Posproceso estadístico

14

DOI: 10.31978/014-18-009-X.14

DAVID QUINTERO PLAZA
D. T. islas Canarias, Agencia Estatal de Meteorología (AEMET)

> The future is already here. It's just not very evenly distributed.
>
> WILLIAM GIBSON

En este capítulo damos una introducción a las técnicas estadísticas que permiten calibrar (y por tanto mejorar) las salidas directas de un modelo meteorológico o un sistema de predicción por conjuntos. Primero hablaremos de las técnicas clásicas *Perfect Prog* y *MOS*, así como las más recientes *BMA* y *ELR* y después reservaremos una parte importante para hablar sobre las novedosas técnicas basadas en *Aprendizaje Automático (Machine Learning)*, que parecen muy prometedoras.

Palabras clave: sistemas de predicción por conjuntos, posproceso estadístico, métodos de posproceso BMA y ELR, aprendizaje automático, machine learning.

Imagen parte superior: *cumulus* y *cirrus* de diferentes tipos en las capas superiores, en las inmediaciones de Valdemorillo. Fotografía de JUAN CARLOS GARCÍA COLOMBO.

14.1 Introducción al posproceso clásico

Cuando un modelo meteorológico termina su normalmente largo proceso de ejecución presenta unas salidas en bruto (*outputs*) que son las variables de interés para un(a) predictor(a). Ejemplos de estas variables son la temperatura en superficie, la presión en distintos niveles, las componentes del viento, etcétera. Aunque estas variables son el resultado de resolver un conjunto de ecuaciones diferenciales que determinan la física atmosférica, dado que algunos procesos son aproximados y el método de resolución es también aproximado, estos outputs no son la última palabra: pueden ser mejorados. Y ahí es donde entra en juego lo que denominamos *posproceso estadístico*.

El posproceso estadístico utiliza una base de datos de predicciones del modelo y las correspondientes observaciones. Pueden ser predicciones recientes, de los últimos 3-4 días, o predicciones anteriores en el tiempo. Mediante alguna técnica se calculan las desviaciones de las predicciones con respecto a las observaciones que luego se midieron y, extrapolando esas desviaciones al futuro, se corrigen las predicciones del modelo para los días venideros. La idea es que las salidas del modelo se parezcan lo más posible a las observaciones. En ocasiones se pueden utilizar también datos climáticos. El algoritmo utiliza algún tipo de *regresión* (lineal o no) para encontrar los parámetros que mejor ajustan el modelo a las observaciones. Cuanto mayor sea la base de datos, mejor, pues más casos son considerados (consultar la caja correspondiente a la regresión lineal en la página 199).

14.1.1 Métodos clásicos de posproceso: Perfect Prog y MOS

Perfect Prog. Por *Perfect Prog* (*Perfect Prognosis* [13]) conocemos al que es quizá el primer método de posproceso estadístico. Perfect Prog utiliza uno o varios *predictores* (variables que sirven como variables independientes para hacer la regresión) para determinar el deseado *predictando* (variable dependiente de las otras). Se sirve para ello de una regresión lineal, multilineal, polinómica o de cualquier otro tipo. Los predictores (ver *predictor*) son las observaciones y (en ocasiones) datos climáticos. Los predictandos son las observaciones. Por tanto, y esta es la clave de Perfect Prog, el modelo no participa en ningún momento en el proceso de regresión. Sobre el modelo se actúa únicamente después de haber encontrado la función de regresión. En términos de *Aprendizaje Automático* (ver 14.2 en la página 198), el modelo no participa en el *entrenamiento*. Esto es debido a que según el paradigma de Perfect Prog se considera al modelo como «perfecto». Ver esquema en Figura 14.1.

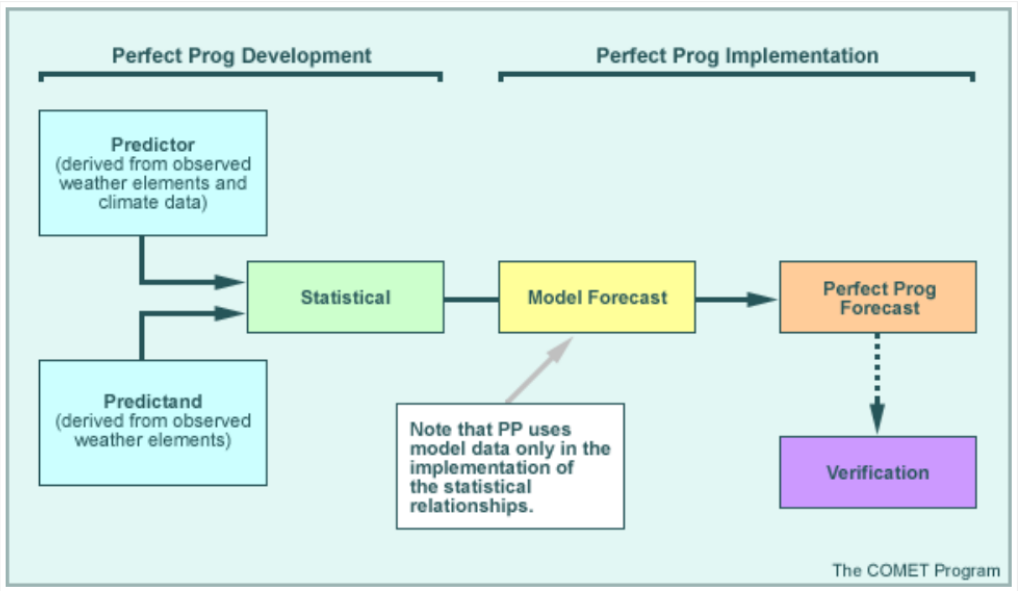

Figura 14.1: Esquema de Perfect Prog. COMET Program: Intelligent Use of Model-Derived Products.

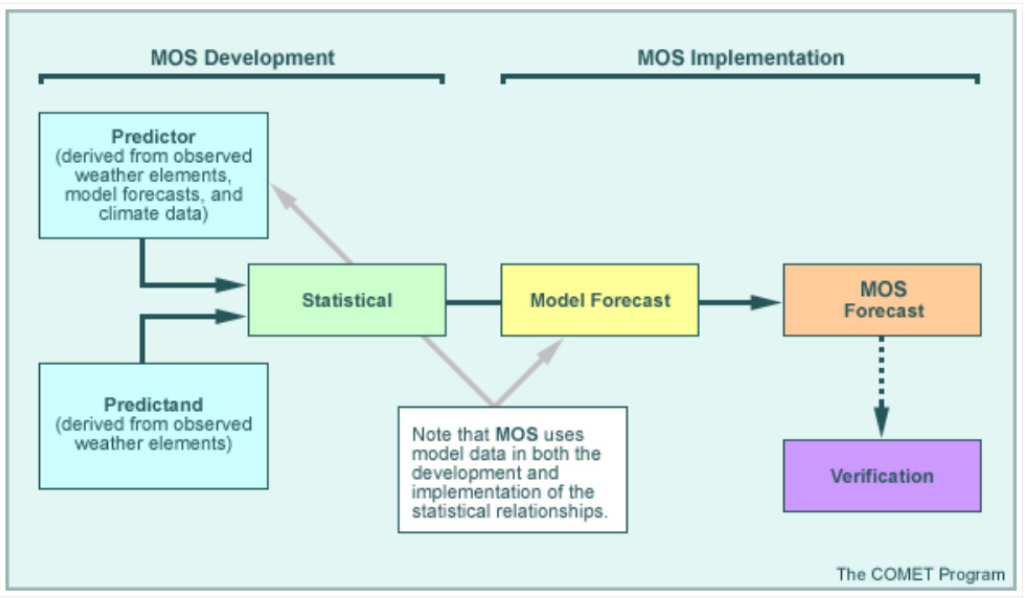

Figura 14.2: Esquema de Model Output Statistics. COMET Program: Intelligent Use of Model-Derived Products.

MOS. Perfect Prog, al no incluir el modelo en la regresión, no puede corregir sus errores sistemáticos, lo cual es una limitación importante. Para superar esta limitación se introdujo el *MOS* (*Model Output Statistics*, [7]), un método muy similar a Perfect Prog pero que utiliza el modelo en la base de entrenamiento; es decir, el modelo aparece también como predictor, junto a observaciones y datos climáticos (ver esquema en Figura 14.2).

MOS es usado por el NCEP, el Centro Nacional de Predicción Medioambiental de los Estados Unidos. Disponen de dos versiones de MOS: una directamente sobre los puntos en los que hay observaciones (*station based MOS*) y otra para los puntos de grid del modelo (*GMOS*). Para MOS utilizan una regresión lineal en múltiples variables, utilizando como predictores un conjunto muy completo de salidas del modelo, datos geoclimáticos y observaciones pasadas. Tras el proceso de entrenamiento (inferencia de los parámetros de la regresión), los resultados que obtienen son bastante buenos [1].

En la Agencia Estatal de Meteorología (AEMET) se han estudiado estos dos métodos en el pasado y se han implementado con resultados positivos [2]. Un ejemplo es el estudio que se hizo con la predicción de temperaturas extremas diarias en varias ciudades españolas durante el mes de octubre de 1992: *Método de predicción perfecta, para adaptación estadística de los diferentes modelos numéricos, utilizando los análisis del modelo de área limitad del INM*. El hecho de elegir la temperatura no fue casual ya que tiende a ser la variable que mejor se comporta. Como era de esperar, los resultados con MOS fueron algo mejores que con Perfect Prog (ver Figura 14.3).

Error absoluto medio en °C de la predicción de temperaturas extremas. Octubre 1992.

	máx (D)	máx (D+1)	mín (D+1)	mín (D+2)
PP	2,1	2,0	2,0	2,1
MOS	1,6	1,5	1,9	2,2
Persistencia		2,5		2,9

Figura 14.3: Comparación entre Perfect Prog y MOS realizada en el INM (antigua AEMET). Los predictandos con los que tanto Perfect Prog como MOS trabajan son variables de tiempo sensible, que son las conocidas por el gran público, tales como la temperatura en superficie, el viento en superficie, la nubosidad, la cantidad de precipitación recogida, la probabilidad de tormentas, etcétera [2].

14.1.2 Métodos más modernos: BMA y ELR

Estos son métodos específicos para calibrar ensembles.

Promediado bayesiano de modelos BMA. El método *Bayesian Model Averaging*, BMA [19] se aplica a ensembles. Busca ajustar una función de densidad de probabilidad (*PDF*, por sus siglas en inglés) para dar cuenta de los errores del ensemble en días pasados y

mejorarlo en días futuros. Dado que un ensemble da predicciones probabilistas, en muchos casos éstas se pueden aproximar por una gaussiana (por ejemplo, la temperatura a 2 metros). Una gaussiana viene dada por dos parámetros: la media (μ) y la desviación estándar (σ). Con el proceso de entrenamiento, se busca aproximar la gaussiana resultante del modelo con la gaussiana que viene dada por la distribución probabilista de las observaciones. El término *Bayesian* aquí se emplea por la similitud de esta técnica con el enfoque bayesiano para la estadística, en el que los modelos y sus parámetros se van actualizando a medida que van apareciendo más datos disponibles.

Distribución gaussiana. Se dice que una variable X sigue una distribución gaussiana cuando $X \sim \mathcal{N}(\mu, \sigma^2)$.

La ecuación de la gaussiana $X \sim \mathcal{N}(\mu, \sigma^2)$ en una dimensión es:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2} \tag{14.1}$$

En la imagen tenemos un ejemplo de la distribución de probabilidad de una gaussiana correspondiente a las observaciones y otra correspondiente a un miembro del ensemble. La idea es corregir la gráfica mostrada en verde para hacerla lo más parecida posible a la de las observaciones, de color azul.

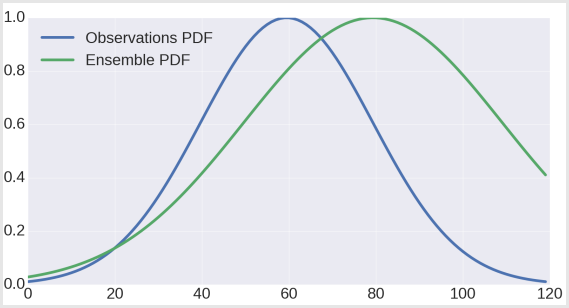

Figura 14.4: Gaussianas.

Puede suceder que una variable no siga una distribución gaussiana de probabilidad. El ejemplo típico es la precipitación. Se puede intentar aproximar en este caso la distribución de probabilidad que tenga la observación con una suma de gaussianas, cada una de ellas con distinto peso, o con una distribución diferente, por ejemplo una *log-normal*.

Bayesian Model Averaging (BMA). La PDF total del ensemble se escribe también como una suma de las PDF de los miembros:

$$PDF(y|f_1,\ldots,f_m;\theta_1,\ldots,\theta_m) = \sum_{i=1}^{m} w_i PDF_i(y|f_i,\theta_i) \tag{14.2}$$

Donde f_i representa la predicción determinista de cada miembro (m miembros en total), y representa la variable a pronosticar, θ_i los parámetros asociados a cada PDF (pueden ser más de uno, es decir, un vector de parámetros; en el caso de la gaussiana son la media y la desviación estándar) y w_i los pesos de cada densidad de probabilidad (que deben sumar 1 para que se cumplan las propiedades de una PDF).

Los pesos w_i y parámetros θ_i y se determinan por el método de *máxima verosimilitud* (*maximum likelihood*), ver caja correspondiente. Desafortunadamente este método no puede resolverse siempre analíticamente y hay que llevar a cabo algún procedimiento numérico, en particular se utiliza el algoritmo *Expectation Maximization*, *EM* [6, 14], ver caja correspondiente en la página siguiente.

Método de máxima verosimilitud. Busca los parámetros óptimos maximizando la probabilidad conjunta que tendrían estos parámetros para ajustarse a los datos. Si tenemos un conjunto de variables independientes e idénticamente distribuidas, podemos escribir la densidad conjunta como:

$$\begin{aligned} f(X_i;\theta) &= f(x_1,x_2,\ldots,x_n \mid \theta) = \\ &= f(x_1 \mid \theta) \times f(x_2 \mid \theta) \times \cdots \times f(x_n \mid \theta) \end{aligned} \tag{14.3}$$

Donde θ es el parámetro o vector de parámetros a resolver. Normalmente se llama $\mathscr{L}$ a la función anterior, aunque a nivel de cálculo se prefiere trabajar con el logaritmo:

$$\mathscr{L}(\theta) = log\mathscr{L} = log\prod_{i=1}^{n} f(X_i;\theta) \tag{14.4}$$

Entonces el parámetro o vector de parámetros se puede escribir como:

$$\hat{\theta}_{MLE} = argmax(\mathscr{L}(\theta;x_1,\ldots,x_n)) \tag{14.5}$$

Este estimador se puede deducir derivando con respecto a θ e igualando a cero, aunque en ocasiones habrá que usar métodos numéricos para resolver el conjunto de ecuaciones resultantes, lo que se describe en la siguiente caja en la página siguiente.

Algoritmo *Expectation Maximization*, *EM*. Este algoritmo permite encontrar un estimador de *máxima verosimilitud* cuando no tenemos opción de resolver las ecuaciones correspondientes de forma analítica. Para ello se utiliza el siguiente artificio: suponemos que tenemos una variable Z, llamada variable *latente*, tal que $f(y,\theta) = \int f(y,z;\theta)dz$ y tal que la función $f(y,z;\theta)$ resulta fácil de maximizar. Cambiemos ligeramente la notación ahora, según ALEXIS ROCHE [20], a quien seguimos en esta derivación: sea L la función logaritmo de la verosimilitud, esto es, $L(\theta) = log\ p(y|\theta)$. Calculemos ahora la diferencia entre dos estimaciones de nuestro parámetro e introduzcamos variables latentes en lugares oportunos:

$$\begin{aligned} L(\theta) - L(\theta') &= log\frac{p(y \mid \theta)}{p(y \mid \theta')} \\ &= log\int \frac{p(z,y \mid \theta)}{p(y \mid \theta')}dz \\ &= log\int \frac{p(z,y \mid \theta)}{p(z,y \mid \theta')}p(z \mid y,\theta')dz \\ &= log\int \frac{p(z \mid \theta)}{p(z \mid \theta')}p(z \mid y,\theta')dz \\ &\geq \int log\frac{p(z \mid \theta)}{p(z \mid \theta')}p(z \mid y,\theta')dz \\ &= Q(\theta,\theta') \end{aligned} \tag{14.6}$$

Donde hemos usado las reglas usuales de probabilidad total y condicionada (anexo G en la página 1019) para simplificar términos y en el último paso se ha usado la *desigualdad de* JENSEN $Eg(X) \leq g(E(X))$ si la función es cóncava, y el logaritmo lo es.

La última función puede escribirse como: $Q(\theta,\theta') = Q(\theta \mid \theta') - Q(\theta' \mid \theta')$, donde

$$\begin{aligned} Q(\theta \mid \theta') &= \int log p(z \mid \theta)p(z \mid y,\theta')dz = \\ &= E(log\, p(Z \mid \theta) \mid y,\theta') \end{aligned} \tag{14.7}$$

Si θ' permanece fijo, maximizar $Q(\theta,\theta')$ es equivalente a maximizar $Q(\theta \mid \theta')$. Podemos introducir una función residuo $R(\theta \mid \theta')$ tal que $L(\theta) = Q(\theta \mid \theta') + R(\theta \mid \theta')$. El algoritmo *EM* reemplaza $L(\theta)$ por $Q(\theta \mid \theta')$, donde el residuo no será importante pues ya hemos visto de las ecuaciones anteriores que $R(\theta \mid \theta') \geq R(\theta' \mid \theta')$.

El algoritmo lo que hace es, en la E de *expectation*: calcular la función $Q(\theta \mid \theta_n)$, que por definición es una esperanza. Nótese que para este paso hay que calcular una distribución $p(z \mid y,\theta_n)$ en función de nuestra variable latente que hemos escogido, Z.

En la fase M de *maximization*: encontrar el parámetro que maximiza la función, esto es,

$$\theta_{n+1} = argmax\ Q(\theta \mid \theta_n) \tag{14.8}$$

Y el proceso se repite iterativamente hasta que se encuentra una convergencia o se llega a un número previsto máximo de iteraciones.

Regresiones logística LR y logística extendida ELR. Comentemos por último la *regresión logística* [10, 11] y la *regresión logística extendida* [27], *LR* y *ELR*, respectivamente. En una regresión logística se trabaja de forma muy parecida a una regresión lineal, aunque el algoritmo en este caso trabaja con umbrales, clasificando si el valor de nuestro predictando cae por debajo o por encima del umbral. La regresión logística extendida es esencialmente lo mismo que la regresión logística pero tratando de corregir errores que tiene la regresión logística: errores de consistencia asociados a diferentes umbrales y a la necesidad de interpolar entre umbrales. La ELR busca también minimizar el tiempo de cálculo.

Regresión logística LR. A pesar del nombre, es realmente un *algoritmo de clasificación*, es decir, la variable dependiente tiene como rango un valor binario (0 ó 1, típicamente). En meteorología se usa un umbral al que llamaremos q tal que tenemos dos posibilidades: estar por debajo o justo en el umbral o estar por encima. La distribución de probabilidad acumulada (*CDF*) sería:

$$CDF(q) = PDF(y \leq q) = \frac{e^{f(x)}}{1+e^{f(x)}} \tag{14.9}$$

Con $f(x) = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$ una *regresión multilineal* donde las x son los predictores y las b los parámetros a estimar. La ecuación para la *CDF* tiene una característica forma de S con valores entre 0 y 1. La palabra logística viene de que la ecuación para la regresión es lineal si vamos a la escala logística (es decir, si tomamos logaritmos):

$$ln\left(\frac{CDF(q)}{1+CDF(q)}\right) = f(x) \tag{14.10}$$

Los predictores que se suelen usar tienden a ser la media del ensemble, su dispersión o una combinación de ellos [4]. Los parámetros q normalmente provienen de cuantiles derivados de la climatología.

Problemas asociados a LR. Existen, sin embargo, algunos problemas asociados a esta utilización de la regresión logística. Por un lado, por cada umbral q hay que llevar a cabo una regresión, lo que aumenta mucho el gasto computacional si queremos calibrar para muchos umbrales. Además, si en algún momento trabajásemos con un umbral intermedio habría que llevar a cabo una interpolación y eso no suele ser estadísticamente muy consistente. Por último, en ocasiones, puede suceder que aunque $q_1 < q_2$, para algunos valores de los predictores tengamos $CDF(q_1) > CDF(q_2)$, debido a la imprecisión inherente a toda regresión, lo cual es lógicamente inconsistente.

Regresión Logística Extendida ELR. Por los problemas asociados a la LR, WILKS propuso la *ELR* (Extended Logistic Regression, [27]), que incluye una función $g(q)$ que aumenta progresivamente con el umbral q:

$$CDF(q) = PDF(y \leq q) = \frac{e^{f(x)+g(q)}}{1+e^{f(x)+g(q)}} \quad (14.11)$$

Este método solucionaría los problemas anteriormente expuestos. Se han hecho pruebas usando como función $g(q)$ (que tiene que ser estrictamente creciente) $g(q) = b\sqrt{q}$ y como $f(x)$ la función $f(x) = b_0 + b_1\overline{(\sqrt{x})}_{ens}$, con resultados bastante buenos [24]

Otros métodos estadísticos modernos de posproceso de SPC, no descritos aquí por sencillez, son la *regresión gaussiana no homogénea*, Non-homogeneous Gaussian Regression [8, 9] y el *revestimiento ensemble*, Ensemble Dressing [23, 26].

14.1.3 Estratificación o agrupación

Con estos nombres se conocen dos técnicas con las que afrontar el proceso de regresión. La *estratificación* reduce el tamaño de los *datasets* para hacerlos más sensibles al tipo de clima, a la estación del año, o incluso a variables meteorológicas tales como un tipo de dirección de viento [**webcomet**].

La *agrupación* o *pooling* hace lo contrario: incluir en el dataset todos los eventos, sin disgregar. La agrupación suele llevar implícito que el lugar geográfico no es una variable importante, cuando de hecho sí lo es. Para corregir esto se puede introducir una variable ficticia de tipo geoclimático que dé cuenta de las características del terreno o incluso de la frecuencia relativa de determinados eventos [**webcomet**].

Desde la perspectiva de una estadística «clásica», ambos métodos tienen su interés. Desde la perspectiva del Aprendizaje Automático quizá se tiende a preferir la agrupación, ya que lo que interesa es que los datasets sean lo más grande posibles para representar cuantos más casos mejor y porque estos algoritmos son capaces de captar detalles que a los seres humanos se nos escapan. No obstante, es posible que un cierto grado de estratificación, sobre todo geográfica, sea necesaria.

14.2 Aprendizaje Automático

Las técnicas de *aprendizaje automático (AA) (Machine Learning)* son uno de los temas más candentes de la actualidad en ciencia y tecnología. Son un subconjunto de la inmensa área de la inteligencia artificial; es decir, de la creación de agentes digitales cada vez más autónomos. Este nuevo conjunto de modelos de inteligencia artificial ha revolucionado el campo de reconocimiento de patrones, la extracción de significado de datos dispersos (buscar conexiones entre datos y medir su importancia) o la predicción de valores y comportamientos en base a aprendizajes previos, por citar unos ejemplos. Muchas son las técnicas que conforman el Aprendizaje Automático pero destaca especialmente la red neuronal, una imitación y una simplificación algorítmica del cerebro humano. La inteligencia artificial venía teniendo una tendencia ascendente desde hace bastantes años, pero cuando un conjunto de redes neuronales derrotó a un jugador humano de élite al juego oriental del *Go* [25], considerado un reto inalcanzable por los programadores durante años, la atención mundial sobre estos algoritmos se disparó.

Parece claro que en AEMET se debe apostar por estas novedosas técnicas, que son no solo el futuro sino también ya el presente, y que, como veremos, están aportando mejoras reales a los modelos.

14.2.1 Brevísimo panorama del Aprendizaje Automático

Lo que se conoce genéricamente como *Aprendizaje Automático* o *Machine Learning* es un conjunto de técnicas basadas en la estadística clásica que permiten crear modelos que aprenden de datos pasados. Este

conjunto de técnicas es bastante dispar y resulta un tanto difícil de sintetizar en pocas líneas. Vayamos por partes, de lo general a lo concreto, para dar una idea lo más completa posible.

el Aprendizaje Automático es básicamente estadística, aunque tiene varios puntos que la diferencian de una concepción más clásica de la estadística. En primer lugar, podemos decir que el Aprendizaje Automático es mucho menos rigurosa que la estadística desde un punto de vista matemático formal; está menos interesada en aportar pruebas o demostraciones de los resultados y hace más énfasis en la aplicación, para lo cual utiliza la capacidad de cálculo que ofrecen no solo los modernos ordenadores sino sobre todo los chips gráficos (*Graphical Process Unit (GPU)*), que aceleran enormemente los cálculos. Hay diferencias más sutiles, por ejemplo, en *Aprendizaje Automático* lo importante es la precisión del modelo construido, mientras que en estadística se hace más hincapié en cómo llevar a cabo un adecuado modelado de los datos basándose en los adecuados principios teóricos [3]. También la estadística tiene un rango de intereses más amplio, estudiando por ejemplo la mejor forma de hacer muestreos (*samplings*) que pueden luego ser útiles a la hora de resolver integrales por métodos como el de *Montecarlo*, o buscando resultados teóricos sin que necesariamente tengan una inmediata aplicación.

En *Aprendizaje Automático* distinguimos tres enfoques: aprendizaje *supervisado* o *no supervisado*, según el algoritmo tenga para cada *input* un *output* con el que debe corresponderse o no disponga de más información que el dato en bruto; y el tercer enfoque es el del *aprendizaje con refuerzo*: un agente interaccionando con un medio, buscando optimizar una función de coste a largo plazo, que es el que más entronca con la inteligencia artificial y el funcionamiento de la mente humana; es muy interesante pero no es de aplicación directa hoy por hoy a la meteorología (en el futuro, quién sabe, probablemente la acabará teniendo). Dentro de la categoría de aprendizaje supervisado hay dos grandes subgrupos: los problemas de *regresión* y los de *clasificación*. En la clasificación buscamos insertar un dato o conjunto de datos en su correspondiente categoría; la salida, por tanto, de un modelo de clasificación es una *variable categórica*. Ejemplos de problemas de clasificación son el reconocimiento de los dígitos del 1 al 9, el reconocimiento facial, o ya en la meteorología el reconocimiento de nubes vistas por el satélite o del tipo de precipitación, por citar solo algunos de muchos ejemplos. En la *regresión*, de la que nos vamos a ocupar aquí, se busca identificar la función que mejor modela unas entradas con sus correspondientes salidas, las cuales se sitúan en un espectro continuo de posible variación.

Si hablamos de algoritmos específicos, tenemos a nuestra disposición todas las técnicas estadísticas clásicas, como regresión lineal (con o sin parámetros de regularización) o regresión polinómica. Dado que la regresión lineal es muy eficiente, detengámonos un momento en ella en el siguiente cuadro.

Regresión lineal. Busca ajustar a una recta los datos minimizando el error cuadrático medio (MSE, sec. 15.3 en la página 213). Esto fue resuelto por GAUSS ya en el siglo XIX. Para minimizar el *overfitting* o sobreajuste, es decir, para que la función aprenda solo las características del conjunto de datos y no su ruido inherente (y por tanto pierda la capacidad de generalizar) existen varias técnicas. Introducir parámetros de regularización es de lo más usado. Si queremos ajustar a un recta tipo

$$y = w_0 + w_1 x \tag{14.12}$$

las w son llamados pesos. Si $\hat{y}$ es el valor deseado, minimizar con regularización es introducir una λ tal que minimizamos

$$MSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{m} w_j^2 \tag{14.13}$$

al segundo término (básicamente un multiplicador de LAGRANGE) se lo conoce como norma L2 y al conjunto total como *Ridge regression*. Existen otras normas, la L1 que sería el valor absoluto de las w (técnica conocida como *Lasso*) o una combinación lineal de L2 y L1 (*Elastic Net*). En cualquier caso se ve que la idea es penalizar que los valores de w, los pesos, crezcan mucho (y por tanto sobreajusten).

Otro grupo importante, sobre todo gracias al Aprendizaje Automático, son los kernels *gaussianos* (SVR en nuestro trabajo). En esta técnica, quizá la más sofisticada matemáticamente, se proyectan los datos a un espacio de dimensión superior a aquella del que se encuentran originalmente. En este nuevo espacio vectorial de dimensión mayor el problema pasa a ser lineal, aunque el precio a pagar es, como hemos dicho, una considerable sofisticación matemática y un incremento computacional (*curse of dimensionality*).

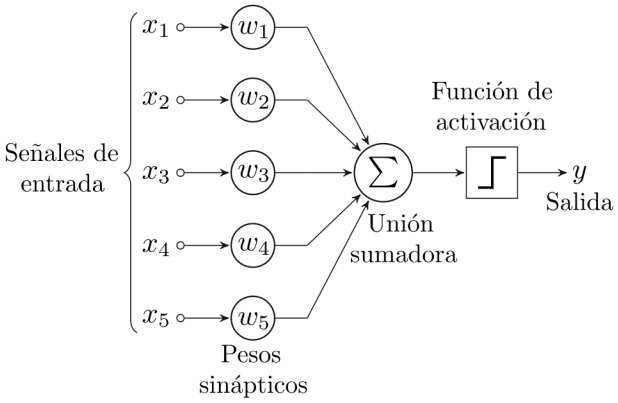

Figura 14.5: El perceptrón*, claramente una copia simplificada de una neurona orgánica (Alejandro Cartas, ver texto)*

La red neuronal es quizá la estrella de la corona. No requiere nada más que álgebra lineal y cálculo, pero es la que más lejos ha llegado, con los resultados más espectaculares. Es una técnica que se conoce desde hace tiempo. Al nivel más básico es el perceptrón (Figura 14.5, Alejandro Cartas https://es.wikipedia.org/wiki/Perceptr%C3%B3n#/media/File:Perceptr%C3%B3n_5_unidades.svg, Licencia Attribution-ShareAlike 4.0 International CC BY-SA 4.0). El perceptrón fue diseñado por FRANK ROSENBLATT en los años 50 y originalmente era una copia digital de como se supone que funciona una neurona en el cerebro humano. Un perceptrón recibe una serie de *inputs* y computa una función con esos *inputs* y unos valores internos llamados *pesos*. Según el resultado de esta función sea mayor o menor que cierto umbral, la neurona (perceptrón) emite o no emite señal. El perceptrón, por tanto, era una algoritmo de clasificación binaria. El aprendizaje se produce comparando la señal emitida por el perceptrón con la señal deseada y actualizando los pesos mediante una astuta regla de asignación.

Sin embargo, en los años 60, MARVIN MINSKY y SEYMOUR PAPERT demostraron que el perceptrón solo resolvía problemas linealmente separables (pusieron el famoso ejemplo de que no podía aprender una función *XOR*) y la técnica quedó un tanto olvidada [15].

Ha adquirido nuevo vigor actualmente con diseños de todo tipo que esquivan las limitaciones del perceptrón: introduciendo una forma de aprendizaje llamada *backpropagation*, permitiendo redes con muchas capas de neuronas entre las que cabe destacar las redes neuronales con recurrencia (RNN) o las convolucionadas (CNN, Figura 14.6), diseños que entrarían en lo que se conoce como *deep learning* y que es uno de los temas más en el candelero de la investigación en este campo en la actualidad.

Por último, y dado que aparecen en algunas gráficas de las que vamos a mostrar, tenemos las técnicas de *boosting*, que son la creación de árboles de decisión en las que se busca el camino que mejor identifica cada input con su output. Estas técnicas han alcanzado gran éxito cuando son combinadas con un algoritmo de descenso del *gradiente* (un método numérico en el que se minimiza la función siguiendo la dirección de máximo descenso del gradiente).

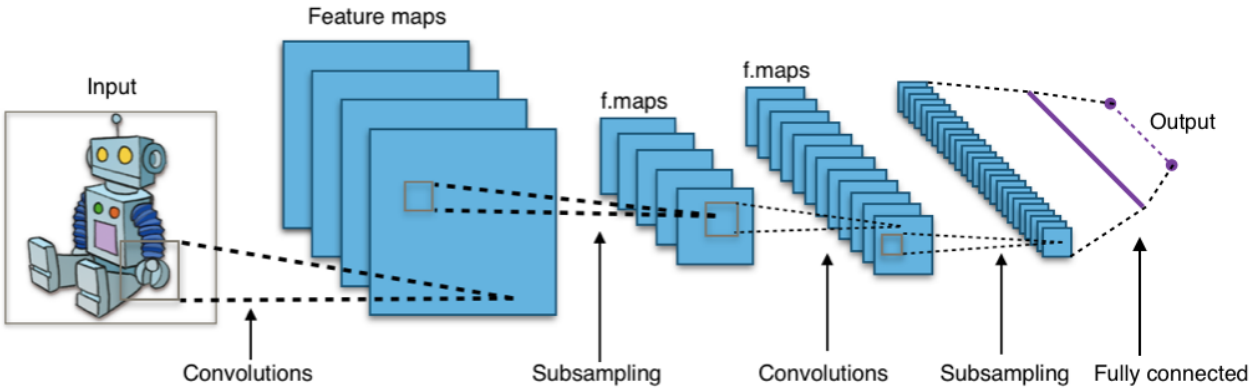

Figura 14.6: Ejemplo de una red neuronal más compleja: una Convolutional Neural Network, de gran éxito para clasificar imágenes. (Aphex34, https://upload.wikimedia.org/wikipedia/commons/6/63/Typical_cnn.png) Licencia Attribution-ShareAlike 4.0 International CC BY-SA 4.0

14.2.2 Aplicación en el posproceso de AEMET-γSREPS

En el futuro SPC de corto plazo de AEMET se dispondrá de un total de 20 miembros, resultado de la combinación de varios modelos con diferentes condiciones de contorno y condiciones iniciales. Cuando estos modelos son ejecutados y resuelven las ecuaciones de la física y la dinámica atmosférica devuelven una salida de cada variable (temperatura, viento, etc.). Estas salidas se pueden mejorar todavía más gracias al posproceso estadístico, que permite captar características a muy pequeña escala que a los modelos se les pueden escapar (sec. 15.4.0.0.4 en la página 218).

Si tenemos un conjunto de datos al que consideramos «bueno», por ejemplo, el análisis o la observación, podemos utilizarlo para realizar una regresión con las salidas de nuestro modelo. Bastaría comparar punto a punto en el espacio (malla, en los modelos) y hora a hora en el tiempo. La comparación hora a hora se puede hacer extrayendo los datos y tabulándolos adecuadamente, la comparación punto a punto es más delicada. Si lo que se hace es comparar con el análisis no hay problema: el punto de predicción y el punto del análisis son coincidentes. Si se compara con la observación es extremadamente improbable que los puntos coincidan. Para resolver este problema se puede apelar a un algoritmo que traslade el punto de predicción al de la observación o viceversa.

Sin embargo, nosotros no hemos optado por este enfoque. Tampoco hemos trabajado con el punto más cercano a la estación que toma los datos. Lo que hemos hecho ha sido hacer un promedio de los cuatro puntos más cercanos al punto deseado y considerar el resultado de ese promedio como el valor más real. Así suavizamos posibles problemas, como el hecho de que el punto caiga en el mar en lugar de caer en tierra, o cambios bruscos e irreales en la orografía de los modelos.

Un comentario importante: aunque hemos trabajado con el análisis, una verdadera calibración debe hacerse con las observaciones, así que los resultados del análisis son orientativos. Los verdaderos resultados de cara a una futura calibración serían siempre frente a las observaciones.

14.2.3 La temperatura: análisis y observación

Para abordar el posproceso de la temperatura se tomaron tres aeropuertos muy importantes del país que representaban diferentes condiciones de orografía, cercanía o no al mar y clima. Estos aeropuertos fueron los de Madrid, Barcelona y Palma de Mallorca. Posteriormente se añadieron los de Málaga y Vigo para el caso de la observación. Se empezó con el análisis, trabajando con el promedio de los cuatro puntos más cercanos a uno dado, comparando valores de análisis con valores de predicción. Posteriormente se trabajó con la observación: el valor observado se consideraba el «real» y no se promediaba con nada, sí se promediaban los valores de predicción de los cuatro puntos más cercanos al punto de observación.

> Los datos de los modelos se insertan en los diversos algoritmos, usando para ello el entorno *Python* [21, 22] con la librería *scikit-learn* [18]. Para la red neuronal se usó una librería de Python llamada *Keras* [5], una especie de capa a alto nivel que permite trabajar con *Theano* o *TensorFlow*, los dos grandes estándares de librerías para redes neuronales.

En el caso del análisis, dado que es la observación suavizada por la predicción de muy corto plazo, se llevó a cabo un control de calidad relativamente básico, consistente en comprobar que no había *outliers* o *valores atípicos* (sec. 13.6.7 en la página 183) en los datos (temperaturas anormalmente altas o bajas). Para la observación sí se implementó un control más exhaustivo: la lectura de temperatura se hacía del *Meteorological Terminal Aviation Routine Weather Report (METAR)* (un parte de observación de distintas variables meteorológicas que emiten las oficinas meteorológicas aeronáuticas, Oficina Meteorológica Aeronáutica (OMA), de los aeropuertos). Como el METAR se obtiene cada hora (y en muchos aeropuertos cada media hora), se descartaron medidas que diferían en más de 5 grados con el promedio de la temperatura a H-3 y a H+3 horas. Es posible que se descartara alguna medida válida, pero estábamos obligados a llevar a cabo un control de calidad exhaustivo pues en ocasiones la observación en bruto puede tener *outliers*. La razón de escoger los intervalos cada 3 horas reside en que el modelo genera salidas precisamente cada 3 horas.

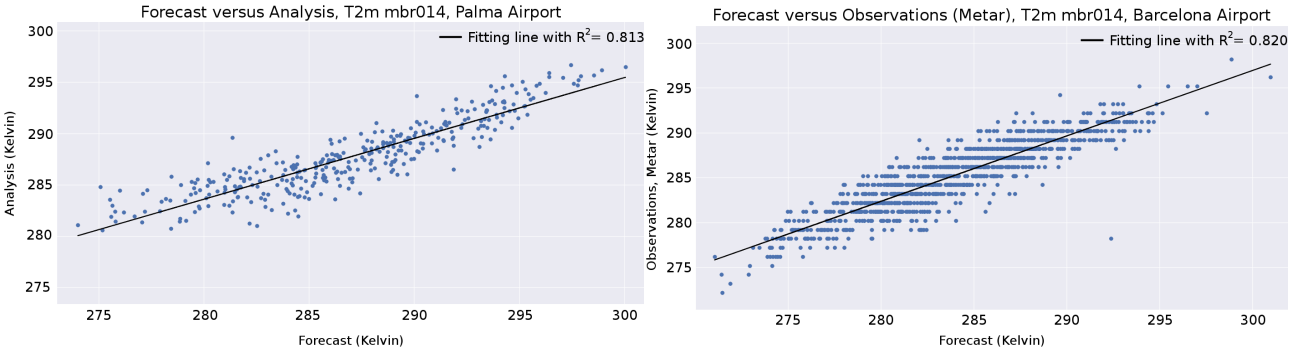

Figura 14.7: Diagramas de dispersión de temperatura. Izquierda: predicción frente a análisis, para Palma. Derecha: predicción frente a observación, para Barcelona

Tanto para el análisis como para la observación se generó un *scatter plot* (sec. 15.2.2 en la página 210) de los datos de observación o análisis frente a los de predicción con un coeficiente R^2 (sec. 15.3 en la página 213) para medir la bondad del ajuste a una recta. Si el modelo fuese perfecto deberíamos ver una recta perfecta con una pendiente de 45°. Lo que vemos es un ajuste bastante razonable a una recta, tanto con el análisis como con la observación (Figura 14.7, izquierda y derecha respectivamente). Nótese que frente a la observación generamos muchos más puntos que frente al análisis, ya que el análisis se genera cada 12 horas y la observación METAR cada hora o incluso media hora (las predicciones, como se dijo, son siempre cada 3 horas).

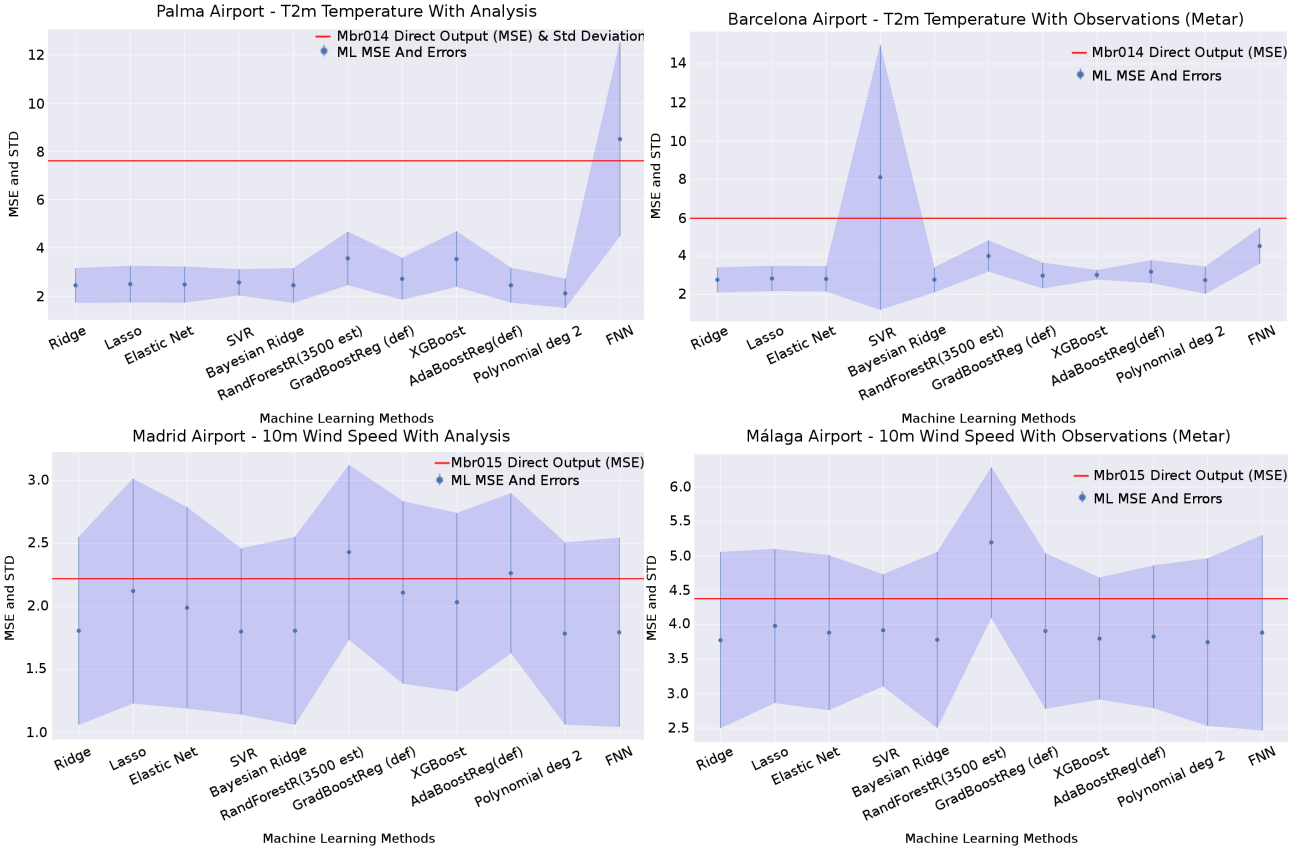

Figura 14.8: Métodos de Aprendizaje Automático.

Posteriormente, se llevaba a cabo el *learning* estadístico. Los datos se pasaban por los diversos modelos de *Aprendizaje Automático* y se les hacía *validación cruzada*. La *validación cruzada* es una técnica estadística consistente en dividir un conjunto de datos en un número n de grupos y utilizar cada vez $n-1$ grupos para hacer el *learning* y el grupo restante para verificar la bondad de dicho *learning*. Dado que se recorren todos los grupos del conjunto, se puede hacer un promedio tanto del error como de su desviación típica, y este resultado será siempre más representativo que tomar una sección cualquiera de los datos y hacer el *learning* y la verificación una sola vez. Podemos ver ejemplos en la Figura 14.8 en la página anterior. En ocasiones se hacía *validación cruzada anidada*, una *validación cruzada* con dos niveles en los que se usa un nivel de *validación cruzada* solo para seleccionar los parámetros que mejor ajustan y con ellos se elabora, en el segundo nivel, la *validación cruzada* propiamente dicha. Esto es debido a que algunas técnicas dependen de multitud de hiperparámetros que tienen un amplio rango de variación y por tanto el *learning* es mucho menos inmediato que con por ejemplo una regresión lineal.

14.2.4 El viento: análisis y observación

Para la variable del viento el patrón de estudio es el mismo que con la temperatura. Nos centramos en la magnitud, no en la dirección, ya que consideramos que la dirección es menos determinante y además distrae parte del *learning* en mejoras de pocos grados, algo que nos parece de escasa importancia frente a los cambios en velocidad. En el momento de escribir este capítulo aún no se había terminado de estudiar el viento con todos los miembros del SPC, pero ciertos esquemas empezaban a verse claros.

En el caso del viento también se llevó a cabo para las observaciones un control de calidad análogo al de la temperatura, aunque en este caso es mucho más difícil poder decir con relativa seguridad cuándo un dato es o no es un *outlier*. La temperatura, incluso en casos de olas de frío o de calor, experimenta una variación que puede ser considerada aproximadamente lineal en el tiempo, pero el viento tiene continuos saltos y discontinuidades en cuestión de segundos. Si se eligen umbrales muy rígidos se corre el riesgo de descartar muchos datos que podrían ser válidos, así que se optó de forma conservadora por descartar solo los *outliers* más gruesos: aquellos que difiriesen en más de 15 m/s en la hora H con el promedio de las horas H-3 y H+3. Aun así, es posible que se hayan descartado datos válidos, o que se hayan incluido datos incorrectos pero que son totalmente indistinguibles de uno bueno. Cualquier persona con experiencia en la observación o la predicción habrá visto en multitud de ocasiones un aeropuerto con dos cabeceras de pista, situadas a poco más de un kilómetro una de la otra, midiendo vientos totalmente diferentes.

Como con la temperatura, generamos *scatter plots* del análisis frente a la predicción y de la observación frente a la predicción. A diferencia del buen comportamiento con la temperatura, vemos que la bondad del ajuste a una recta es bastante peor. Dado que el coeficiente R^2 es independiente de la escala, nos sirve como regla estándar para calibrar ajustes. R^2 tiene sus propias limitaciones [16]: se basa en la comparación del modelo de trabajo con un modelo constante, el de la media, no con el verdadero modelo; no obstante, obviaremos estos refinamientos y usaremos R^2.

Un peor ajuste como el observado puede indicar que o bien la relación deja de ser lineal o el modelo predice algo peor el viento que la temperatura o ambas cosas. Teniendo en cuenta las discontinuidades del viento de las que antes hablábamos o su extrema dependencia del punto en cuestión (dependencia asociada a cambios orográficos, obstáculos, etc.) parece probable que el modelo recoja el viento con más imprecisión; lo que no es óbice para pensar también que la relación entre las variables pueda no ser del todo lineal.

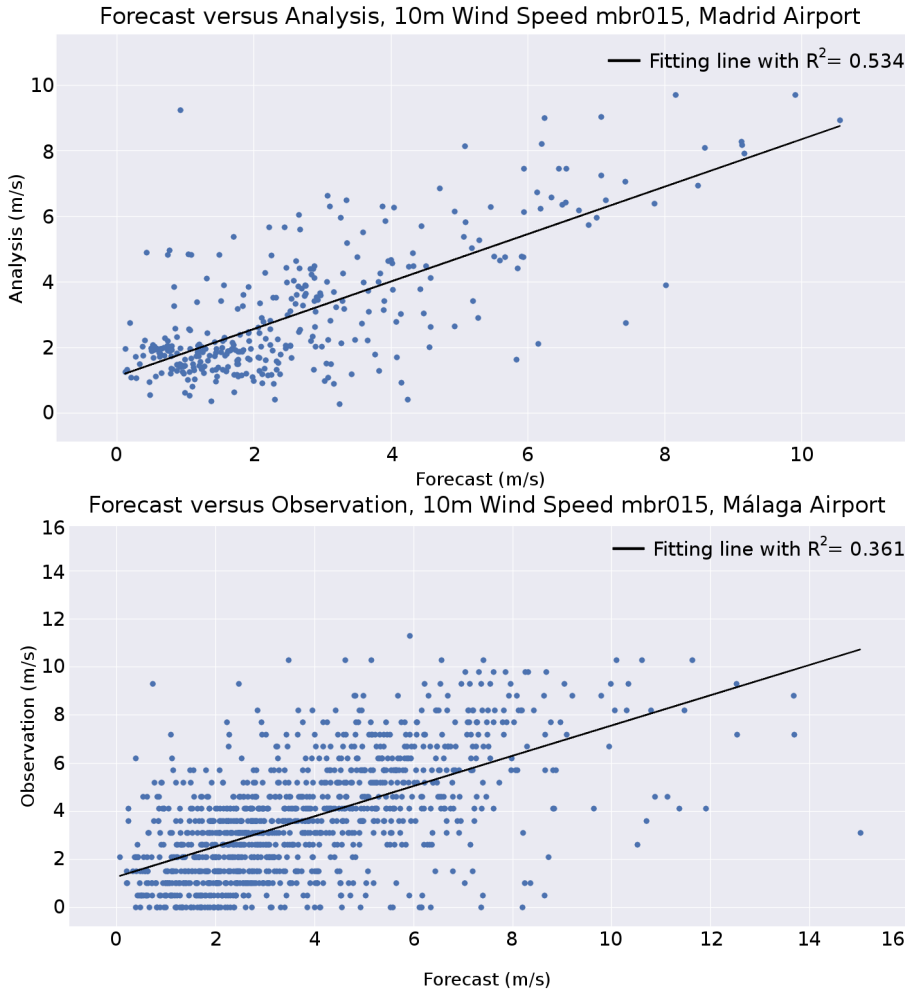

Figura 14.9: Diagramas de dispersión. Arriba: predicción frente a análisis, para Madrid. Abajo: predicción frente a observación, para Málaga.

Tenemos dos ejemplos en la Figura 14.9. El último paso es introducir los diferentes métodos de *Aprendizaje Automático*, Figura 14.8 en la página 202, parte inferior.

14.3 Conclusiones

Nuestro objetivo en este capítulo era, además de presentar las técnicas clásicas y modernas de posproceso estadístico aplicable a las salidas de los modelos meteorológicos, ilustrar cómo se pueden usar las técnicas de *Aprendizaje Automático,* un caso particular, para mejorar las salidas de los citados modelos. Estas técnicas se han usado en conjunción con las técnicas estadísticas más habituales y conocidas anteriormente presentadas.

Hemos mostrado unos pocos ejemplos de entre los muchos que se están generando para cada modelo del γSREPS y para varios puntos (aeropuertos) de España. Parece que el viento se comporta algo peor que la temperatura. Parece también que el Aprendizaje Automático mejora claramente las salidas brutas de los modelos. Es verdad que hay casos (no mostrados aquí) en los que no se consigue mejorar al modelo, pero son una minoría. Hasta ahora se puede decir que la regresión lineal clásica o con regularización (*Ridge*) es lo que en líneas generales mejor está funcionando, al menos en términos del error cuadrático medio (MSE, sec. 15.3 en la página 213); esto es especialmente cierto en el caso de la temperatura y en el caso del viento estaría a la par con la red neuronal, aunque pensamos que tiende también a superarla.

Por supuesto, todo este trabajo aún está en pleno proceso de elaboración. Hay muchas cosas que pueden probarse. Tras el viento se pretende abordar la precipitación, variable especialmente delicada, pues sabemos no sigue en absoluto una distribución normal y es aún más sensible a cambios en pocos metros de terreno de lo que es el viento. Veremos si los métodos de *Aprendizaje Automático* son capaces de ofrecer alguna mejora frente a la salida en bruto del modelo. También se

pretende probar nuevos y diversos diseños, sobre todo de la red neuronal, un área que crece cada día que pasa y en la que el número de arquitecturas posibles junto con el de sus hiperparámetros asociados es todo un mundo a explorar. Al fin y al cabo, las técnicas más sofisticadas (básicamente el *deep learning*) son las que acumulan los mayores éxitos, aunque también es cierto que se centran más en la clasificación que en la regresión. De todas formas es difícil asegurar qué técnica va a ir mejor a priori, y de hecho hay un teorema de *Aprendizaje Automático* que afirma que no existe un algoritmo único y universal que valga para todos los problemas [12].

Por último, quisiéramos mencionar que todo lo que se ha hecho es para puntos concretos, y que los parámetros aprendidos para un determinado lugar (por ejemplo, Vigo) difícilmente van a ser exportables a otro lugar (por ejemplo, Madrid) con características físicas y climáticas totalmente distintas. Sería deseable poder generalizar los parámetros aprendidos a regiones más extensas. Existen diversas ideas en este sentido. Estaríamos ya hablando de un plan sumamente ambicioso que requeriría un equipo de varias personas y grandes recursos de computación, pero sin duda una prometedora línea de desarrollo e investigación para el futuro.

Anexo. Poco antes de la publicación de este libro, un artículo demostraba cómo un sistema de AA superaba, por primera vez en la historia, a un modelo físico [17]. El caso estudiado fue el de la propagación de un frente de incendio, un sistema altamente caótico. Uno de los investigadores llegó a sugerir que un día toda la predicción del tiempo se hará con modelos de AA en lugar de con modelos físicos, aunque otros investigadores fueron más comedidos [28]. Anteriormente, algunos algoritmos habían igualado a modelos físicos; por ejemplo, un sistema de inteligencia artificial aprendió la dinámica del tiro parabólico exclusivamente de los datos, sin haberle programado ni leyes de NEWTON, ni gravedad, ni nada físico. En esta ocasión, sin embargo, el algoritmo de AA no solo ha aprendido sino que ha mejorado las predicciones de un complicado modelo basado en la física. Un resultado prometedor y que demuestra la atención que merecen estos algoritmos. Eso sí, probablemente para adquirir verdadera comprensión (o lo que entendemos por comprensión humana) sea imprescindible recurrir al conocimiento de la física. El debate está abierto y trasciende los dominios de la física o las matemáticas para adentrarse en terrenos, tal vez más resbaladizos, de teoría del conocimiento y filosofía.

14.4 Referencias

[1] ANTOLIK, Mark S y BRANCH, Statistical Modeling. "Model Output Statistics (MOS) - Objective Interpretation of NWP model output". En: *University of Maryland USA* (2003) (citado en página 195).

[2] AYUSO, Juan José. *Método de predicción perfecta, para adaptación estadística de los diferentes modelos numéricos, utilizando los análisis del modelo de área limitada del INM*. Informe técnico. 1996 (citado en página 195).

[3] BREIMAN, Leo y col. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". En: *Statistical science* 16.3 (2001), páginas 199-231 (citado en página 199).

[4] CALLADO, Alfons y col. "Ensemble Forecasting". En: *Climate Change and Regional/Local Responses*. Editado por RAY, Pallav. InTech, mayo de 2013. ISBN: 978-953-51-1132-0. DOI: 10.5772/55699 (citado en página 197).

[5] CHOLLET, François y col. *Keras*. 2015. URL: https://github.com/fchollet/keras (citado en página 201).

[6] DEMPSTER, Arthur P, LAIRD, Nan M y RUBIN, Donald B. "Maximum likelihood from incomplete data via the EM algorithm". En: *Journal of the royal statistical society. Series B (methodological)* (1977), páginas 1-38 (citado en página 196).

[7] GLAHN, Harry R y LOWRY, Dale A. "The use of model output statistics (MOS) in objective weather forecasting". En: *Journal of applied meteorology* 11.8 (1972), páginas 1203-1211 (citado en página 195).

[8] GNEITING, Tilmann y RAFTERY, Adrian E. "Weather Forecasting with Ensemble Methods". En: *Science* 310.October (2005), páginas 248-249 (citado en página 198).

[9] GNEITING, Tilmann y col. "Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation". En: *Monthly Weather Review* 133.5 (2005), páginas 1098-1118. ISSN: 0027-0644. DOI: 10.1175/MWR2904.1 (citado en página 198).

[10] HAGEDORN, Renate y col. "Probabilistic forecast calibration using ECMWF and GFS ensemble reforecasts. Part II: Precipitation". EN. En: *Monthly Weather Review* 136.7 (jul. de 2008), páginas 2620-2632. ISSN: 0027-0644. DOI: 10.1175/2007MWR2410.1 (citado en página 197).

[11] HAMILL, Thomas M., WHITAKER, Jeffrey S. y WEI, Xue. *Recalibrating mrf re-forecasts: a logistic regression approach*. Informe técnico 303. 2001, páginas 1-8 (citado en página 197).

[12] HO, Y.C. C y PEPYNE, D.L. L. "Simple Explanation of the No-Free-Lunch Theorem and Its Implications". En: *Journal of Optimization Theory and Applications* 115.3 (dic. de 2002), páginas 549-570. ISSN: 0022-3239. DOI: 10.1023/A:1021251113462 (citado en página 205).

[13] KLEIN, William H, LEWIS, Billy M y ENGER, Isadore. "Objective prediction of five-day mean temperatures during winter". En: *Journal of Meteorology* 16.6 (1959), páginas 672-682 (citado en página 194).

[14] MCLACHLAN, Geoffrey J y KRISHNAN, Thriyambakam. *The EM algorithm and extensions*. Volumen 382. John Wiley & Sons, 2007, página 359. ISBN: 0471123587. DOI: 10.2307/1271189 (citado en página 196).

[15] MINSKY, Marvin, PAPERT, Seymour y BOTTOU, Leòn. *Perceptrons : an introduction to computational geometry*. 1969, página 292. ISBN: 9780262534772 (citado en página 200).

[16] MYLES WHITE, J. *Why I'm Not a Fan of R-Squared*. 2016. URL: http://www.johnmyleswhite.com/notebook/2016/07/23/why-im-not-a-fan-of-r-squared/ (visitado 23-07-2016) (citado en página 203).

[17] PATHAK, Jaideep y col. "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data". En: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.12 (2017), página 121102 (citado en página 205).

[18] PEDREGOSA, Fabian y col. "Scikit-learn: Machine learning in Python". En: *Journal of Machine Learning Research* 12.Oct (2011), páginas 2825-2830 (citado en página 201).

[19] RAFTERY, Adrian E. y col. "Using Bayesian model averaging to calibrate forecast ensembles". En: *Monthly Weather Review* 133.5 (2005), páginas 1155-1174. ISSN: 0027-0644. DOI: 10.1175/MWR2906.1 (citado en página 195).

[20] ROCHE, Alexis. "EM algorithm and variants: An informal tutorial". En: *arXiv preprint arXiv:1105.1476* (2011) (citado en página 197).

[21] ROSSUM, G van. "Python tutorial, Technical Report CS-R9526". En: (1995). URL: https://docs.python.org/3/tutorial/ (citado en página 201).

[22] ROSSUM, G van. "Python Language Reference, version 2.7". En: (2016). URL: https://docs.python.org/2/reference/index.html (citado en página 201).

[23] ROULSTON, Mark S. y SMITH, Leonard A. "Combining dynamical and statistical ensembles". En: *Tellus A* 55.1 (2003), páginas 16-30. DOI: 10.3402/tellusa.v55i1.12082 (citado en página 198).

[24] SCHMEITS, Maurice J y KOK, Kees J. "A Comparison between Raw Ensemble Output, (Modified) Bayesian Model Averaging, and Extended Logistic Regression Using ECMWF Ensemble Precipitation Reforecasts". En: *Monthly Weather Review* 138.11 (2010), páginas 4199-4211. ISSN: 0027-0644. DOI: 10.1175/2010MWR3285.1 (citado en página 198).

[25] VV. AA. "La inteligencia artificial gana al campeón humano de Go". En: *El País* 15-03-2016 (2016). URL: https://elpais.com/elpais/2016/03/15/ciencia/1458034897_194344.html (citado en página 198).

[26] WANG, Xuguang y BISHOP, Craig H. "Improvement of ensemble reliability with a new dressing kernel". En: *Quarterly Journal of the Royal Meteorological Society* 131.607 (2005), páginas 965-986. DOI: 10.1256/qj.04.120 (citado en página 198).

[27] WILKS, Daniel S. "Extending logistic regression to provide full-probability-distribution MOS forecasts". En: *Meteorological Applications* 16.3 (2009), páginas 361-368. ISSN: 13504827. DOI: 10.1002/met.134 (citado en páginas 197, 198).

[28] WOLCHOVER, Natalie. "Machine Learning's .Amazing.Ability to Predict Chaos". En: (2018). URL: https://www.quantamagazine.org/machine-learnings-amazing-ability-to-predict-chaos-20180418/ (citado en página 205).