mu = 0, \sigma^2)$.

El espacio funcional en el que se realiza la búsqueda de hipótesis es el conjunto de polinomios

$$h(x_n;\theta) = \sum_{m=0}^{M} \theta_m x_n^m. \tag{2.6}$$

La complejidad de este modelo depende únicamente del hiperparámetro M, el grado del polinomio. Los coeficientes θ_m se determinan minimizando la función de pérdida, en este caso el error cuadrático medio.

En la primera gráfica se muestra como una recta ($M = 1$) no es capaz de reflejar la dependencia sinusoidal en los datos. Es un ejemplo de ajuste insuficiente o infraajuste. En la segunda gráfica, se puede observar que un polinomio de grado $M > 30$ presenta sobreajuste. En concreto, al intentar modelizar el ruido inyectado, el modelo predice fluctuaciones de una frecuencia mucho mayor que el seno. En la última se observa que un polinomio de grado $M = 3$ capta la tendencia de los datos sin aprender el ruido. En este caso, el modelo cúbico es, dentro de la familia polinómica considerada el que presenta un grado de complejidad suficiente para representar, en el rango de valores considerados, la relación entre la variable predictora y la objetivo sin que se produzca sobreajuste.

Existen varias formas de evitar el sobreajuste. En concreto, se puede introducir un término que penalice la complejidad del modelo,

$$\theta^* = \arg\min_{\theta} \left[\text{Loss}\left(\{h(\mathbf{x}_n;\theta), y_n\}_{n=1}^{N} \right) + \alpha \text{Complexity}(\theta) \right]. \tag{2.7}$$

El hiperparámetro $\alpha \geq 0$ establece la importancia relativa del término de complejidad respecto al de pérdida. Cuanto mayor sea su valor menor será la tendencia al sobreajuste. No obstante, es necesario limitar su

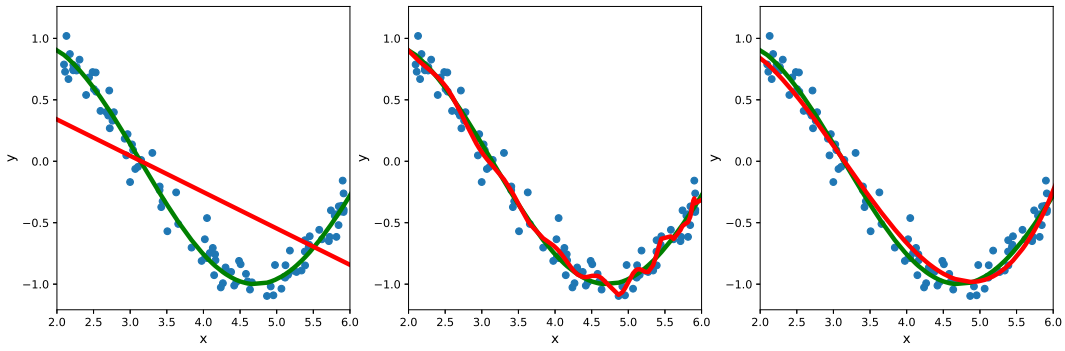


Figura 2.1: Ilustración del fenómeno de sobreajuste en un problema de regresión. La línea verde traza la función original (la función seno). Los puntos marcan lo ejemplos a partir de los que se ha realizado el aprendizaje. La línea roja corresponde a la predicción dada por el modelo.

tamaño para evitar que el modelo sea rígido en exceso y se produzca infraajuste. El valor de este hiperparámetro se puede determinar utilizando técnicas de validación.

En este trabajo se utilizan los siguientes modelos para regresión: redes neuronales, máquinas de vectores soporte, procesos gaussianos, bosques aleatorios, potenciación del gradiente y potenciación extrema del gradiente. En las siguientes secciones procederemos a describir sus características y los elementos que es necesario especificar en su diseño.

2.2. Redes neuronales

Una red neuronal es un sistema inspirado en el cerebro. El cerebro es el órgano central del sistema nervioso de los animales superiores. Santiago Ramón y Cajal descubrió que el sistema nervioso no es un continuo, sino que está compuesto por células nerviosas o neuronas [5]. Las neuronas del cerebro forman una red cuyas conexiones se denominan sinapsis. A través de estas conexiones sinápticas, las neuronas se comunican entre sí mediante impulsos electrícos y químicos. Una neurona biológica es una unidad de integración y disparo: recibe y acumula señales nerviosas que provienen del disparo de otras neuronas conectadas con ellas por medio de una sinapsis. Una vez la intensidad acumulada sobrepasa un cierto umbral, la neurona genera un impulso, el cual, a su vez, es transmitido a través de conexiones sinápticas a otras neuronas. En una neurona biológica se pueden distinguir en tres partes:

- Dendritas: terminales de la neurona a través de los cuales se reciben los impulsos nerviosos que provienen del disparo de otras neuronas conectadas por una sinapsis.
- Soma: Zona central de la célula en la que se encuentra el núcleo y en la que se realiza el proceso de integración (acumulación) de señales que son recibidas desde otras neuronas a través de las dendritas.
- Axón: terminal alargado de la neurona, que se inserta en el soma y por el que se transmite la señal de disparo hacia otras neuronas.

En la figura 2.2 se muestra de manera esquemática una neurona artificial [26]. Dicha neurona recibe como entradas los valores $\{x_1, x_2, \ldots, x_m\}$ y genera como salida el valor $g(\sum_{i=0}^{m} w_i x_i)$, donde $\{w_i\}_{i=0}^{m}$ es el conjunto de pesos sinápticos y $g(z)$ es una función de activación que realiza una transformación no lineal. Las entradas incluyen una señal constante $x_0 = 1$ a la que se denomina sesgo.

A continuación se describen funciones de activación que se utilizan habitualmente:

- La función sigmoidal o logística (Ec. 2.8) es una función de activación con la propiedad de transformar la recta real a un intervalo $[0, 1]$. Esto se puede considerar como una transformación del dominio de las variables a un espacio de probabilidad. Esta función se utiliza no solo como función de activación de las capas intermedias sino también como función de activación de la neurona de salida en

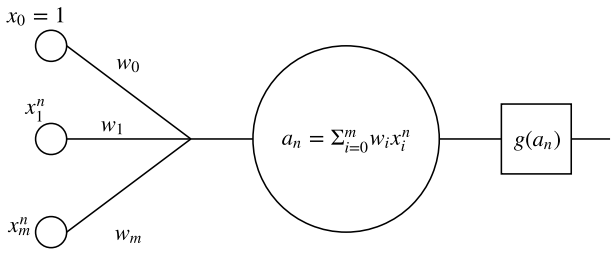


Figura 2.2: Estructura de una neurona artificial.

problemas de clasificación binaria. Aunque esta función sea muy utilizada, presenta problemas en redes neuronales con muchas capas, ya que la función satura en una asíntota rápidamente y la derivada en esos intervalos es nula, con el resultado de que deja de aprender. Esto se puede evitar con una inicialización de los pesos adecuada.

$$g(z) = \frac{1}{1+e^{-z}} \tag{2.8}$$

- La función tangente hiperbólica (ecuación 2.9) tiene una forma similar a la función logística pero el intervalo al que transforma la recta real es $[-1,1]$. Debido que el recorrido de la función no tiene traducción en un espacio de probabilidades y es una buena aproximación de la función escalón, la función se utiliza en capas intermedias. No obstante, esta función también tiene el problema de aprendizaje en redes profundas y de inicialización que presentaba la función logística . Por tanto, también comparte la misma solución.

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.9}$$

- Para evitar que las neuronas dejen de aprender debido a que la derivada tienda rápidamente a 0, se utiliza la función ReLU (*Rectified Linear Units*), que tiene la forma mostrada en la ecuación 2.10. Esta función tiene la propiedad de que es 0 en la parte negativa de la recta real y la función identidad en la parte positiva de la recta real. Con esto se consigue que mientras la neurona esté activa, es decir, con un valor no nulo, la derivada no tienda a cero y el proceso de aprendizaje continue. Esta función de activación es utilizada habitualmente en capas intermedias de redes neuronales profundas.

$$g(z) = \text{máx}(0,z) \tag{2.10}$$

- La función identidad (ecuación 2.11) es aquella que deja invariante la entrada. Se usa principalmente en las neuronas de la capa de salida en problemas de regresión.

$$g(z) = z \tag{2.11}$$

La arquitectura de un perceptrón multicapa con dos capas ocultas se muestra la figura 2.3. Cada una de las flechas que aparece en el esquema representa una conexión entre dos neuronas. Estas conexiones corresponden a pesos sinápticos entrenables.

Desde el punto de vista funcional, se puede considerar que el cerebro es un sistema que recibe estímulos de entrada y genera respuestas. Las neuronas que componen el sistema nervioso se organizan en capas. En la primera capa, se recibe como entrada un estímulo externo (por ejemplo, estímulos lumínicos en la retina). Esta señal se transmite a las siguientes capas para su procesamiento. En una capa intermedia dada las neuronas realizan un procesamiento de las señales que provienen de la capa anterior. Finalmente, como resultado del procesamiento de la capa de salida se genera una respuesta.

Esta arquitectura, de inspiración biológica, es la que se usa en un perceptrón multicapa, que es el tipo de red neuronal más común, y es el que se utilizará en el presente estudio. En la capa de entrada de este tipo de red las neuronas transmiten los valores de las variables predictoras a las neuronas de la siguiente

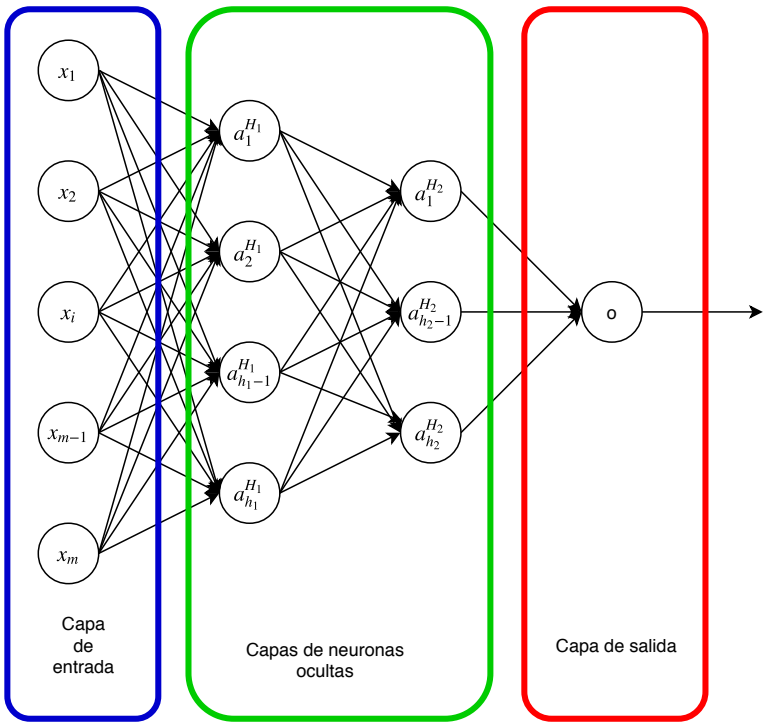


Figura 2.3: Esquema de una red neuronal con dos capas ocultas H_1 y H_2 con un número de neuronas h_1 en la primera capa y h_2 en la segunda capa.

capa. Las capa de entrada está conectada a la primera capa oculta mediante los pesos $\mathbf{w}^{(i)}$. La dimensión de $\mathbf{w}^{(\mathbf{i})}$ tiene el tamaño del número de variables predictoras más un peso adicional correspondiente al término de sesgo. En las capas intermedias de la red, conocidas como capas ocultas, cada neurona recibe la salida de las neuronas de la capa anterior. El procesamiento que lleva a cabo una neurona de una capa oculta consiste en realizar una combinación lineal de los valores de entrada que provienen de las neuronas de la capa anterior. Los pesos de esta combinación lineal $\mathbf{w}^{(H_1)}$ y $\mathbf{w}^{(H_2)}$ son los parámetros a determinar en el proceso de aprendizaje. Posteriormente, en la neurona oculta se realiza una transformación generalmente no lineal de la señal resultante. Para un problema de regresión univariante, la capa de salida posee una única neurona que únicamente realiza una combinación lineal de los valores de salida de las neuronas de la última capa oculta. La fórmula asociada a cada una de las neuronas de cada capa oculta es la siguiente:

$$a_j^{H_1} = g^{H_1}\left(\sum_{k=0}^{h_1} w^{(i)} x_k\right); \tag{2.12}$$

$$a_i^{H_2} = g^{H_2}\left(\sum_{j=0}^{h_1} w^{(H_1)} a_j^{(H_1)}\right). \tag{2.13}$$

Donde $g(\mathring{u})$ representa la función de actiivación de la capa correspondiente. Teniendo finalmente la función de regresión la expresión:

$$h(\mathbf{x},\mathbf{w}) = g^{(o)}\left(\sum_{i=0}^{h_2} w_i^{(H_2)} a_i^{(H_2)}\right), \tag{2.14}$$

Donde $\mathbf{w}$ esla matriz formada por los pesos que conforman la red neuronal $\mathbf{w} = \left(\mathbf{w}^{(i)}, \mathbf{w}^{(H_1)}, \mathbf{w}^{(H_2)}\right)$

El primer paso en el diseño de una red neuronal es determinar su arquitectura. Es decir, el número de capas, las neuronas que pertenecen a cada una de estas capas y el tipo de función de activación. La arquitectura de la red neuronal debe ser elegida para cada conjunto de datos. La selección de una arquitectura adecuada para un problema dado se puede realizar mediante un método de validación, como por ejemplo, validación cruzada.

Una vez determinada la arquitectura, es necesario determinar el valor de los pesos sinápticos $\mathbf{w}$ se realiza minimizando una función de coste. La función de coste utilizada en este trabajo corresponde a:

$$\text{Cost}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(h(\mathbf{x}_n;\mathbf{w}) - y_n)^2 + \frac{\alpha}{2}||\mathbf{w}||_2^2 \tag{2.15}$$

Esta función de coste tiene dos términos. El primero es el error cuadrático medio que cuantifica cuán cerca está la predicción de la red, $h(x_n;\mathbf{w})$ del valor real de la variable objetivo y_n. Adicionalmente, se incluye un término de regularización, que es proporcional al cuadrado del módulo de los pesos sinápticos $\mathbf{w}$. Este término es necesario para que los pesos de la red neuronal no sean muy altos y para limitar el sobreajuste a los datos de entrenamiento. Este término no es necesario, ya que existen otros métodos para evitar el sobreajuste (por ejemplo, eliminando de forma aleatoria algunas neuronas en cada paso del entrenamiento). Se ha añadido debido a que las redes neuronales que se utilizan en este trabajo tienen este término de penalización de la complejidad. Para una arquitectura dada, el parámetro que controla la complejidad del modelo predictivo es $\alpha > 0$. La tendencia al sobreajuste será tanto mayor cuanto más capas ocultas y neuronas en cada capa oculta haya, y cuanto menor sea el valor de α. Si α es muy pequeño , la red tenderá a memorizar los ejemplos de entrenamiento, lo que tiende a reducir la capacidad de generalización. Por el contrario, si α tiene un valor excesivamente alto, la mayoría de los pesos serín próximos a cero, y la red no tendrá suficiente capacidad expresiva para predecir. El valor del hiperparámetro α se determina de manera conjunta con la arquitectura de la red, y también requiere de un proceso de validación.

Para minimizar esta función de coste se pueden utilizar algoritmos de optimización no lineal, como gradientes conjugados o métodos cuasi-Newton, o de descenso por gradiente, generalmente descenso por gradiente estocástico [20].

Una ventaja de las redes neuronales es el teorema de aproximación universal,[9]. Este dice que una red neuronal de una sola capa con una capa oculta y un número finito de neuronas puede aproximar cualquier función que pertenezca a $\mathbb{R}$, siempre y cuando se cumplan una serie de condiciones para las funciones de activación. Sin embargo, esto no siempre es verdad cuando se evalúa empíricamente este método.

La teoría y descripción de las redes neuronales es muy amplia y aunque existen más tipos de capas, funciones de activación y otras formas de regularizar [23].

2.3. Máquinas de vectores soporte

En esta sección se presenta una descripción de las máquinas de soporte vectorial para abordar problemas de regresión [36, 40]. En este tipo de problemas se dispone de un conjunto de entrenamiento $\mathscr{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, donde $\mathbf{x}_n \in \mathscr{X} \subset \mathbb{R}^D$ es el vector de atributos que caracteriza el ejemplo n-ésimo e $y_n \in \mathbb{R}$ es la variable a predecir. El objetivo es inducir a partir de estos datos una función de regresión

$$f : \mathscr{X} \to \mathbb{R} \tag{2.16}$$

de forma que la predicción $f(\mathbf{x}_n)$ difiera como mucho una distancia ϵ del valor de y_n para cada uno de los ejemplos del conjunto de entrenamiento [40]. La magnitud de ϵ puede ser interpretada como el nivel de ruido de las observaciones. Inicialmente supondremos que esta función es lineal en el espacio de características en el que está formulado el problema

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b, \quad \text{donde } \mathbf{x}, \mathbf{w} \in \mathscr{X}, \text{y} b \in \mathbb{R}. \tag{2.17}$$

En esta ecuación $\langle \cdot, \cdot \rangle$ representa el producto escalar en $\mathscr{X}$. Posteriormente extenderemos el modelo, suponiendo que la relación es lineal en un espacio de características extendido, de forma que se contemple la posibilidad de que la dependencia entre los atributos y la variable objetivo sea no lineal en el espacio original.

Una de las características deseables para la función de regresión es que sea lo más plana posible, con el fin de limitar el problema del sobreajuste. Una manera de alcanzar este objetivo es minimizar la norma l_2 del vector de coeficientes de la regresión lineal, $||\mathbf{w}||^2 = \langle \mathbf{w}, \mathbf{w} \rangle$. El problema resultante es

$$\begin{aligned} &\text{minimizar} && \tfrac{1}{2}||\mathbf{w}||^2 \\ &\text{sujeto a:} && \begin{cases} y_n - \langle \mathbf{w}, \mathbf{x}_n \rangle - b \leq \epsilon \\ \langle \mathbf{w}, \mathbf{x}_n \rangle + b - y_n \leq \epsilon \end{cases}, \quad n = 1, \ldots, N. \end{aligned} \tag{2.18}$$

Las restricciones impuestas en la minimización reflejan el objetivo de que $f(\mathbf{x}_n)$ no se separe más que una distancia ϵ de la variable dependiente, y_n, para los ejemplos del conjunto de entrenamiento. Sin embargo, no siempre va a ser posible cumplir esta condición, ya que podría ser restrictiva en exceso. Para solucionar esto, se puede permitir una mayor flexibilidad en el modelo haciendo que el margen sea menos rígido. En concreto, se permitirá que las predicciones $f(\mathbf{x}_n)$ puedan presentar una desviación mayor que ϵ de los valores y_n, si bien tales desviaciones se penalizarán de manera lineal. Para ello, se introducen las variables de holgura ξ_n y ξ_n^*. Con estas modificaciones, el problema de optimización se transforma en

$$\begin{array}{ll} \text{minimizar} & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{n=1}^{N}\left(\xi_n + \xi_n^*\right) \\ \text{sujeto a:} & \left\{\begin{array}{lll} y_n - \langle \mathbf{w}, \mathbf{x}_n\rangle - b & \leq & \epsilon + \xi_n \\ \langle \mathbf{w}, \mathbf{x}_n\rangle + b - y_n & \leq & \epsilon + \xi_n^* \quad n = 1,\ldots,N. \\ \xi_n, \xi_n^* & \geq & 0 \end{array}\right. \end{array} \tag{2.19}$$

En la función objetivo se ha introducido el hiperparámetro $C > 0$, que regula el peso relativo entre el término de regularización (mediante la minimización de $||\mathbf{w}||^2$) y la flexibilidad permitida en cuanto a que puedan existir desviaciones mayores que ϵ. En concreto, las desviaciones δy de las predicciones del modelo respecto a los valores observados para la variable objetivo se penalizan con una función de pérdida $\epsilon-$insensible [36, 40], la cual tiene la forma

$$|\delta y|_\epsilon := \left\{\begin{array}{ll} 0 & si\ |\delta y| \leq \epsilon \\ |\delta y| - \epsilon & si\ |\delta y| > \epsilon \end{array}\right. . \tag{2.20}$$

El problema primal planteado en (2.19) es un problema de optimización con restricciones en el espacio de variables $\left(\mathbf{w}, b, \xi_n, \xi_n^*\right)$. Para abordar este problema, construiremos la función lagrangiana, $\mathscr{L}$. Esta función es la cantidad que se quiere minimizar en el primal menos la suma de cada una de las restricciones multiplicadas por unos coeficientes llamados multiplicadores de Lagrange. Cada restricción tiene asociada un multiplicador de Lagrange distinto. Estos coeficientes deben ser mayores o iguales que cero.

La función lagrangiana para una SVR es

$$\begin{aligned} \mathscr{L} &= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{n=1}^{N}\left(\xi_n + \xi_n^*\right) - \sum_{n=1}^{N}\left(\eta_n\xi_n + \eta_n^*\xi_n^*\right) \\ &\quad - \sum_{n=1}^{N}\left(\alpha_n\left(\epsilon + \xi_n - y_n + \langle \mathbf{w}, \mathbf{x}_n\rangle + b\right)\right) - \sum_{n=1}^{N}\left(\alpha_n^*\left(\epsilon + \xi_n^* + y_n - \langle \mathbf{w}, \mathbf{x}_n\rangle - b\right)\right). \end{aligned} \tag{2.21}$$

Como se puede ver, la lagrangiana incluye la función objetivo del primal, $\frac{1}{2}||\mathbf{w}||^2 + C\sum_{n=1}^{N}\left(\xi_n + \xi_n^*\right)$. Los términos de cada restricción aparecen multiplicados por el correspondiente multiplicador de Lagrange. La restricciones ξ_n y $\xi_n^* \geq 0$ tienen asociados los términos $\eta_n\xi_n$, y $\eta_n^*\xi_n^*$, respectivamente. Los correspondientes multiplicadores de Lagrange son η_n y η_n^*. Las condiciones de que $f(\mathbf{x}_n)$ no se separe una distancia $\epsilon + \xi_n^{(*)}$ de y_n, donde $\xi_n^{(*)}$ puede ser o bien ξ_n o bien ξ_n^*, tiene asociados los términos $\alpha_n\left(\epsilon + \xi_n - y_n + \langle \mathbf{w}, \mathbf{x}_n\rangle + b\right)$ y $\alpha_n^*\left(\epsilon + \xi_n^* + y_n - \langle \mathbf{w}, \mathbf{x}_n\rangle - b\right)$. Los valores α_n y α_n^* son los mutiplicadores de Lagrange para cada una de estas restricciones.

Como se ha dicho antes, los multiplicadores de Lagrange están sujetos a la restricción de ser no negativos:

$$\alpha_n, \alpha_n^*, \eta_n, \eta_n^* \geq 0, \quad n = 1,\ldots,N. \tag{2.22}$$

Se puede demostrar que en la solución de (2.19), la lagrangiana tiene un punto de silla: presenta un mínimo respecto a las variables del primal (en este caso, $\mathbf{w}, b, \xi_n, \xi_n^*$) y un máximo respecto a las del dual (en este caso, $\alpha_n, \alpha_n^*, \eta_n, \eta_n^*$). Las condiciones de punto de silla se pueden obtener derivando la lagrangiana respecto a las variables primales e igualando a cero:

$$\partial_b\mathscr{L} = \sum_{n=1}^{N}\left(\alpha_n - \alpha_n^*\right) = 0; \tag{2.23}$$

$$\partial_{\mathbf{w}}\mathscr{L} = \mathbf{w} - \sum_{n=1}^{N}\left(\alpha_n - \alpha_n^*\right)\mathbf{x}_n = 0; \tag{2.24}$$

$$\partial_{\xi_n^{(*)}}\mathscr{L} = C - \alpha_n^{(*)} - \eta_n^{(*)} = 0, \quad n = 1,\ldots,N. \tag{2.25}$$

A partir de la igualdad (2.24) se pueden expresar los pesos de la función regresora en función de las variables duales α_n y α_n^*:

$$\mathbf{w} = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*)\mathbf{x}_n. \tag{2.26}$$

Asimismo, la ecuación (2.25) permite expresar $\eta_n^{(*)}$ en función de $\alpha_n^{(*)}$:

$$\eta_n = C - \alpha_n, \quad \eta_n^* = C - \alpha_n^*, \quad n = 1, \ldots, N. \tag{2.27}$$

Sustituyendo en la lagrangiana los valores de $\mathbf{w}$ y $\eta_n^{(*)}$ obtenidos en función de $\alpha_n^{(*)}$, se obtiene el problema dual:

$$\text{maximizar} \quad -\tfrac{1}{2}\textstyle\sum_{n,m=1}^{N}\left(\alpha_n - \alpha_n^*\right)\left(\alpha_m - \alpha_m^*\right)\langle \mathbf{x}_n, \mathbf{x}_m\rangle - \epsilon\sum_{n=1}^{N}\left(\alpha_n + \alpha_n^*\right) + \sum_{n=1}^{N} y_n\left(\alpha_n - \alpha_n^*\right) \tag{2.28}$$

$$\text{sujeto a:} \quad \begin{cases} \sum_{n=1}^{N}\left(\alpha_n - \alpha_n^*\right) & = & 0 \\ \alpha_n, \alpha_n^* & \in & [0, C], \quad n = 1, \ldots, N. \end{cases} \tag{2.29}$$

La primera restricción proviene de la igualdad (2.23). La segunda se deriva a partir de las condiciones de que tanto $\eta_n^{(*)}$ como $\alpha_n^{(*)}$ son no negativos (por ser multiplicadores de Lagrange) y están relacionados por (2.27).

Se ha comentado previamente que si existe una solución, esta tiene que ser un punto de silla. Para que esto ocurra se han de satisfacer las condiciones de Karush-Kuhn-Tucker (KKT). De estas condiciones se establece, entre otras relaciones, que

$$\alpha_n \alpha_n^* = 0, , \quad n = 1, \ldots N. \tag{2.30}$$

De las condiciones KKT y la restricción de que α_n y α_n^* no pueden tomar valores negativos, se establece la siguiente clasificación para los ejemplos de entrenamiento:

- Ejemplos para los cuales $\alpha_n = \alpha_n^* = 0$. En este caso, el ejemplo correspondiente no es utilizado para la predicción.
- Ejemplos para los cuales $\alpha_n = 0$ y $\alpha_n^* > 0$ o $\alpha_n^* = 0$ y $\alpha_n > 0$. Estos ejemplos son los llamados *vectores soporte*. Son los que se utilizan para realizar predicciones. Un caso especial de vectores soporte es cuando $\begin{pmatrix} \alpha_n = 0 \\ \alpha_n^* = C \end{pmatrix}$ o $\begin{pmatrix} \alpha_n = C \\ \alpha_n^* = 0 \end{pmatrix}$. En este caso, $\mathbf{x}_n$ se encuentra fuera de la región en la que no son penalizadas las desviaciones de las predicciones respecto a los valores observados (zona $\epsilon-$insensible).

La función regresora de una máquina de vectores soporte es:

$$f(\mathbf{x}) = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*)\langle \mathbf{x}_n, \mathbf{x}\rangle + b. \tag{2.31}$$

El siguiente paso consiste en realizar una formulación no lineal para la SVR. Con este fin, haremos la hipótesis que la función de regresión es lineal en un espacio extendido de características, $\mathscr{F}$. Este espacio se obtiene mediante una transformación no lineal de los atributos en el espacio original:

$$\Phi: \quad \mathscr{X} \to \mathscr{F} \tag{2.32}$$

$$\mathbf{x} \to \Phi(\mathbf{x}). \tag{2.33}$$

El modelo resultante es

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b, \tag{2.34}$$

donde $\langle \cdot, \cdot \rangle$ es el producto ecalar en $\mathscr{F}$.

Una vez formulado el modelo lineal en el espacio de características aplicaremos el llamado truco del kernel. Este parte de la observación de que en el espacio de atributos original el problema de optimización dual depende únicamente de productos escalares en $\mathscr{X}$

$$\langle \mathbf{x}, \mathbf{x}^* \rangle. \tag{2.35}$$

Por lo tanto, en el espacio extendido, el problema dual puede ser expresado únicamente en función del kernel

$$k(\mathbf{x},\mathbf{x}^*) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}^*) \rangle, \tag{2.36}$$

que está definido como el producto escalar en $\mathscr{F}$. Basta con sustituir en Eq. (2.28) los productos escalares $\langle \mathbf{x}_n, \mathbf{x}_m \rangle$ por evaluaciones del kernel $k(\mathbf{x}_n, \mathbf{x}_m)$. El problema dual resultante es:

$$\begin{aligned} &\text{maximizar} \quad -\tfrac{1}{2}\sum_{n,m=1}^{N}(\alpha_n - \alpha_n^*)(\alpha_m - \alpha_m^*)\,k(\mathbf{x}_n,\mathbf{x}_m) - \epsilon\sum_{n=1}^{N}(\alpha_n + \alpha_n^*) + \sum_{n=1}^{N} y_n(\alpha_n - \alpha_n^*) \\ &\text{sujeto a:} \quad \begin{cases} \sum_{n=1}^{N}(\alpha_n - \alpha_n^*) &= 0 \\ \alpha_n, \alpha_n^* &\in [0, C], \quad n = 1, \ldots, N \end{cases} . \end{aligned} \tag{2.37}$$

De manera análoga al caso lineal, los pesos de la función de regresión se pueden expresar en este nuevo espacio de atributos

$$\mathbf{w} = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)\Phi(\mathbf{x}_n). \tag{2.38}$$

Al igual que en la SVR lineal, aquellos $\Phi(\mathbf{x}_n)$ para los cuales α_n o α_n^* sean distintos que 0, son los vectores soporte. A pesar de que en la expresión del vector de pesos aparece explícitamente el vector de características, la función de regresión en este espacio extendido solo depende del kernel

$$f(\mathbf{x}) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)k(\mathbf{x}_n,\mathbf{x}) + b. \tag{2.39}$$

Desde el punto de vista computacional es importante que las cantidades de interés puedan expresarse como evaluaciones del kernel y que no sea necesario trabajar en el espacio $\mathscr{F}$, el cual podría tener dimensión infinita.

Para finalizar, se describen dos tipos de kernels que son utilizados comúnmente en máquinas de vectores soporte:

- kernel polinómico.

  $$K(\mathbf{x}_n,\mathbf{x}_m) = (\mathbf{x}_n \cdot \mathbf{x}_m + c)^d \tag{2.40}$$

  Los hiperparámetros de este kernel son el grado del polinomio, d, y el término independiente, c.

- kernel gaussiano.

  $$K(\mathbf{x}_n,\mathbf{x}_m) = e^{-\gamma||\mathbf{x}_n - \mathbf{x}_m||^2} \text{ donde } \gamma = \frac{1}{2\sigma^2}. \tag{2.41}$$

  Este kernel corresponde a un espacio de características de dimensión infinita. Depende del hiperparámetro σ, el cual determina el tamaño de la zona la influencia de un ejemplo de entrenamiento.

Los valores de C, σ_ϵ, el tipo de kernel y los hiperparámetros del kernel normalmente se determinan mediante validación cruzada. En concreto, se suele realizar una búsqueda en rejilla en el espacio que resulta de discretizar el espacio de hiperparámetros de la siguiente forma:

- El hiperparámetro C se puede discretizar en una escala logarítmica de base 2 con valores que van desde 2^{-5} hasta 2^6.
- El hiperparámetro ϵ suele ser una fracción de la desviación estándar de la variable y, σ_y.
- En caso de que se utilice un kernel gaussiano, el valor de γ es una potencia de $\frac{1}{D}$. Donde D es la dimensión del espacio de atributos $\mathscr{X}$.

Para finalizar, cabe destacar que existen varias implementaciones de este método de regresión. IBM publicó su implementación en la librería OSL. Otros paquetes son CPLEX [21], LOQO y livsvm [6]. Este último es uno de los más recientes y estables a nivel computacional. La implementación utilizada en este trabajo fin de máster es la de *scikit-learn* [29]. Esta implementación de la máquina de vectores soporte para regresión está basada en la librería livsvm.

2.4. Procesos Gaussianos

En esta sección se describe el uso de modelos basados en procesos gaussianos para abordar problemas de regresión [32, 34].

Un proceso gaussiano es un proceso estocástico que define una distribución sobre un espacio de funciones, de tal forma que si tomamos observaciones de esas funciones en un conjunto finito de puntos, los valores observados tienen una distribución conjunta gaussiana multivariante.

Para abordar el problema de regresión disponemos de un conjunto de datos de entrenamiento $\mathscr{D} = (\mathbf{X}, \mathbf{y}) = \{\mathbf{x}_n, y_n\}_{n=1}^N$, donde $\mathbf{x}_n \in \mathscr{X} \subset \mathbb{R}^D$ es el vector de atributos del ejemplo n-ésimo e $y_n \in \mathbb{R}$ es el valor observado de la variable dependiente. Partiremos de la hipótesis de que estos datos han sido generados por un modelo aditivo

$$y_n = f(\mathbf{x}_n) + \epsilon_n, \quad n = 1, 2, \ldots, N. \tag{2.42}$$

Los valores $\{\epsilon_n\}_{n=1}^N$ son ruido blanco gaussiano; es decir, son valores aleatorios independientes e idénticamente distribuidos de acuerdo con $\epsilon_n \sim \mathcal{N}(0, \sigma_\epsilon)$. La presencia de ruido aditivo en el modelo refleja la suposición de que existe cierta incertidumbre en las observaciones. Supondremos también que los valores $f(\mathbf{x}_n)$ son evaluaciones de una función aleatoria que es una realización de un proceso gaussiano:

$$f \sim \mathbb{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{2.43}$$

De manera análoga a una distribución gaussiana $\mathcal{N}(\mu, \sigma)$, que está caracterizada por los parámetros media μ y varianza σ^2, un proceso gaussiano se caracteriza por una función de media $m(\mathbf{x})$ y una función de covarianza $k(\mathbf{x}, \mathbf{x}')$. La media evaluada en $\mathbf{x}$ es el valor esperado de las funciones aleatorias que son realizaciones del proceso gaussiano evaluadas en dicho punto:

$$m(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right]. \tag{2.44}$$

Por simplicidad en las derivaciones que realizaremos supondremos que esta media es nula, $m(\mathbf{x}) = 0$. En la práctica, esto se consigue restando la media estimada del conjunto de entrenamiento a todas las observaciones.

La función de covarianza $k(\mathbf{x}, \mathbf{x}')$ refleja las dependencias entre los valores de una función aleatoria que es una realización del proceso gaussiano en dos puntos de su dominio $\mathbf{x}$ y $\mathbf{x}'$. La función de covarianza o kernel tiene la siguiente expresión:

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \mathbb{E}\left[\left(f\left(\mathbf{x}\right) - m\left(\mathbf{x}\right)\right)\left(f\left(\mathbf{x}'\right) - m\left(\mathbf{x}'\right)\right)\right] \tag{2.45}$$

La elección del kernel del proceso gaussiano está basada en suposiciones sobre las características de la función de regresión, f. Por ejemplo, se puede suponer que la función es suave o que es derivable hasta cierto orden. Un ejemplo de propiedad que se asume comúnmente es que la covarianza de la función evaluada en dos puntos decae con la distancia entre dichos puntos. Ejemplos de distintos tipos de kernel se introducirán al final de esta sección.

Las funciones que son realizaciones de un proceso gaussiano dependen de un parámetro continuo, $\mathbf{x}$. Sin embargo, en la práctica, tendremos observaciones únicamente en un número finito de puntos. Por ejemplo, dado un conjunto de ejemplos de test $\{\mathbf{x}_n^*\}_{n=1}^{N_{test}}$, el vector de predicciones es

$$\mathbf{f}^* = \left(f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_{N_{test}}^*)\right). \tag{2.46}$$

Dado que la función de regresión es una realización del proceso gaussiano, la distribución de este vector es una gaussiana multivariante. Suponiendo que el proceso es de media nula, podemos caracterizar dicha distribución utilizando únicamente el kernel: sea $\mathbf{X}^* = \left(\mathbf{x}_1^*, \ldots, \mathbf{x}_{N_{test}}^*\right)$ una matriz cuyas columnas son los vectores de atributos de ejemplos del conjunto de test. La distribución conjunta de los valores de la función evaluada en esos puntos es una gaussiana de media $\mathbf{0}$ y matriz de covarianzas de dimensiones $N_{test} \times N_{test}$

$$k\left(\mathbf{X}^*, \mathbf{X}^*\right) = \begin{bmatrix} k\left(\mathbf{x}_1^*, \mathbf{x}_1^*\right) & k\left(\mathbf{x}_1^*, \mathbf{x}_2^*\right) & \cdots & k\left(\mathbf{x}_1^*, \mathbf{x}_{N_{test}}^*\right) \\ k\left(\mathbf{x}_2^*, \mathbf{x}_1^*\right) & k\left(\mathbf{x}_2^*, \mathbf{x}_2^*\right) & \cdots & k\left(\mathbf{x}_2^*, \mathbf{x}_{N_{test}}^*\right) \\ \vdots & \vdots & \ddots & \vdots \\ k\left(\mathbf{x}_{N_{test}}^*, \mathbf{x}_1^*\right) & k\left(\mathbf{x}_{N_{test}}^*, \mathbf{x}_2^*\right) & \cdots & k\left(\mathbf{x}_{N_{test}}^*, \mathbf{x}_{N_{test}}^*\right) \end{bmatrix}. \tag{2.47}$$

Por lo tanto, la distribución del vector de observaciones es la gaussiana multivariante:

$$\mathbf{f}^* \sim \mathcal{N}\left(\mathbf{0}, k(\mathbf{X}^*, \mathbf{X}^*)\right). \tag{2.48}$$

Esta distribución de probabilidad es el prior de $\mathbf{f}^*$, que no tiene en cuenta los valores de las observaciones realizadas.

La distribución de la variable objetivo para las observaciones realizadas, $\mathbf{y} = \{y_n\}_{n=1}^N$, se define de manera análoga, excepto que es necesario añadir en los elementos de la diagonal de la matriz de covarianzas un término σ_ϵ^2 procedente del ruido blanco gaussiano que, hemos supuesto, contamina las observaciones

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{0}, k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}\right), \tag{2.49}$$

donde $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ es una matriz cuyas columnas son los vectores de atributos de los ejemplos del conjunto de entrenamiento, $k(\mathbf{X}, \mathbf{X})$ es la matriz de Gram

$$k(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}, \tag{2.50}$$

σ_ϵ^2 es la varianza del ruido aditivo e $\mathbf{I}$ es la matriz identidad de tamaño $N \times N$.

El siguiente paso es adaptar la distribución de probabilidad para $\mathbf{f}^*$ para que tenga en cuenta los ejemplos del conjunto de entrenamiento $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$. La distribución de probabilidad conjunta de los ejemplos de entrenamiento y de los de test es

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I} & k(\mathbf{X}, \mathbf{X}^*) \\ k(\mathbf{X}^*, \mathbf{X}) & k(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix}\right), \tag{2.51}$$

donde $k(\mathbf{X}^*, \mathbf{X})$ es la matriz de covarianzas entre los valores de la función en los ejemplos de test y los de entrenamiento y $k(\mathbf{X}, \mathbf{X}^*)$ es la de los puntos de entrenamiento y los de test.

El objetivo es obtener la distribución posterior de $\mathbf{f}^*$, que incorpore la información proporcionada por las observaciones. Para ello, calcularemos la distribución condicional $p(\mathbf{f}^* | \mathbf{X}, \mathbf{y}, \mathbf{X}^*)$. Esta distribución es una gaussiana multivariante cuya media es

$$k\left(\mathbf{X}^*, \mathbf{X}\right)\left[k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}\right]^{-1} \mathbf{y}, \tag{2.52}$$

y cuya matriz de covarianzas es

$$k\left(\mathbf{X}^*, \mathbf{X}^*\right) - k\left(\mathbf{X}^*, \mathbf{X}\right)\left[k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}\right]^{-1} k\left(\mathbf{X}, \mathbf{X}^*\right). \tag{2.53}$$

El valor que utilizaremos para predecir en test es la media de esta distribución.

Se puede ver que la predicción depende de σ_ϵ y del tipo y parámetros del kernel. Para encontrar una combinación de valores de estos hiperparámetros adecuada para nuestro problema se podría utilizar validación cruzada. El inconveniente de este procedimiento es que es es muy costoso computacionalmente. Por ello, el valor de estos hiperparámetros se suele determinar maximizando el logaritmo de la verosimilitud marginal

$$\log(p(\mathbf{y} \mid \mathbf{X}, \theta)) = -\frac{1}{2}\mathbf{y}^T \mathbf{K}_y^{-1} \mathbf{y} - \frac{1}{2} \mid \mathbf{K}_y \mid - \frac{N}{2} \log(2\pi). \tag{2.54}$$

En esta expresión, $\mathbf{K}_y = k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}$ es la matriz de covarianzas de la variable dependiente en los ejemplos de entrenamiento teniendo en cuenta el ruido. El término $\frac{1}{2}\mathbf{y}^T \mathbf{K}_y^{-1} \mathbf{y}$ es una medida de la calidad con la que el proceso gaussiano aproxima la distribución de la variable objetivo en el conjunto de entrenamiento. El término $\frac{1}{2} \mid \mathbf{K}_y \mid$, proporcional al determinante de la matriz de covarianzas $\mathbf{K}_y$, representa una penalización por complejidad. Finalmente, el término $\frac{N}{2}\log(2\pi)$ corresponde a la constante de normalización de la distribución de probabilidad.

En este trabajo fin de máster se utiliza una combinación de los siguientes kernels para abordar los problemas de regresión con procesos gaussianos:

- Ruido Blanco. Se trata de un kernel diagonal de la forma

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = c\delta(\mathbf{x}_n, \mathbf{x}_{n'}). \tag{2.55}$$

  Este kernel se utiliza para modelizar el ruido en las observaciones.

- Kernel exponencial cuadrático o RBF (*Radial Basis Function*):

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = e^{-\frac{||\mathbf{x}_n - \mathbf{x}_{n'}||^2}{2l^2}}. \tag{2.56}$$

  Es un kernel infinitamente diferenciable. Por ello, se utiliza para aproximar funciones de regresión suaves. Depende de un parámetro de escala, l, que determina cuan rápido se atenúa la influencia de un punto sobre otro a medida que aumenta la distancia entre ellos.

- Kernel Matérn:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}||\mathbf{x}_n - \mathbf{x}_{n'}||_2}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}||\mathbf{x}_n - \mathbf{x}_{n'}||_2}{l} \right). \tag{2.57}$$

  Este kernel depende de un parámetro ν, que determina la suavidad del proceso correspondiente. Cuanto menor es ν menos suaves son las aproximaciones. En concreto, las funciones aleatorias que son realizaciones de un proceso gaussiano con este kernel son diferenciables $(\nu - 1)$ veces. Valores habituales para este parámetro son $\nu = 3/2$ (funciones diferenciables una vez) o $\nu = 5/2$ (funciones diferenciables dos veces). El kernel también dependen de un parámetro de escala, l. Este parámetro tiene una interpretación análoga al correspondiente en un kernel RBF.

La combinación los distintos kernels es una buena idea para capturar las distintas características de una función. En la figura 2.4 se muestra este hecho. En la primera fila se ha representado los priors para un kernel RBF, mátern y la suma de ambos. En la segunda fila se representa la función posterior despues de incorporar la información del conjunto de entrenamiento, que esta formado por un conjunto de números aleatorios. Se puede observar, que el kernel RBF es demasiado suave y no es capaz de capturar la aleatoriedad de los datos. El kernel mátern, es capaz de capturar esta aletoriedad pero no la componente más suave de los datos. Una suma poderada de ambos, mostrada en la columna 3, es la que tiene el menor error

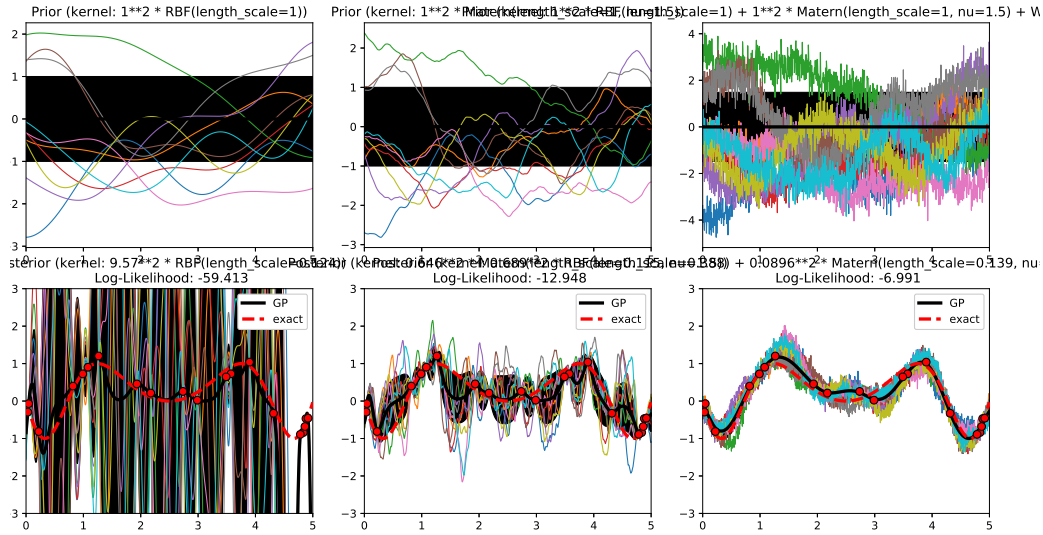


Figura 2.4: Combinación los distintos kernels. En la primera fila se ha representado los priors para un kernel RBF, mátern y la suma de ambos. En la segunda fila se representa la función posterior despues de incorporar la información del conjunto de entrenamiento.

Para determinar los hiperparámetros del problema mediante la maximización de la verosimilitud marginal se utilizan métodos numéricos de optimización. En la implementación utilizada, la de sklearn [29], el algoritmo utilizado es *L-BFGS-B* [25]. Este algoritmo de optimización pertenece a la familia de métodos

quasi-Newton, donde no se realiza el calculo explícito de la matriz hessiana o matriz de las derivadas segundas. En los métodos BFGS, la matriz hessiana se aproxima a partir del gradiente de la función de forma iterativa. Una diferencia entre L-BFGS-B y BFGS es que el primero utiliza una cantidad limitada de memoria y es capaz de resolver problemas con restricciones simples, del tipo $l_i \leq \theta_i < u_i$, donde θ_i es la i-ésima componente de vector de parámetros a optimizar.

2.5. Bosque Aleatorio

Un Bosque Aleatorio o *Random Forest* (RF) es un algoritmo de aprendizaje automático basado en combinar las predicciones de un conjunto de árboles aleatorios. Individualmente, cada uno de estos árboles de decisión puede que no sea capaz de realizar una buena predicción. Sin embargo, si los errores de las predicciones individuales son independientes, la predicción combinada debería ser mejor [4]. En regresión esta combinación se hace mediante el promedio de predicciones.

Un árbol de decisión es modelo de predicción basado dividir el problema original en una colección de problemas de predicción más simples, cada uno de ellos definido en una región distinta del espacio de atributos. Para ello, se realiza una partición de este espacio mediante una secuencia jerárquica de preguntas sobre los valores de dichos atributos. Cada uno de los nodos internos del árbol tiene asociado uno de estos tests. Las hojas del árbol se corresponden con las regiones en las que se ha dividido el espacio de atributos. Cada una de estas regiones está caracterizada por las respuestas a la secuencia de preguntas sobre los valores de los atributos en los nodos internos de la trayectoria que enlaza el nodo raíz con la correspondiente hoja.

Para predecir la etiqueta de un ejemplo caracterizado por su vector de atributos se aplica dicha secuencia de tests, de forma el ejemplo sea asignado a una hoja del árbol. El valor predicho se determina a partir del subconjunto de datos de entrenamiento que son asignados a dicha hoja por la secuencia de tests. Habitualmente se utilizan modelos simples. Por ejemplo, en clasificación, se suele utilizar para la predicción la etiqueta de la clase mayoritaria entre los ejemplos de entrenamiento asignados a tal hoja. En regresión es habitual utilizar como predicción el promedio de la variable dependiente para dichos ejemplos.

Sea $\mathscr{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$ el conjunto de entrenamiento utilizado para construir el árbol de decisión. Suponiendo que el árbol tiene J hojas, la partición del espacio de atributos es $\{R(l_j)\}_{j=1}^J$, donde $R(l_j)$ es la región asociada al nodo hoja l_j. Sea $\hat{y}(l_j)$ es la predicción del árbol en la región $R(l_j)$. La predicción del árbol de decisión para un ejemplo caracterizado por el vector de atributos $\mathbf{x}$ es

$$h(\mathbf{x}; \{R(l_j), \hat{y}(l_j)\}_{j=1}^J = \sum_{j=1}^J \hat{y}(l_j) \mathscr{I}\left[\mathbf{x} \in R(l_j)\right], \tag{2.58}$$

donde $\mathscr{I}[\cdot]$ es la función indicatriz ($\mathscr{I}[True] = 1, \mathscr{I}[False] = 0$). En regresión, la predicción en la hoja l_j se calcula como el promedio de los valores de la variable dependiente

$$\hat{y}(l_j) = \frac{\sum_{i=1}^N y_i \mathscr{I}\left[\mathbf{x}_n \in R(l_j)\right]}{\sum_{i=1}^N \mathscr{I}\left[\mathbf{x}_n \in R(l_j)\right]}. \tag{2.59}$$

La construcción de un árbol de decisión parte del nodo raíz, al que se asignan todos los ejemplos del conjunto de entrenamiento. Supongamos que el árbol ha sido generado hasta un cierto nivel. Para hacer crecer el árbol, se parte de un nodo hoja n, al que está asociada la región del espacio de atributos $R(n)$. Sea $\mathscr{D}(n) \subset \mathscr{D}$, el conjunto de ejemplos de entrenamiento en dicha región. Centrándonos en el problema de regresión, la predicción de este nodo es el promedio de los valores de la variable dependiente para los ejemplos de entrenamiento asignados a dicho nodo

$$\hat{y}(n) = \frac{1}{|\mathscr{D}(n)|} \sum_{i \in \mathscr{D}(n)} y_i$$

, donde $|\mathscr{D}(n)|$ es el tamaño de $\mathscr{D}(n)$, . El objetivo es convertir este nodo hoja en un nodo interno definiendo una partición del el espacio de atributos. En caso de que el árbol sea binario, como los considerados en este trabajo, la partición se define mediante un test Booleano sobre el vector de atributos. En concreto, los ejemplos para los cuales el test evalúa a *Verdadero* son asignados a n_L, el hijo izquierdo de n. Ejemplos para

los cuales el test evalúa a *Falso* son asignados a n_R, el hijo derecho de n. Esta partición segmenta el conjunto $\mathscr{D}(n) = \mathscr{D}(n_L) \cup \mathscr{D}(n_R), \quad \mathscr{D}(n_L) \cap \mathscr{D}(n_R) = \emptyset$ en dos subconjuntos disjuntos. En regresión, las predicciones en cada uno de estos nodos son

$$\hat{y}(n_L) = \frac{1}{|\mathscr{D}(n_L)|} \sum_{i \in \mathscr{D}(n_L)} y_i \tag{2.60}$$

$$\hat{y}(n_R) = \frac{1}{|\mathscr{D}(n_R)|} \sum_{i \in \mathscr{D}(n_R)} y_i, \tag{2.61}$$

respectivamente. La partición óptima será aquella que minimiza el error de predicción

$$ECM(n; n_L, n_R) = \sum_{j \in \{L,R\}} \frac{1}{\left|\mathscr{D}(n_j)\right|} \sum_{i \in \mathscr{D}(n_j)} \left(y_i - \hat{y}(n_j)\right)^2. \tag{2.62}$$

Los árboles que forman parte de un bosque aleatorio son generados por una modificación de este procedimiento, que involucra dos tipos de aleatorización:

- En lugar del conjunto de entrenamiento original se utiliza una muestra bootstrap de tamaño N obtenida por remuestreo con repetición a partir de $\mathscr{D}$ [13].
- En cada nodo interno se seleccionan $d \leq D$ atributos de manera aleatoria y se busca una partición óptima basada en realizar un test Booleano únicamente sobre los valores de las d variables seleccionadas.

Cada nodo interno de cada árbol aleatorio recibe distinta información, ya que no reciben el mismo conjunto de datos. Esta forma de entrenar garantiza que los árboles presenten cierta diversidad, de manera que, al promediar las predicciones de una cantidad elevada de arboles se reduzca la varianza. En [4], se demuestra que cuando el número de regresores utilizados tiende a infinito, el resultado converge al error mínimo que produciría un regresor del tipo base elegido, suponiendo nulo el término de varianza.

El pseudocódigo del bosque aleatorio es el siguiente:

Algoritmo 1 Bosque aleatorio

Entrada:

- Un conjunto de entrenamiento $\mathscr{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$
- El número de variables a considerar en las particiones internas del árbol, d
- El número de estimadores, M.

Salida: Bosque Aleatorio entrenado y la predicción $h(\mathbf{x}_n)$

BosqueAleatorio($\mathscr{D}$, d, M):
  $H \leftarrow \emptyset$;
para $i \in 1, ..., M$ **hacer**
  $\mathscr{D}^{[i]} \leftarrow$ muestra *bootstrap* de $\mathscr{D}$;
  $h_i \leftarrow$ ÁrbolAleatorio($\mathscr{D}^{[i]}, d$);
  $H \leftarrow H \cup h_i$;
fin para
$H(\mathbf{x}) \leftarrow \frac{1}{M} \sum_{i=1}^M h_i(\mathbf{x})$;

Adicionalmente a las mejoras que se obtienen en las predicciones, el bosque aleatorio tienen la ventaja de que es fácil de paralelizar.

Otra ventaja añadida es que permite realizar de manera natural un proceso de selección de variables. Este algoritmo, llamado *feature importance*, realiza un ranking de las variables conforme a su relevancia en la predicción [4]. La forma de determinar si una variable predictora es relevante para la predicción es si aumenta mucho el error de predicción al permutar los valores que toma dicha variable en los ejemplos

de entrenamiento. Una vez realizada una permutación en la variable seleccionada, se reentrena el bosque aleatorio. Esta permutación aleatoria elimina la dependencia efectiva entre la variable objetivo y la variable regresora, por lo que típicamente aumentará el error de predicción. Cuanto más aumente la métrica del error, más relevante es esa variable. De esta forma, se puede establecer un ranking de variables de acuerdo con su relevancia.

Este método de selección de variables se utilizará como método estadístico en el segundo grupo de experimentos del capítulo 3 para compararlo con una selección de variables basada en criterios meteorológicos.

2.6. Potenciación del Gradiente y Potenciación Extrema del Gradiente

La potenciación del gradiente es otro método de predicción de conjuntos que utiliza árboles como regresores base. Dispondremos de un conjunto de datos $\mathscr{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$ donde $\mathbf{x}_n$ es la variable de atributos de dimensión D. La variable objetivo y_n es aquella que se quiere predecir [15].

En métodos de regresión utilizados hasta ahora el objetivo ha sido encontrar la función $H \in \mathscr{F}$ que cumpla la condición

$$H^* = \arg\min_{H\in\mathscr{F}} \mathbb{E}_{Y,\mathbf{X}} L(y, H(\mathbf{x})). \tag{2.63}$$

donde $\mathscr{F}$ representa un espacio funcional. Debido a que el conjunto de entrenamiento $\mathscr{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$ es finito, el valor esperado de la función de pérdida $L(y, H(\mathbf{x}))$ se puede aproximar mediante el promedio muestral, de tal forma que el problema se transforma en

$$H^* = \arg\min_{H\in\mathscr{F}} \frac{1}{N}\sum_{n=1}^{N} L(y_n, H(\mathbf{x}_n)). \tag{2.64}$$

Como ya se dijo anteriormente, dos ejemplos clásicos de las funciones de pérdida en regresión son el error cuadrático medio y el error absoluto medio:

$$L_{ECM}(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2; \tag{2.65}$$

$$L_{EAM}(y, h(\mathbf{x})) = |y - h(\mathbf{x})|. \tag{2.66}$$

Como encontrar una función genérica que satisfaga esta condición es un reto muy ambicioso, lo que normalmente se hace es seleccionar una familia paramétrica $h(\mathbf{x};\theta)$ y se encuentran los parámetros óptimos tales que

$$\theta^* = \text{argmin}_\theta \mathbb{E}_{Y,\mathbf{X}}\left[L(y, h(\mathbf{x};\theta))\right] \tag{2.67}$$

En potenciación del gradiente, se supone que el modelo es dado por la serie:

$$h(\mathbf{x}; \{\beta_m, \mathbf{a}_m\}_{m=1}^M) = \sum_{m=1}^{M} \beta_m h(\mathbf{x}; \mathbf{a}_m) \tag{2.68}$$

Es decir, la función regresora viene dada por la suma de regresores, en nuestro caso serían árboles de regresión o *CART*, con parámetros $\mathbf{a}_m$ para los test de los nodos internos y predicciones en las hojas.

Los árboles de decisión utilizados en este algoritmo son distintos de los utilizados en el bosque aleatorio. Los árboles aquí utilizados no son aleatorios. Estos regresores realizan J divisiones binarias utilizando las variables explicativas $\mathbf{x}$. Cada regresor se puede expresar como

$$h(\mathbf{x}; \{b_j, R_j\}_{j=1}^J) = \sum_{j=1}^{J} b_j 1(\mathbf{x} \in R_j) \tag{2.69}$$

donde $\{R_j\}_{j=1}^J$) son las regiones disjuntas creadas mediante las divisiones binarias que minimizan el error cuadrático medio realizadas por árbol sobre el espacio de características al que pertenece $\mathbf{x}$. La función $1(\cdot)$ es la función indicatriz que representa la divisiones binarias realizadas en los J nodos. La predicción que realiza el árbol en regresión es la media de las etiquetas y_n de los ejemplos $x_n \in R_j$.

Como hemos asumido que los regresores H es una expansión aditiva y que el conjunto de entrenamiento tiene un tamaño finito N se puede demostrar que el problema que se quiere minimizar se transforma en

$$\{\beta_m, \mathbf{a}_m\} = \arg\min_{\{\beta'_m, \mathbf{a}'_m\}} \sum_{n=1}^{N} L\left(y_n, \sum_{n=1}^{N} \beta'_m h(\mathbf{x}_n; \mathbf{a}'_m)\right). \tag{2.70}$$

Sin embargo, este problema de optimización tiene una dimensionalidad demasiado alta. Por lo que se aplicará intentará llegar a una solución codiciosa mediante una aproximación por etapas y para cada uno los $m = 1, \ldots, M$ regresores se puede aproximar el problema dejando $m-1$ regresores constantes tal que:

$$\left(\beta_m, \mathbf{a}_m\right) = \arg\min_{\beta, \mathbf{a}} \sum_{n=1}^{N} L\left(y_n, H_{m-1}(\mathbf{x}_n) + \beta h(\mathbf{x}_n; \mathbf{a})\right). \tag{2.71}$$

Entonces el regresor $m-$ésimo tendrá la forma $H_m(\mathbf{x}) = H_{m-1}(\mathbf{x}) + \beta_m h(\mathbf{x}; \mathbf{a_m})$. Cada función $h(\mathbf{x}; \mathbf{a}_m)$ realiza una regresión sobre los residuos generados por los $m-1$ regresores anteriores. En este contexto, el parámetro β_m puede ser entendido como la tasa de aprendizaje del algoritmo de optimización de descenso por gradiente.

El pseudo código de este algoritmo es el siguiente

Algoritmo 2 Potenciación del gradiente para un problema de regresión con ECM como función de pérdida

Entrada: Un conjunto de entrenamiento $\mathscr{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, número de estimadores M y la tasa de aprendizaje β

Salida: Prediccion H_M

1: Sea $H_0(\mathbf{x}_n) = \bar{y}$
2: Potenciacion del gradiente($\mathscr{D}$, M, β):
3: **para** $m \in 1, ..., M$ **hacer**
4: $\quad \tilde{y}_n \leftarrow y_n - H_{m-1}(\mathbf{x}_n)$ para $n = 1, ..., N$
5: $\quad (\beta_m, \mathbf{a}_m) = \arg\min_{\mathbf{a}, \beta} \sum_{n=1}^{N} [\tilde{y}_n - \beta h(\mathbf{x}_m; \mathbf{a})]^2$
6: $\quad H_m(\cdot) \leftarrow H_{m-1}(\cdot) + \beta_m h(\cdot; \mathbf{a}_m)$
7: **fin para**

Según la descripción de potenciación del gradiente, este método tiene una compotente secuencial, por lo que no se puede paralelizar. El algoritmo de potenciación extrema del gradiente no aporta una idea nueva sobre el método de potenciación del gradiente. Sin embargo, aporta una serie de mejoras a la hora de realizar la regresión. Esas mejoras incluyen la utilización de submuestreo en el espacio de atributos. Las dos mejoras más relevantes que aporta la potenciación extrema del gradiente frente a la potenciación del gradiente son las siguientes:

- Término de regularización Ω en la función de pérdida de cada árbol m que depende del número de hojas J_m y de las predicciones realizadas por cada árbol $\hat{y}_{jm}$:

$$\Omega_m = \gamma J_m + \frac{\lambda}{2} \sum_{j=1}^{J} \hat{y}_{jm}^2 \tag{2.72}$$

- Optimización del gradiente mediante el método de Newton que utiliza una derivada de segundo orden aproximada

El resultado de implementar estos cambios es un modelo más robusto con un conjunto de entrenamiento grande, más rápido y paralelizable.

Para la potenciación del gradiente se ha utilizado la implementación de scikit-learn [29] y para la potenciación extrema del gradiente la del paquete *XGBoost* [8].

2.7. Comparación de los métodos.

En la introducción se describieron los modelos de predicción meteorológica. En estos modelos se resuelven numéricamente las ecuaciones que describen la dinámica de un sistema meteorológico. Entre sus salidas se encuentran las componentes del viento en varios niveles de la atmósfera para distintas localizaciones que están dispuestas en una rejilla. Estos modelos son útiles para la predicción a largo plazo, ya que captan muy bien fenómenos a gran escala, como por ejemplo, el alcance de un frente de bajas presiones. Sin embargo, a pequeña escala su error es generalmente mayor. En nuestro caso, esto supone una desventaja, ya que nuestro objetivo es predecir el viento en una localización que en general no coincide con ningún punto de la rejilla en el que el modelo meteorológico proporciona predicciones. Además, hay que añadir el inconveniente de que los cómputos necesarios para serlver númericamente las ecuaciones que describen la evolución de la atmósfera son costosos desde el punto de vista computacional.

En este capítulo se han descrito métodos de aprendizaje automático que pueden ser utilizados para abordar el problema de la predicción a menores escalas. A continuación se resume las ventajas e inconvenientes de algunos uno de ellos y cómo han sido utilizados en la literatura [38].

Las redes neuronales tienen ventaja de que son modelos muy flexibles. Se puede utilizar esta flexibilidad para reflejar las características funcionales del viento. Por ejemplo, la serie temporal de velocidades se puede descomponer en señales de distintas frecuencias. A modo de ilustración, en [41] y [46] se realiza una transformada de Fourier para obtener las distintas componentes de frecuencia de la serie temporal de viento y utilizar los coeficientes de Fourier como entrada de la red neuronal. También es posible utilizar redes neuronales recurrentes para la predicción del viento, [12, 42, 45].

Las redes neuronales tienen la ventaja de que son adaptativas y pueden entrenar *online*. Es decir, los pesos de la red pueden modificarse a medida que están disponibles los ejemplos de entrenamiento. Esto implica que se podría tener un modelo que incorpora la información más reciente en las predicciones. Sin embargo existen algunas desventajas. El proceso de entrenamiento podría ser costoso. Adicionalmente puede que sea difícil determinar los valores adecuados de los hiperparámetros y la arquitectura de la red. A pesar de eso, algunos autores utilizan redes neuronales profundas, [11, 18, 31], con un número de capas muy alto. Este tipo de modelos requiere gran cantidad de datos para realizar un buen entrenamiento y evitar el sobreajuste. El esfuerzo computacional se puede mitigar si se consideran otras arquitecturas para la red. Un ejemplo de esto son las redes de aprendizaje extremo (ELM) en las que se construye una red neuronal con una única capa oculta. El primer conjunto de pesos que conectan la capa de entrada con la capa oculta se inicializa de forma aleatoria. Los pesos entre las neuronas de la capa oculta y de la capa de salida se determinan mediante una regresión lineal. Este método también ha sido ampliamente utilizado para la predicción de viento [27, 30, 44, 49].

Un método bastante utilizado en la literatura para la predicción de viento son las máquinas de vectores soporte [24, 28, 47]. Por ejemplo, en [19] se hace un estudio para determinar cuál es el algoritmo mejor para la predicción de viento en un parque eólico en Osorio (Brasil). Los métodos comparados fueron un perceptrón multicapa, una máquina de vectores de soporte, sistemas de inferencia de lógica difusa, optimización por enjambre de partículas o algoritmos genéticos. El resultado fue que la máquina de vectores de soporte realizaba las predicciones más precisas. Las máquinas de vectores soporte se caracterizan por tener una gran capacidad de generalización, siendo muy robustas. Sin embargo, su entrenamiento requiere un esfuerzo computacional alto, sobre todo en la parte de selección de hiperparámetros.

Otro método que se ha utilizado para la predicción de viento a corto plazo es la regresión mediante procesos gaussianos [48]. Al igual que las SVM, estos sistemas de predicción son costosos de entrenar, por lo que no se pueden utilizar directamente en conjunto de tamaño medio o grande (> 1000 ejemplos). No obstante existen técnicas, como el uso de puntos de inducción (*inducing points*) para reducir su coste computacional [16]. Para que los modelos basados en procesos gaussianos sean eficaces es importante elegir kernels apropiados que reflejen la estructura de los datos a predecir. El kernel utilizado en [48] es la suma del kernel RBF y un kernel exponencial para la componente no diferenciable en la serie de valores de la velocidad del viento.

En lugar de utilizar un único modelo para predecir, se puede utilizar una combinación de modelos de distinto tipo. En [14] se propone usar una combinación de modelos en dos capas. La primera capa está formada por una red neuronal, una máquina de vectores soporte, un bosque aleatorio y un potenciación del gradiente. La segunda capa tiene como entrada las predicciones de los métodos anteriores y utiliza otra

vez uno de esos métodos para realizar una regresión por conjuntos. El método seleccionado es aquel que minimiza el error cuadrático medio. Otro artículo con una idea similar es [7], donde lo que se propone es un algoritmo compuesto por tres capas: la primera es un predicción por conjuntos de LSTM, en la segunda se le añade una SVM y se acaba con una EO (*extreme optimization*). Esta red recibe el nombre de EnsemLSTM.

Entre los métodos considerados en este trabajo se encuentran los de predicción por conjuntos: bosque aleatorio, potenciación del gradiente y potenciación extrema del gradiente. La potenciación por gradiente y la potenciación extrema son más díficiles de entrenar que el bosque aleatorio ya que es necesario determinar más hiperparámetros. El bosque aleatorio tiene la ventaja adicional de que los árboles aleatorios que forman parte del conjunto se pueden construir en paralelo. Esto hace que el bosque aleatorio sea el método más eficiente. En este trabajo fin de máster no se han tratado conjuntos de datos lo suficientemente grandes como para que la potenciación extrema del gradiente mejore a la potenciación del gradiente sin la aceleración estocástica, que suelen ser más potentes con un mayor número de datos. En [22] se propone el uso del algoritmo de predicción por conjuntos más utilizado, el bosque aleatorio. Este algoritmo tiene la ventaja de que tiene un número reducido de hiperparámetros. [39] muestran que los métodos por conjuntos (bosque aleatorio, potenciación del gradiente y potenciación extrema del gradiente) obtienen resultados comparables a las máquinas de vectores soporte en problemas de predicción de viento.

En el siguiente capítulo realizaremos una evaluación empírica de algunos de estos métodos en la predicción de viento en un parque eólico a partir de datos de reanálisis.

Capítulo 3

Evaluación empírica

Tras la revisión de modelos meteorológicos y la descripción de los métodos de aprendizaje automático realizadas en en los capítulos anteriores, este capítulo se centra en la evaluación empírica de dichos métodos en un problema de predicción de la intensidad del viento a partir de datos de reanálisis. Los métodos considerados son redes neuronales, máquinas de vectores soporte, bosques aleatorios, procesos gaussianos, potenciación del gradiente y potenciación extrema del gradiente.

La red neuronal se ha escogido porque es un método potente y muy flexible. Las máquinas de vectores soporte han demostrado ser un método robusto y que proporciona buenos resultados en este campo. Los métodos de predicción por conjuntos son ampliamente utilizados en este campo. Finalmente los procesos gaussianos son muy eficaces para la resolución de problemas de regresión.

3.1. Introducción y descripción general de los experimentos

El objetivo es determinar cuál es el mejor método para la predicción de viento en un parque eólico a partir de datos de reanálisis. Adicionalmente, se analizará cómo depende la precisión de los distintos sistemas de predicción con el tamaño del conjunto de entrenamiento. Finalmente, se realizará una reducción de la dimensión mediante la selección de las variables predictoras más importantes. Este último proceso es importante, ya que mejora la interpretabilidad del modelo.

Los modelos meteorológicos de predicción numérica tienen que discretizar el espacio. El problema de dicha discretización es que las soluciones entre puntos contiguos son interpolaciones suaves que no tienen en cuenta que el viento es un fenómeno muy local y dependiente de la topología del terreno. Para corregir las predicciones de los modelos númericos de predicción meteorológica, se pueden utilizar modelos de aprendizaje automático.

La localización del parque eólico no tiene porque coincidir con las coordenadas del espacio discretizado del modelo meteorológico. La gestión del parque eólico y de la electricidad generada es muy dependiente de las predicciones meteorológicas y necesitan una mayor predicción que no ofrecen las predicciones interpoladas del modelo meteorológico. Una vez se han descrito los métodos que se van a utilizar durante este trabajo, se procede a la evaluación empírica de los mismos.

El protocolo empírico adoptado es el siguiente: De manera previa a la aplicación de los métodos de aprendizaje automático, es necesario llevar a cabo un preprocesamiento de los datos. Una vez completado dicho preprocesamiento, se realiza un análisis exploratorio para determinar la configuarción y los valores de los hiperparámetros adecuados para cada método. Con la configuración y los valores de los hiperparámetros seleccionados se procede a realizar la evaluación empírica de los distintos métodos de aprendizaje automático. En concreto, se comparan los errores de cada método estimados por validación cruzada y en el conjunto de test.

Para el análisis subsiguiente se selecciona el sistema predictor más preciso. Una vez realizado este grupo de experimentos, se procede a realizar una selección de variables para el método seleccionado. De esta forma, no solo se obtendrá un sistema de predicción preciso para este problema sino que se entenderá cuales son las variables clave a la hora de realizar predicciones.

Para finalizar, se incluye una sección en la que se exponen las conclusiones alcanzadas gracias al análisis de los resultados de este estudio. En concreto, se obtienen excelentes resultados con árboles aleatorios. Las

variables más relevantes corresponden a las componentes de velocidad paralelas a la superficie terrestre para una altura intermedia (el nivel de 850 hPa, que corresponde aproximadamente a 1500 m), por encima de la capa límite. En menor medida, la temperatura superficial también aporta información relevante para la predicción. Las mejoras obtenidas al incluir el resto de variables son de menor importancia.

3.2. Descripción del conjunto de datos.

El problema abordado consiste en la predicción de la magnitud del viento en el parque eólico de Peñaparda (Salamanca) a partir de datos que proporciona el modelo de reanálisis ERA-Interim [10] en los 4 puntos de la rejilla considerada en dicho modelo más próximos al parque.

Se han manejado dos fuentes de datos separadas. El primer conjunto de datos contiene las mediciones realizadas en el propio parque de la velocidad y dirección del viento y la potencia eléctrica generada. Estos datos tienen una extensión temporal de ocho años (1995-2003). Para cada una de las variables consideradas se toman cuatro medidas diarias (6h, 12h, 18h y 24h). El número total de medidas consideradas asciende a 22815 datos.

Existen instantes para los que no se dispone de mediciones en el parque, por lo que es necesario realizar algún tipo de imputación en la fase de preprocesamiento. Esta ausencia de información puede ser debida a diversos problemas como obras en el parque o incidencias en el sistema de medición. El procedimiento que se ha utilizado para la imputación de los valores no disponibles, es promediar los datos cercanos correpondientes dos días de diferencia del mismo rango horario. Es decir, si hay un dato faltante a las 15h, se seleccionan los datos medidos a las 15h de los dos días anteriores y posteriores y se realiza el promedio. Esto se debe a que el viento es un fenómeno muy sensible al ciclo diurno. Promediando datos medidos en la misma hora, se respeta esta dependencia.

De entre la información disponible, tomaremos la velocidad del viento como la variable objetivo para los métodos de aprendizaje autómatico supervisado descritos en el capítulo 2.

El segundo conjunto de datos proviene del modelo de reanálisis ERA-Interim [10]. Los valores son proporcionados en los mismos instantes de tiempo que en la base de datos anterior. A continuación se explica en más detalle las características de los datos procedentes de reanálisis.

3.2.1. Descripción de la base de datos de reanálisis

Un modelo de reanálisis integra información que proviene de modelos meteorológicos con observaciones. El proceso de integración es conocido como *asimilación de datos*. La motivación de utilizar este proceso de integración es, por una parte, paliar las deficiencias de los modelos numéricos de predicción y, por otra, completar las observaciones, que pueden que, en general, no están disponibles en una rejilla espacio-temporal regular y suelen contener errores (por ejemplo por limitaciones o defecto de los instrumentos de medida).

El proceso de asimilación es iterativo. Se parte de las predicciones numéricas de un modelo, en este caso el modelo IFS versión *Cy31r2* del Centro Europeo de previsiones Meteorológicas a Plazo Medio. Las predicciones de estos modelos son a menudo inestables y poco fiables a largo plazo, dada la dificultad de establecer con precisión las condiciones iniciales para la simulación en todos y cada uno de los puntos de la rejilla espacial considerada. Para mejorar la calidad de las estimaciones, estas predicciones son modificadas a la luz de las observaciones realizadas, así como de predicciones anteriores que han sido consideradas como válidas. El objetivo es disponer de estimaciones estables y fiables en todos los puntos de la rejilla. De esta manera, se obtienen valores que son comparables en el tiempo y que permiten realizar estudios a gran escala y largo plazo. En este trabajo, el objetivo es determinar si estos datos de reanálisis también son útiles a pequeña escala para el problema de predecir la magnitud del viento en un punto que no está incluido en la rejilla original. De hecho, los datos originales pasan por dos filtros de calidad: tienen que ser coherentes con lo que ocurre en la realidad y los parámetros que se optimizan con esos datos tienen que ser compatibles con la física conocida. Es decir, un día de verano no puede hacer una temperatura de -10ºC si en todos los termómetros de alrededor se han tomado unas medidas mayores y si hay un día soleado de tiempo estable, no puede haber medidas de viento huracanado. Este control de calidad es necesario porque los sensores se estropean y/o tienen sesgos. Por ejemplo, hay mediciones de la temperatura del nivel del mar (SST) desde

el casco de los barcos. Pero estas medidas tienen que corregirse, ya que cuando el barco surca el mar, el casco se calienta provocando una alteración en la medida. Estos sesgos también afectan a satélites u otros sensores.

ERA-Interim [10] proporciona datos en una rejilla cuya resolución espacial de 0.125º, que corresponde aproximadamente 15 km. En cada uno de los puntos de la rejilla se conocen las variables de presión, temperatura, viento zonal, meridional y vertical en distintos niveles atmosféricos. El viento zonal es la componente del viento paralela a la superficie terrestre a lo largo un paralelo, en dirección Oeste a Este. La componente meridional del viento es la componente de la velocidad horizontal a lo largo de un meridiano, de Sur a Norte. Se proporcionan valores a 10 m. por encima de la superficie terrestre, a 850 hPa (a una altura de, aproximadamente, 1500 m) y 500 hPa (aproximadamente, 5500 m de altura). Estos niveles proporcionan información complementaria del estado de la atmósfera: el nivel superficial está asociado a efectos característicos de la capa límite. Los valores proporcionados en el nivel más elevado corresponden a una situación de la atmósfera en la que no hay efectos de turbulencia asociados a la superficie. En concreto, los valores estimados deberían aproximarse a los del viento geostrófico, que es el que se obtendría por equilibrio entre la fuerza de Coriolis y el gradiente de presión. Los valores de velocidad y temperatura en el nivel medio, de 850 hPa, proporciona información acerca del estado de la atmósfera a una escala más local. Los identificadores y una somera descripción de las variables regresoras se muestran en la tabla 3.1. Para la predicción se utilizan los datos de los cuatro puntos de la rejilla de reanálisis más próximos al parque eólico.

Nombre de la variable	Descripción
skt	Temperatura en la superficie
sp	Presión atmósferica en la superficie
u_{10}	Componente zonal de viento a 10 metros de altura
v_{10}	Componente meridional de viento a 10 metros de altura
temp1	Temperatura del aire a 500 hPa
up1	Componente zonal de viento a 500 hPa
vp1	Componente meridional de viento a 500 hPa
wp1	Componente vértical de viento a 500 hPa
temp2	Temperatura del aire a 850 hPa
up2	Componente zonal de viento a 850 hPa
vp2	Componente meridional de viento a 850 hPa
wp2	Componente vertical de viento a 850 hPa

Cuadro 3.1: Variables predictoras consideradas en cada punto del modelo de reanálisis ERA-Interim [10].

A continuación se procederá a realizar una descripción más amplia las variables que conforman este conjunto de datos procedente de reanálisis.

3.2.2. Descripción de las variables de reanálisis.

La base de datos de reanálisis está compuesta por las variables mostradas en la tabla 3.1. En esta sección se explicará de forma más detallada cada una de estas variables. Las variables se dividen en tres niveles. Estos niveles son el de 10 metros de altura, el de 850 hPa y el de 500hPa.

El primer nivel es el nivel superficial. En este nivel se proporcionan cuatro variables: la temperatura superficial, la presión en superficie y las componentes zonal y meridional del viento a 10 metros de altura. Este es el único punto donde se mide la presión en superficie. Como ya se ha visto durante la explicación de los modelos meteorológicos, el viento se crea mediante las diferencias espaciales de presión. Debido a la interpolación en los datos de reanálisis, esta diferencia no es especialmente significativa. Para que lo fuera serían necesarios más puntos de reanálisis. Sin embargo, la temperatura en superficie, puede ser relevante, ya que con temperaturas en superficie elevadas suele haber convección. La convección es el desplazamiento de masas de aire debido a la diferencia de temperatura. Si el aire en superficie se calienta mucho, asciende porque es más ligero.

También se dan sólo las componentes zonal y meridional del viento y no la vertical y medidas a 10 metros de altura. Esto se debe a que el viento en la superficie, a 0 metros de altura, es nulo debido al rozamiento.

Por tanto, existe la necesidad de elevar los anemómetros para obtener una medida fiable. Por eso, se miden a 10 metros de la superficie. La velocidad vertical del viento en este nivel es muy pequeña y por eso no se mide.

El segundo y el tercer nivel tienen la misma estructura. Aquí no se proporciona la variable de presión, debido a que se selecciona la altura para una presión determinada. Las variables que hay en cada nivel son la temperatura del aire a esa altura y las tres componentes del viento (zonal, meridional y vertical). Aquí sí que se incluye la velocidad vertical del viento, porque a pesar, de ser menor que las componentes zonal y meridional, es más significativa que en el caso de la superficie.

Cada nivel tiene una importancia distinta en la predicción. En el experimento de selección de variables, se ilustrará que no todos los niveles contribuyen de la misma manera a la predicción. El nivel más relevante será el nivel intermedio, el de 850 hPa. Tampoco todas las variables serán igualmente informativas. Serán más relevantes las del viento y en menor medida, la temperatura en superficie.

3.3. Experimentos

El objetivo de este trabajo es obtener un buen modelo de predicción con la cantidad mínima de datos. Para ello, se han realizado dos grupos de experimentos. En el primer grupo, se comparan seis algoritmos de aprendizaje supervisado: redes neuronales, regresión con máquinas de vectores soporte, procesos gaussianos, bosque aleatorio o Random Forest (RF), Potenciación del Gradiente o Gradient Boosting (GB) y Potenciación Extrema del Gradiente o Extreme Gradient Boosting (XGB). Los experimentos se han realizado en una máquina con un procesador Intel®Core™i7-3612QM con una frecuencia de 2.10GHz. En todos los casos, las variables predictoras han sido estandarizadas,

$$\mathbf{z}_n = \frac{\mathbf{x}_n - \bar{x}}{\sigma_x}. \tag{3.1}$$

donde $\bar{x}$ es la media y σ_x la desviación típica. Ambos estadísticos han sido estimadas en el conjunto de entrenamiento. Se ha comprobado, no obstante, que el uso otros tipos de normalización habituales (por ejemplo, usar la mediana para centrar los datos y el intervalo intercuartílico para escalarlos) no tiene una gran influencia en la precisión que se obtiene con los distintos métodos. Las métricas utilizadas para cuantificar la precisión de una hipótesis $h(\mathbf{x})$ son la raíz del error cuadrático medio:

$$RECM = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - h(\mathbf{x}_n))^2}, \tag{3.2}$$

y el error absoluto medio

$$EAM = \frac{1}{N}\sum_{n=1}^{N}|y_n - h(\mathbf{x}_n)|. \tag{3.3}$$

Se han escogido estas dos métricas debido a que son muy utilizadas en regresión. Además, la función que minimiza el RECM es la media y la que minimiza el EAM es la mediana. La media es una medida de centralidad ampliamente utilizada aunque es subceptible a valores extremos. La mediana es más robusta en este sentido pero tiene un mayor coste computacional.

La elección del modelo y de los hiperparámetros se realiza mediante validación cruzada con 10 subconjuntos. En los experimentos realizados se proporciona tanto el error de validación cruzada como el de test. De esta manera, se puede comprobar que, aunque los valores de estas dos estimaciones son distintos, las ordenaciones de los modelos de acuerdo con el error de validación cruzada y con el error de test son similares. Por lo tanto, en este problema, validación cruzada permite identificar los modelos de aprendizaje automático que son más precisos en el conjunto de test.

Una vez seleccionado en el experimento anterior el método de aprendizaje automático más preciso, se procederá a determinar qué variables de entre las disponibles en el estudio de reanálisis son más relevantes y cuales pueden eliminarse debido a que proporcionan información redundante o que no son relevates. Se comprueba que se obtienen resultados similares cuando, en lugar de utilizar datos de los 4 puntos más próximos al parque eólico, se utiliza solo uno de ellos. En concreto, el deterioro de la calidad de las predicciones que conlleva utilizar solo uno de los puntos de la rejilla espacial en la que disponen de datos de reanálisis es

muy leve. También se hará incluido una análisis de la importancia de las variables predictoras por niveles de la atmósfera. y se exploran las combinaciones de variables que proporcionan mejores resultados [28].

3.3.1. Dependencia del error con el tamaño del conjunto de entrenamiento.

El objetivo de este experimento es determinar cual es el mejor modelo de predicción con el menor número de datos. El número total de datos es 22815. Pero realizar experimentos con un conjunto de entrenamiento grande es inabarcable computacionalmente con la máquina utilizada.

Por esta razón, se han considerado cuatro proporciones distintas de entrenamiento-test. Se espera que, aunque el tamaño del conjunto de entrenamiento sea reducido, las conclusiones (cuál es el mejor predictor o cuales son las variables importantes) sigan siendo válidas. En concreto se han considerado los conjuntos de entrenamiento con el 5 %, el 10 %, el 15 % y el 20 % de los ejemplos disponibles. Como la división de entrenamiento-test se ha hecho de forma aleatoria, los resultados pueden variar, sobre todo en el caso en el que el conjunto de entrenamiento es pequeño. La solución encontrada ha sido repetir la división aleatoria 50 veces y realizar el promedio para reducir las fluctuaciones estadísticas. Además del promedio, se proporcionan también ± 2 desviaciones estándar. Para que los conjuntos de entrenamiento y test sean iguales para todos los métodos, la semilla utilizada ha sido el número de repetición del experimento.

Se proporcionan medidas del error para validación cruzada y en test para los cuatro subconjuntos con dos métricas distintas: el error absoluto medio y la raíz del error cuadrático medio.

Configuración del aprendizaje y selección de hiperparámetros.

La configuración del método de aprendizaje automático y los valores de los hiperparámetros considerados han sido determinados mediante un análisis exploratorio. En el caso de hiperparámetros continuos, se exploran intervalos suficientemente grandes de modo que el proceso de validación cruzada no seleccione valores en la frontera.

A continuación se explica para cada método de aprendizaje las configuraciones y los valores de hiperparámetros considerados. Finalmente, se indica la configuración óptima seleccionada y se lleva a cabo un análisis de la calidad de las predicciones.

Redes neuronales.

El primer algoritmo considerado es un perceptrón multicapa dada por la siguiente clase de *scikit-learn* de Python:

```
class sklearn.neural_network.MLPRegressor(hidden_layer_sizes,
activation, solver='adam', alpha)
```

Se han considerado valores diferentes de los fijados por omisión tres elementos de la configuración: la función de activación, el número de neuronas de la capa oculta y el hiperparámetro α que determina la importancia de regularización L_2, respecto al término de error.

Para la activación se ha considerado tanto las unidades ReLU, como la función tangente hiperbólica. El número de neuronas de la capa oculta (*hidden_layer_sizes*) se ha variado entre 70 y 190 neuronas. Para el hiperparámetro α se ha explorado una rejilla logarítmica con 5 valores entre 10^{-5} y 10^{-1}.

En cuanto al optimizador, se realizaron varias pruebas con los distintos optimizadores que tiene *scikit-learn* implementados. El optimizador que mejor funcionaba era adam (*Adaptive Moment Estimation*) [20]. Es un optimizador que realiza un descenso por gradiente teniendo en cuenta la inercia del error durante el descenso por gradiente. En este sentido utiliza los promedios del primer y del segundo momento del gradiente para actualizar los parámetros. Los hiperparámetros relacionados están asociados a los pesos de los momentos. Los parámetros que vienen por defecto suelen tener una buena ejecución según los experimentos realizados en el artículo donde está definido el método de optimización.

Los hiperparámetros seleccionados y el error de validación cruzada para cada tamaño del conjunto de entrenamiento son los siguientes:

En cuanto a la calidad de las predicciones realizadas por las redes neuronales, se puede observar en la figura 3.1 que tampoco es capaz de reproducir las colas de la función. Estos valores son las predicciones de test con una proporción del 20 % en el conjunto de entrenamiento. La arquitectura ha sido seleccionada

Tamaño del conjunto entrenamiento	'alpha'	activation	hidden_layer_sizes	$RECM_{10-cv}$
5 %	1.47e-05	ReLU	170	2.853 ± 0.087
10 %	0.046	logistic	130	2.689 ± 0.047
15 %	0.046	logistic	170	2.65 ± 0.039
20 %	0.003	logistic	100	2.624 ± 0.035

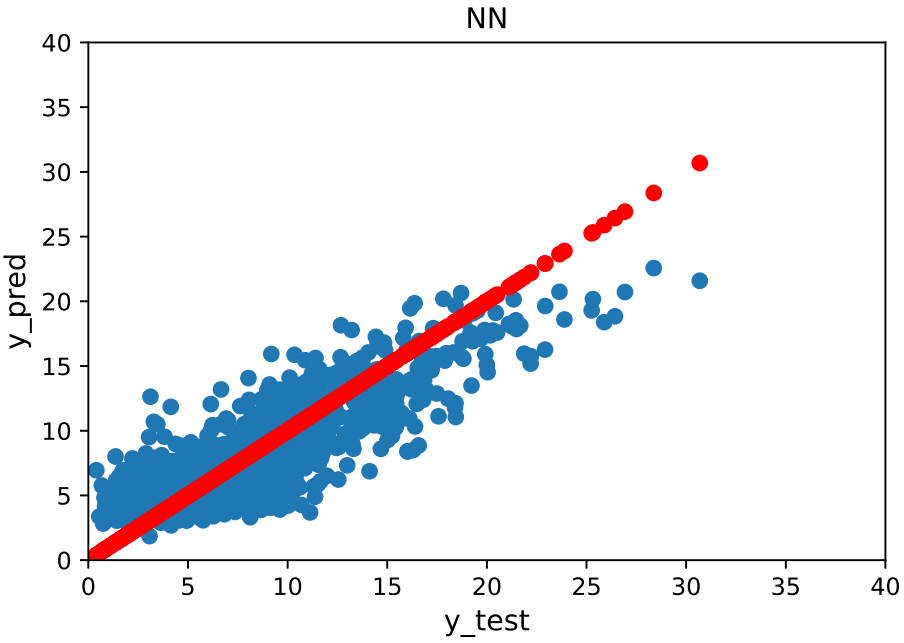


Figura 3.1: Predicciones realizadas por una red neuronal frente a los valores reales.

por validación cruzada en el espacio de parámetros definido anteriormente. En cuanto a la estructura del error (figura 3.2), se puede observar que la red neuronal no es capaz de reproducir las colas de la función. La predicción es velocidades del viento bajas es mucho mayor mientras que en velocidades altas es mucho menor. El error aquí representado se ha calculado como

$$\text{Error} = \mathbf{y} - \text{h}(\mathbf{x}). \tag{3.4}$$

A pesar de no ser capaz de reproducir los extremos de la distribución de probabilidad, sí que es capaz de predecir los valores intermedios, ya que el histograma se encuentra centrado en 0.

Máquinas de vectores soporte, SVR

Una máquina de vectores soporte es un método de aprendizaje automático que convierte el problema en lineal mediante transformaciones del espacio original a espacios de dimensión superior mediante un kernel $K(\mathbf{x}_n, \mathbf{x}_{n'})$. En clasificación, surge para separar el espacio entre cada clase y la frontera de clasificación. Sin embargo en regresión, se minimiza el error eliminando la penalización en torno a un intervalo $\pm\epsilon$. Para que no caiga en el sobreajuste, se admite cierto error en los datos, que viene marcado por el hiperparámetro C.

En el caso de las SVR se ha escogido un kernel RBF

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = e^{-\gamma ||\mathbf{x}_n - \mathbf{x}_{n'}||^2}. \tag{3.5}$$

La rejilla de hiperparámetros explorada el siguiente:

- C: $[2^{-10}, 2^{-8}, 2^{-6}, 2^{-4}, 2^{-2}, 2^0, 2^2, 2^4, 2^6, 2^8, 2^{10}, 2^{12}]$.
- ϵ: $[(2^{-10}, 2^{-8}, 2^{-6}, 2^{-4}, 2^{-2})\sigma_y]$
- γ: $[(\frac{1}{4D})^{-2}, (\frac{1}{4D})^{-1}, (\frac{1}{4D})^0, (\frac{1}{4D})^1, (\frac{1}{4D})^2, (\frac{1}{4D})^3]$,

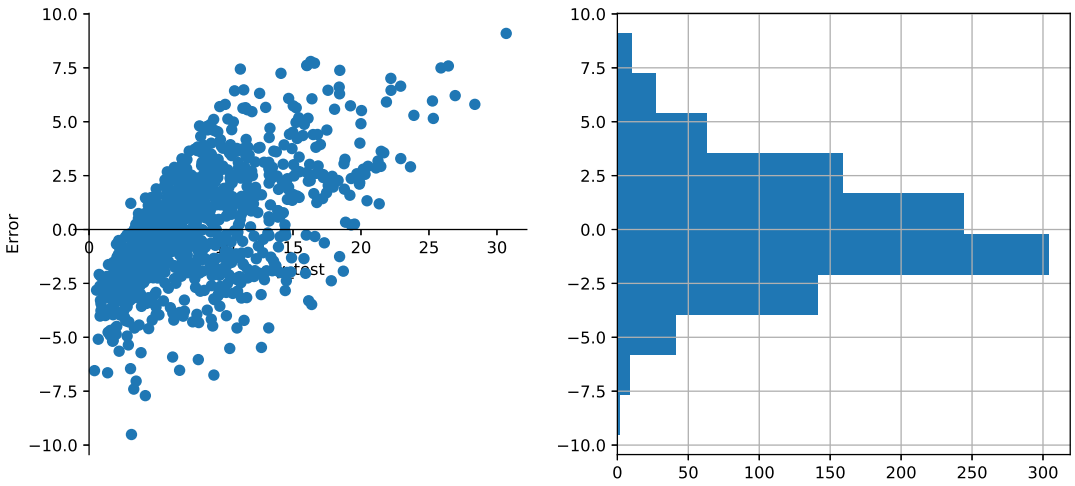


Figura 3.2: Representación de los errores ,$\mathbf{y} - h(\mathbf{x})$, generados por una red neuronal. En la figura de la izquierda se han representado frente a la velocidad del viento real y en la figura de la derecha se representa el histograma de los errores.

donde D es el número de variables predictoras y σ_y es la desviación estándar de la variable objetivo cuyo valor estimado en el conjunto de entrenamiento es 4.84 ± 0.06 m/s.

Debido a que para cada tamaño del subconjunto de entrenamiento el mínimo puede estar en una distinta combinación de hiperparámetros, se ha seleccionado para cada tamaño. La selección de hiperparámetros se hace mediante una validación cruzada de 10 hojas, donde la mejor combinación para cada tamaño del conjunto de entrenamiento es la siguiente: En esta tabla, el tamaño del conjunto de entrenamiento se

Tamaño del conjunto entrenamiento	C	ϵ	γ	$RECM_{10-cv}$
5%	16.0	1.20	0.02	2.837 ± 0.071
10%	4.0	1.20	0.02	2.738 ± 0.048
15%	16.0	1.20	0.02	2.689 ± 0.037
20%	16.0	1.20	0.02	2.661 ± 0.037

expresa como el porcentaje de ejemplos del conjunto completo del que disponemos.

En cuanto a la calidad de las predicciones, se ha representado el valor de la velocidad del viento predicho frente al valor real en la figura 3.3. Los datos mostrados son para una SVR con una proporción del conjunto de entrenamiento del 20% del tamaño del conjunto de datos original y los hiperparámetros han sido seleccionados por validación cruzada del espacio de parámetros definidos anteriormente. En esta ocasión tampoco es capaz de reproducir viento por debajo de los 2 m/s mientras que los vientos por encima de los 25 m/s tampoco es capaz de predecirlos con precisión. En cuanto a la estructura del error, representado en la figura 3.4, se puede observar el mismo comportamiento que en la figura 3.3 y que en las figuras 3.9 y 3.2. Sin embargo, la disrtibución de los errores aquí tiene unas colas más pesadas, ya que tiene incluso errores de 20 m/s cuando en realidad la velocidad del viento es 0.

Procesos gaussianos.

Un proceso gaussiano es el proceso estocástico en el que cualquier conjunto finito de variables aleatorias tiene una distribución normal. Este proceso se caracteriza por la función media y la función de covarianza o kernel:

$$h(\mathbf{x}_n) = \mathbb{GP}(m(\mathbf{x}_n), k(\mathbf{x}_n, \mathbf{x}_{n'})). \tag{3.6}$$

Para la evaluación empírica de los procesos gaussianos se ha utilizado la clase de *scikit-learn*

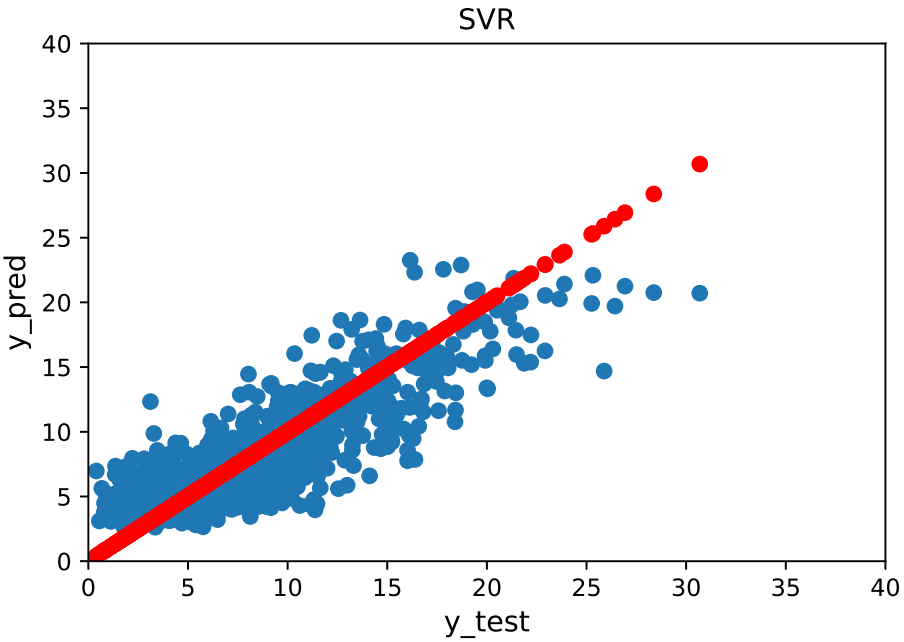


Figura 3.3: Predicciones realizadas por una máquina de vectores soporte frente a los valores reales.

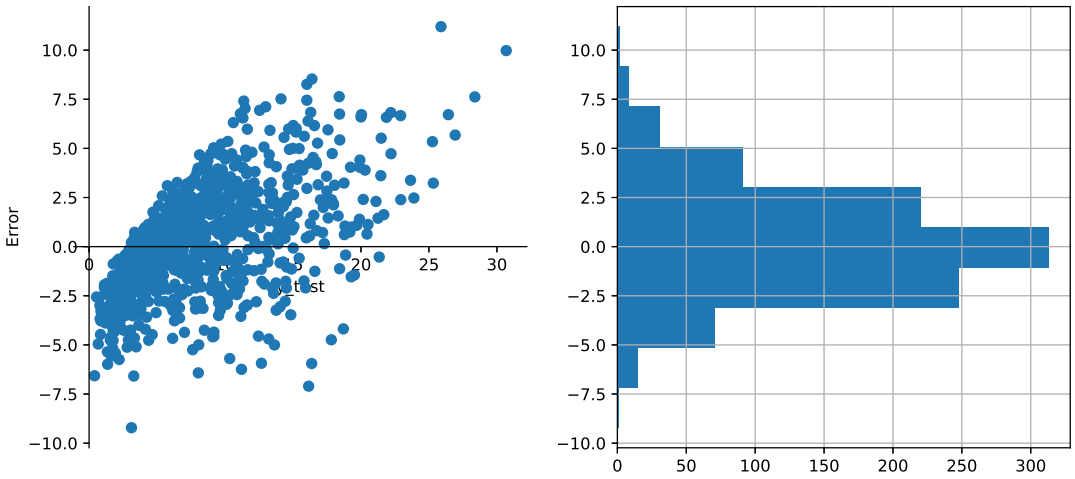


Figura 3.4: Representación de los errores ,$\mathbf{y} - h(\mathbf{x})$, generados por una máquina de vectores de soporte. En la figura de la izquierda se han representado frente a la velocidad del viento real y en la figura de la derecha se representa el histograma de los errores.

```
class sklearn.gaussian_process.GaussianProcessRegressor(kernel, normalize_y=True,
n_restarts_optimizer= 5)
```

Con el objeto de minimizar la dependencia de los resultados con los valores iniciales, se utiliza el parámetro n_restart_optimizer, que reinicia el optimizador un número de veces con distintas condiciones iniciales, con lo cual no es necesario.

El kernel utilizado ha sido una suma de los distintos kernels que tiene la biblioteca de *scikit-learn*:

- White. Este kernel permite modelizar el ruido. Su ecuación es la siguiente:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = c. \tag{3.7}$$

- RBF. Este kernel ha sido seleccionado para capturar las variaciones suaves de dichas series temporales. Su ecuación es la siguiente:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = e^{-\frac{||\mathbf{x}_n - \mathbf{x}_{n'}||^2}{2l^2}}. \tag{3.8}$$

- Matérn 3/2 y Matérn 5/2. Estos kernels han sido escogidos debido a que se espera que capturen las características de intermitencia y no diferenciabilidad de las series temporales de intensidad del viento. Su ecuación es la siguiente:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}||\mathbf{x}_n - \mathbf{x}_{n'}||_2}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}||\mathbf{x}_n - \mathbf{x}_{n'}||_2}{l} \right). \tag{3.9}$$

Debido a que se ha utilizado la suma de varios tipos de kernels, el espacio de hiperparámetros es muy grande. En lugar de seleccionar los valores de dichos hiperparámetros por validación cruzada, se han determinado maximizando la verosimilitud logarítmica marginal para cada conjunto de datos de entrenamiento considerado. No obstante, por completitud, en los resultados se proporciona el error de validación cruzada junto con el de test.

Sin embargo, por completitud y por mantener una estructura, el error de validación cruzada en 10 hojas para cada uno de los tamaños del conjunto de entrenamiento es el siguiente:

Tamaño del conjunto entrenamiento	$RECM_{10-cv}$
5%	2.744 ± 0.064
10%	2.675 ± 0.047
15%	2.639 ± 0.033
20%	2.619 ± 0.033

A continuación se muestran las predicciones en la figura 3.5. Los valores son los correspondientes al conjunto de test para un modelo entrenado con el 20% de los datos. En la figura se puede observar que el

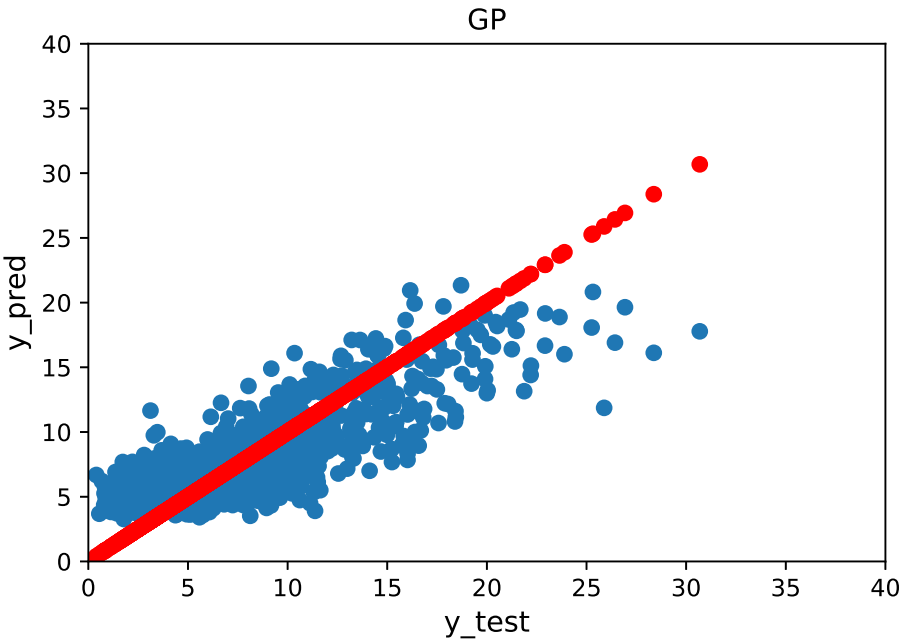


Figura 3.5: Predicciones realizadas por un proceso gaussiano frente a los valores reales.

proceso gaussiano no es capaz de reproducir las colas de una distribución de error no gaussiana. Esto es normal y es debido a que no se cumplen el axioma del proceso gaussiano de que los datos deben tener una distribución normal.

Sin embargo, en cuanto a la estructura del error, se puede observar en la figura 3.6 que efectivamente no es capaz de reproducir las colas correctamente. Sin embargo, la distribución de los errores es bastante

simétrica. Sin embargo, la contribución de las cola inferior viene dada por las velocidades bajas del viento que es incapaz de predecir y la cola superior es debido a las altas velocidades que tampoco es capaz de predecir correctamente.

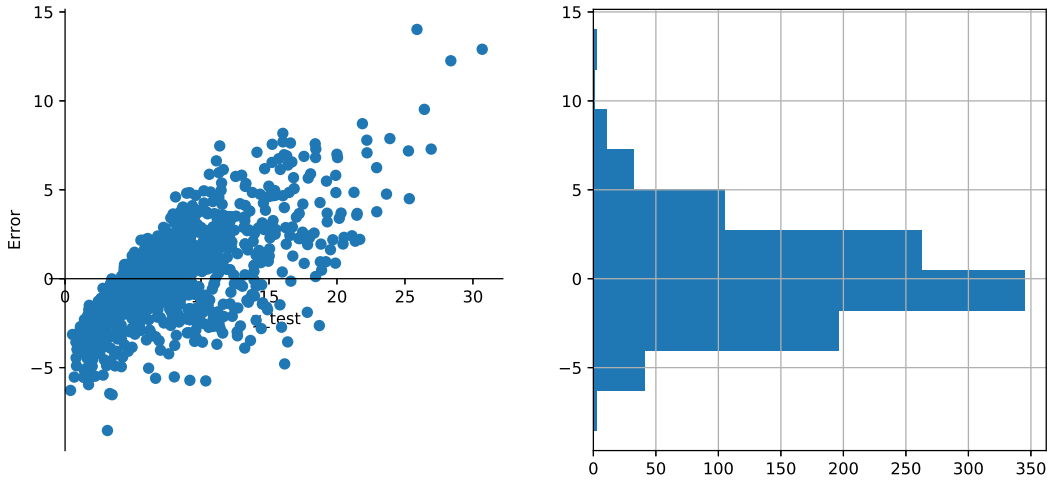


Figura 3.6: Representación de los errores ,$\mathbf{y} - h(\mathbf{x})$, generados por un proceso gaussiano. En la figura de la izquierda se han representado frente a la velocidad del viento real y en la figura de la derecha se representa el histograma de los errores.

Árboles aleatorios.

El siguiente algoritmo considerado es el bosque aleatorio (*Random Forest* o RF). Una de las ventajas de este algoritmo es que tiene pocos hiperpámetros para elegir y que proporciona buenas predicciones para un rango amplio de los valores de dichos hiperparámetros.

Las conclusiones son similares para conjuntos de entrenamiento de menor tamaño. En el implementación realizada se utiliza la clase de *scikit-learn* [29] de Python

```
class sklearn.ensemble.RandomForestRegressor(n_estimators, max_features)
```

donde *n_estimators* es el número de árboles aleatorios del conjunto, y *max_features* es el número de variables que se consideran en cada uno de los tests asociados a los nodos internos de los árboles de regresión que componen el conjunto.

El resto de parámetros de esta clase de *scikit-learn* son los relacionados con la arquitectura interna de cada árbol aleatorio considerado. Estos parámetros no se han modificado.

Los resultados se muestran en la figura 3.7. En ella se muestran errores de validación cruzada. La división en los conjuntos de entrenamiento-test se ha realizado 50 veces de forma aleatoria y se han promediado los resultados. Las curvas en dicha figura trazan la dependencia del error de predicción con el tamaño del bosque aleatorio. Cada una de las curvas corresponde a un valor distinto del parámetro *max_features*: 'sqrt', valor para el que *max_features* = $\sqrt{D}$, 'log2', para *max_features*= $\log_2 D$ y 'auto', configuración en la que selecciona D, el número total de variables predictoras. El número de arboles varía entre 1 y 500. En este apartado sólo se ha tenido en cuenta el RECM ya que se trata de un análisis exploratorio. Las curvas del EAM muestran un comportamiento similar.

De los resultados obtenidos se observa que tanto la opción 'sqrt' como 'log2' obtienen resultados muy similares, mejores que considerar todas las variables. En los experimentos se utilizará 'sqrt'. Como es de esperar, el error de predicción de los bosques aleatorios generados disminuye de manera monótona a medida que se incrementa el número de regresores. De las curvas que se muestran en la figura 3.7 se puede considerar que se ha alcanzado la convergencia para 500 regresores. Este es valor elegido en los experimentos subsiguientes.

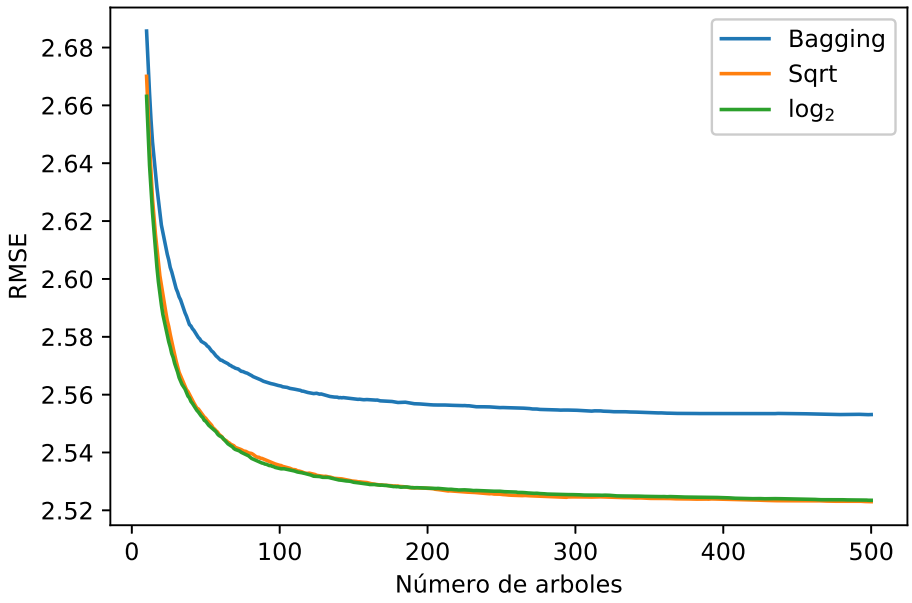


Figura 3.7: Error de validación cruzada en función del tamaño del bosque aleatorio para las tres opciones de *max_features*.

El error en validación cruzada con 10 hojas para cada una de las proporciones del conjunto de entrenamiento ha sido:

Tamaño del conjunto entrenamiento	$RECM_{10-cv}$
5%	2.744 ± 0.064
10%	2.675 ± 0.047
15%	2.639 ± 0.033
20%	2.619 ± 0.033

Para estudiar la calidad de las predicciones, se ha seleccionado una proporción del conjunto de entrenamiento del 20%. En la figura 3.8 se han representado las predicciones frente a los valores reales de velocidad del viento. En esta figura se muestra que el bosque aleatorio no es capaz de reproducir adecuadamente ninguna de las dos colas de la distribución de probabilidad de valores del viento. No es capaz de predecir valores muy bajos ni muy altos. Esto indica que no es capaz ni de predecir valores del viento incapaces de mover las aspas de los aerogeneradores ni de predecir valores que romperían las aspas del aerogenerador. Sin embargo, los valores intermedios los reproduce mucho mejor.

Si observamos como se distribuye el error (figura 3.9), se comprueba esta hipótesis. La primera figura representa el error en función de la velocidad del viento predicha, mientras que la segunda representa el histograma de errores.

En este caso se vuelve a observar como con velocidades pequeñas de velocidad del viento, el modelo predice valores $h(\mathbf{x})$ mayores, por lo que el error es negativo y en velocidades altas, predice valores $h(\mathbf{x})$ menores, con lo que el error es negativo. Sin embargo el histograma de los errores está centrado en 0 y la distribución del error no es simétrica: se observa una frecuencia mayor de errores por exceso que por defecto. Por tanto, la distribución que se quiere predecir está caracterizada por unas colas muy pesadas.

Potenciación del Gradiente (*Gradient Boosting*). Para el algoritmo de Potenciación del Gradiente, se utilizará la clase proporcionada por la librería *scikit-learn* [29]:

```
class sklearn.ensemble.GradientBoostingRegressor(learning_rate, n_estimators,
max_features)
```

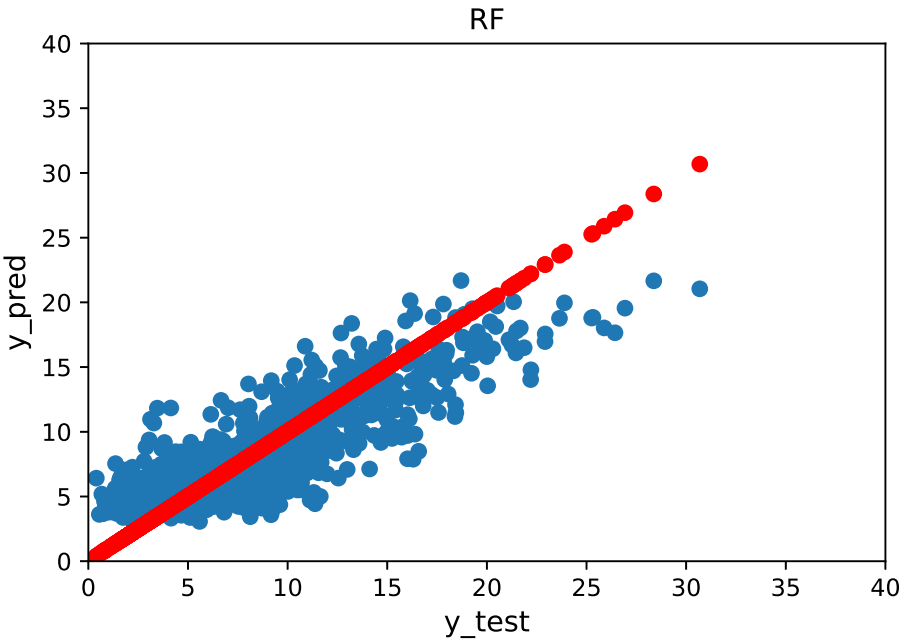


Figura 3.8: Predicciones realizadas por un bosque aleatorio frente a los valores reales de la velocidad del viento para un tamaño del conjunto de entrenamiento del 20 % del tamaño del conjunto de datos.

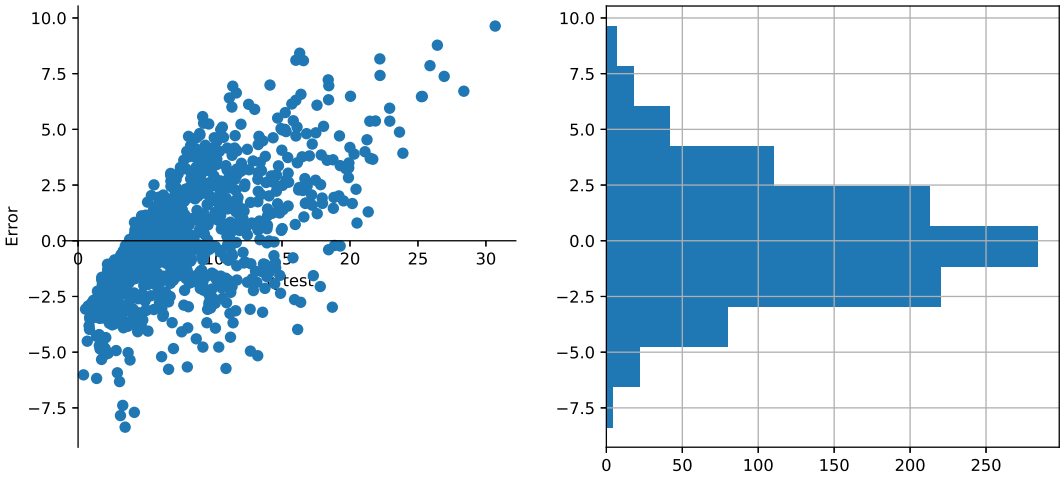


Figura 3.9: Representación de los errores ,$\mathbf{y} - h(\mathbf{x})$, generados por un bosque aleatorio. En la figura de la izquierda se han representado frente a la velocidad del viento real y en la figura de la derecha se representa el histograma de los errores.

Se explorarán tres parámetros diferentes: el número de iteraciones del algoritmo, el número máximo de variables que se consideran cada test en los nodos internos de los árboles que componen el conjunto y, finalmente, la tasa de aprendizaje. En este algoritmo la optimización se hace mediante suma de funciones en vez de modificar los parámetros asociados a una familia de funciones. Es decir, cada función realiza la regresión de los residuos de la función anterior. La tasa de aprendizaje es el peso que se le da a esta suma de funciones en cada paso del proceso de optimización. El espacio de hiperparámetros se ha sacado de [39].

La rejilla del modelo de Potenciación del Gradiente es la siguiente:

- número de estimadores:{500,600,900, 1200].
- número máximo de variables: [’sqrt’, ’auto’],

- tasa de aprendizaje: [0.01, 0.05, 0.1, 0.15].

Los hiperparámetros se determinan mediante una validación cruzada de 10 hojaspara cada una de los experimentos realizados. Los valores obtenidos son:

Tamaño del conjunto de entrenamiento	Tasa de aprendizaje	Max_features	Número de estimadores	$RECM_{10-cv}$
5%	0.01	'auto'	900	2.802 ± 0.084
10%	0.01	'sqrt'	1200	2.715 ± 0.045
15%	0.05	'sqrt'	600	2.681 ± 0.036
20%	0.05	'sqrt'	900	2.66 ± 0.033

Como en el apartado anterior, en esta tabla, el tamaño del conjunto de entrenamiento se expresa como el porcentaje de ejemplos del conjunto completo del que disponemos.

En cuanto a las predicciones realizadas por una potenciación del gradiente, se han representado los valores predecidos frente a los valores reales en la figura 3.10. Estos valores han sido obtenidos para una proporción del conjunto de entrenamiento-test del 20-80%. Los hiperparámetros han sido seleccionados por validación cruzada.

Aunque no es capaz de reproducir los valores extremos de la distribución del viento, si que parece que se ajustan más a la recta que en las predicciones realizadas por los métodos anteriores.

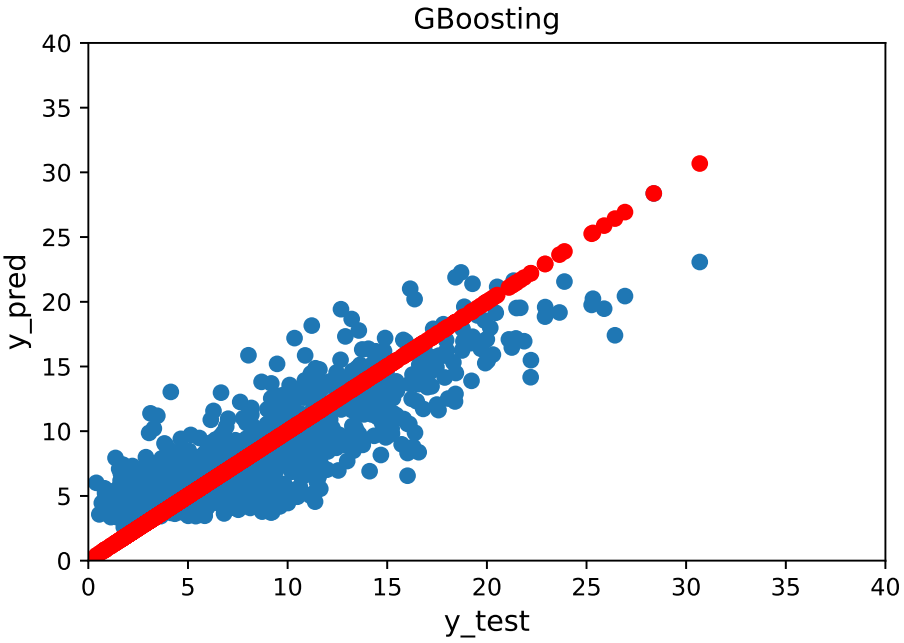


Figura 3.10: Predicciones realizadas por una potenciación del gradiente frente a los valores reales.

Si estudiamos más a fondo la estructura de los errores (figura 3.11), se sigue mostrando que la contribución de los errores negativos grandes es debido a la cola inferior y los errores positivos grandes son debidos a la cola superior. Sin embargo, los errores están centrados en 0 de una forma mucho más simétrica que con los métodos anteriores.

Potenciación Extrema del Gradiente. Para la Potenciación Extrema del Gradiente, se ha usado la implementación del paquete de Python, *xgboost* [8]:

```
class xgboost.XGBRegressor(max_depth, n_estimators, min_child_weight, colsample_bylevel)
```

Debido a que la Potenciación Extrema del Gradiente, es un método parecido a la Potenciación del Gradiente y también usa árboles de decisión como regresor individual, los hiperparámetros son aquellos que determinan la arquitectura de estos árboles y el número de estimadores necesarios para llegar al error mínimo. El espacio de hiperparámetros se ha sacado de [39].

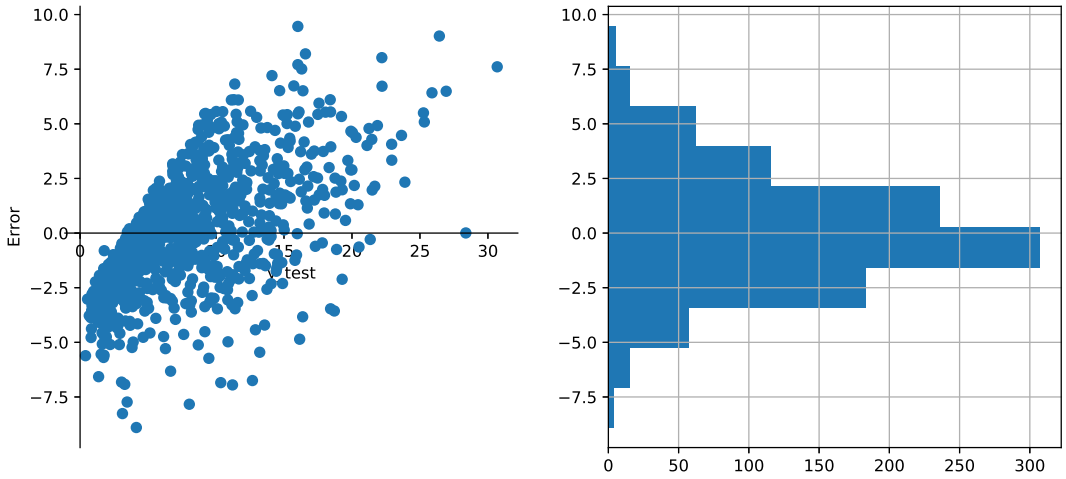


Figura 3.11: Representación de los errores ,$\mathbf{y} - h(\mathbf{x})$, generados por una potenciación del gradiente. En la figura de la izquierda se han representado frente a la velocidad del viento real y en la figura de la derecha se representa el histograma de los errores.

- Número de estimadores: [200,400,600,800],
- *colsample_bylevel*: [0.5, 0.75, 1],
- *min_child_weight*: [1,2,4],
- *max_depth*:[2,4,8]

Para cada tamaño del subconjunto de entrenamiento, se han determinado los hiperparámetros mediante una validación cruzada de 10 hojas. Los valores así obtenidos son:

Tamaño del conjunto entrenamiento	'colsample_bylevel'	'max_depth'	'min_child_weight'	'n_estimators'	$RECM_{10-cv}$
5%	0.5	2	2	200	2.817 ± 0.072
10%	0.5	4	2	200	2.734 ± 0.045
15%	0.5	4	1	200	2.688 ± 0.038
20%	0.75	4	4	200	2.666 ± 0.037

La calidad de las predicciones realizadas por la potenciación extrema del gradiente se muestra en la figura 3.12. En ella se vuelve a observar el hecho de no ser capaz de reproducir los vientos muy bajos o los que son muy altos.

Y esto se ve en la estructura del error mostrada en la figura 3.13, donde se muestra que las mayores desviaciones provienen de los valores pequeños de viento. Las desviaciones positivas vienen debido a la incapacidad de reproducir las altas velocidades del viento.

Selección del modelo de aprendizaje automático.

Para saber cual es el tamaño mínimo del subconjunto de entrenamiento, ha variado entre 5%, 10%, 15%, 20%. Se han repetido 50 veces cada división entrenamiento-test y se ha promediado. La semilla usada en cada separación, ha sido el número de repetición. Para separlo se ha usado la clase de sklearn train_test_split.

Se han probado distintos modelos: red neuronal, procesos gaussianos, bosque aleatorio, potenciación del gradiente, potenciación extrema del gradiente y máquinas de vectores de soporte. Se han repetido 50 veces cada experimento y se ha promediado. La selección de parámetros de cada modelo, se ha relizado en la primera iteracción. El resultado se muestra en las figuras 3.14 y 3.17.

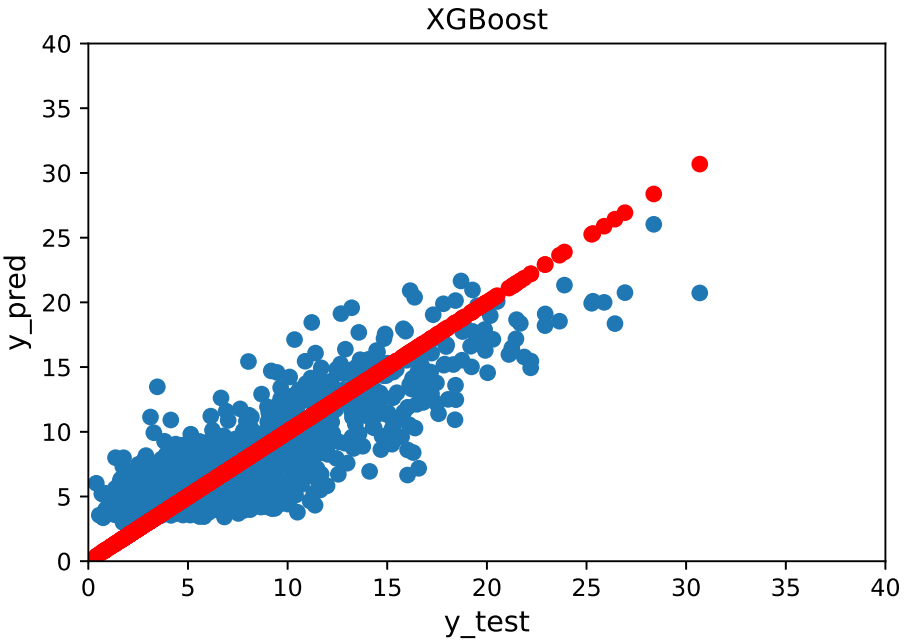


Figura 3.12: Predicciones realizadas por una potenciación extrema del gradiente frente a los valores reales.

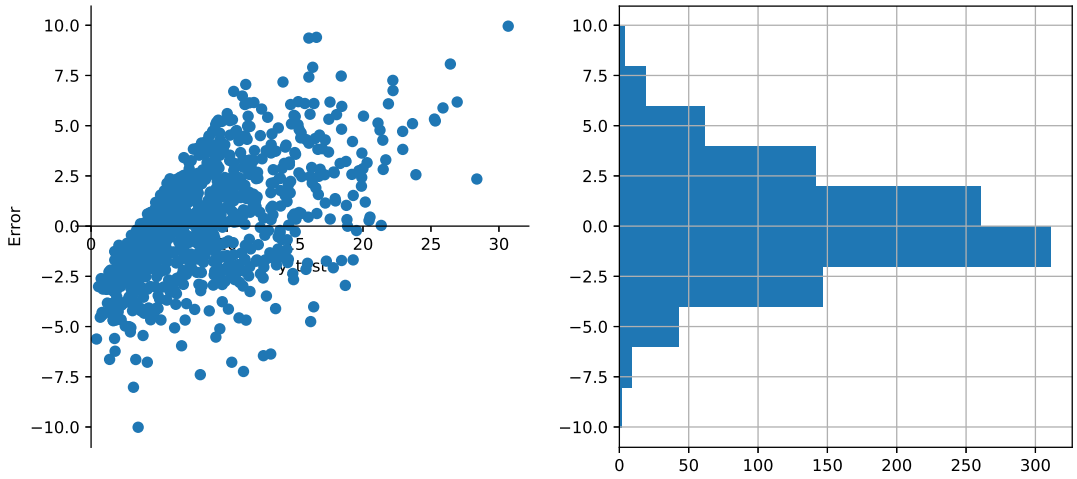


Figura 3.13: Representación de los errores ,$\mathbf{y} - h(\mathbf{x})$, generados por una potenciación extrema del gradiente. En la figura de la izquierda se han representado frente a la velocidad del viento real y en la figura de la derecha se representa el histograma de los errores

Se puede observar que modelo que tiene mejor resultado es el bosque aleatorio (RF). También es interesante ver como el ranking de modelos se mantiene en distintos tamaños de entrenamiento. Por ejemplo, para el RECM el ranking es bosque aleatorio, redes neuronales, Potenciación del Gradiente, Potenciación Extrema del Gradiente, Máquinas de vectores soporte (SVR) y el proceso gaussiano. Para el EAM, parece que la red neuronal y el bosque aleatorio tienen el mismo resultado en 3 tamaños del conjunto de entrenamiento, sin embargo en el conjunto de test se produce el desempate. En validación cruzada con un tamaño del 5% la red neuronal tiene un error especialmente alto pero en el conjunto de test muestra que la selección de parámetros está bien hecha.

En cuanto a las distribuciones de las predicciones en test (figura 3.15) con una proporción de un 20% del conjunto de entrenamiento, se puede evidenciar lo que ya habían mostrado las figuras anteriores: no son

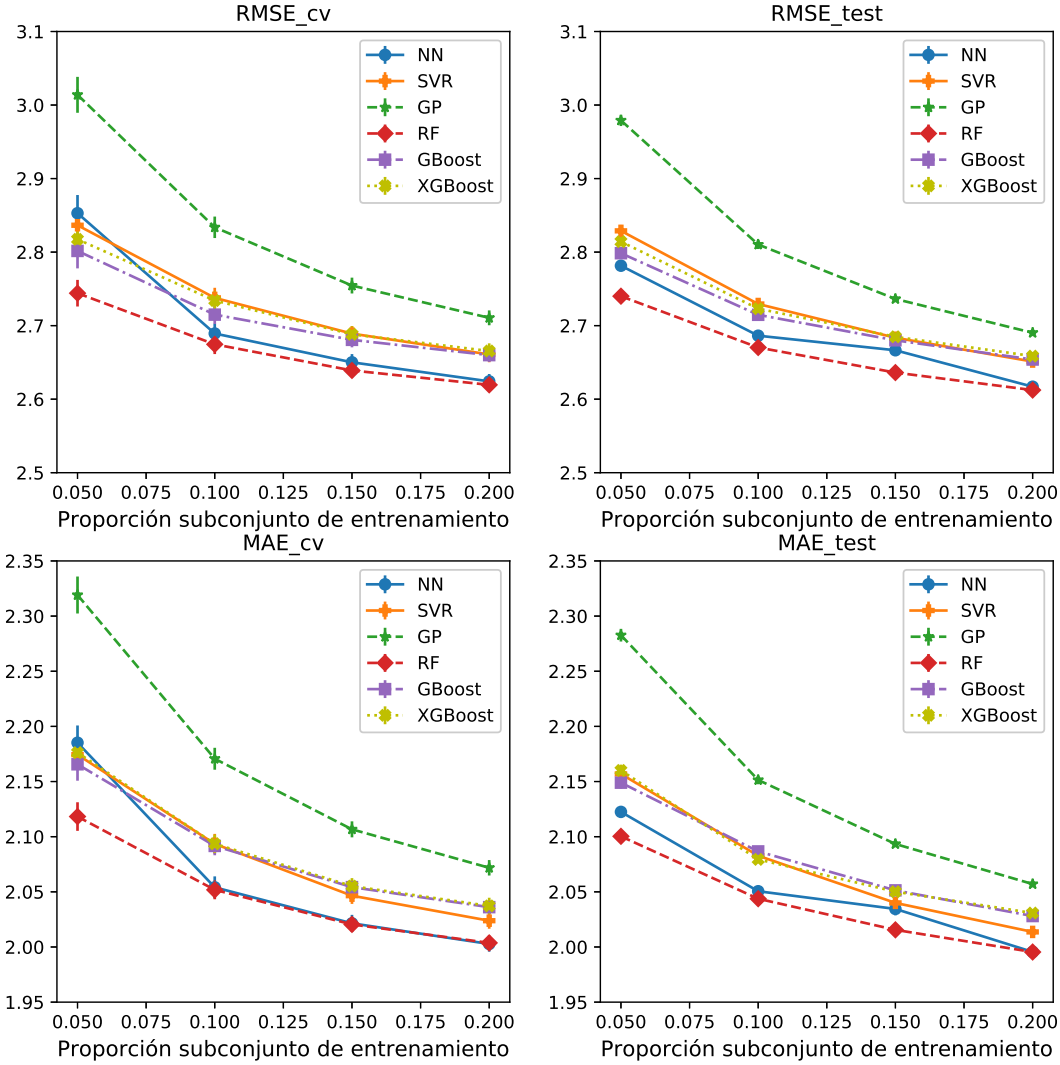


Figura 3.14: RECM y EAM en validación cruzada (10-cv) y en el conjunto de test frente al tamaño del subconjunto de entrenamiento para NN, RF, SVM, GBoosting, XGBoost y GP.

capaces de reproducir la distribución de probabilidad del conjunto de test. Se observa ninguno es capaz de reproducir la cola inferior que representan vientos con velocidades muy bajas. Los que más se aproximan serían la potenciación del gradiente (*GBoost*) y las máquinas de vectores soporte (*SVR*).

En cuanto a las distribuciones de errores (figura 3.16) en el conjunto de test con una proporción del 20 % de conjunto de entrenamiento, se observa que la potenciación del gradiente es también el algoritmo que mayores errores comete. Mientras que el bosque aleatorio es el que más concentrado tiene el error.

Si además de la evolución del error, se analiza el tiempo (Fig. 3.17), se observa que el modelo que menor tiempo necesita es el bosque aleatorio independientemente del tamaño del subconjunto de entrenamiento. Este aspecto tiene en cuenta no sólo el tiempo de ejecución de cada modelo, sino que además tiene en cuenta el tiempo de búsqueda de la mejor combinación de parámetros. Esa es la razón por la que las redes neuronales tienen un coste mucho mayor.

La siguiente figura (Fig. 3.18) es un adelanto del siguiente capítulo donde los métodos se evaluarán empíricamente. Pero ya se puede ver que el modelo que comete menos error en el mínimo tiempo es el bosque aleatorio. El modelo menos eficiente es la red neuronal pero esto es debido a la explicación anteriormente

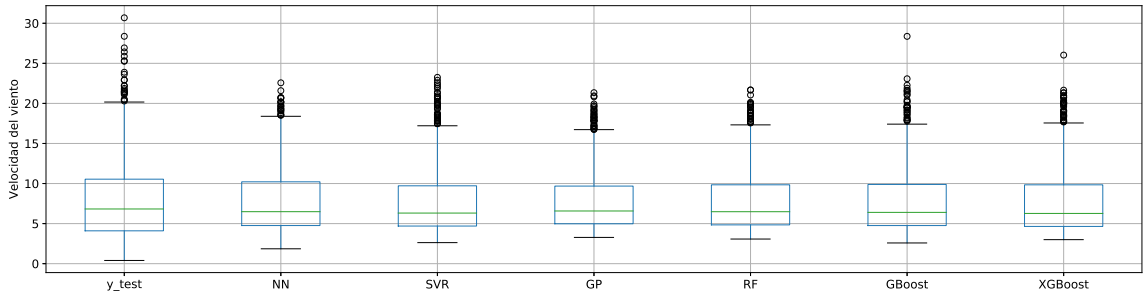


Figura 3.15: Distribuciones de probabilidad del viento en el conjunto de test con un 80 % de los datos y para las predicciones realizadas por los métodos utilizados.

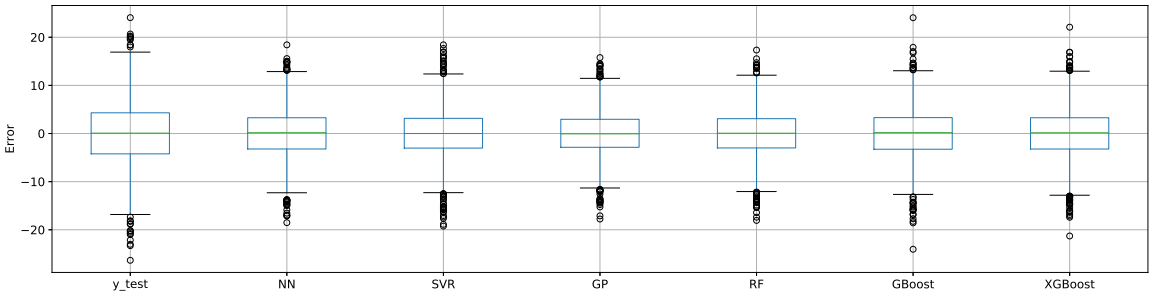


Figura 3.16: Distribuciones de probabilidad del error de predicción en el conjunto de test con un 80 % de los datos de los métodos utilizados.

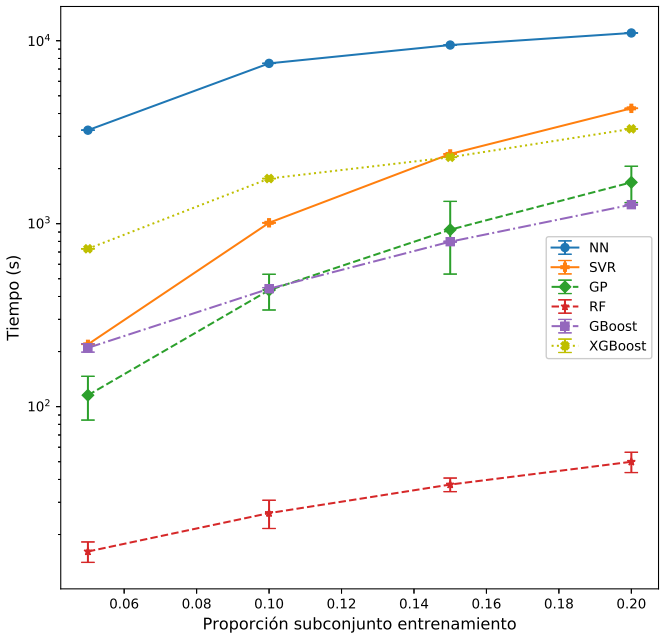


Figura 3.17: Tiempo de calculo para los modelos de redes neuronales (NN), máquinas de vectores soporte (SVR), bosque aleatorio (RF), procesos gaussianos (GP), GBoosting (GBoosting) y Extreme Gradient Boosting (XGBoost).

dada. Por otro lado en cuanto a eficiencia el resto de métodos están en una zona intermedia. Si analizamos que modelo es el que realiza menos error, existe un empate. O el bosque aleatorio o la red neuronal. Sin embargo, el que mayor error comete es el proceso gaussiano.

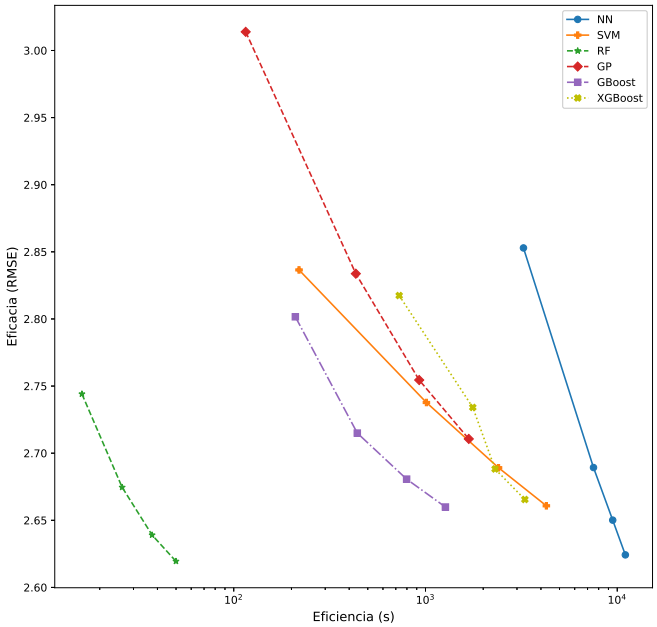


Figura 3.18: Error respecto al tiempo de ejecución para los modelos de redes neuronales (NN), máquinas de vectores soporte (SVM), bosques aleatorios (RF), procesos gaussianos (GP), potenciación del gradiente (GBoost) y potenciación extrema del gradiente (XGBoost)

Una vez hemos encontrado los mejores parámetros para cada modelo, se ha analizado la evolución del error a lo largo de distintos tamaños del subconjunto de entrenamiento y cuanto es el coste temporal a la hora de usar cada modelo. El elegido para el siguiente experimento es el bosque aleatorio con un tamaño del subconjunto de entrenamiento del 20 %.

3.3.2. Selección de variables

Una vez analizado el error usando todas las variables predictoras, ahora se va a analizar que ocurre si se reduce el número de variables seleccionando aquellas que sean más relevantes para predecir. De acuerdo con los resultados del bloque de experimentos anterior, utilizaremos un bosque aleatorio, que es el método que proporciona mejores predicciones con un bajo coste computacional. Para entrenar los bosques aleatorios se utilizaran conjuntos con un 20 % de los ejemplos disponibles. Aparte de la calidad de las predicciones de este algoritmo, tiene la ventaja de que la configuración por defecto funcionan muy bien. Se ha realizado cada experimento 50 veces y se han promediando los resultados. Todas las medidas del error vienen dados con una desviación estándar.

En este proceso de selección de variables se han utilizado dos enfoques diferentes. En el primero se utilizan aproximaciones meteorológicas mientras que en el segundo se utiliza un método estadístico de selección de variables.

Selección de variables mediante aproximaciones meteorológicas.

En este experimento se ha utilizado conocimiento experto a la hora de realizar la selección de variables. Primero se comprobará si los cuatro puntos de reanálisis tienen una información redundante. Después se separarán priorizarán las variables de viento y por último, se separarán por niveles.

Es necesaria una referencia para poder comparar. En la tabla de resultados 3.2, se ha añadido la referencia de la desviación estándar del error, el resultado que se obtendría con una regresión lineal y el bosque aleatorio con las 48 variables predictoras, es decir, con las 12 variables para cada uno de los cuatro puntos circuncidantes al parque. Este error nos servirá de referencia a la hora de reducir el número de variables predictoras. Aunque se reduzca mucho el número de variables, el error no puede ser mayor que la desviación estándar o la regresión líneal.

La primera aproximación que se realiza es comprobar si los cuatro puntos de reanálisis tienen información repetida o redundante. Para ello, lo primero que se ha hecho es mirar si existe alguna estructura en los datos, para ello, se observa la matriz de correlación (Fig. 3.19). En la figura 3.19 se han ordenado las varia-

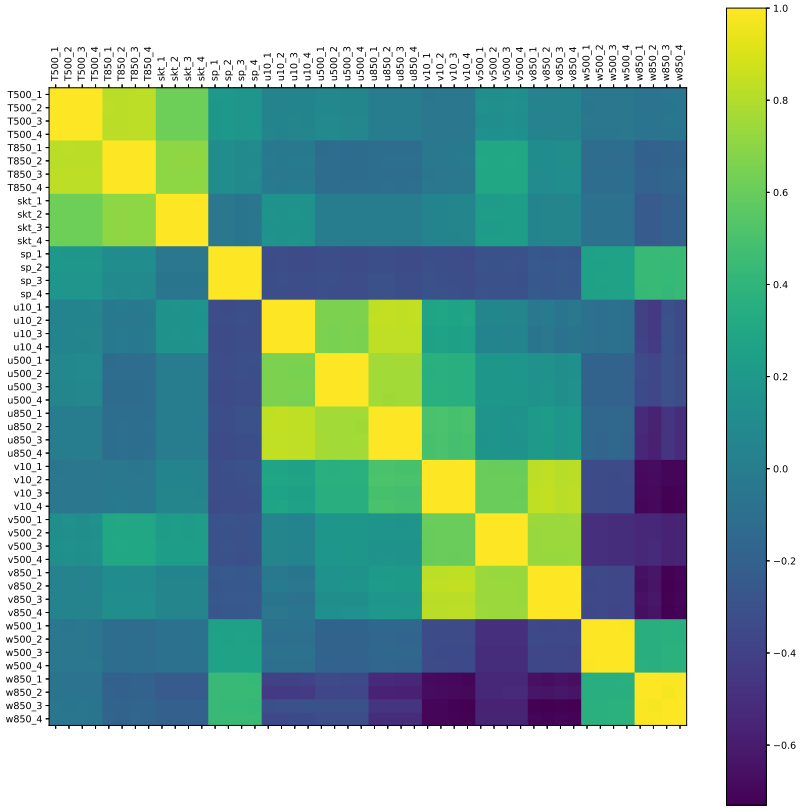


Figura 3.19: Matriz de correlación de las variables predictoras mostradas en tabla 3.1.

bles en bloques de cuatro correspondientes a la misma variable pero para cada uno de los cuatro puntos que rodean al parque. Se puede ver que existe una correlación de 1 entre las mismas variables de cada uno de los cuatro puntos, que es lo que significan los cuadrados de 4 puntos por 4 puntos en amarillo. Cada uno de esos cuadrados está compuesto por 16 cuadrados, correspondientes a las 16 correlaciones. Esto significa que los cuatro puntos proporcionan información similar. Por tanto, se ha seleccionado las variables en uno solo de los puntos.

Usando las variables de un sólo punto, el RECM se muestra en la tabla 3.2. Este experimento es el que tiene 12 variables meteorológicas y se ha realizado para cada uno de los puntos. El error es muy similar respecto al experimento con el conjunto de datos completo. Aunque el error es pequeño, el aumento del error es significativo. Para determinar si las diferencias entre errores son significativas se ha utilizado la aproximación de Bonferroni [3, 35, 43] al test de hipótesis de diferencia de medias. Esta modificación tiene en cuenta que las medias podrían provenir de muestras muy similares y cambia el umbral de significación. Por último, se ha comprobado que si se entrenan cuatro bosques aleatorios, cada uno con un punto diferente,

y se promedian las predicciones (tabla 3.2, $D = 12*$), el resultado es un poco mejor en validación cruzada e igual en el subconjunto de test. Por tanto, a la hora de reducir dimensionalidad, usar sólo un punto y sus 12 variables meteorológicas es una buena elección ya que los cuatro puntos proporcionan información similar.

D	$RECM_{10-cv}$	$RECM_{test}$
cte (std)	4.842 ± 0.057	4.825 ± 0.014
48 (LR)	4.153 ± 0.046	4.133 ± 0.012
48 (RF)	2.623 ± 0.033	2.612 ± 0.011
12 (punto 1)	2.648 ± 0.032	2.638 ± 0.013
12 (punto 2)	2.657 ± 0.031	2.644 ± 0.013
12 (punto 3)	2.650 ± 0.031	2.640 ± 0.012
12 (punto 4)	2.658 ± 0.033	2.646 ± 0.013
12*	2.634 ± 0.012	2.646 ± 0.032
9	2.651 ± 0.032	2.643 ± 0.011
8	2.685 ± 0.035	2.678 ± 0.012
7	2.656 ± 0.034	2.648 ± 0.011
6	2.680 ± 0.037	2.675 ± 0.011

Cuadro 3.2: Resultados de validación cruzada y de test del bosque aleatorio (RF) para la varianza, las 48 variables con una regresión linear y con RF; 12 variables de cada punto con RF y con 9, 8 ,7 y 6 variables. El modelo 12* consiste en entrenar 4 RF cada uno con un punto y promediar las predicciones. Las diferencias entre cada modelo y el bosque aleatorio de 48 variables son significativas con un α igual a 5% con una corrección de Bonferroni.

Ante estos resultados, se ha seleccionado sólo un punto de los cuatro para usarlo como variables predictora (el punto 3). Dentro de las variables meteorológicas de este punto, se han seleccionado las 8 variables de viento. También se ha añadido la temperatura de la superficie. Esta temperatura puede dar información sobre la estabilidad de la atmósfera y si se produce convección o no. Los resultados se muestran en la tabla 3.2 en las entradas con un número de variables 8 (todas las variables de viento) y 9 (todas las variables del viento y la temperatura en superficie). Aunque el aumento de error al pasar de 12 a 8 variables sigue siendo significativa, la diferencia se encuentra en el segundo decimal, así que todavía es pequeña.

Como se comentó en la introducción del trabajo, la atmósfera está estratificada. Esto quiere decir que está compuesta por capas de aire. La inestabilidad en la atmósfera la suelen provocar corrientes ascendentes de aire, pero estas corrientes tienen una magnitud mucho más pequeña que la velocidad en la propia capa dada por las componentes zonal y meridional del viento. Por tanto, analizando la importancia de cada una de las componentes del viento, la componente vertical se ha eliminado de las variables predictoras. También se ha realizado el experimento contando con la temperatura de la superficie y sin ella. Los resultados son las entradas de la tabla 3.2 7 (componentes zonal y meridional de los tres niveles de viento y temperatura en superficie) y 6 (componenetes zonal y meridional del viento para cada uno de los tres niveles), respectivamente.

Por último, se ha querido observar si al tener tres niveles de altura (superficie, 850hPa y 500hPa), al escoger sólo uno de ellos, el error no aumenta demasiado. Ya que tenemos tres niveles bien separados, se pueden considerar independientes. Igual hay algún nivel que aporta más a la predicción del viento que los demás. El resultado se muestra en la tabla 3.3. En ella se puede observar, que el mejor nivel es el viento en 850 hPa más la temperatura de la superficie. Hay que tener en cuenta que todos las diferencias son significativas.

El hecho de que añadiendo la temperatura de la superficie el error disminuya, significa que es una variable con peso en la predicción. Esto indica que la convección que pueda haber en la cercanía del parque es relevante. Esto tiene sentido, ya que el parque se localiza en una montaña.

Las variables predictoras que mejor funcionan son las componentes del viento a 850 hPa. Esto se debe a que es la capa más cercana a la superficie pero sin efectos asociados al rozamiento. Con otra combinación de dos variables, el error puede ser peor que usando una regresión lineal con todas las variables.

Nivel de la atmósfera	D	$RECM_{10-cv}$	$RECM_{test}$
10m	3	3.321 ± 0.052	3.305 ± 0.015
10m	2	3.735 ± 0.058	3.717 ± 0.015
850 hPa	3	2.881 ± 0.033	2.876 ± 0.010
850 hPa	2	3.005 ± 0.041	3.004 ± 0.010
500 hPa	3	4.283 ± 0.061	4.270 ± 0.013
500 hPa	2	4.490 ± 0.065	4.483 ± 0.015

Cuadro 3.3: Resultados de validación cruzada del bosque aleatorio usando las 2 variables horizontales de viento para cada nivel del punto 3 y las 2 variables horizontales de viento en cada nivel más la temperatura de superficie.

Selección de variables mediante métodos estadísticos.

Una ventaja del bosque aleatorio, es que incorpora un método estadístico que devuelve la importancia de las variables durante el entrenamiento. Para poder comparar cada una de las aproximaciones realizadas, se ha seleccionado las variables más relevantes según el bosque aleatorio y se ha hecho una predicción para cada número de variables. Esto permitirá comprobar si el método de reducción de dimensionalidad empleado es mejor que uno realizado mediante aprendizaje automático.

El bosque aleatorio tiene un algoritmo propio para determinar la importancia de las variables [4]. Este método consiste en la permutación de los valores de una variable sin cambiar nada más en el conjunto de entrenamiento. Esta permutación elimina la relación entre dicha variable y la varible que se quiere predecir. Si se realiza una predicción con la variable permutada y la métrica utilizada empeora, se determina que esta variable es importante a la hora de predecir. Si la métrica apenas cambia, significa que esta variable es irrelevante. Teniendo en cuenta esto, el bosque aleatorio realiza una ordenación de las variables en función de su impacto en la predicción.

Se han seleccionado las n-variables más relevantes para la predicción de viento. El número de variables ha sido escogido para que coincidan con las tablas 3.2 y 3.3. Esto facilitará la comparación entre ambos métodos de selección de variables.

Para cada número de variables se ha repetido 50 veces el experimento con tamaño del conjunto de entrenamiento del 20 %. Los resultados se muestran en la siguiente tabla.

D	$RECM_{10-cv}$	$RECM_{test}$
cte (std)	4.842 ± 0.057	4.825 ± 0.014
48 (LR)	4.153 ± 0.046	4.133 ± 0.012
48 (RF)	2.623 ± 0.033	2.612 ± 0.011
12	2.836±0.12	2.838 ± 0.012
9	2.872±0.127	2.873 ± 0.011
8	2.882±0.125	2.883 ± 0.022
7	2.892±0.128	2.891 ± 0.023
6	2.91±0.128	2.907 ± 0.019
3	3.007±0.124	3.01 ± 0.012
2	3.101±0.130	3.109 ± 0.014

Cuadro 3.4: Resultados de validación cruzada y de test del bosque aleatorio (RF) para la varianza, las 48 variables con una regresión linear y con RF; 12 variables de cada punto con RF y con 9, 8 ,7 y 6 variables. El modelo 12* consiste en entrenar 4 RF cada uno con un punto y promediar las predicciones. Las diferencias entre cada modelo y el bosque aleatorio de 48 variables son significativas con un α igual a 5 % con una corrección de Bonferroni.

Utilizando esta selección de variables el error es mayor. No obstante el RECM se no varía mucho según se reduce el número de variables D. Mientras que las diferencias de error en el otro método son mayores a medida que se disminuye D.

En resumen, se ha realizado una selección de variables al conjunto de datos inicial. Esta selección ha

sido comparada con un método estadístico de selección de variables y como se ha visto, el método estadístico obtiene un error mayor. Este conjunto de datos estaba formado por 12 variables para cada uno de los cuatro puntos del parque. En primer lugar se ha comprobado que no es necesario introducir los cuatro puntos ya que la información es en gran medida redundante. Con un único punto con sus 12 variables es suficiente. Después se han diferenciado las variables de viento del resto. Se han combinado con la temperatura superficial, que se ha demostrado que es una variable importante en al predicción. Después se ha analizado la contribución de cada una de las componentes del viento al módulo de la velocidad del viento. La componente vertical era la menos significativa y se ha eliminado de los dos niveles que la tenían. En este punto, teníamos las variables estructuradas por niveles, y el mejor ha sido el nivel de 850 hPa combinado con la temperatura en superficie. Esta es la combinación más relevante de variables predictoras.

Capítulo 4

Conclusiones

La predicción del viento es importante para el funcionamiento de un parque eólico. Si el parque eólico dispone de predicciones fiables es capaz de conocer de cuanta energía va a disponer para venderla en el mercado eléctrico. El parque eólico también necesita esta información para gestionar el funcionamiento de los propios aerogeneradores, ya que si la velocidad del viento es demasiado alta, estos se pueden dañar y causar un perjuicio económico al parque.

Existen modelos meteorológicos de predicción númerica que resuelven la dinámica de la atmósfera y que son capaces de realizar predicciones de viento precisas. Sin embargo, estos modelos tienen inconvenientes: son computacionalmente muy costosos y debido a la discretización espacial que hay que realizar, puede ocurrir para la localización del parque no se realice ninguna predicción.

Existen otros algoritmos de predicción que pueden solucionar este problema ya que los parques eólicos disponen de datos históricos de velocidad del viento. Aprovechando esta serie de datos, se pueden realizar predicciones mediante algortimos de aprendizaje automático. Estos modelos son capaces de mejorar las predicciones realizadas por los modelos meteorológicos de predicción numérica. Los algoritmos utilizados en este Trabajo Fin de Máster se han explicado en el capítulo 2. Estos algoritmos se utilizan frecuentemente en la literatura y son las redes neuronales, las máquinas de vectores soporte, los procesos gaussianos, el bosque aleatorio, la potenciación del gradiente y la potenciación extrema del gradiente.

Una vez seleccionados algoritmos, se han realizado dos experimentos en el capítulo 3. El primero consiste en comparar el error en validación cruzada y en test en función de la proporción entre el conjunto de entrenamiento y el conjunto de test para saber cual es el mejor modelo posible. El segundo experimento consiste en determinar cuales son las varibles más influyentes a la hora de realizar la predicción de la velocidad del viento.

En la comparación de errores se han comparado el error de validación cruzada y el de test. Además, se han estudiado la calidad de las predicciones de cada método y la distribución del error. El algoritmo que menos error tuvo en menor tiempo fue el bosque aleatorio, que es el mejor modelo para este problema. Sin embargo, el algoritmo que mejor reproducía la distribución de probabilidad del viento era la potenciación del gradiente. Como el error es más pequeño y el tiempo de ejecución es menor en el caso del bosque aleatorio que en el de potenciación del gradiente, ha determinado que el mejor modelo es el bosque aleatorio.

El segundo experimento indica la importancia que tiene el nivel de 850hPa, ya que se trata del nivel más cercano a la superficie pero en el que no hay efectos de la capa límite atmosférica asociados al rozamiento que se produce en la superficie. También es relevante la temperatura superficial, ya que es un indicador de la convección que puede haber ese día en la atmósfera. El hecho de que al utilizar estas tres variables como variables predictoras el error no tenga una variación tan pronunciada respecto al error del bosque aleatorio con todas las variables del conjunto de datos indica que son las más influyentes a la hora de realizar la predicción.

Bibliografía

[1] Acheson D (1990) Elementary Fluid Dynamics. Comparative Pathobiology - Studies in the Postmodern Theory of Education, Clarendon Press, URL https://books.google.es/books?id=IGfDBAAAQBAJ

[2] Arya P, Holton J (2001) Introduction to Micrometeorology. International Geophysics, Elsevier Science, URL https://books.google.es/books?id=tRE0sBZ8u4YC

[3] Bonferroni C (1935) Il calcolo delle assicurazioni su gruppi di teste. Tipografia del Senato, URL https://books.google.es/books?id=xnQ4cgAACAAJ

[4] Breiman L (2001) Random forests. Machine Learning 45(1):5–32, DOI 10.1023/A:1010933404324, URL https://doi.org/10.1023/A:1010933404324

[5] Cajal SRY (1894) The Croonian lecture: La fine structure des centres nerveux. Proceedings of the Royal Society of London 55:444–468, URL http://www.jstor.org/stable/115494

[6] Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2:27:1–27:27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[7] Chen J, Zeng GQ, Zhou W, Du W, Lu KD (2018) Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Conversion and Management 165:681–695

[8] Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, KDD '16, pp 785–794, DOI 10.1145/2939672.2939785, URL http://doi.acm.org/10.1145/2939672.2939785

[9] Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2:303–314

[10] Dee DP, Uppala SM, Simmons A, Berrisford P, Poli P, Kobayashi S, Andrae U, Balmaseda M, Balsamo G, Bauer dP, et al (2011) The ERA-Interim reanalysis: Configuration and performance of the data assimilation system. Quarterly Journal of the royal meteorological society 137(656):553–597

[11] Deo RC, Ghorbani MA, Samadianfard S, Maraseni T, Bilgili M, Biazar M (2018) Multi-layer perceptron hybrid model integrated with the firefly optimizer algorithm for windspeed prediction of target site using a limited set of neighboring reference station data. Renewable energy 116:309–323

[12] Du P, Wang J, Guo Z, Yang W (2017) Research and application of a novel hybrid forecasting system based on multi-objective optimization for wind speed forecasting. Energy Conversion and Management 150:90–107

[13] Efron B, Tibshirani R (1994) An Introduction to the Bootstrap. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Taylor & Francis, URL https://books.google.es/books?id=gLlpIUxRntoC

[14] Feng C, Cui M, Hodge BM, Zhang J (2017) A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. Applied Energy 190:1245–1257

[15] Friedman JH (2001) Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29(5):1189–1232, URL http://www.jstor.org/stable/2699986

[16] Gal Y, van der Wilk M, Rasmussen CE (2014) Distributed variational inference in sparse gaussian process regression and latent variable models. 1402.1389

[17] Holton JR (2004) An introduction to dynamic meteorology, 4th edn. International Geophysics Series, Elsevier Academic Press,, Burlington, MA, URL http://books.google.com/books?id=fhW5oDv3EPsC

[18] Khodayar M, Kaynak O, Khodayar ME (2017) Rough deep neural architecture for short-term wind speed forecasting. IEEE Trans Ind Inform 13:2770–2779

[19] Khosravi A, Machado L, Nunes R (2018) Time-series prediction of wind speed using machine learning algorithms: A case study osorio wind farm, brazil. Applied Energy 224:550–566

[20] Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980

[21] Laborie P, Rogerie J, Shaw P, Vilím P (2018) Ibm ilog cp optimizer for scheduling. Constraints 23(2):210–250

[22] Lahouar A, Slama JBH (2017) Hour-ahead wind power forecast based on random forests. Renewable energy 109:529–541

[23] LeCun Y, Bengio Y, Hinton G (2015) Deep learning. nature 521(7553):436

[24] Li DY, Cai WC, Li P, Jia ZJ, Chen HJ, Song YD (2016) Neuroadaptive variable speed control of wind turbine with wind speed estimation. IEEE Transactions on Industrial Electronics 63(12):7754–7764

[25] Liu D, Nocedal J (1989) On the limited memory bfgs method for large scale optimization. Mathematical Programming 45(1-3):503–528, DOI 10.1007/BF01589116

[26] McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5(4):115–133, DOI 10.1007/BF02478259, URL https://doi.org/10.1007/BF02478259

[27] Mi Xw, Liu H, Li Yf (2017) Wind speed forecasting method using wavelet, extreme learning machine and outlier correction algorithm. Energy Conversion and Management 151:709–722

[28] Ouyang T, Zha X, Qin L (2017) A combined multivariate model for wind power prediction. Energy Conversion and Management 144:361–373

[29] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830

[30] Peng X, Zheng W, Zhang D, Liu Y, Lu D, Lin L (2017) A novel probabilistic wind speed forecasting based on combination of the adaptive ensemble of on-line sequential orelm (outlier robust extreme learning machine) and tvmcf (time-varying mixture copula function). Energy Conversion and Management 138:587–602

[31] Qureshi AS, Khan A, Zameer A, Usman A (2017) Wind power prediction using deep neural network based meta regression and transfer learning. Applied Soft Computing 58:742–755

[32] Rasmussen CE, Williams CKI (2005) Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press

[33] Red Eléctrica de España (????) Un modelo energetico sostenible. URL http://www.ree.es/es/red21/un-modelo-energetico-sostenible

[34] Schulz E, Speekenbrink M, Krause A (2018) A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. Journal of Mathematical Psychology 85:1 – 16, DOI https://doi.org/10.1016/j.jmp.2018.03.001, URL http://www.sciencedirect.com/science/article/pii/S0022249617302158

[35] Seabold S, Perktold J (2010) Statsmodels: Econometric and statistical modeling with python. In: 9th Python in Science Conference

[36] Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Statistics and Computing 14(3):199–222, DOI 10.1023/B:STCO.0000035301.49549.88, URL https://doi.org/10.1023/B:STCO.0000035301.49549.88

[37] Stocker T (2014) Climate change 2013: the physical science basis: Working Group I contribution to the Fifth assessment report of the Intergovernmental Panel on Climate Change. Cambridge University Press

[38] Tascikaraoglu A, Uzunoglu M (2014) A review of combined approaches for prediction of short-term wind speed and power. Renewable and Sustainable Energy Reviews 34:243 – 254, DOI https://doi.org/10.1016/j.rser.2014.03.033, URL http://www.sciencedirect.com/science/article/pii/S1364032114001944

[39] Torres-Barrán A, Álvaro Alonso, Dorronsoro JR (2019) Regression tree ensembles for wind energy and solar radiation prediction. Neurocomputing 326-327:151 – 160, DOI https://doi.org/10.1016/j.neucom.2017.05.104, URL http://www.sciencedirect.com/science/article/pii/S0925231217315229

[40] Vapnik V, Guyon I, Hastie T (1995) Support vector machines. Mach Learn 20(3):273–297

[41] Wang Hz, Li Gq, Wang Gb, Peng Jc, Jiang H, Liu Yt (2017) Deep learning based ensemble approach for probabilistic wind power forecasting. Applied energy 188:56–70

[42] Wang J, Du P, Niu T, Yang W (2017) A novel hybrid system based on a new proposed algorithm—multi-objective whale optimization algorithm for wind speed forecasting. Applied Energy 208:344–360

[43] Weisstein EW (2004) Bonferroni correction. Wolfram Research, Inc.

[44] Yin H, Dong Z, Chen Y, Ge J, Lai LL, Vaccaro A, Meng A (2017) An effective secondary decomposition approach for wind power forecasting using extreme learning machine trained by crisscross optimization. Energy Conversion and Management 150:108–121

[45] Yu C, Li Y, Zhang M (2017) An improved wavelet transform using singular spectrum analysis for wind speed forecasting based on elman neural network. Energy Conversion and Management 148:895–904

[46] Yu C, Li Y, Xiang H, Zhang M (2018) Data mining-assisted short-term wind speed forecasting by wavelet packet decomposition and Elman neural network. Journal of Wind Engineering and Industrial Aerodynamics 175:136–143

[47] Yuan X, Tan Q, Lei X, Yuan Y, Wu X (2017) Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine. Energy 129:122–137

[48] Zhang C, Wei H, Zhao X, Liu T, Zhang K (2016) A Gaussian process regression based hybrid approach for short-term wind speed prediction. Energy Conversion and Management 126:1084–1092

[49] Zhang C, Zhou J, Li C, Fu W, Peng T (2017) A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting. Energy Conversion and Management 143:360–376