Aprendizaje Automático

Práctica 1

PREDICCIÓN DE LA PRODUCCIÓN DE ENERGÍA SOLAR

3,5 puntos

INTRODUCCIÓN

El propósito de esta primera práctica es practicar con diferentes métodos de aprendizaje automático y ajuste / optimización de hiperparámetros (HPO). 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), 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**).

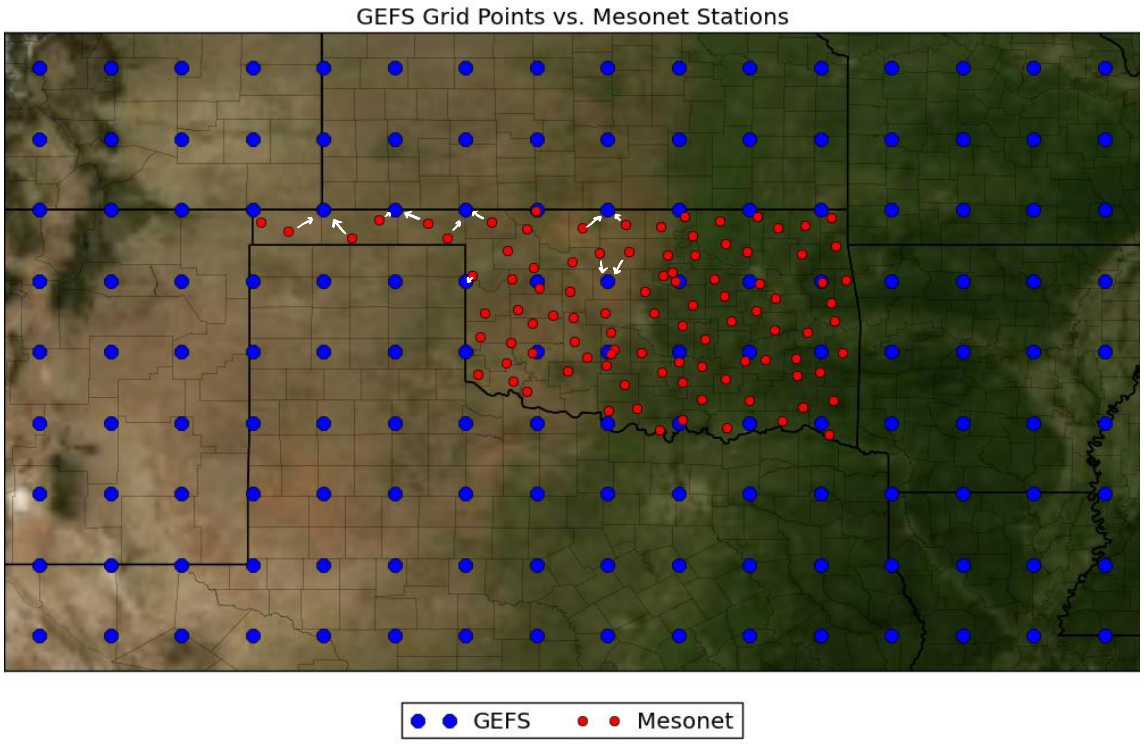
Hoy en día, las redes eléctricas de los países avanzados dependen cada vez más de fuentes de energía renovables no gestionable (no «despachables»), principalmente eólica y solar. Sin embargo, para integrar las fuentes de energía en la red eléctrica, se requiere que la cantidad de energía a generar se prevea 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 la demanda durante el día siguiente[1].

Esto no es un problema para las fuentes de energía tradicionales (gas, petróleo, energía hidroeléctrica, ...) porque se pueden generar (gestionar) a voluntad (quemando más gas, por ejemplo). Pero las energías solar y eólica no están bajo el control del operador energético, porque dependen de la meteorología. La única alternativa es predecirlas con la mayor precisión posible. Esto se puede lograr hasta cierto punto mediante pronósticos meteorológicos, los cuales recurren a la simulación de la atmósfera mediante modelos físico-matemáticos. El *Global Forecast System* (GFS, EE.UU., también llamado Global Ensemble Forecast System o GEFS) y el *European Centre for Medium-Range Weather Forecasts* (ECMWF) son dos de los modelos de predicción numérica del tiempo (NWP: Numerical Weather Prediction) más importantes.

Sin embargo, aunque los NWP son muy buenos para predecir variables como «promedio de flujo radiativo de onda larga descendente en la superficie», relacionado con la radiación solar, la relación entre esas variables y la electricidad realmente producida en las plantas solares no es sencilla. Los modelos de aprendizaje automático se pueden usar para esta última tarea.

En particular, vamos a utilizar variables meteorológicas predichas por GFS como atributos de entrada a un modelo de aprendizaje automático que es capaz de estimar cuánta energía solar se producirá en plantas solares del estado de Oklahoma. En la figura a continuación, los puntos rojos son las plantas solares y los puntos azules son localizaciones para las cuales GFS proporciona predicciones meteorológicas.

[1] https://es.wikipedia.org/wiki/Mercado_eléctrico



Para este conjunto de datos en particular, GFS da un pronóstico todos los días a las 00:00 UTC para el día siguiente. Concretamente, realiza predicciones para los siguientes 5 momentos del día siguiente: 1 (12 h), 2 (15 h), 3 (18 h), 4 (21 h), 5 (24 h), para 15 variables meteorológicas (apcp_sfc, dlwrf_sfc, ...) en cada uno de los puntos azules de la cuadrícula (se puede ver lo que significa cada variable en el apéndice). En esta práctica sólo utilizaremos las variables predichas en un punto azul, el más cercano al punto rojo para el que queremos predecir la energía solar en dicho punto. Por tanto, el número de variables de entrada es de 5 instantes de tiempo x 15 variables meteorológicas = 75 variables de entrada. El modelo de aprendizaje automático a construir viene representado por *f* en la siguiente ecuación:

1 - 12h
2 - 15h
3 - 18h
4 - 21h
5 - 24h

$$ES = f(apcp\_sfc_1, apcp\_sfc_2, \dots, apcp\_sfc_5, dlwrf\_sfc_1, dlwrf\_sfc_2, \dots, uswrf\_sfc_4, uswrf\_sfc_5)$$

En esta ecuación, *ES* es la energía solar acumulada al día siguiente. Este valor se puede aproximar mediante una función *f* que depende de las 75 variables especificadas, donde el subíndice determina el momento de tiempo. Por ejemplo, si se considera la variable *apcp_sf*, entonces $apcp\_sf_i$ representa el momento de tiempo *i*, donde *i = 1, 2, 3, 4, 5.*

Se proporcionan dos archivos, «disponibles» y «competición». El fichero «disponibles» contiene datos de 12 años (un día por fila), con 75 variables de entrada, más la variable de respuesta («salida», la cual representa la energía eléctrica solar **acumulada** producida durante el día correspondiente). Por su parte, «competición» contiene 2 años de datos (también, un dato para cada día), con 75 variables de entrada, pero sin la variable de respuesta, dado que habrá que utilizar el modelo para predecirla.

CONSIDERACIONES GENERALES

1. Los datos han sido extraídos de la competición planteada en Kaggle: https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest.
2. 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 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.

3. Los resultados deben ser reproducibles. Por lo tanto, hay que fijar la semilla de números aleatorios en los lugares adecuados. Se usará como semilla el NIA de uno de los miembros del grupo o bien el número del grupo de prácticas.
4. Para cada grupo, se proporcionan dos archivos de conjunto de datos (**xx** representa el número de grupo):
    a. disp_st**xx**ns1.txt.bz2: datos **disponibles** (para entrenar, HPO, evaluar y construcción del modelo final). Los datos disponibles contienen los 75 atributos y la variable de respuesta ("salida"). Contiene 12 años de datos, una instancia por día (años de 365 días)**.**
    b. comp_st**xx**ns1.txt.bz2: datos de la **competición**, sobre los que usar el modelo final para hacer predicciones, que podríamos enviar a una competición. Se trata de 2 años de datos (una instancia por día) con las 75 variables de entrada, pero **sin la variable de respuesta** (dado que cada grupo usará su modelo final para hacer predicciones).

DESARROLLO DE LA PRÁCTICA

1) (**0.5 puntos**) Preparar un repositorio privado en GitHub para poder hacer los *commits* semanales de lo realizado en la práctica cada semana. Haciendo al menos un *commit* cada semana se obtienen 0.5 puntos. Se recomienda que el nombre del repositorio sea vuestro número de grupo de prácticas seguido con el literal "Practica1". Por ejemplo, si sois el grupo 13 de prácticas, el repositorio se llamará "Grupo13-Practica1". Enviar el enlace del repositorio al profesor de prácticas por e-mail.
2) Leer los conjuntos de datos. Cada grupo usará una planta solar distinta (punto rojo), sustituyendo **xx** por el número de grupo:

```
disp_df = pd.read_csv("disp_stxxns1.txt.bz2",
                      compression="bz2",
                      index_col=0)
comp_df = pd.read_csv("comp_stxxns1.txt.bz2",
                      compression="bz2",
                      index_col=0)
```

3) (**0.25 puntos**) Hacer un Análisis Exploratorio de Datos (EDA).
4) Dividir los datos en «train» (los 10 primeros años) y «test» (los 2 últimos). Siguiendo la metodología planteada en la competición, no desordenaremos los datos antes de partir en entrenamiento y test, sino que respetaremos el orden temporal. Importante: los datos de «test» se reservan para la evaluación final en el apartado 8 de la práctica, no se puede utilizar para tomar decisiones durante el resto de puntos de la práctica (es decir, desde los apartados 4 a 7, habrá que evaluar los distintos métodos sin usar dicho conjunto de test).
5) (**1.00 puntos**) Métodos básicos: aquí se considerarán los siguientes métodos básicos: KNN, árboles de regresión, regresión lineal. Las métricas de evaluación son RMSE y MAE. Aparte de evaluar las métricas, también se medirá el tiempo que tarda su entrenamiento.
    a. Se evaluarán dichos modelos con sus hiperparámetros por omisión.
    b. Después, se ajustarán los hiperparámetros más importantes de cada método y se obtendrá su evaluación.

   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*/*baseline*? ¿El ajuste de hiperparámetros mejora con respecto a los valores por omisión? ¿Hay algún equilibrio entre tiempo de ejecución y mejora de resultados? Etc.

6) (**0.75 puntos**) ¿Es posible reducir la dimensionalidad del problema? (aquí no tiene por qué utilizarse una técnica estándar, sino algo que se os ocurra para que en los datos haya menos atributos sin empeorar resultados).
7) (**0.75 puntos**) Métodos avanzados**:** SVMs, Random Forests.
   a. Se evaluarán dichos modelos con sus hiperparámetros por omisión.
   b. Después, se ajustarán los hiperparámetros más importantes de cada método y se obtendrá su evaluación.
   c. Interpretar la importancia de los atributos según aquellas técnicas que lo permitan.
   d. Conclusiones hasta el momento.
8) (**0.25 puntos**) Seleccionar el mejor método, evaluarlo, construir modelo final, hacer predicciones para la competición.
   a. Seleccionar el mejor método de los evaluados en los puntos anteriores.
   b. Usar la partición de test para evaluar ese mejor método. 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 (*comp*). Guardar estas predicciones en un fichero (llamado «predicciones.csv»).

QUÉ ENTREGAR

- Código con dos notebooks:
  - 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.
  - Otro que cargue el modelo final y lo use para hacer predicciones en los datos de la competición.
- El archivo conteniendo el modelo final (llamado «modelo_final.pkl») y el archivo conteniendo las predicciones (predicciones.csv).
- El código y los archivos (modelo y predicciones) se entregarán finalmente en Aula Global en un fichero .zip.
- 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
SIGNIFICADO DE ATRIBUTOS DE ENTRADA

Variable	Description	Units
apcp_sfc	3-Hour accumulated precipitation at the surface	kg m-2
dlwrf_sfc	Downward long-wave radiative flux average at the surface	W m-2
dswrf_sfc	Downward short-wave radiative flux average at the surface	W m-2
pres_msl	Air pressure at mean sea level	Pa
pwat_eatm	Precipitable Water over the entire depth of the atmosphere	kg m-2
spfh_2m	Specific Humidity at 2 m above ground	kg kg-1
tcdc_eatm	Total cloud cover over the entire depth of the atmosphere	%
tcolc_eatm	Total column-integrated condensate over the entire atmos.	kg m-2
tmax_2m	Maximum Temperature over the past 3 hours at 2 m above the ground	K
tmin_2m	Mininmum Temperature over the past 3 hours at 2 m above the ground	K
tmp_2m	Current temperature at 2 m above the ground	K
tmp_sfc	Temperature of the surface	K
ulwrf_sfc	Upward long-wave radiation at the surface	W m-2
ulwrf_tatm	Upward long-wave radiation at the top of the atmosphere	W m-2
uswrf_sfc	Upward short-wave radiation at the surface	W m-2