PREDICCIÓN DE LA PRODUCCIÓN DE ENERGÍA EÓLICA CON SCIKIT-LEARN (3.5 PUNTOS)

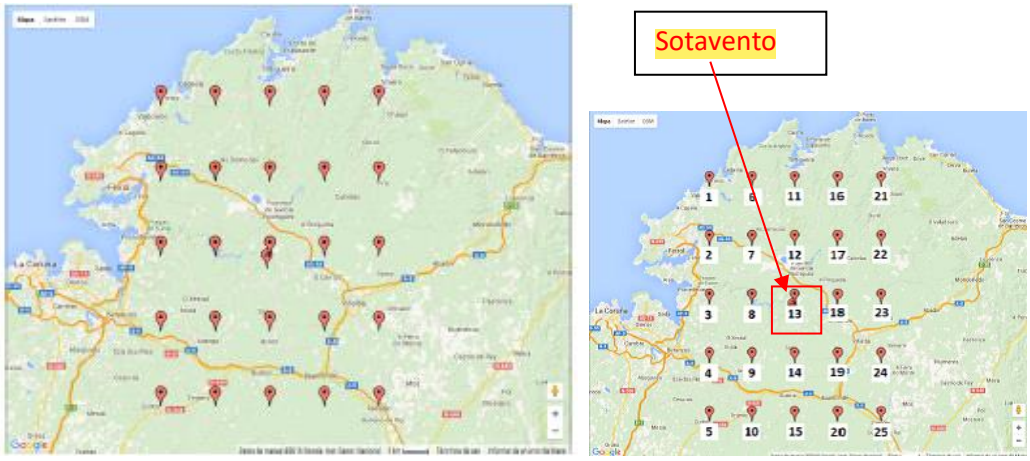
INTRODUCCIÓN

El propósito de esta primera práctica es practicar con diferentes métodos de aprendizaje automático. Además, se trata de practicar todo el proceso: determinar el mejor método para un conjunto de datos (**selección de modelo**, incluido el ajuste de hiperparámetros/HPO), estimar el rendimiento futuro del mejor método (**evaluación de modelo**) y **construir el modelo final** y usarlo para hacer nuevas predicciones sobre nuevos datos (**uso del modelo**).

Actualmente, las redes eléctricas de los países avanzados dependen cada vez más de fuentes de energía renovable, principalmente eólica y solar. Para integrar estas fuentes de energía en la red eléctrica, es necesario prever la cantidad de electricidad que se generará con 24 horas de anticipación, de modo que las plantas de energía conectadas a la red eléctrica puedan planificarse y prepararse para satisfacer la oferta y demanda durante el día siguiente. Esto no representa un problema para las fuentes de energía tradicionales (gas, petróleo, hidroeléctrica, etc.) porque pueden generarse a voluntad. Pero la energía solar y eólica no están bajo el control del operador energético, ya que dependen de la meteorología y, por tanto, la única alternativa para integrarlas en la red, es hacer buenos pronósticos meteorológicos, como los proporcionados por modelos NWP (Numeral Weather Prediction). Estos modelos predicen variables meteorológicas, como la componente de viento U a 100 metros, relacionada con la velocidad del viento. Sin embargo, la relación entre esas variables y la electricidad finalmente producida por la planta eólica o solar, es complicada. Los modelos de Aprendizaje Automático pueden utilizarse para esta tarea. En esta práctica, utilizaremos variables meteorológicas pronosticadas por el ECMWF (European Centre for Medium-Range Weather Forecasts: https://www.ecmwf.int/) como atributos de entrada para un modelo de aprendizaje automático. La variable de respuesta será la energía eléctrica que se producirá en el parque eólico experimental de Sotavento (https://www.sotaventogalicia.com/).

Parque eólico de Sotavento (Lugo).

En los datos se dan los valores de 22 variables meteorológicas. El modelo meteorológico ECMWF proporciona estas variables en los puntos de una cuadrícula 5x5 alrededor de Sotavento. Es por eso que, por ejemplo, el atributo iews aparece 25 veces en el conjunto de datos (iews.1, iews.2, ..., iews.13, ..., iews.25). Aunque las 22*5*5 variables están disponibles en los datos, en esta práctica sólo usaremos las variables estimadas sobre la localización de Sotavento (la 13).



Cuadrícula de 5x5 alrededor de Sotavento.

CONSIDERACIONES GENERALES:

1. Los resultados deben ser reproducibles. Usar como semilla el número NIA de uno de los miembros del grupo o el número de grupo.
2. Se proporcionan dos archivos de conjunto de datos:
   a. datos disponibles:

      wind_ava = pd.read_csv('wind_available.csv.gzip', compression="gzip")

   b. datos de competición *'wind_competition.csv.gzip'*, sobre los que usar el modelo final para hacer predicciones. No contienen la variable de respuesta (dado que cada grupo usará su modelo final para hacer predicciones).
3. Para realizar la práctica, los estudiantes emplearán un **repositorio de código en GitHub**. Para ello, cada grupo debe crear un repositorio de código privado y agregar como «colaborador» al profesor de prácticas (que indicará a los estudiantes su nombre de usuario en GitHub). Durante la primera semana, el grupo hará llegar al profesor de prácticas el enlace al repositorio de GitHub donde se harán los *commits* (debe haber un único repositorio por grupo). Se espera que cada grupo haga al menos un *commit* semanal del código de la práctica. Esta parte de la práctica se valorará con **0.5 puntos**. Además, también habrá que entregar el cuaderno (notebook) final a través de Aula Global.
4. Será necesario explicar cómo se ha usado ChatGPT en esta práctica. Se pueden incluir prompts (y respuestas) relevantes, casos en los que ChatGPT estaba equivocado, etc.

QUÉ HACER:

1. (**0.3 puntos**) Realizar un EDA simplificado, principalmente para determinar cuántas características e instancias hay, qué variables son categóricas/numéricas, si hay valores faltantes (missing values) y qué variables los tienen, si hay columnas constantes (que deberían eliminarse) y si es un problema de regresión o clasificación. Se puede analizar otras cuestiones que se consideren relevantes. Tened en cuenta que la variable de respuesta es "energía". **Importante:** de los datos originales, hay que quitar todas las variables meteorológicas que no correspondan a la localización de Sotavento (la localización 13).
2. (**0.1 puntos**) Decidir cómo se va a llevar a cabo la evaluación *outer* (estimación de rendimiento futuro / evaluación de modelo) y la evaluación *inner* (para comparar diferentes alternativas y ajustar hiper-parámetros). Decidir qué métrica(s) se van a usar. Justificar las decisiones.

3. (**0.2 puntos**) Decidir, usando KNN el método de escalado más apropiado para este problema y usarlo de aquí en adelante cuando sea necesario.
4. (**1.2 puntos**) A continuación, se considerarán estos métodos: KNN, árboles de regresión, regresión lineal (la normal y al menos, la variante Lasso) y SVM:
    a. Se evaluarán dichos modelos con sus hiperparámetros por omisión. También se medirán los tiempos que tarda el entrenamiento.
    b. Después, se ajustarán los hiperparámetros más importantes de cada método y se obtendrá su evaluación. Medir tiempos del entrenamiento, ahora con HPO.
    c. Obtener algunas conclusiones, tales como: ¿cuál es el mejor método? ¿Cuál de los métodos básicos de aprendizaje automático es más rápido? ¿Los resultados son mejores que los regresores triviales/naive/dummy? ¿El ajuste de hiperparámetros mejora con respecto a los valores por omisión? ¿Hay algún equilibrio entre tiempo de ejecución y mejora de resultados? ¿Es posible extraer de alguna técnica qué atributos son más relevantes? etc.
5. (**0.2 puntos**) Seleccionar el mejor método (usando la evaluación *inner*), evaluarlo, construir modelo final, hacer predicciones para la competición.
    a. Seleccionar la mejor alternativa de las evaluadas en los puntos anteriores.
    b. Estimar el rendimiento / desempeño futuro del modelo (evaluación *outer*). Esta es una estimación de cómo se desempeñaría el modelo en la competición.
    c. Entrenar el modelo final. Guardarlo en un fichero (llamado «modelo_final.pkl»).
    d. Utilizar el modelo final para obtener predicciones para el conjunto de datos de la competición. Guardar estas predicciones en un fichero (llamado «predicciones.csv»).
6. (**0.8 puntos**) Sotavento está interesada en saber si las predicciones de los modelos son de más calidad cuando la energía producida es baja o cuando es alta. Primero, se pide comprobar con el mejor modelo obtenido hasta el momento, si las predicciones para valores altos son peores que para valores bajos. Además, se nos propone convertir el problema de regresión en uno de clasificación, de la siguiente manera: cuando la energía sea menor que el tercer cuantil, se considerará clase "baja", y cuando sea mayor, clase "alta". Resolver el problema utilizando ahora algún método para clasificación, eligiendo métricas adecuadas, intentando obtener los mejores resultados y alcanzando conclusiones.
7. (**0.2 puntos**) Explicar brevemente cómo se ha usado ChatGPT en esta práctica. Se pueden incluir prompts (y respuestas) relevantes, casos en los que ChatGPT estaba equivocado, etc. No más de 2 páginas en el informe.
8. (**0.5 puntos**) Se recuerda que además de la entrega final, cada semana hay que hacer al menos un commit en el **GitHub** privado de cada grupo.

QUÉ ENTREGAR:

1. Código con dos notebooks:
    1. Uno que haga el EDA, ajuste de hiperparámetros, selección de modelo, etc. El notebook tiene que tener explicaciones de los procesos, análisis de los resultados, justificaciones de las decisiones, etc., preferiblemente usando tablas y gráficos.
    2. Otro que cargue el modelo final y lo use para hacer predicciones en los datos de la competición.
    3. Otro donde se resuelva el problema transformado en clasificación.
2. El archivo conteniendo el modelo final (llamado «modelo_final.pkl») y el archivo conteniendo las predicciones ("predicciones.csv").
3. El código y los archivos (modelo y predicciones) se entregarán finalmente en Aula Global en un .zip
4. Se recuerda que además de la entrega final, cada semana hay que hacer al menos un commit en el **GitHub** privado de cada grupo **(0.5 puntos).**

APÉNDICE: NOMBRE DE LOS ATRIBUTOS / VARIABLES (METEOROLÓGICAS) DE ENTRADA AL MODELO:

- t2m: 2 metre temperature
- u10: 10 metre U wind component
- v10: 10 metre V wind component
- u100: 100 metre U wind component
- v100: 100 metre V wind component
- cape: Convective available potential energy
- flsr: Forecast logarithm of surface roughness for heat
- fsr: Forecast surface roughness
- iews: Instantaneous eastward turbulent surface stress
- inss: Instantaneous northward turbulent surface
- lai_hv: Leaf area index, high vegetation
- lai_lv: Leaf area index, low vegetation
- u10n: Neutral wind at 10 m u-component
- v10n: Neutral wind at 10 m v-component
- stl1: Soil temperature level 1
- stl2: Soil temperature level 2
- stl3: Soil temperature level 3
- stl4: Soil temperature level 4
- sp: Surface pressure
- p54.162: Vertical integral of temperature
- p59.162: Vertical integral of divergence of kinetic energy
- p55.162: Vertical integral of water vapour