MÁSTER EN INGENIERÍA DE SISTEMAS DE DECISIÓN

Escuela de Másteres Oficiales de la Universidad Rey Juan Carlos

Curso académico 2019-2020

Trabajo Fin de Máster

Previsión de energía eólica mediante técnicas de Machine Learning

Tutor: Javier Martínez Moguerza
Autor: Víctor Chaparro Parra

Resumen

El uso de energías renovables, como la eólica o la solar, se ha convertido en una pieza clave en el actual escenario de cambio climático que experimenta el planeta. Los métodos de predicción son herramientas clave para conseguir que este tipo de fuentes de energía sean competitivas en el mercado energético actual. En este trabajo se han utilizado técnicas de Machine Learning para prever la energía eólica en varios parques ubicados en distintas localizaciones geográficas. Se obtuvieron las predicciones mediante cuatro algoritmos: máquinas de vector soporte (SVM), k vecinos más próximos (KNN), bosques aleatorios (RF) y splines de regresión adaptativa multivariante (MARS). Se ha desarrollado una aplicación de software flexible y configurable que automatiza la construcción de los modelos y facilita su análisis. Los resultados obtenidos, aunque tienen un amplio margen de mejora, han demostrado la eficacia de estas técnicas en el problema planteado.

Abstract

Renewable energies, such as wind or solar power, have become a key element in the current climate change scenario that the planet is experiencing. Forecasting methods are key tools to make this type of energy source competitive in today's energy market. In this work, Machine Learning techniques are applied to wind power forecasting in several wind farms. The predictions have been obtained using four different algorithms: Support Vector Machines (SVM), k-Nearest Neighbors (KNN), Random Forests (RF), and Multivariate Adaptive Regression Splines (MARS). A flexible and configurable software tool has been developed to easily build and analyze the models. The results, although having a wide margin for improvement, have demonstrated the effectiveness of these techniques in the problem posed.

A mi familia...

Índice

1 Introducción

En el contexto actual de cambio climático, las energías renovables, como alternativa a las de origen fósil, están empezando a cobrar un papel fundamental, siendo la energía eólica una de las más relevantes y prometedoras. Sin embargo, su naturaleza aleatoria e intermitente constituye un reto a superar debido a que la demanda energética tiene un carácter básicamente continuo.

La mejora en los métodos de predicción de energía eólica se presenta como una de las soluciones clave a este problema, ya que permite reducir costes y mejorar la fiabilidad en la integración de este tipo de producción energética [7]. Estas predicciones son necesarias en varios contextos. Por ejemplo, para el mercado energético, resulta de utilidad la predicción total en una región, así como para la planificación de instalación de un parque en un punto geográfico concreto, o tareas relacionadas con el mantenimiento de los parques.

En este trabajo se aborda el problema de la previsión de energía mediante la utilización de técnicas de aprendizaje automático, analizando y comparando la precisión de distintos algoritmos para un conjunto de parques eólicos situados en diferentes localizaciones geográficas, desarrollándose una aplicación que automatiza todo el proceso necesario para construir los modelos a partir de los datos en bruto.

1.1 Planteamiento del problema

El problema objeto de estudio ha sido propuesto en el contexto de una competición[1] de Ciencia de Datos organizada anualmente por dos instituciones educativas francesas, la École Normale (ENS)[2] y el Collège de France[3]. Varias empresas y laboratorios proponen retos de aprendizaje automático supervisado, facilitando datos reales y abarcando un amplio espectro de aplicaciones y escenarios (consumo energético, energías renovables, medicina, logística, etc.).

De entre todos los retos propuestos, el elegido es el que ofrece el operador de energías renovables Compagnie Nationale du Rhone (CNR) con el título *Wind power forecasting for the day-ahead energy market*. El reto se centra en la predicción de producción de energía eólica en seis de sus parques eólicos. Diariamente, CNR participa en el *mercado energético* vendiendo su producción estimada para el día siguiente. La predicción de la producción energética a ese plazo de tiempo se convierte, por tanto, en pieza clave para poder realizar una buena operación de venta.

1.2 Objetivos

Se persiguen dos objetivos principales:

1. Demostrar la eficacia de los métodos de aprendizaje automático para la predicción de la producción de energía eólica mediante su aplicación a un problema concreto del mercado energético en el que se busca obtener predicciones lo más precisas posibles.

2. Desarrollo de un software que permita automatizar el proceso de análisis y aplicación de los distintos algoritmos a los datos del problema. Se busca que dicha aplicación sea flexible en cuanto a la configuración de experimentos y que admita la inclusión de nuevos algoritmos de manera sencilla, haciendo uso de lenguajes de programación y herramientas actuales que permitan el desarrollo e implementación de la aplicación utilizando buenas prácticas de Ingeniería del Software.

[1] `https://challengedata.ens.fr/`
[2] `https://www.ens.psl.eu/`
[3] `https://www.college-de-france.fr/site/en-college/index.htm`

2 Marco teórico

2.1 Fundamentos de energía eólica y su predicción.

Los parques eólicos están formados por un conjunto de turbinas que transforman la energía cinética del viento en energía eléctrica. Existen varios tipos de turbinas, siendo las más comunes las de eje horizontal, en las que el eje de rotación de las palas es paralelo a la dirección del flujo de viento (véase Fig. 1).

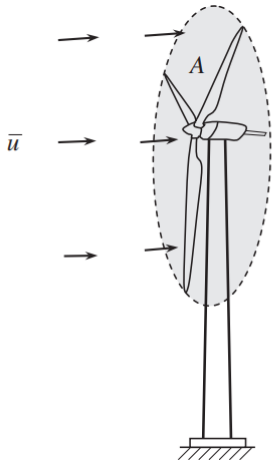

Figura 1: Esquema de una turbina de eje horizontal [Imagen obtenida de [17]].

Se parte de una formulación fundamental en ciencia de datos para describir la producción de energía eólica dada por

$$f_t(y) = \int_{\vec{x}} f_t\left(y|\vec{x}\right) f_t\left(\vec{x}\right) d\vec{x}, \tag{1}$$

donde $f_t(\cdot)$ es una función de densidad de probabilidad que en general varía con el tiempo, para la que se necesita conocer la distribución de probabilidad $f_t(\vec{x})$ del vector de variables meteorológicas $\vec{x}$, así como la distribución de probabilidad condicionada $f_t\left(y|\vec{x}\right)$, esto es, la distribución de la potencia eólica dadas unas condiciones de viento y medioambientales $\vec{x}$ [3].

Variables meteorológicas típicas son:

- V, velocidad del viento.
- D, dirección del viento.
- ρ, densidad del aire relacionada con la temperatura T a través de $\rho = P/RT$, siendo P la presión atmosférica y R la constante de los gases ideales.
- H, humedad relativa del aire.
- I, intensidad de las turbulencias del viento dada por $I = \hat{\sigma}/\bar{V}$.
- S, cizalladura, esto es, la variación de la velocidad del viento en función de la altura. Conociendo la velocidad V_1, V_2 a dos alturas distintas h_1 y h_2, se tiene que $S = \ln\left(\frac{V_2}{V_1}\right) \ln\left(\frac{h_2}{h_1}\right)$.

Tal y como se detalla en [3], el objetivo es modelizar la distribución $f_t\left(y|\vec{x}\right)$ con el fin de realizar predicciones de la producción eólica en función de las variables meteorológicas. Cuando $\vec{x}$ se reduce a la velocidad del viento, el valor medio de f, $\mathbb{E}\left(y|\vec{x}\right)$, es la **curva de potencia** (Fig. 2).

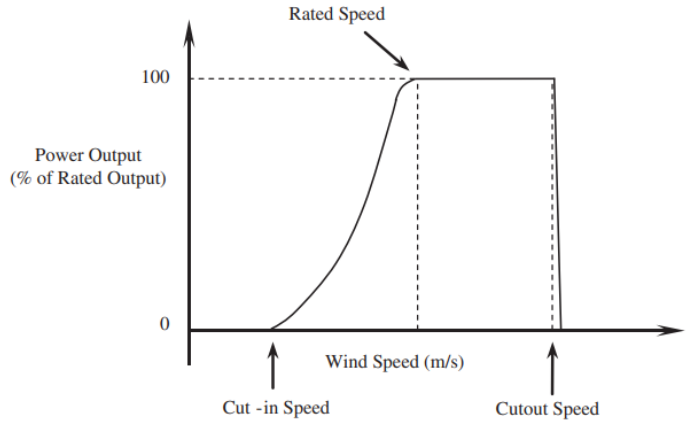

Figura 2: Curva de potencia típica de una turbina eólica [Imagen obtenida de [17]].

Suele ser facilitada por el fabricante de la turbina y determina la producción generada en función de la velocidad del viento. *Cut-in* es la velocidad mínima a la que se empieza a generar potencia, *Rated speed* es la velocidad a la que se alcanza la potencia nominal y *Cut-out* es la velocidad de corte a la que la turbina se detiene para evitar daños.

La ecuación física que proporciona la potencia producida por el viento en función de la velocidad del mismo es

$$y = \frac{1}{2} C_p \rho A V^3, \tag{2}$$

donde A es el área barrida por las palas y C_p es el coeficiente de potencia [17]. Este coeficiente depende de V, entre otros factores, y se determina experimentalmente en función de la densidad del aire y la velocidad del viento. Es responsable de que exista una complejidad añadida al aparecer otras relaciones no lineales entre la potencia y la velocidad, además de la que viene dada por el cubo de ésta. Otro factor de complejidad añadido es el control de las turbinas, por ejemplo, apagándolas cuando se alcanza la velocidad *cut-out*, generando una curva como la comentada en la figura 2.

De la ecuación (2) se pueden extraer las siguientes observaciones [10]:

1. Existen al menos tres variables importantes: V, D y ρ.
2. La relación funcional entre las variables de entrada y la respuesta es no lineal, ya que no se conoce de forma explícita la relación de C_p con V, D y ρ. Por esto, no es posible establecer la forma funcional de la curva de potencia.
3. Las variables meteorológicas aparecen multiplicándose, por lo que existen interacciones entre ellas.

La falta de conocimiento sobre la forma funcional explícita entre la potencia y las variables meteorológicas justifica el uso de métodos de aprendizaje automático, concretamente de métodos no paramétricos. Estos métodos, frente a los paramétricos que son más rígidos, ya que asumen de entrada una forma funcional concreta, permiten determinar esta dependencia a través de los datos.

2.2 Métodos de aprendizaje automático

El problema analizado se enmarca dentro de un problema de regresión no lineal supervisada[4]. Existen diferentes algoritmos de aprendizaje automático o *Machine Learning* para abordar este tipo de problemas. A continuación, se detallan los utilizados en este trabajo.

2.2.1 k vecinos más próximos (KNN)

El método de los k vecinos más próximos consiste en determinar una vecindad del punto $\vec{x}_0$ para el que se quiere conocer $\hat{y}$ mediante la definición de una distancia y del número de puntos que conformarán dicha vecindad, de tal manera que las vecindades estarán determinadas por los k puntos que estén más próximos entre sí midiendo esa proximidad con la distancia establecida [6]. El valor de $\hat{y}(x_0)$ vendrá dado simplemente por

$$\hat{y}(x_0) = \frac{1}{k} \sum_{x_i \in \mathfrak{N}_k(x_0)} y_i, \tag{3}$$

donde $\mathfrak{N}_k(x_0)$ representa el conjunto de k vecinos de x_0. Existen varias distancias que se pueden utilizar. En este trabajo se han usado las siguientes:

- Euclídea. La distancia entre dos puntos P y Q viene dada por $d_{PQ} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$, siendo p_i y q_i la coordenada i-ésima de P y Q respectivamente [19].
- Manhattan. La distancia entre P y Q se define como la suma de la diferencia absoluta de las coordenadas, esto es, $d_{PQ} = \sum_{i=1}^{n} |p_i - q_i|$ [21].

El *hiperparámetro*[5] más relevante en este algoritmo es k, siendo necesario encontrar el valor del mismo que permita un buen balance entre la varianza y el sesgo, ya que modelos demasiado complejos se ajustan muy bien a los datos de entrenamiento pero generalizan mal mientras que modelos demasiado simples no son capaces de ajustarse correctamente ni a los datos de entrenamiento ni a los de test.

Una aplicación de este método puede encontrarse en trabajos como el de Mangalova y Agafonov [13], donde se utiliza una combinación de métodos heurísticos y estadísticos para la construcción de un modelo basado en el algoritmo KNN, capaz de predecir la energía eléctrica generada por una planta eólica a medio y largo plazo.

2.2.2 Splines de regresión adaptativa multivariante (MARS)

El método de splines de regresión adaptativa multivariante es una técnica de regresión que busca de manera automática interacciones y relaciones no lineales entre las variables [5]. Este método divide el espacio de predictores en regiones en las que se ajusta un modelo lineal base de una variable. El modelo final es una combinación de estas funciones base o *splines*. En el caso de una dimensión, el modelo viene dado por:

$$\hat{f}(x) = \alpha_0 + \sum_i \alpha_i B_i(x), \tag{4}$$

[4]Los problemas de aprendizaje automático supervisado son aquellos en los que los datos de entrenamiento incluyen los valores de las variables de entrada y su correspondiente valor de salida, tratándose de una regresión si éste es numérico y continuo. Si además la relación entre las variables de entrada y salida no se ajusta a una recta, se habla de regresión no lineal [1].

[5]Los hiperparámetros son parámetros que los modelos no pueden aprender de manera automática durante la fase de entrenamiento y que deben ser introducidos manualmente. Normalmente, los valores óptimos se determinan de forma experimental, mediante prueba y error, o técnicas de validación cruzada.

siendo $\hat{f}$ el valor estimado de la variable respuesta, α_0 la ordenada en el origen, B_i las funciones base y α_i los respectivos coeficientes del ajuste lineal.

Cada función base está dada por el producto de uno o más de los siguientes términos: una constante, funciones lineales de las variables originales de entrada o funciones *hinge* de las mismas. La función hinge es como sigue:

$$h(x-t) = \begin{cases} x-t, & \text{si } x > t, \\ 0, & \text{si } x \leq t. \end{cases}$$

El algoritmo consta de dos etapas:

1. Paso *forward*, donde se buscan términos minimizando el error cuadrático de los datos de entrenamiento.
2. Paso de *pruning* o poda, que selecciona el subconjunto de estos términos de tal manera que se obtenga un valor mínimo en el proceso de validación cruzada generalizada (GCV).

Algunos de los principales hiperparámetros que se pueden ajustar son el número máximo de términos del modelo en la etapa 1, el coeficiente de penalización b para utilizar en el GCV, o el grado máximo de interacción de los términos.

Corrección: En [12], Li et al. aplican esta técnica a la predicción de la producción de energía solar, proponiéndola como una alternativa a otros métodos más tradicionales como la regresión lineal, demostrando su fiabilidad en este tipo de problemas donde la estocasticidad inherente a estas fuentes de energía resulta en relaciones no lineales entre las variables que intervienen.

2.2.3 Máquinas de vector soporte (SVM)

Las máquinas de vector soporte constituyen un método de aprendizaje automático potente y versátil aplicable a problemas de regresión y de clasificación lineales y no lineales.

En el caso de la regresión [4][14] el objetivo es transformar el espacio de predictores en un espacio de dimensión superior para poder realizar un ajuste lineal en dicho espacio, de tal manera que la función de regresión tiene la forma

$$f(x) = \sum_{i=1}^{d} = w_i \phi_i(x) + b, \tag{5}$$

donde d es la dimensión del espacio de predictores transformado, $\{\phi_i(x)\}_{i=1}^{d}$ es el conjunto de predictores transformado, b y $\{w_i\}_{i=1}^{d}$ son los coeficientes a estimar a partir de los datos. De esta manera, un problema de regresión no lineal en el espacio original de los predictores se convierte en un problema lineal en el espacio transformado. Estos coeficientes se determinan minimizando la función

$$R(w) = \frac{1}{N} \sum_{i=1}^{N} |f(x_i) - y_i|_\epsilon + C \|w\|^2, \tag{6}$$

donde C es una constante de regularización y L_ϵ es la función de coste denominada ϵ-insensitiva que se define como

$$L_\epsilon = \begin{cases} |f(x_i) - y_i| - \epsilon, & |f(x_i) - y_i| \geq \epsilon, \\ 0, & \text{en otro caso.} \end{cases} \tag{7}$$

De la minimización de $R(w)$ se obtiene que

$$f\left(x, \alpha, \alpha^*\right) = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right) K(x_i, x) + b, \tag{8}$$

donde α_i, α_i^* son multiplicadores de Lagrange no negativos, cumpliéndose que $\alpha_i \alpha^* = 0$, y $K(x_i, x)$ es una *función kernel* dada por el producto interno

$$K(x, y) = \sum_{j=1}^{d} \phi_j(x) \phi_j(y). \tag{9}$$

Dos de las funciones kernel más utilizadas son la polinomial

$$K(x, y) = \gamma \left(a + \langle x, y \rangle\right)^p,$$

y la radial gaussiana

$$K(x, y) = \exp\left(-\sigma \left\|x - y\right\|^2\right),$$

donde σ y γ son factores de escala.

En cuanto a los hiperparámetros relevantes en este algoritmo, se tienen C y ϵ para ajustar el balance entre varianza y sesgo, y además los que aparecen en las funciones kernel como γ, σ y p.

Aplicaciones de este método en el campo de la previsión energética pueden consultarse en trabajos como el de Li et al. [11]. Se utilizan técnicas de minería de datos para la eliminación de valores anómalos, con el objetivo de conseguir predicciones más ajustadas aplicando regresión SVM con horizontes de predicción a muy corto plazo, de entre diez minutos y seis horas.

2.2.4 Bosques aleatorios (RF)

Los bosques aleatorios o *Random Forest* están basados en la combinación de árboles de decisión para lidiar con el problema de sobreajuste que se da normalmente en estos [8].

El procedimiento consiste en dividir el conjunto de datos original en particiones, para cada una de las cuales se ajusta un árbol de decisión. Para determinar los nodos de decisión de cada árbol se selecciona aleatoriamente un número k de predictores del conjunto original, de entre los que se elige el que resulta en el menor valor de la impureza, que está dada por el error cuadrático medio. El valor de la predicción para la variable respuesta se obtiene promediando las respuestas de todos los árboles.

Algunos de los hiperparámetros que se pueden controlar en este algoritmo son:

- El número de árboles.
- El número de k predictores seleccionados en la creación de cada árbol.
- La complejidad de cada árbol.

En [9] se analizan los factores meteorológicos más influyentes en la producción de energía eólica con el fin de utilizarlos paras realizar predicciones usando bosques aleatorios. Se demuestra la poca afectación que tiene sobre este tipo de modelos el hecho de incluir variables no relevantes.

2.3 Transformación de variables

En muchas ocasiones es necesario la transformación de los predictores por razones diversas, como por ejemplo, algunos algoritmos requirieren que todos los predictores tengan la misma escala, o características concretas de los datos que empeoran los resultados del modelo, como la presencia de valores anómalos o distribuciones asimétricas.

Una transformación muy utilizada es la *estandarización*, consistente en restar la media y dividir por la desviación estándar a cada uno de los valores del predictor, obteniendo así una distribución con media cero y desviación estándar uno.

Para convertir distribuciones asimétricas en distribuciones más gaussianas, se pueden utilizar métodos estadísticos como las *transformaciones de Box-Cox*, que solo son aplicables en datos estrictamente positivos, o las de *Yeo-Johnson* válidas para datos positivos y negativos [20]. Estas últimas han sido las utilizadas en este trabajo por existir variables con valores nulos. Se definen como sigue:

$$x_i^{(\lambda)} = \begin{cases} [(x_i+1)^\lambda - 1]/\lambda & ,\lambda \neq 0, x_i \geq 0, \\ \ln(x_i+1) & ,\lambda = 0, x_i \geq 0 \\ -[(-x_i+1)^{2-\lambda} - 1]/(2-\lambda) & ,\lambda \neq 2, x_i < 0, \\ -\ln(-x_i+1) & ,\lambda = 2, x_i < 0, \end{cases} \tag{10}$$

donde λ se estima a partir de los datos x_i mediante el método de la máxima verosimilitud.

2.4 Cuantificación de la importancia de variables

Cuantificar la importancia que tienen los predictores sobre la variable respuesta es muy útil cuando se quiere reducir el número de atributos para obtener modelos lo más simples e interpretables posible, eliminando los predictores menos influyentes. Para ello se pueden aplicar varios métodos, como el *análisis de correlación*, que mide el grado dependencia lineal entre dos variables, o el *método de información mutua* [2].

La información mutua I es una medida de dependencia estadística entre dos variables aleatorias que cuantifica la cantidad de información obtenida acerca de una de ellas cuando se observa la otra. Formalmente, si X e Y son dos variables aleatorias discretas cuya distribución de probabilidad conjunta es $P(X,Y)$, la información mutua media viene dada por

$$I(X;Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\log\frac{P(x_i,y_j)}{P(x_i)P(y_j)}. \tag{11}$$

2.5 Métricas

Con el objetivo de medir la capacidad predictiva de los modelos, se hace necesario definir *métricas* que cuantifiquen la precisión de las predicciones. En este trabajo se han utilizado las siguientes:

- **Raíz del error cuadrático medio** (RMSE). Es una función de los residuos del modelo que mide cuánto difieren en media los valores reales de los estimados. Se calcula como

$$\text{RMSE} = \left(\mathbb{E}(\widehat{y}-y)\right)^{1/2}, \tag{12}$$

  y viene dado en las unidades de la variable observada y.

- **Porcentaje de error absoluto acumulado** (CAPE). Propuesta en el concurso descrito en la sección (1.1), es una forma relativa del error absoluto y se define como

$$\mathrm{CAPE}_k\left(\widehat{y}_k, y_k\right) = 100 \times \frac{\sum_{i=1}^{N_k} |y_{i,k} - \widehat{y}_{i,k}|}{\sum_{i=1}^{N_k} y_{i,k}}, \tag{13}$$

  donde CAPE_k es la métrica del parque k (en %), N_k el tamaño de la muestra de test para el parque k, $y_{i,k}$ e $\widehat{y}_k$ la producción energética observada y estimada respectivamente para el parque k en la hora i (en MW o MWh).

Otra métrica comúnmente utilizada en regresión es el **coeficiente de determinación** R^2, que en este trabajo se ha calculado como [15]

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{14}$$

donde $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ [15]. Representa la cantidad de información o varianza explicada por el modelo y hay que tener presente que no es una medida de precisión sino de correlación entre la variable respuesta y las predictoras.

3 Metodología

La metodología empleada se basa en el flujo típico que se utiliza en el aprendizaje automático para abordar problemas de modelización predictiva. Consta de varias etapas diferenciadas [16]:

1. **Análisis exploratorio**. El análisis exploratorio es muy importante para tener una visión general del tipo de datos, relaciones entre ellos así como para identificar posibles problemas a resolver en la etapa siguiente de preprocesado.

2. **Preprocesado de los datos**. Es muy improbable que los datos en bruto vengan en un formato apropiado para la aplicación de algoritmos de aprendizaje automático. Pueden existir valores perdidos, datos anómalos, escalas dispares entre los atributos o atributos altamente correlacionados, entre otros problemas. El preprocesado de los datos para manejar y corregir este tipo de problemas se hace imprescindible si se quiere obtener un rendimiento óptimo de los algoritmos.

3. **Aprendizaje o entrenamiento**. En esta etapa es donde se entrena al algoritmo utilizando los datos de entrenamiento preparados en la fase anterior. Normalmente se entrenan y comparan distintos algoritmos para seleccionar después el que mejor resultados proporciona en base a la métrica elegida.

4. **Evaluación y predicción**. Una vez seleccionado el modelo, este se utiliza sobre el conjunto de test con el objetivo de comprobar su precisión sobre datos nuevos y estimar así el error de generalización. Si el rendimiento es bueno se podrá utilizar el modelo para realizar predicciones en base a nuevos datos.

En la figura 3 se muestra un diagrama que resume el flujo completo.

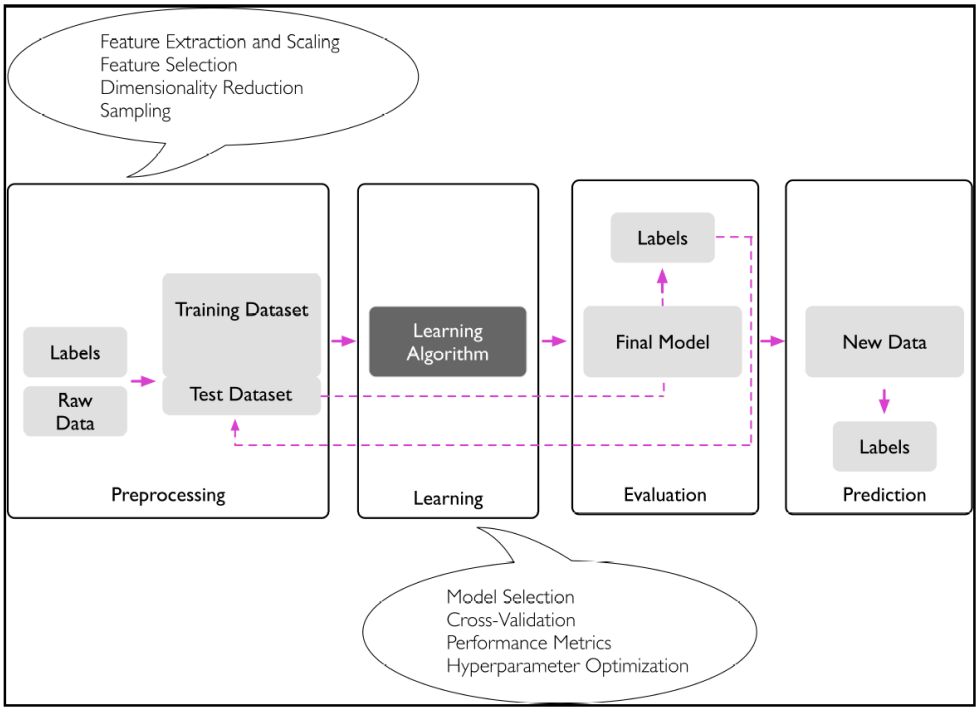

Figura 3: Flujo típico empleado en la aplicación del aprendizaje automático para problemas de modelización predictiva [Imagen obtenida de [16]].

3.1 Origen y descripción de los datos

Los datos proporcionados por la empresa CNR consisten en los siguientes ficheros:

- Datos de entrenamiento sobre los que se entrenará el modelo.

  `X_train.csv` correspondiente a las variables de entrada o predictores.

  `y_train.csv` correspondiente a la variable de salida.

- Datos de test sobre los que se validará el modelo.

  `X_tet.csv` para las variables de entrada.

Existe también un fichero `y_test.csv` para la variable de salida a la que los participantes no tienen acceso, ya que son los valores que deben predecir en base a los cuales se medirá la precisión de su modelo y, por ende, su posición en el ranking de la competición. El resultado de la predicción solo puede comprobarse dos veces al día a través de la web del concurso. Esta restricción supone una ralentización en la realización de este trabajo, ya que reduce el número de pruebas que se pueden realizar en un tiempo determinado. Para salvar esta limitación se han utilizado los datos de entrenamiento para crear a su vez dos subconjuntos de entrenamiento y validación. Para la construcción de los modelos se utilizarán estos dos subconjuntos.

Los datos de entrenamiento proporcionados contienen la producción eólica horaria desde el 01-05-2018 hasta el 15-01-2019 (ocho meses y medio) para seis parques eólicos distintos. En la partición realizada se han utilizado siete meses y medio para construir el conjunto de entrenamiento, dejando el resto para test. Se realizarán, por tanto, predicciones con frecuencia horaria para el mes siguiente.

Se facilitan las variables predictoras consistentes en la predicción horaria de distintas variables meteorológicas, proporcionadas por cuatro modelos numéricos de predicción meteorológica y ejecutados

por distintos servicios nacionales de predicción. Los nombres de estos modelos no se dan por motivos de confidencialidad, nombrándose de manera genérica como NWP1, ..., NWP4.

A continuación, se detallan las variables proporcionadas:

- Variables contenidas en los ficheros `X_train.csv` y `X_test.csv`:
  - **Time** (UTC): Fecha y hora a la que se observa el valor de la producción.
  - **WF**: Identificador para cada parque eólico, siendo los posibles valores WF1, WF2, $\cdots$,WF6.
  - **ID**: Identificador único para cada fila. Cada `ID` corresponde a una pareja Time-WF. Los ID del conjunto de test son consecutivos a los de entrenamiento.
  - **U** y **V** (m/s): Componente zonal y meridional de la velocidad del viento en m/s. Están dadas a 100 m de altura sobre el suelo para los modelos NWP1, NWP2 y NWP3, y a 10 m para NWP4. Aunque se proporcionan cada hora, es importante señalar que son los valores medios de los últimos 10 minutos de cada hora.
  - **T** (K): Temperatura media del aire para cada hora, se proporciona únicamente por los modelos meteorológicos NWP1 y 3.
  - **CLCT** (%): Porcentaje de cielo cubierto por las nubes, variando entre el 0 % (cielo totalmente despejado) y el 100 % (cielo totalmente cubierto). Es un valor instantáneo tomado al inicio de cada hora. Solo se proporciona en el modelo NWP4.

  Las predicciones meteorológicas son facilitadas varias veces al día (ejecuciones), típicamente a las `00h`, `06h`, `12h` y `18h`. Esto implica que se pueden tener varias predicciones para cada variable meteorológica para una misma hora. De esta manera, el formato del nombre para las variables meteorológicas viene dado por

  `NWPi_HoraEjecucion_DiaEjecucion_<U,V,T,CLCT>`,

  donde `NWPi` es cada uno de los cuatro modelos numéricos de predicción meteorológica, `HoraEjecucion` es la hora de ejecución, `DiaEjecucion` es el día de la ejecución, siendo `D-2` antes de ayer, `D-1` ayer y `D` el día en curso (ver tabla 1).

	ID	WF	Time	NWP1_00h_D-2_U	NWP1_00h_D-2_V
0	37376	WF1	16/01/2019 01:00	-4.5500	-1.5148
1	37377	WF1	16/01/2019 02:00	-5.7942	0.4186
2	37378	WF1	16/01/2019 03:00	-5.9803	1.0079
3	37379	WF1	16/01/2019 04:00	-6.1664	0.4983
4	37380	WF1	16/01/2019 05:00	-6.9187	0.2228

Tabla 1: Extracto del conjunto de datos de entrenamiento en bruto.

- Variables contenidas en los ficheros `y_train.csv` y `y_test.csv` (además del ID comentado más arriba que identifica cada observación):
  - **Production** (MWh): Potencia media horaria suministrada a la red eléctrica por cada parque.

El tamaño de cada conjunto de datos se resume en la tabla 2.

Datos en bruto	Filas	Columnas
`X_train.csv`	37375	105
`X_test.csv`	36529	105
`y_train.csv`	37375	2

Tabla 2: Tamaño de los cojuntos de datos de partida.

3.2 Análisis exploratorio

Con el objetivo de ilustrar el análisis exploratorio, se muestran los resultados solo para alguno de los parques ya que en el resto se ha procedido de forma similar. Para facilitar este análisis ha sido necesario transformar la estructura del conjunto de entrenamiento para obtener el formato que se puede ver en la tabla3.

WF	time	NWP	fc_day	run	U	V	T	CLCT
WF5	20/09/2018 09:00	1	D-1	00h	4.232100	5.915200	291.970000	NaN
WF6	14/01/2019 15:00	3	D-1	06h	-0.315288	-9.496811	279.399994	NaN
WF1	05/05/2018 17:00	3	D	00h	NaN	NaN	NaN	NaN
WF2	02/12/2018 23:00	1	D	18h	10.554000	5.631900	286.960000	NaN

Tabla 3: Extracto aleatorio del conjunto de datos generado para el análisis exploratorio.

Un hecho importante que se observa es la existencia de valores perdidos en todas las variables meteorológicas siguiendo aproximadamente un mismo patrón en su distribución en todos los parques (Fig. 4). La presencia de estos valores está condicionada por la disponibilidad y frecuencia horaria con la que se obtienen los datos de los diferentes servicios meteorológicos de predicción.

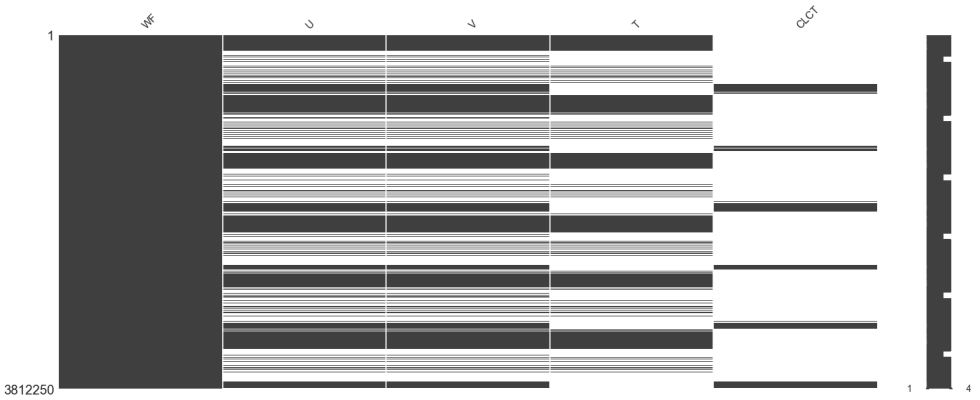

Figura 4: Distribución de los valores perdidos existentes en las variables meteorológicas.

En la figura 5 se recogen los diagramas de cajas de las variables meteorológicas en cada uno de los parques en función de los modelos numéricos de predicción. Como se puede observar, los valores medios que proporcionan son similares para todas las variables a excepción de la componente de la velocidad V, donde se observa que en el parque WF6 las medias difieren en mayor medida.

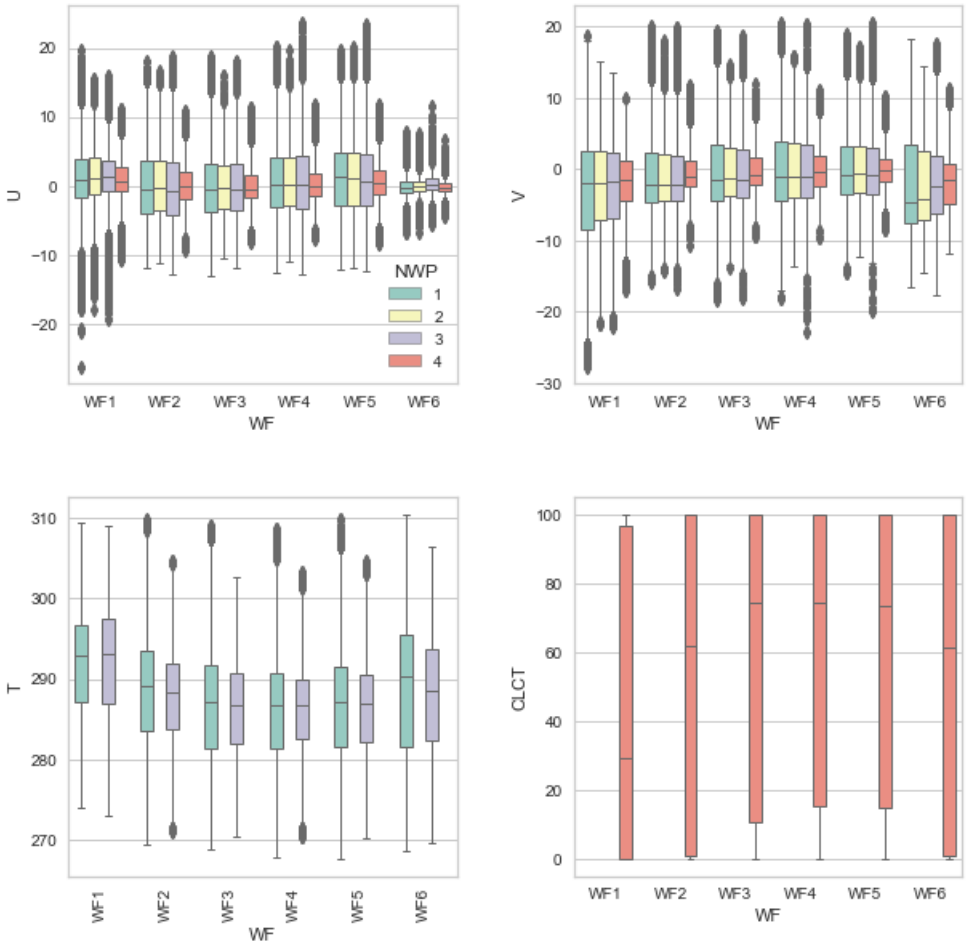

Figura 5: Diagramas de caja para cada una de las variables meteorológicas en cada uno de los seis parques eólicos.

A falta de más información sobre cuál de los cuatro modelos de predicción meteorológica es mejor, se decide tomar la media de los valores de los métodos NWP1, 2 y 3 para las componentes de las velocidades, teniendo en cuenta que estos tres proporcionan la velocidad a la misma altura (100 m), mientras que el NWP4 lo hace a 10 m. La razón reside en que la velocidad del viento varía con la altura, dando lugar a la *cizalladura* (ver sección 2.1), por lo que es conveniente usar velocidades medidas o estimadas a la misma altura.

Para la temperatura se ha utilizado la media de las predicciones dadas por los modelos NWP1 y 3, que son los únicos que la proporcionan, y para la cobertura del cielo (CLCT), la única de la que se dispone dada por el modelo NWP4.

En la figura 6 se muestra la distribución de la producción eólica en cada parque. En todos los casos son distribuciones muy asimétricas por la derecha, con valores máximos en torno a 10 KW, si bien en WF4 y WF6 es menor, en torno a 8 KW y 4 KW respectivamente.

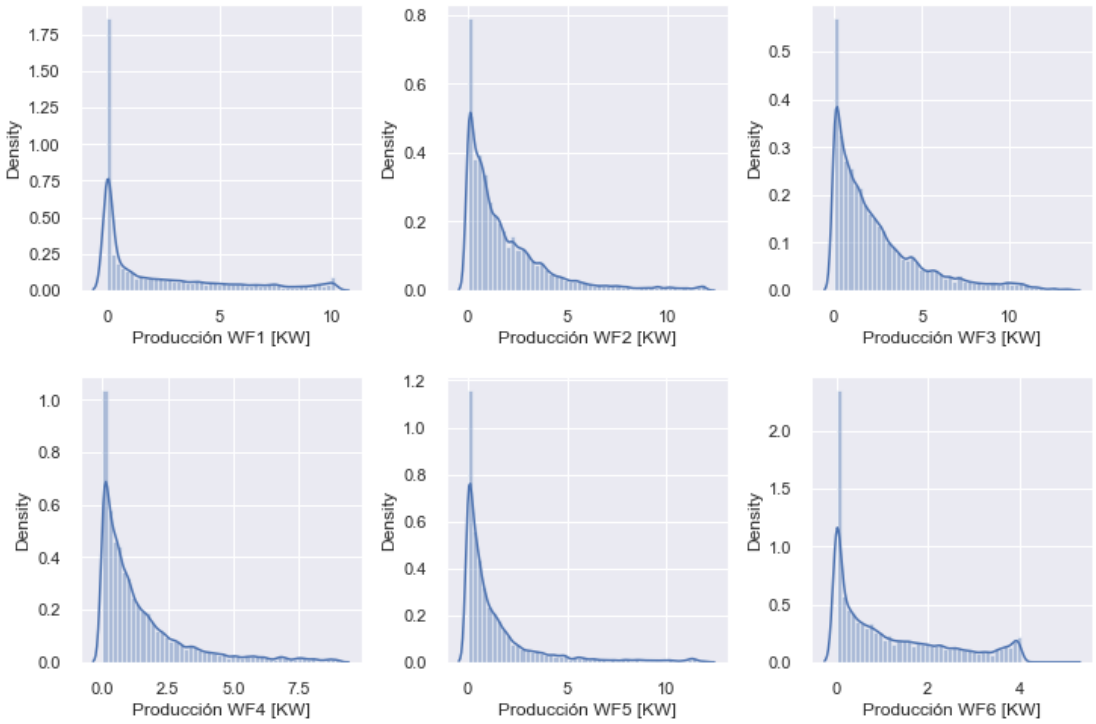

Figura 6: Distribución de la producción eólica en cada parque.

La figura 7 recoge la matriz de correlaciones entre las distintas variables para el parque WF1 como ejemplo del comportamiento que se da también en el resto de parques. Las mayores correlaciones se dan entre U y V, como era de esperar al ser componentes de un mismo vector. Además, en cuanto a la producción, hay correlación negativa con la componente V y con la temperatura. Esto quiere decir, por ejemplo, en el caso de la temperatura, que un aumento de ésta implica una disminución en la producción de energía. Esto es consistente con la física del problema dada por la ecuación (2), donde se ve que la potencia es inversamente proporcional a la temperatura.

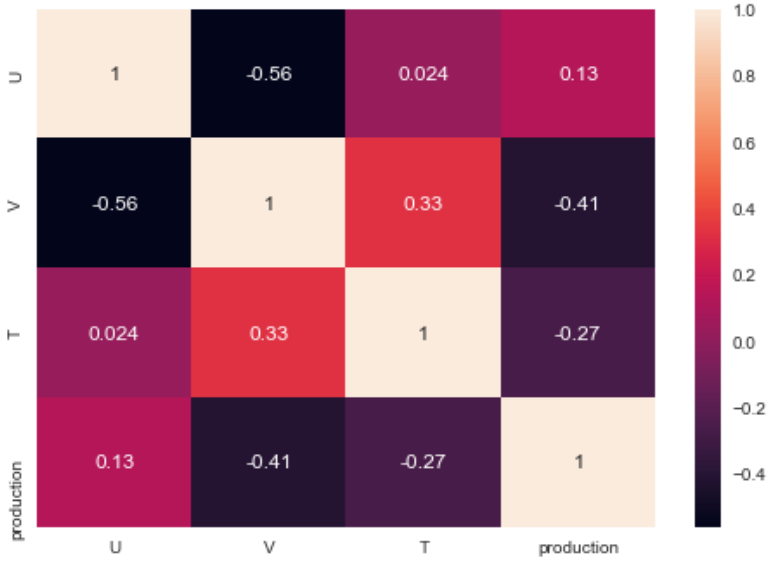

Figura 7: Matriz de correlaciones para el parque eólico WF1.

3.3 Preparación de los datos

En esta sección se describen los pasos que han sido necesarios para transformar los datos originales y obtener un conjunto de entrenamiento apto para el aprendizaje de los algoritmos.

3.3.1 Limpieza

La limpieza de los datos originales ha consistido principalmente en la eliminación de valores perdidos y la identificación y eliminación de valores anómalos.

Valores perdidos. Los dos escenarios de valores perdidos encontrados son, por un lado, la frecuencia con la que los servicios de predicción meteorológica facilitan las predicciones, y por otro, valores perdidos de manera aleatoria existiendo casos en los que faltan días completos. Respectivamente, para rellanar estos valores se ha procedido como sigue:

1. Interpolación lineal basada en el índice temporal dado por la fecha y hora de cada observación.
2. Teniendo en cuenta que las predicciones meteorológicas son más precisas cuanto más corto es el plazo de predicción, se ha consolidado como dato horario el más cercano al día para el que se quiere predecir, ya que se dispone de las estimaciones numéricas emitidas en los tres días anteriores.

Valores anómalos. La detección de valores anómalos o *outliers* se ha basado en la elaboración de la curva de potencia (sección 2.1, figura 2). En este punto se necesita calcular el módulo de la velocidad del viento a partir de las componentes U y V. Se han definido cinco tipos de puntos en función de su posición respecto a la curva de potencia:

- Tipo 1 (normal). Puntos que se ajustan a la curva de potencia.
- Tipo 2 (outlier). Puntos que tienen una baja velocidad del viento pero una alta producción.
- Tipo 3 (outlier). Puntos con velocidad del viento negativa.
- Tipo 4 (outlier). Puntos con producción negativa.
- Tipo 5 (outlier). Puntos que tienen baja producción con un valor alto de velocidad del viento.

En el caso de estudio solo se han encontrado datos de tipo 1, 2 y 5. Esta clasificación se ha basado en [18] donde se proponen, además de esta, otras tres aproximaciones para la clasificación de estos puntos, como puede verse en la figura 8.

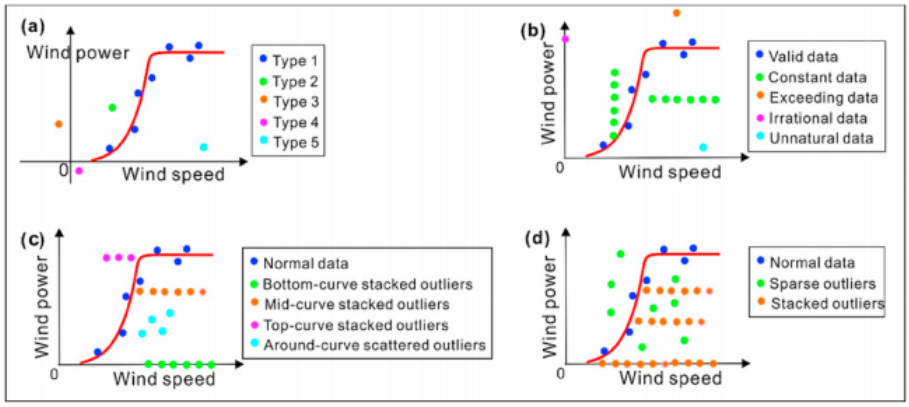

Figura 8: Distintas clasificaciones de los datos respecto a la curva de potencia. Imagen obtenida de [18].

El método utilizado para la detección de dichos valores consiste en dividir en secciones o *bins* los datos respecto al valor de la producción. Para cada bin se marcan como outliers aquellos puntos cuya velocidad esté fuera del entorno con centro la mediana y radio el 80 % de la desviación típica. Se ha elegido la mediana como centro por ser más robusta que la media.[6]

En las gráficas de la figura 9 se muestran los outliers detectados de esta manera en cada uno de los parques. Cabe señalar que la función utilizada requiere de la selección de varios parámetros como el tamaño del bin, o los valores máximos de la producción y la velocidad para los que se quiere hacer el barrido. No es un método automático, sino que se trata de ir ajustando estos parámetros basándose en la visualización de la curva de potencia.

Por último, conviene aclarar que en la partición reservada para el conjunto de test no se han limpiado los valores anómalos para reproducir lo más fielmente el escenario planteado en el concurso, donde al no disponer de los valores de test para la producción, no es posible aplicar este método de detección al conjunto de test para los predictores.

[6]Este método está disponible en la librería `OpenOA` (`https://openoa.readthedocs.io/en/latest/index.html`), donde se implementan varias funciones para el tratamiento de series de datos de viento.

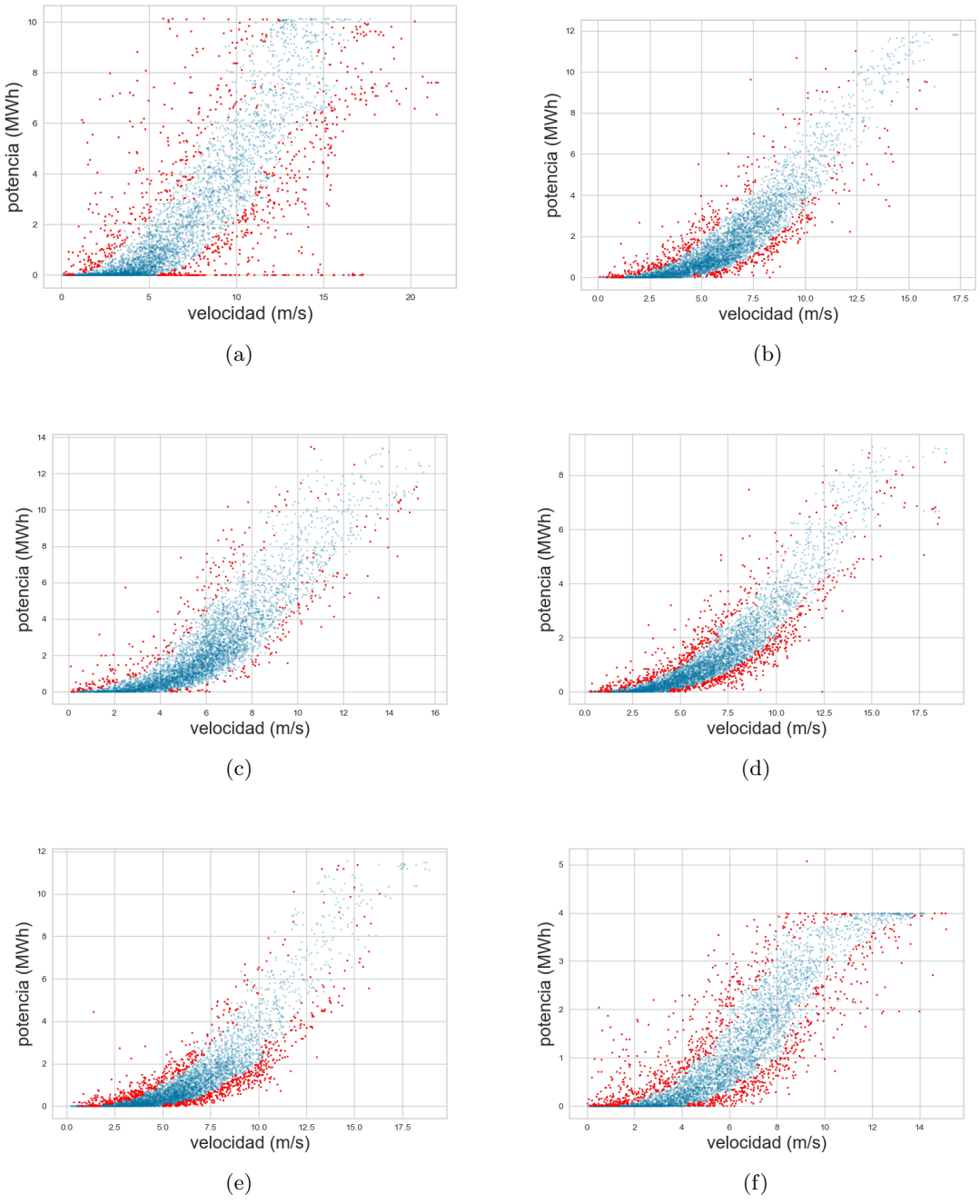

Figura 9: Curva de potencia y outliers detectados (en rojo) para (a) WF1, (b) WF2, (c) WF3, (d) WF4, (e) WF5 y (f) WF6.

3.3.2 Ingeniería de variables predictoras

La ingeniería de variables es el proceso de generación y selección de predictores mediante técnicas de análisis y minería de datos, con el objetivo de mejorar la precisión de los algoritmos de aprendizaje automático. En este trabajo esta tarea se ha basado en la física del problema, generando dos conjuntos de datos de entrenamiento distintos con el propósito de hacer un estudio posterior de las más influyentes en la variable de salida:

- Conjunto de datos 1 (básico), que consta de las variables fundamentales que influyen en la potencia generada por el viento atendiendo a la ecuación (2), además de predictores derivados de la fecha, como el mes y la hora para captar posibles estacionalidades en la producción. Como se puede ver en [17], la velocidad del viento es proporcional a la intensidad de la radiación solar, dando lugar a una estacionalidad horaria. De la misma manera, en [17] se indica que existe una estacionalidad mensual en la velocidad del viento ya que es inversamente proporcional a la temperatura media del mes.
- Conjunto de datos 2 (extendido). Incluye interacciones entre las variables meteorológicas relevantes, como la temperatura, la velocidad y la dirección del viento, basándonos en el hecho de que aparecen relacionados mediante su producto en la ecuación (2). Además, se dota de carácter vectorial a las variables cíclicas como la dirección del viento, el mes y la hora, y se añaden sus componentes como nuevas variables tomando como base las recomendaciones dadas por Bishop en [1, pág. 105-110].

3.3.3 Selección de los predictores más influyentes.

Utilizando el método de Información Mutua descrito en la sección 2.4, se obtiene la importancia entre cada predictor y la variable objetivo para los dos conjuntos de datos. Como ejemplo, en las figuras 10 y 11 se muestra la importancia de cada variable para el parque WF1.

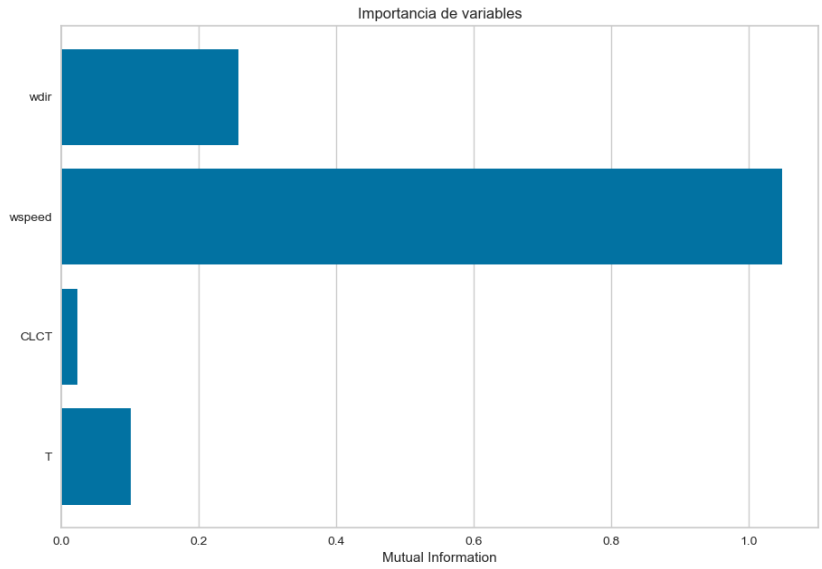

Figura 10: Importancia de las variables predictoras para el conjunto de entrenamiento básico del parque WF1.

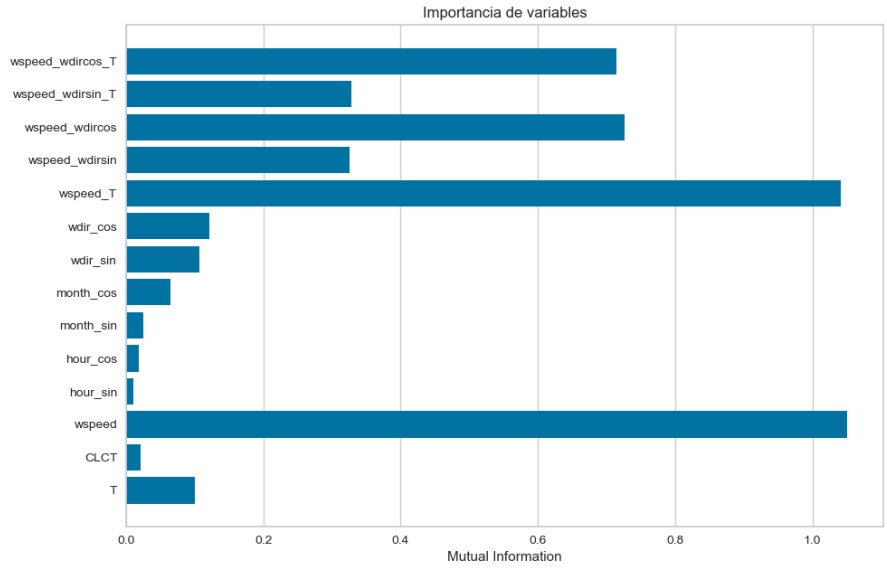

Figura 11: Importancia de las variables predictoras para el conjunto de entrenamiento extendido del parque WF1.

Atendiendo a estas gráficas, para entrenar a los algoritmos se han escogido las dos variables más importantes para el conjunto de datos básico y las seis más importantes para el extendido, ya que el resto, con un valor de la importancia mutua por debajo del 0.2, no parecen tener una influencia relevante sobre la producción. Así, en el ejemplo del parque WF1, en el conjunto básico quedarían incluidas la velocidad y dirección del viento, mientras que en el conjunto extendido se seleccionarían la velocidad y su interacción con la temperatura y con las componentes cíclicas de la dirección de viento, además de la interacción de la temperatura con dichas componentes del viento.

3.4 Entrenamiento y optimización de hiperparámetros.

Tal y como se comenta en la sección 3.1, el conjunto de datos original se ha dividido en dos subconjuntos, uno para entrenamiento abarcando siete meses, y otro para test con el mes siguiente. Se han entrenado los siguientes algoritmos para todos los parques y para los dos conjuntos de datos descritos:

- k vecinos más próximos (KNN).
- Splines de regresión adaptativa multivariante (MARS).
- Bosques aleatorios (RF).
- Máquinas de vector soporte (SVM).

Para evitar el sobreajuste a los datos de entrenamiento, se ha utilizado el método de validación cruzada o *Cross Validation* (CV) combinado con la búsqueda en rejilla o *Grid Search* para encontrar los valores óptimos de los hiperparámetros de cada algoritmo.

La validación cruzada consiste en dividir el conjunto de entrenamiento en un número k de particiones, entrenando cada vez con todas las particiones menos una que se deja para validación. De esta manera, se realizan k entrenamientos y validaciones para el ajuste del modelo, seleccionándose el que

tenga el menor error sobre el conjunto de validación. Es importante señalar que, en este caso, al tratarse de observaciones temporales y, por tanto, no independientes, las particiones se realizan respetando el orden temporal de las observaciones con el objetivo de no utilizar valores futuros en la predicción de valores pasados. Tras este proceso, se han utilizado los mejores valores de los hiperparámetros obtenidos para reentrenar el algoritmo con todos los datos. La métrica a minimizar es el CAPE, aunque también se calculan el RMSE y el R^2 en cada validación cruzada.

Como ejemplo, en la figura 12 se muestra el resultado del proceso de validación cruzada en el parque WF6 después de entrenar el algoritmo SVM. Se han realizado seis particiones para todos los parques con el objetivo de que cada partición de validación recogiera aproximadamente un mes de datos, teniendo así un tamaño lo más parecido posible al del conjunto de test.

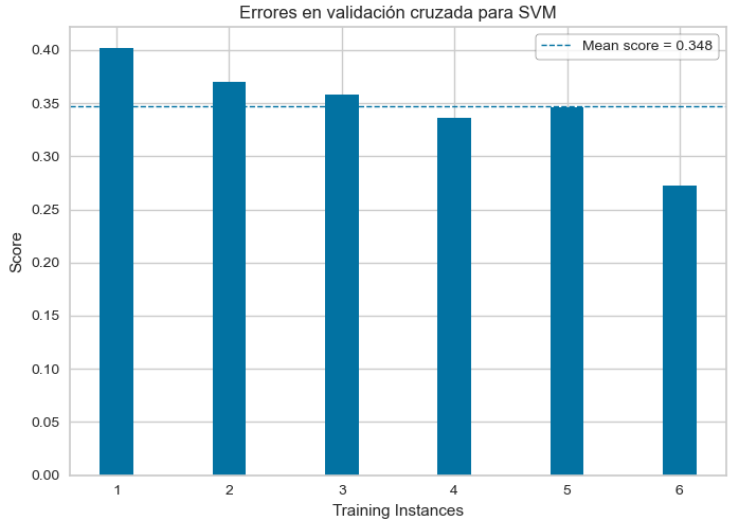

Figura 12: RMSE obtenido en el entrenamiento de un SVM para el parque WF6 en el proceso de validación cruzada.

En el proceso de Grid Search se establece un conjunto discreto de valores para cada hiperparámetro, probando todas las combinaciones posibles y seleccionando el valor que minimice el error del modelo en base a la métrica elegida. La elección de los valores del intervalo se ha determinado en todos los casos utilizando la curva de evolución del error en función del hiperparámetro en cuestión como referencia. Por ejemplo, para el caso de un KNN en el parque WF6, el error del modelo varía en función del número de vecinos k como se muestra en Fig. 13. Se observa un mínimo para $k \approx 15$, por lo que se toma un conjunto de valores en torno a ese valor.

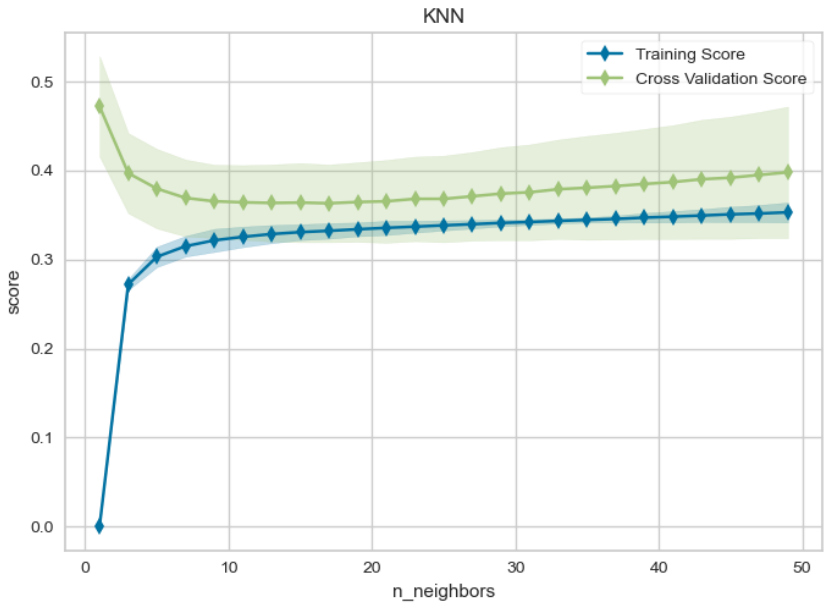

Figura 13: Variación del RMSE en función de k del modelo KNN en WF6.

3.5 Selección del modelo y predicción

Con el objetivo de realizar predicciones sobre los datos de test, se ha elegido el modelo que menor CAPE proporciona en la etapa de validación cruzada. Además, se utilizan las curvas de aprendizaje (error vs. tamaño de la muestra) para comprobar la estabilidad del modelo, verificando que no exista sobreajuste o sesgo. Como ejemplo, en la figura 14 se puede ver la curva de aprendizaje tanto para el error de validación como el de entrenamiento en el caso de un bosque aleatorio para el parque WF6.

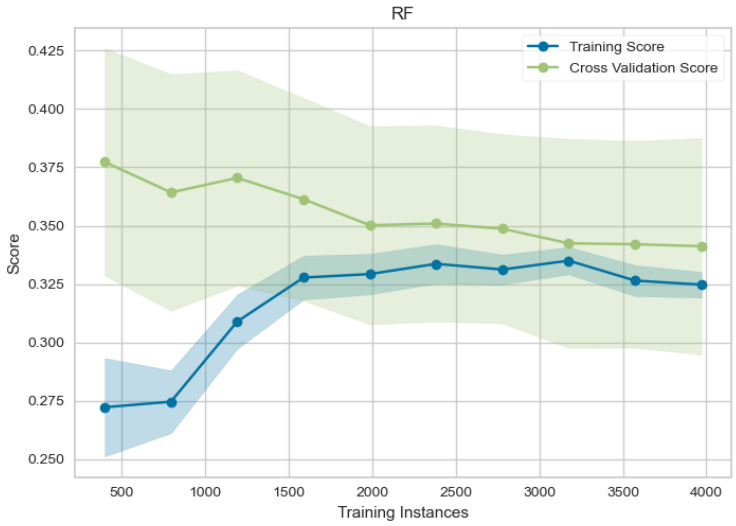

Figura 14: Variación del RMSE de validación y entrenamiento en función del número de observaciones del modelo RF para WF6.

Se puede ver cómo los errores convergen a valores parecidos a medida que aumenta el número de observaciones. Esto indica que el modelo es estable y que no existe sobreajuste ni sesgo. Una vez elegido el mejor modelo para cada parque, se procede a realizar la predicción sobre el conjunto de test (Fig. 15).

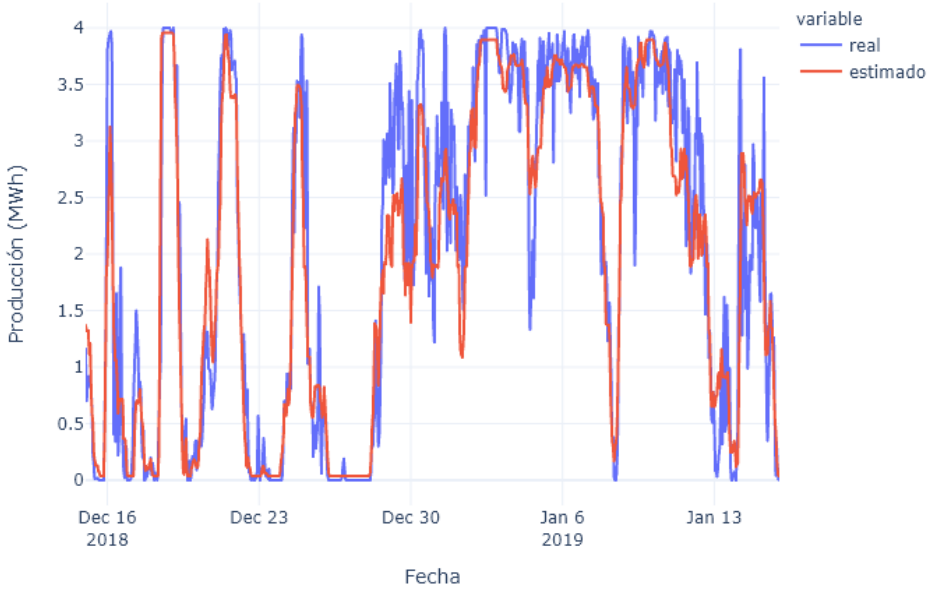

Figura 15: Valores de la producción eólica estimada y real para el modelo RF en WF6.

4 Herramientas utilizadas

4.1 Lenguaje de programación y librerías

Uno de los objetivos principales de este trabajo es el desarrollo de un software lo suficientemente flexible y configurable para automatizar lo máximo posible la realización de pruebas con distintos modelos y parámetros. Para conseguirlo se ha elegido **Python** como lenguaje de programación ya que, siendo un software de carácter libre, posee una amplia variedad de librerías para aprendizaje automático y mucha versatilidad a la hora de desarrollar aplicaciones.

Las principales librerías utilizadas han sido:

- **Numpy**: librería de cálculo numérico computacional que permite la vectorización de operaciones utilizando arrays multidimensionales.
- **Pandas**: librería para el análisis y manipulación de datos en forma de tablas o *data frames*.
- **Scikit-Learn**: contiene infinidad de librerías que implementan métodos de Machine Learning en Python, incluyendo algoritmos de clasificación, regresión y técnicas de agrupamiento.
- **Matplotlib** y **Plotly**: librerías para la elaboración de gráficas.

Además se han utilizado los siguientes frameworks:

- **Kedro**[7]: permite aplicar buenas prácticas de ingeniería de software para crear flujos o *pipelines* de Machine Learning.
- **MLflow Tracking**[8]: es un componente de la plataforma MLflow utilizado para organizar y registrar los resultados de los experimentos.

Como herramienta para el control de versiones del software se ha empleado **Git** y **Github** como repositorio en la nube.

4.2 Descripción del software desarrollado

El software desarrollado se organiza en módulos denominados *pipelines* que se forman a partir de funciones o *nodos* que realizan secuencialmente los pasos de la metodología descrita. Los pipelines principales implementados son:

1. Transformación de los datos en bruto en un formato adecuado para el análisis exploratorio.
2. Preparación de los datos a consumir por los algoritmos.
3. Ingeniería de predictores.
4. Modelización y predicción.

En cada uno de los pipelines se generan gráficas relevantes para el análisis de los modelos, ficheros de datos transformados, así como ficheros binarios con la información de los modelos, permitiendo su posterior uso (por ejemplo, para predecir sobre nuevos datos, reentrenarlos, etc.).

A nivel de usuario el software dispone de varios comandos de consola (CLI) que permiten incluir diferentes opciones para la ejecución de los pipelines. Además, por cada pipeline existe un fichero de parámetros de configuración independiente para que el usuario pueda configurar los experimentos que desee sin tener que modificar el código de la aplicación, pudiéndose además especificar cada parámetro a través de los comandos de consola, reescribiendo el valor configurado en los ficheros si así se desea. Por ejemplo, si se quiere construir el modelo SVM para el parque WF1 utilizando los dos predictores más importantes, partiendo de los datos en bruto la secuencia de comandos sería la siguiente:

1. Pipeline *data engineering* (de): `kedro run --pipeline de --params wf:WF1`
2. Pipeline *feature engineering* (fe): `kedro run --pipeline fe --params wf:WF1,k_max=2`
3. Pipeline *modeling* (mdl): `kedro run --pipeline mdl --params wf:WF1,alg:SVM`

Para más detalles se puede acceder al código y documentación de la herramienta a través del siguiente enlace de Github:

`https://github.com/vchaparro/MasterThesis-wind-power-forecasting`

[7] `https://kedro.readthedocs.io/en/stable/`
[8] `https://mlflow.org/`

En Fig. 16, 17 y 18 se ilustra el flujo para cada pipeline, mostrando los nodos de los que se compone, así como los parámetros y datos de entrada en cada uno de ellos. Los nodos se representan con la letra f y los parámetros con una doble flecha. El símbolo restante identifica a los conjuntos de datos.

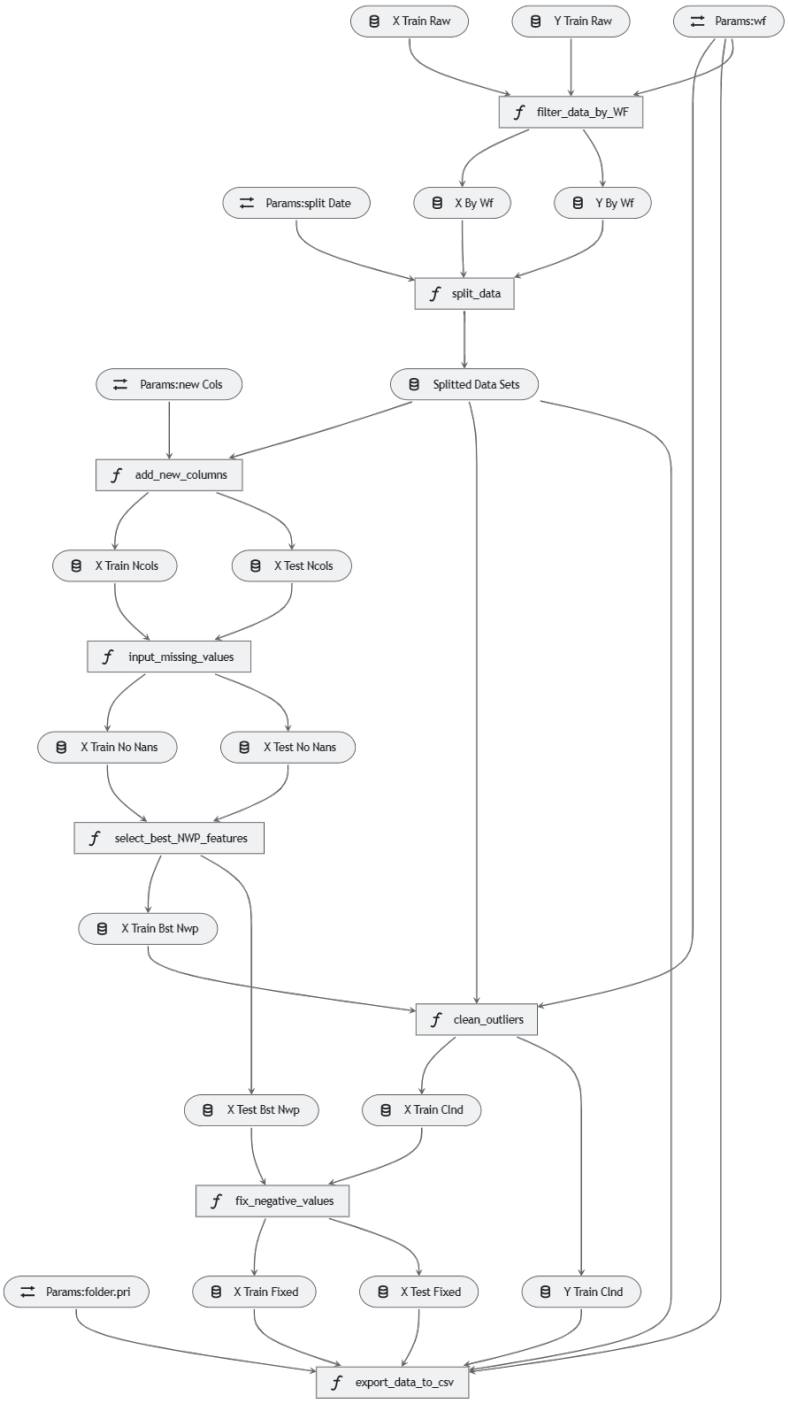

Figura 16: Flujo implementado en el pipeline de preparación de datos.

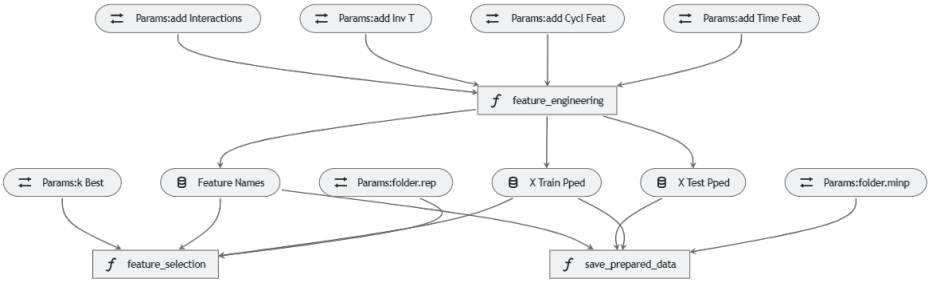

Figura 17: Flujo implementado en el pipeline de ingeniería de predictores.

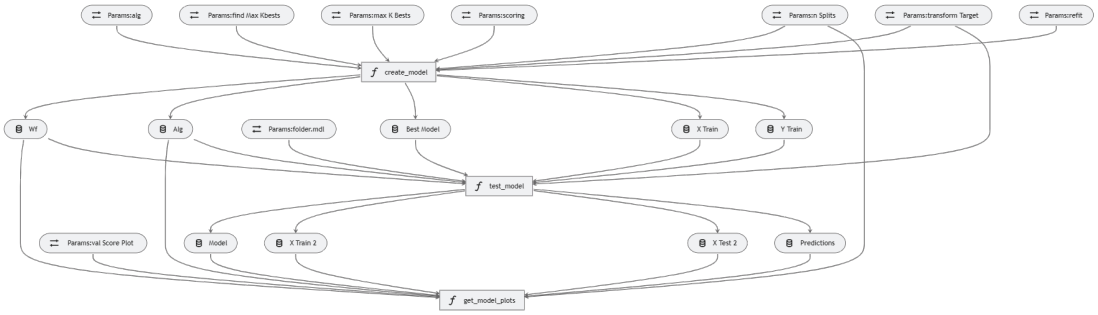

Figura 18: Flujo implementado en el pipeline de modelización y predicción.

5 Resultados y discusión

En esta sección se exponen los resultados obtenidos tras la aplicación de la metodología a los seis parques eólicos en cada uno de los dos experimentos realizados, esto es, entrenamiento con el conjunto de datos básico y extendido. En ambos casos, para cada parque y para el conjunto de variables de entrada más importantes encontrado, se ha entrenado cada algoritmo incluyendo progresivamente las variables en orden de importancia. El objetivo es comprobar la precisión de los modelos en función del número de predictores.

Para la evaluación de la precisión de los modelos se incluyen tres métricas: R^2, RMSE y CAPE, siendo esta última la utilizada para la elección del mejor modelo.

5.1 Versión básica del conjunto de entrenamiento

En las figuras 19 a 24 aparecen los diagramas de barras con el valor de la importancia mutua para cada variable.

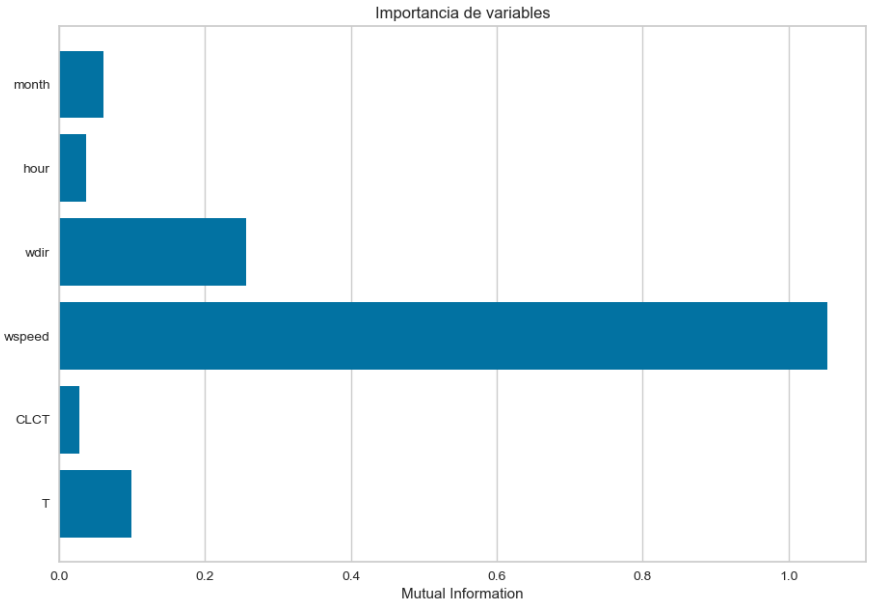

Figura 19: Importancia de las variables del conjunto de entrenamiento básico para WF1.

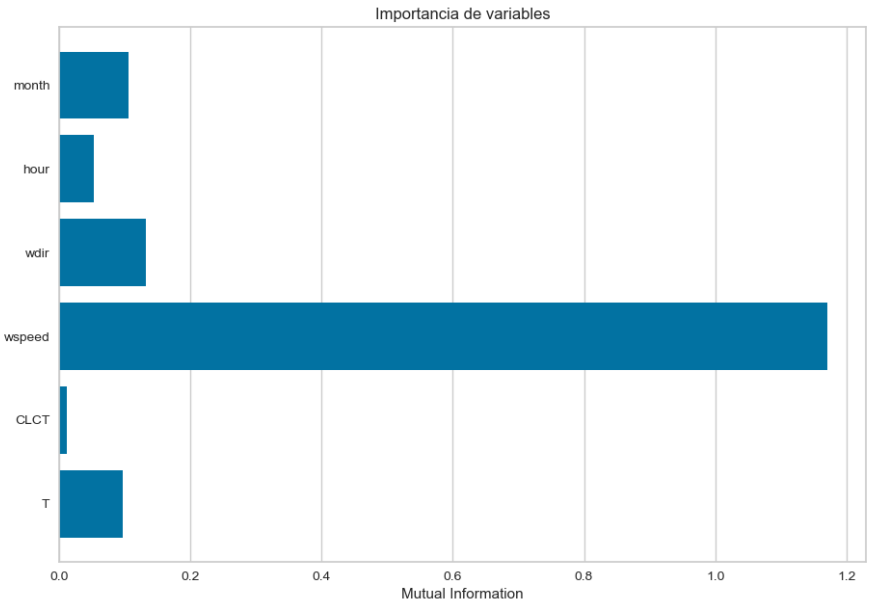

Figura 20: Importancia de las variables del conjunto de entrenamiento básico para WF2.

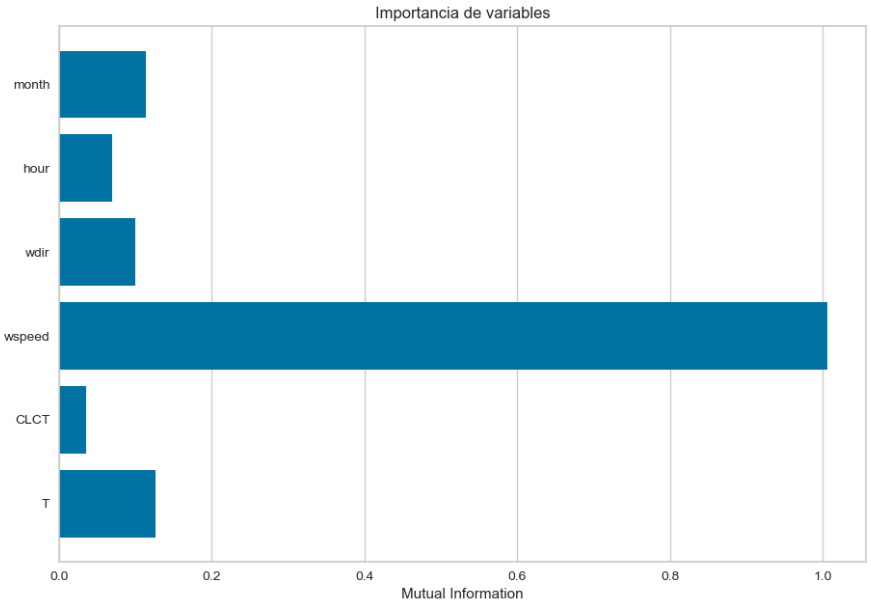

Figura 21: Importancia de las variables del conjunto de entrenamiento básico para WF3.

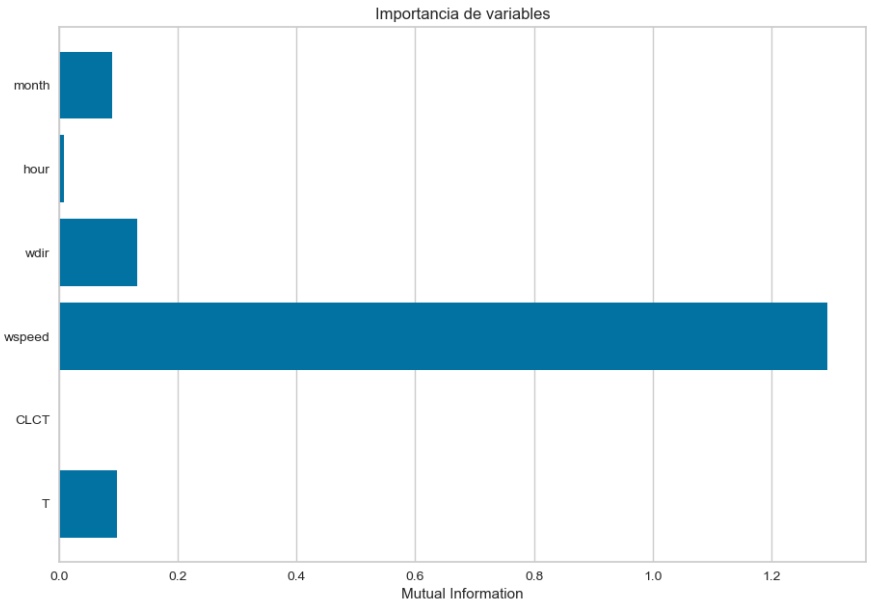

Figura 22: Importancia de las variables del conjunto de entrenamiento básico para WF4.

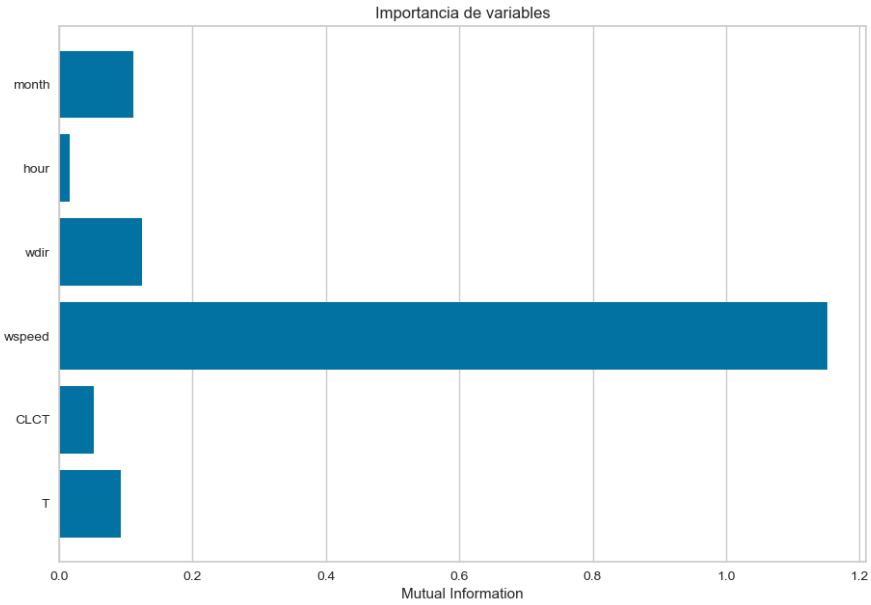

Figura 23: Importancia de las variables del conjunto de entrenamiento básico para WF5.

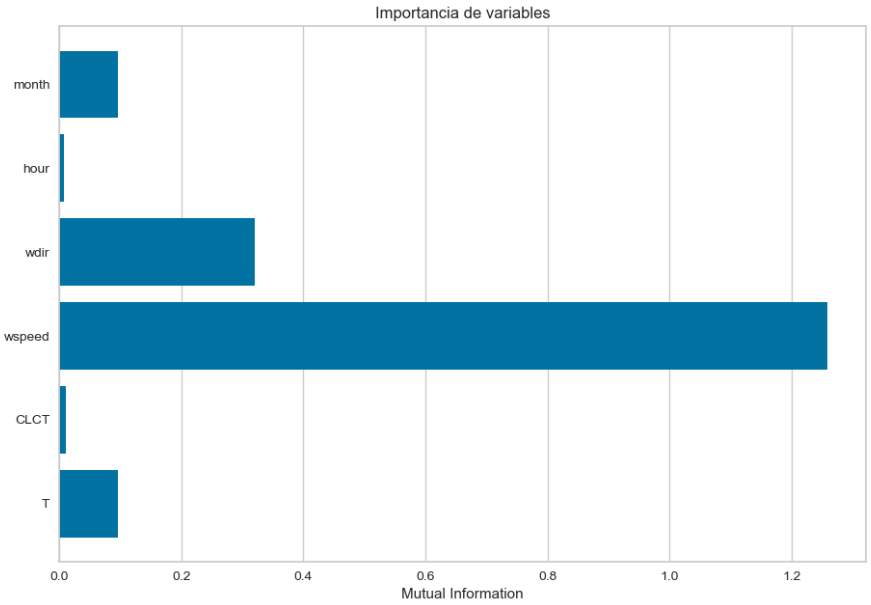

Figura 24: Importancia de las variables del conjunto de entrenamiento básico para WF6.

La variable más influyente en la producción ha resultado ser, en todos los casos, la velocidad del viento, con una importancia mutua mucho mayor que el resto de variables. La segunda variable más importante es la dirección, en todos los parques a excepción del WF3 en el que la temperatura supera a la dirección. Variables como el mes, la hora o el CLCT apenas influyen en la producción.

Las tablas 4 y 5 recogen los resultados obtenidos en la aplicación de los modelos al conjunto de test,

para los casos de una y dos variables predictoras.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF1	KNN	1	**29.995**	0.741	1.930
		2	32.840	0.697	2.089
	MARS	1	**30.196**	0.744	1.917
		2	33.381	0.708	2.050
	RF	1	**29.915**	0.744	1.920
		2	31.753	0.719	2.009
	SVM	1	34.497	0.619	2.341
		2	**31.734**	0.709	2.045
WF2	KNN	1	26.707	0.884	0.880
		2	**26.011**	0.882	0.886
	MARS	1	26.564	0.885	0.876
		2	**26.000**	0.888	0.862
	RF	1	26.468	0.884	0.879
		2	**25.578**	0.889	0.858
	SVM	1	26.355	0.884	0.880
		2	**25.473**	0.890	0.855
WF3	KNN	1	**30.427**	0.719	1.353
		2	31.466	0.674	1.457
	MARS	1	**30.499**	0.714	1.364
		2	31.143	0.696	1.406
	RF	1	**30.677**	0.709	1.375
		2	32.450	0.627	1.557
	SVM	1	**29.491**	0.731	1.322
		2	32.496	0.584	1.646

Tabla 4: Resultados del error de predicción y precisión obtenidos para los parques WF1, WF2 y WF3 con los distintos algoritmos en función del número de predictores.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF4	KNN	1	**29.223**	0.834	0.977
		2	29.245	0.830	0.989
	MARS	1	29.119	0.833	0.981
		2	**29.026**	0.831	0.985
	RF	1	29.182	0.830	0.988
		2	**28.998**	0.828	0.995
	SVM	1	**28.428**	0.828	0.994
		2	29.387	0.815	1.032
WF5	KNN	1	28.966	0.865	1.198
		2	**26.467**	0.889	1.085
	MARS	1	**27.757**	0.873	1.161
		2	27.898	0.879	1.134
	RF	1	29.057	0.861	1.212
		2	**27.555**	0.878	1.139
	SVM	1	**27.745**	0.868	1.182
		2	27.848	0.874	1.155
WF6	KNN	1	**17.044**	0.885	0.507
		2	17.523	0.875	0.531
	MARS	1	**16.850**	0.887	0.503
		2	17.216	0.881	0.518
	RF	1	**16.941**	0.885	0.508
		2	17.114	0.883	0.513
	SVM	1	17.060	0.883	0.512
		2	**16.912**	0.883	0.513

Tabla 5: Resultados del error de predicción y precisión obtenidos para los parques WF4, WF5 y WF6 con los distintos algoritmos en función del número de predictores.

El mejor modelo con esta versión del conjunto de entrenamiento se ha conseguido con el método MARS para el parque WF6, obteniéndose un CAPE del 16.85 %, RMSE de 0.503 MW y un R^2 del 88.7 %, con la velocidad del viento como única variable de entrada. El peor modelo se ha dado en el parque WF1 con el SVM, con un CAPE ligeramente superior al 31 %, RMSE en torno a 2 MW y un R^2 del 70 %, con la velocidad y dirección como predictores.

5.2 Versión extendida del conjunto de entrenamiento

En las figuras 25 a 30 aparece el diagrama de barras con el valor de la importancia mutua para cada variable.

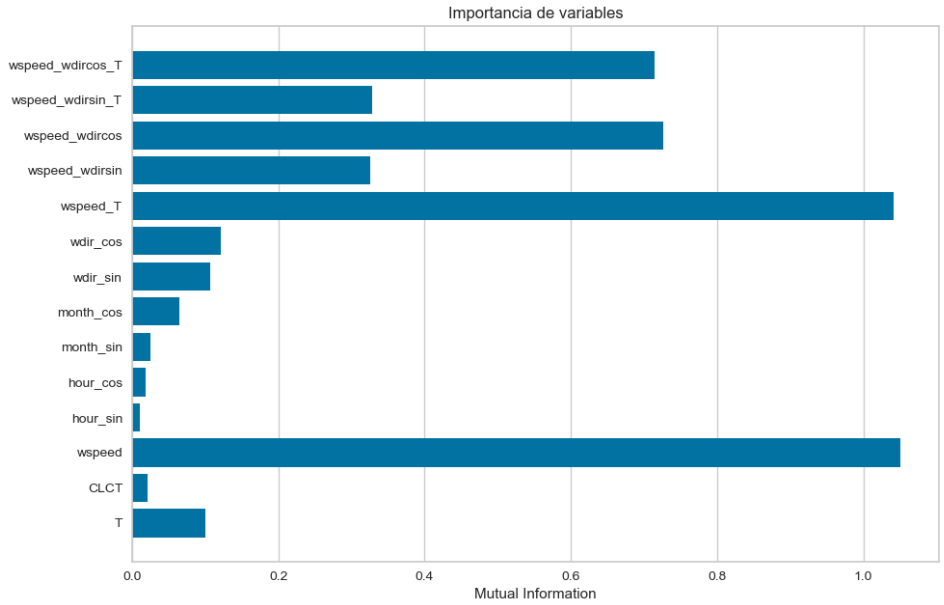

Figura 25: Importancia de las variables del conjunto de entrenamiento extendido para WF1.

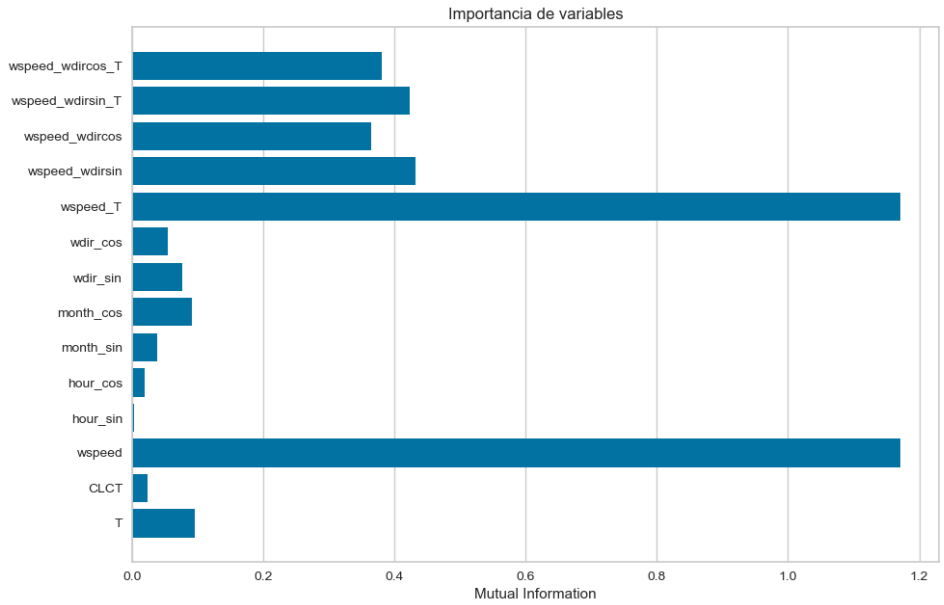

Figura 26: Importancia de las variables del conjunto de entrenamiento extendido para WF2.

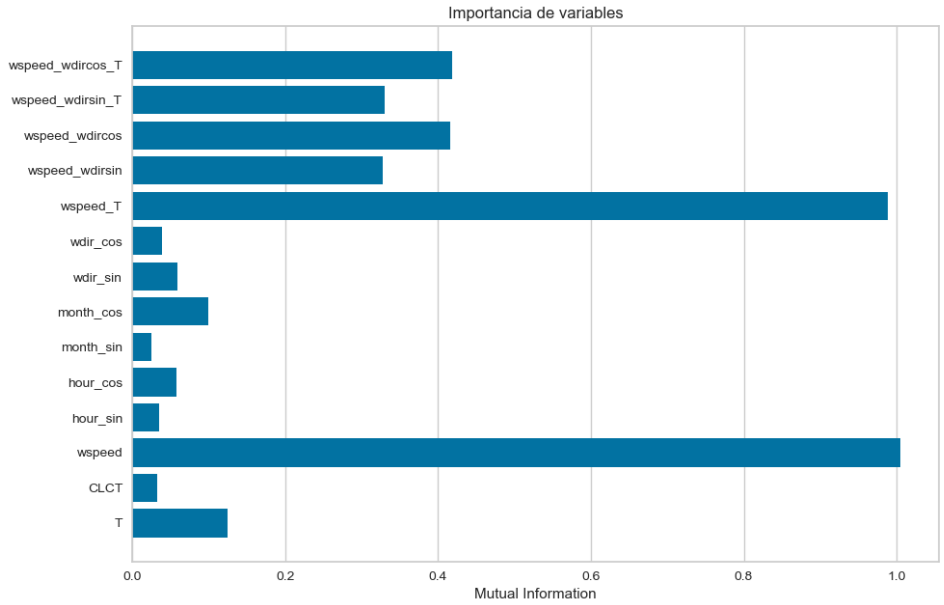

Figura 27: Importancia de las variables del conjunto de entrenamiento extendido para WF3.

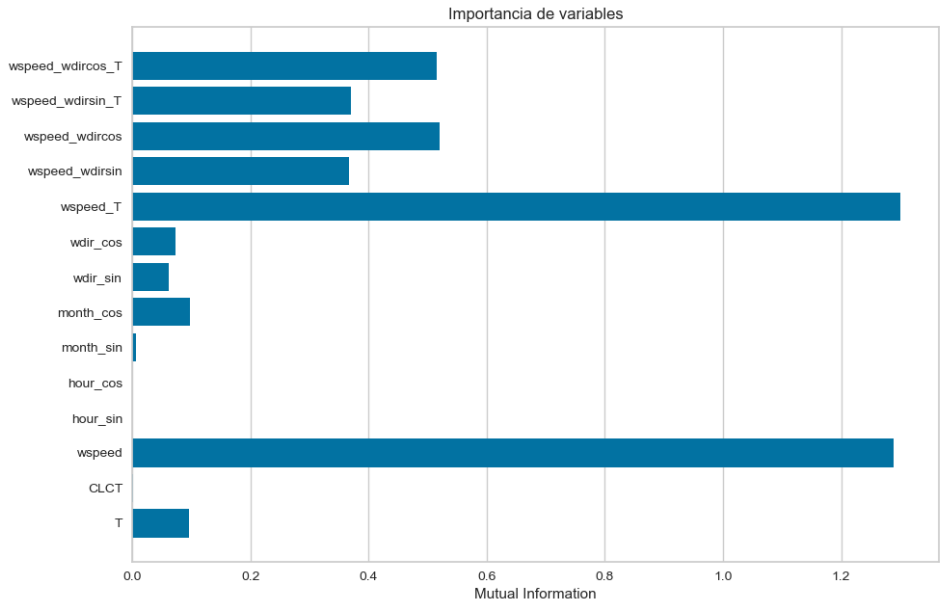

Figura 28: Importancia de las variables del conjunto de entrenamiento extendido para WF4.

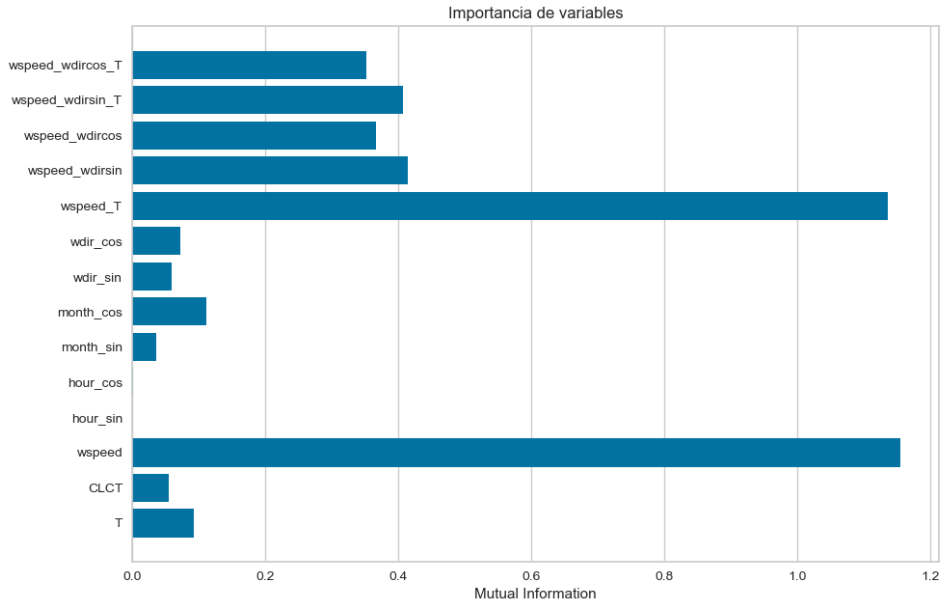

Figura 29: Importancia de las variables del conjunto de entrenamiento extendido para WF5.

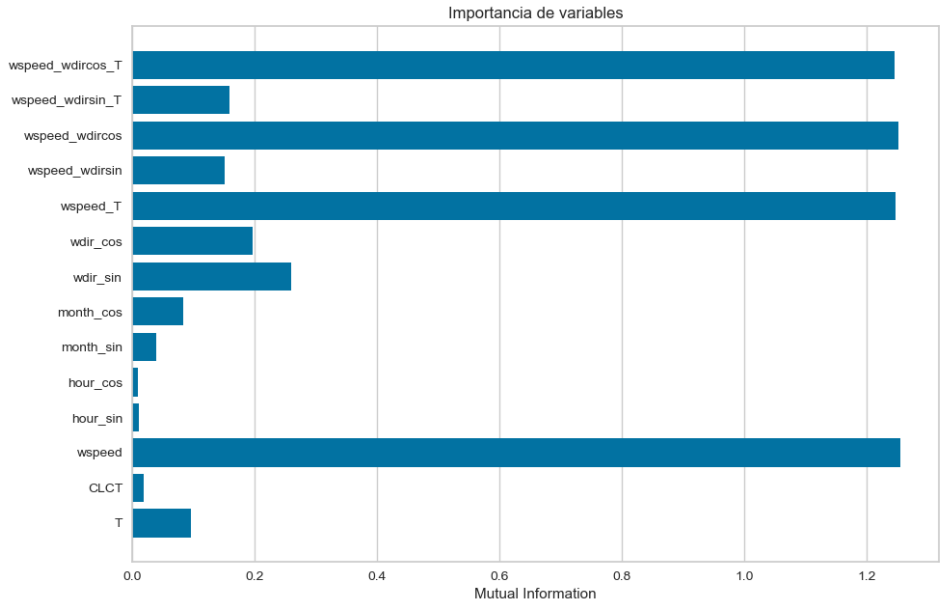

Figura 30: Importancia de las variables del conjunto de entrenamiento extendido para WF6.

Las seis variables más importantes han resultado ser la velocidad y las interacciones de ésta, tanto con la temperatura como con las componentes seno y coseno de la dirección. Cabe señalar que, en este caso, la velocidad sigue siendo la más importante pero le siguen muy de cerca las interacciones, no como ocurre en el conjunto básico, donde la velocidad siempre es la predominante con mucha diferencia sobre las demás. La temperatura por si sola es mucho menos influyente que sus interacciones con la velocidad

y la dirección. Además, se comprueba que el hecho de codificar las variables mes y hora en componentes seno y coseno no resulta en un aumento en su influencia sobre la producción. La cobertura del cielo CLCT, al igual que en el conjunto básico, sigue teniendo muy poca influencia sobre la producción.

Las tablas 6 a 11 recogen los resultados obtenidos tras la aplicación de los modelos en el conjunto de test en función del número de predictores.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF1	KNN	1	29.995	0.741	1.930
		2	**29.735**	0.747	1.909
		3	31.071	0.724	1.991
		4	31.103	0.728	1.978
		5	32.175	0.717	2.018
		6	32.264	0.716	2.022
	MARS	1	30.196	0.744	1.917
		2	**29.025**	0.744	1.918
		3	32.468	0.710	2.042
		4	31.615	0.720	2.006
		5	32.045	0.724	1.992
		6	31.631	0.720	2.007
	RF	1	29.915	0.744	1.920
		2	**29.547**	0.756	1.873
		3	31.627	0.720	2.007
		4	31.692	0.719	2.010
		5	31.511	0.723	1.996
		6	31.792	0.722	1.998
	SVM	1	34.497	0.619	2.341
		2	**30.475**	0.704	2.064
		3	33.694	0.693	2.102
		4	34.893	0.674	2.164
		5	31.935	0.710	2.042
		6	31.779	0.718	2.013

Tabla 6: Resultados del error de predicción obtenido para WF1 con los distintos algoritmos en función del número de predictores. La lista en orden de importancia es: 1. wspeed, 2. wspeed_T, 3. wspeed_wdircos, 4. wspeed_wdircos_T, 5. wspeed_wdirsin_T y 6. wspeed_wdirsin.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF2	KNN	1	26.7776	0.8788	0.8986
		2	26.6493	0.8823	0.8853
		3	**26.1515**	0.8826	0.8843
		4	26.3119	0.8816	0.8880
		5	26.5863	0.8803	0.8931
		6	26.6086	0.8815	0.8885
	MARS	1	26.7105	0.8806	0.8918
		2	26.1146	0.8860	0.8713
		3	25.2177	0.8932	0.8436
		4	25.1958	0.8933	0.8432
		5	**25.1925**	0.8933	0.8431
		6	25.1925	0.8933	0.8431
	RF	1	26.8914	0.8775	0.9034
		2	26.5014	0.8827	0.8838
		3	25.8197	0.8875	0.8658
		4	25.7382	0.8875	0.8657
		5	**25.6741**	0.8880	0.8636
		6	25.7065	0.8876	0.8653
	SVM	1	26.5598	0.8785	0.8995
		2	25.9989	0.8843	0.8779
		3	25.3901	0.8881	0.8632
		4	25.3310	0.8888	0.8607
		5	**25.1591**	0.8894	0.8584
		6	25.2384	0.8892	0.8589

Tabla 7: Resultados del error de predicción obtenido para WF2 con los distintos algoritmos en función del número de predictores. La lista en orden de importancia es 1. wspeed, 2. wspeed_T, 3. wspeed_wdirsin, 4. wspeed_wdirsin_T, 5. wspeed_wdircos_T y 6. wspeed_wdircos.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF3	KNN	1	30.427	0.719	1.353
		2	30.509	0.718	1.355
		3	30.818	0.717	1.357
		4	30.530	0.726	1.335
		5	28.719	0.753	1.267
		6	**28.208**	0.780	1.197
	MARS	1	30.499	0.714	1.364
		2	32.825	0.651	1.506
		3	32.526	0.674	1.456
		4	33.475	0.671	1.464
		5	29.509	0.713	1.366
		6	**29.506**	0.696	1.407
	RF	1	30.677	0.709	1.375
		2	30.610	0.714	1.364
		3	30.671	0.714	1.364
		4	31.098	0.705	1.385
		5	**28.384**	0.723	1.342
		6	28.444	0.722	1.345
	SVM	1	29.491	0.731	1.322
		2	29.522	0.724	1.339
		3	31.132	0.703	1.390
		4	30.503	0.716	1.359
		5	28.059	0.763	1.242
		6	**28.035**	0.765	1.235

Tabla 8: Resultados del error de predicción obtenido para WF3 con los distintos algoritmos en función del número de predictores. La lista en orden de importancia es 1. wspeed, 2. wspeed_T, 3. wspeed_wdircos_T, 4. wspeed_wdircos, 5. wspeed_wdirsin_T y 6. wspeed_wdirsin.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF4	KNN	1	**28.232**	0.839	0.961
		2	28.799	0.838	0.966
		3	29.632	0.828	0.994
		4	30.983	0.814	1.036
		5	29.627	0.827	0.998
		6	29.473	0.829	0.993
	MARS	1	28.118	0.838	0.965
		2	28.695	0.832	0.983
		3	28.311	0.837	0.970
		4	28.311	0.837	0.970
		5	**27.905**	0.824	1.008
		6	27.928	0.823	1.008
	RF	1	**27.785**	0.838	0.965
		2	28.270	0.836	0.970
		3	28.264	0.834	0.977
		4	28.937	0.829	0.991
		5	28.440	0.833	0.980
		6	28.715	0.832	0.983
	SVM	1	27.787	0.833	0.979
		2	27.861	0.831	0.987
		3	28.296	0.827	0.999
		4	28.626	0.823	1.010
		5	26.784	0.835	0.973
		6	**26.616**	0.839	0.964

Tabla 9: Resultados del error de predicción obtenido para WF4 con los distintos algoritmos en función del número de predictores. La lista en orden de importancia es 1. wspeed_T, 2. wspeed, 3. wspeed_wdircos, 4. wspeed_wdircos_T, 5. wspeed_wdirsin_T y 6. wspeed_wdirsin.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF5	KNN	1	28.966	0.865	1.198
		2	27.275	0.873	1.162
		3	26.574	0.881	1.122
		4	26.831	0.879	1.133
		5	**26.224**	0.885	1.104
		6	26.677	0.881	1.121
	MARS	1	27.757	0.873	1.161
		2	26.664	0.876	1.145
		3	25.171	0.892	1.072
		4	**24.624**	0.895	1.057
		5	25.582	0.889	1.085
		6	25.764	0.888	1.089
	RF	1	29.057	0.861	1.212
		2	27.824	0.869	1.178
		3	26.827	0.879	1.131
		4	26.675	0.881	1.124
		5	26.454	0.884	1.107
		6	**26.309**	0.885	1.104
	SVM	1	27.745	0.868	1.182
		2	25.946	0.879	1.135
		3	24.641	0.895	1.057
		4	24.330	0.897	1.045
		5	**23.806**	0.892	1.070
		6	23.854	0.893	1.066

Tabla 10: Resultados del error de predicción obtenido para WF5 con los distintos algoritmos en función del número de predictores. La lista en orden de importancia es 1. wspeed, 2. wspeed_T, 3. wspeed_wdirsin, 4. wspeed_wdirsin_T, 5. wspeed_wdircos y 6. wspeed_wdircos_T.

Parque	Algoritmo	# Predictores	CAPE	R^2	RMSE
WF6	KNN	1	17.044	0.885	0.507
		2	**16.874**	0.887	0.504
		3	17.301	0.880	0.520
		4	17.560	0.877	0.526
		5	18.238	0.867	0.546
		6	18.344	0.866	0.550
	MARS	1	16.850	0.887	0.503
		2	**16.715**	0.888	0.502
		3	17.021	0.890	0.497
		4	17.021	0.890	0.497
		5	17.382	0.885	0.508
		6	17.213	0.884	0.510
	RF	1	16.972	0.885	0.508
		2	16.886	0.886	0.507
		3	**16.659**	0.889	0.499
		4	17.095	0.882	0.515
		5	17.399	0.879	0.522
		6	17.400	0.879	0.520
	SVM	1	16.901	0.884	0.511
		2	16.851	0.885	0.509
		3	**16.577**	0.889	0.498
		4	16.712	0.889	0.500
		5	17.264	0.879	0.521
		6	17.246	0.879	0.521

Tabla 11: Resultados del error de predicción obtenido para WF6 con los distintos algoritmos en función del número de predictores. La lista en orden de importancia es 1. wspeed, 2. wspeed_wdircos, 3. wspeed_T, 4. wspeed_wdircos_T, 5.wdir_sin y 6. wdir_cos_T.

Con esta versión del conjunto de datos también se confirma que WF1 y WF6 son los parques donde peor y mejor resultado se obtiene respectivamente. Concretamente, las mejores predicciones se dan para el algoritmo SVM con un CAPE del 16.58 %, RMSE de 0.498 MW y un R^2 de casi el 89 %, con los tres predictores más importantes. Las peores predicciones se corresponden con un CAPE del 34.89 %, RMSE de 2.16 MW y un R^2 del 67.4 %, utilizando un SVM con los cuatro predictores más importantes.

5.3 Comparativa: conjunto básico vs. conjunto extendido

En la tabla 12 se compara la métrica CAPE obtenida para cada tipo de conjunto de datos. Así, en la columna 'Variación', un porcentaje negativo indica una disminución del CAPE.

Parque	KNN			MARS			RF			SVM		
	Básico	Extendido	Variación (%)	Básico	Extendido	Variación (%)	Básico	Extendido	Variación (%)	Básico	Extendido	Variación (%)
WF1	29.995	29.735	-0.87	30.196	29.025	-3.88	29.915	29.547	-1.23	31.734	30.475	-3.97
WF2	26.011	26.1515	0.54	26	25.1925	-3.11	25.578	25.6741	0.38	25.473	25.1591	-1.23
WF3	30.427	28.208	-7.29	30.499	29.506	-3.26	30.677	28.384	-7.47	29.491	28.035	-4.94
WF4	29.223	28.232	-3.39	29.026	27.905	-3.86	28.998	27.785	-4.18	28.428	26.616	-6.37
WF5	26.467	26.224	-0.92	27.757	24.624	-11.29	27.555	26.309	-4.52	27.745	23.806	-14.20
WF6	17.044	16.874	-1.00	16.85	16.715	-0.80	16.941	16.659	-1.66	16.912	16.577	-1.98

Tabla 12: Variación del CAPE obtenido con el conjunto extendido respecto al básico, para cada parque y modelo.

Salvo para el parque WF2 con los modelos KNN y RF, la variación del CAPE es negativa en todos los casos indicando, por tanto, una mejora en la precisión del modelo cuando se utiliza el conjunto de datos extendido. Esto puede ser atribuible a que las variables de interacción, que tienen una importancia alta sobre la variable respuesta, son capaces de captar información relevante de los datos que no detectan las variables individuales.

El valor del CAPE para todos los modelos, parques y versiones del conjunto de entrenamiento no baja del 23 %, a excepción del parque WF6, donde se han conseguido siempre valores inferiores al 20 %. Como se comentaba anteriormente, los peores resultados se han dado con mayor frecuencia en WF1, obteniéndose valores intermedios y muy parecidos entre los parques WF2, 3, 4 y 5. La razón se puede intuir atendiendo a las gráficas de la figura 8. Las distribuciones de datos son parecidas en todos los parques excepto en WF1 y WF6, donde en el primer caso se tiene una distribución que no se ajusta bien a la curva de potencia ideal, ocurriendo lo contrario en el segundo caso.

Otra razón para que WF1 presente siempre los peores resultados es que, como se puede ver en la gráfica de la figura 31, entre el 16 y el 23 de diciembre la producción fue exactamente cero, lo que probablemente indique que durante ese tiempo las turbinas de ese parque estuvieron paradas por algún motivo, por ejemplo, por mantenimiento. En el resto de parques no se dan periodos tan largos de inactividad, por lo que se consiguen errores medios más bajos. Estos periodos de inactividad son prácticamente imposibles de captar por los modelos si no se tiene información extra que sirva para identificarlos.

En cuanto a las métricas, se observa que un mínimo en el CAPE implica un mínimo en el RMSE y un máximo en R^2. Tanto el CAPE como el RMSE son medidas de error en la predicción, mientras que el R^2 es un indicador de la cantidad de varianza en la producción que el modelo es capaz de captar debida a la variación de los predictores , teniendo que un menor valor implica una bondad mayor en el ajuste.

Las predicciones horarias para el mes de test del mejor modelo en cada parque se muestran en las figuras 31 a 36. El mejor algoritmo ha sido siempre el SVM entrenado con la versión extendida del conjunto de datos, excepto para WF1 donde el ganador es el MARS.

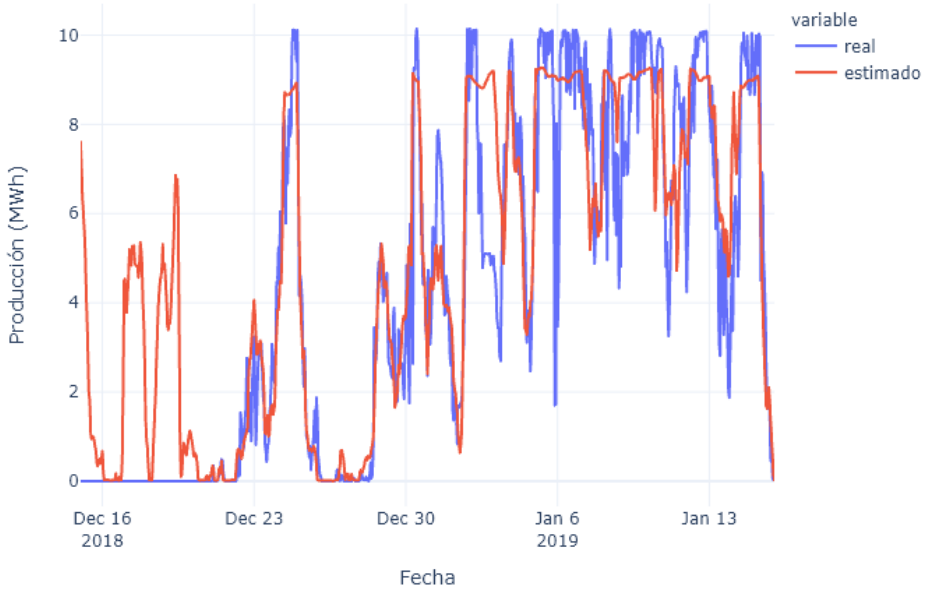

Figura 31: Predicciones del mejor modelo para WF1: MARS con dos predictores.

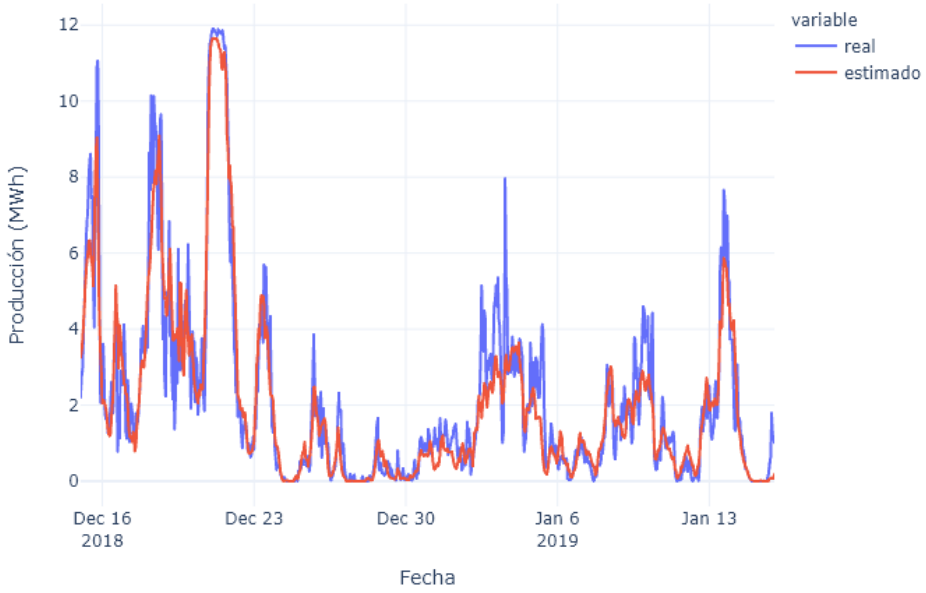

Figura 32: Predicciones del mejor modelo para WF2: SVM con cinco predictores.

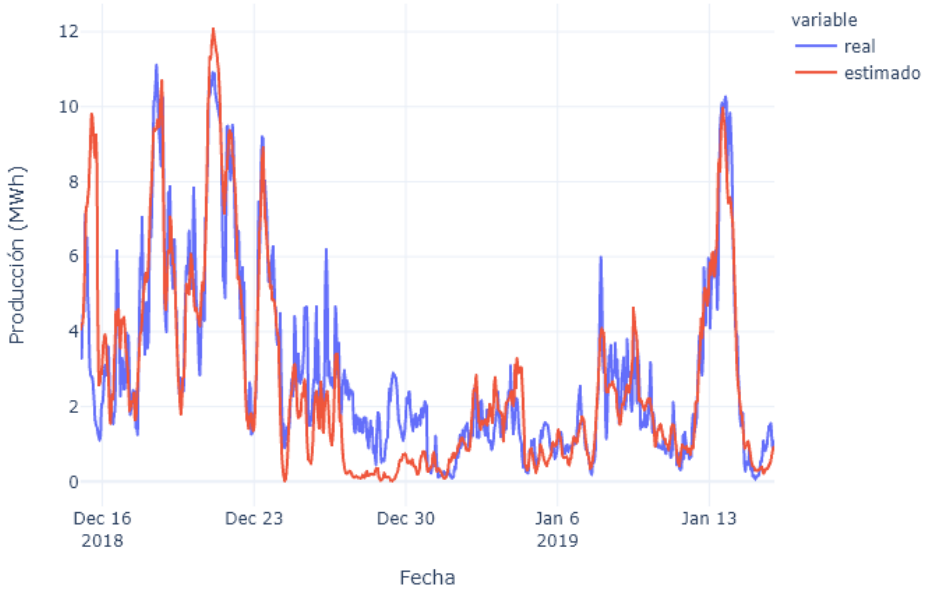

Figura 33: Predicciones del mejor modelo para WF3: SVM con seis predictores.

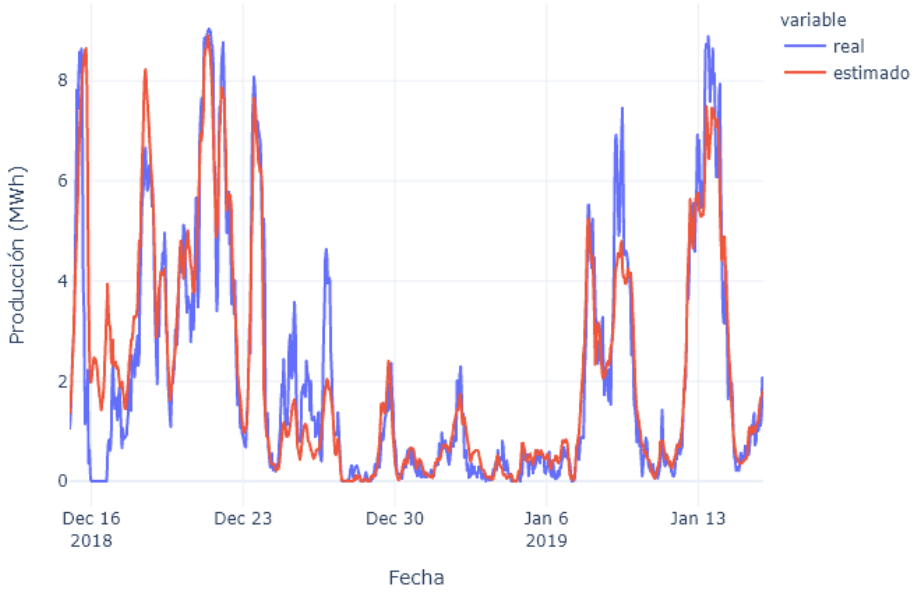

Figura 34: Predicciones del mejor modelo para WF4: SVM con seis predictores.

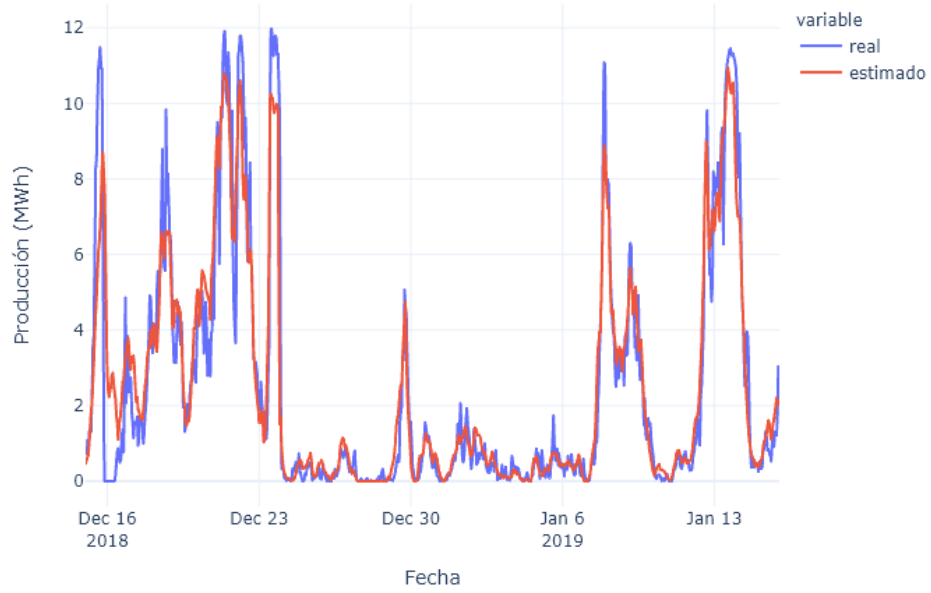

Figura 35: Predicciones del mejor modelo para WF5: SVM con cinco predictores.

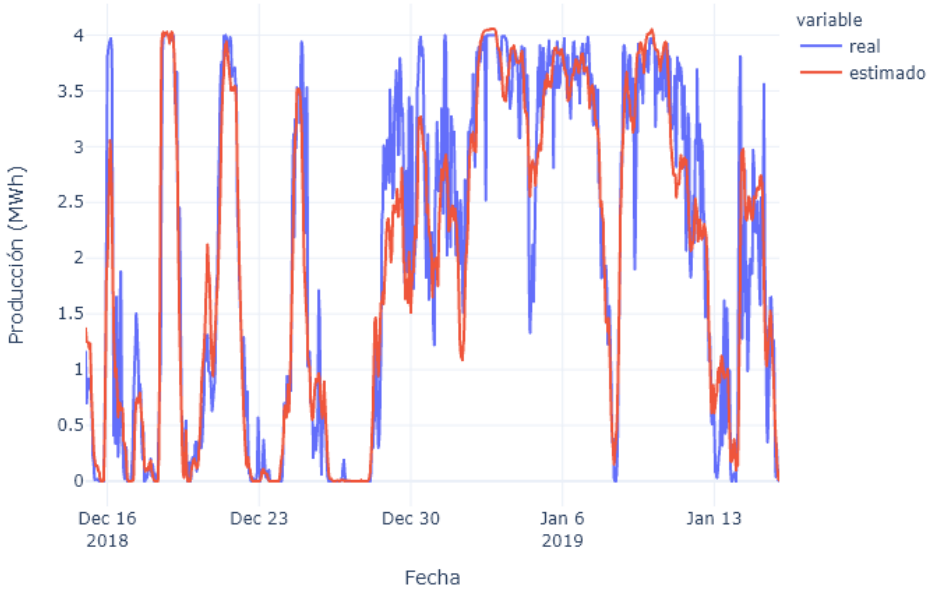

Figura 36: Predicciones del mejor modelo para WF6: SVM con tres predictores.

Resultados obtenidos en la competición. El *benchmark* establecido por CNR para el CAPE es del 31.73 % utilizando una red neuronal. Con los mejores modelos obtenidos en este estudio en cada parque eólico se ha conseguido un CAPE del 35 %. Hay que tener en cuenta que la métrica en la competición la calculan de forma global como la media del CAPE de los seis parques, por lo que esto

resulta en una contribución no homogénea al valor final, siendo el CAPE más sensible a los parques con los valores más altos de la producción.

6 Conclusiones y trabajos futuros

En este trabajo se han aplicado métodos de Machine Learning para la previsión de energía eólica en seis parques de diferentes localizaciones francesas. Estos métodos han sido implementados mediante el desarrollo de un software que ha permitido configurar y automatizar los experimentos realizados de manera sencilla, demostrando ser una herramienta muy útil para el análisis del problema estudiado, tanto por su flexibilidad de configuración, como por su grado de automatización, dando la posibilidad de realizar gran variedad de experimentos. Se ha invertido un tiempo aproximado de 700 horas para la realización de este proyecto, desde las etapas de estudio previo, investigación y documentación sobre el estado del arte, pasando por el desarrollo del software (que ha supuesto un alto porcentaje del tiempo invertido), hasta la fase de experimentación y elaboración del informe. Siguiendo la metodología descrita, se han entrenado cuatro algoritmos de aprendizaje automático (KNN, MARS, RF y SVM) con dos conjuntos de entrenamiento distintos, uno que incluye variables básicas como la velocidad y dirección del viento y la temperatura, y otro extendido que incluye interacciones entre estas variables. A la vista de los resultados obtenidos, se pueden extraer las siguientes conclusiones:

1. Es muy importante realizar una buena limpieza y preparación de los datos, eliminando valores anómalos y completando valores perdidos.

2. El escalado y centrado de los datos es muy recomendable de cara al entrenamiento de los algoritmos para evitar dar mayor influencia a unas variables que a otras, dependiendo del rango de valores en cada una.

3. La selección de variables predictoras en base a la física del problema ha redundado en modelos que explican satisfactoriamente los datos, consiguiendo valores de R^2 entre el 70 y el 90 %.

4. Este problema de predicción es totalmente dependiente del parque considerado, es decir, la relación estadística entre las variables de entrada y la de salida difiere de un parque a otro, por lo que se hace necesario entrenar los algoritmos por parque y no de de manera global.

5. No se puede concluir que un número de predictores determinado proporcione el mejor modelo en todos los parques. En algunas ocasiones, una mayor complejidad ha redundado en mejores resultados, aunque en un buen número de experimentos, modelos más simples han funcionado de manera aceptable.

A pesar del margen de mejora que existe, los resultados obtenidos son prometedores, consiguiendo precisiones no muy lejanas de la de referencia establecida en la competición de Ciencia de Datos donde se propone el problema estudiado.

Como posibles trabajos futuros se proponen los siguientes:

- Explorar nuevas técnicas de detección de valores anómalos, como pueden ser los algoritmos de agrupamiento no supervisados para obtener curvas de potencia más ajustadas.

- Aplicación de métodos de aprendizaje profundo, como las redes neuronales.

- Estudio de los modelos de *stacking* (combinación de predictores) para conseguir predicciones más precisas.

Referencias

[1] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 2006.

[2] Brillinger, David «Some data analyses using mutual information». *Brazilian Journal of Probability and Statistics*, 2004, **18**.

[3] Ding, Y. *Data Science for Wind Energy*, 2019.

[4] Drucker, Harris; Burges, C.; Kaufman, L.; Smola, Alex y Vapnik, V. «Support Vector Regression Machines». En: *NIPS*, , 1996.

[5] Freidman, J. «Multivariate adaptive regression splines», 1991.

[6] James, G.; Witten, D.; Hastie, T. y Tibshirani, R. *An introduction to statistical learning*, 2013.

[7] Jung, Jaesung y Broadwater, R. «Current status and future advances for wind speed and power forecasting». *Renewable & Sustainable Energy Reviews*, 2014, **31**, pp. 762–777.

[8] Kuhn, M. y Johnson, K. *Applied Predictive Modeling*, 2013.

[9] Lahouar, Ali y Ben Hadj Slama, J. «Hour-ahead wind power forecast based on random forests». *Renewable Energy*, 2017, **109**, pp.–. doi: 10.1016/j.renene.2017.03.064.

[10] Lee, Giwhyun; Ding, Yu; Genton, M. y Xie, Le «Power Curve Estimation With Multivariate Environmental Factors for Inland and Offshore Wind Farms». *Journal of the American Statistical Association*, 2015, **110**, pp. 56 – 67.

[11] Li, Cunbin; Lin, Shuaishuai; Xu, Fangqiu; Liu, Ding y Liu, Jicheng «Short-term wind power prediction based on data mining technology and improved support vector machine method: A case study in Northwest China». *Journal of Cleaner Production*, 2018, **205**. doi: 10.1016/j.jclepro.2018.09.143.

[12] Li, Yanting; He, Yong; Su, Yan y Shu, Lianjie «Forecasting the daily power output of a grid-connected photovoltaic system based on multivariate adaptive regression splines». *Applied Energy*, 2016, **180**, pp. 392–401. doi: 10.1016/j.apenergy.2016.07.052.

[13] Mangalova, Ekaterina y Agafonov, Evgeny «Wind power forecasting using the -nearest neighbors algorithm». *International Journal of Forecasting*, 2013, **30**. doi: 10.1016/j.ijforecast.2013.07.008.

[14] Mohandes, M.; Halawani, T.; Rehman, S. y Hussain, A. A. «Support vector machines for wind speed prediction». *Renewable Energy*, 2004, **29**, pp. 939–947.

[15] 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. y Duchesnay, E. «Scikit-learn: Machine Learning in Python». *Journal of Machine Learning Research*, 2011, **12**, pp. 2825–2830.

[16] Sebastian Raschka, Vahid Mirjalili *Python Machine Learning*. Packt Publishing, 2017.

[17] Tong, Wei Qin «Fundamentals Of Wind Energy». *WIT Transactions on State-of-the-art in Science and Engineering*, 2010, **44**.

[18] Wang, Yun; Hu, Qinghua; Li, Linhao; Foley, Aoife y Srinivasan, Dipti ≪Approaches to wind power curve modeling: A review and discussion≫. *Renewable and Sustainable Energy Reviews*, 2019, **116**. doi: 10.1016/j.rser.2019.109422.

[19] Wikipedia contributors ≪Euclidean distance — Wikipedia, The Free Encyclopedia≫, 2020. https://en.wikipedia.org/w/index.php?title=Euclidean_distance&oldid=976383156. [Online; accessed 18-October-2020].

[20] —— ≪Power transform — Wikipedia, The Free Encyclopedia≫, 2020. https://en.wikipedia.org/w/index.php?title=Power_transform&oldid=972178635. [Online; accessed 19-October-2020].

[21] —— ≪Taxicab geometry — Wikipedia, The Free Encyclopedia≫, 2020. https://en.wikipedia.org/w/index.php?title=Taxicab_geometry&oldid=960454083. [Online; accessed 18-October-2020].