

APRENDIZAJE AUTOMÁTICO

MEMORIA PRÁCTICA 1

SAMUEL FERNÁNDEZ FERNÁNDEZ (100432070) Y CARLOS GARCÍA GARCÍA
(100432251)

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).
 - a. Tras realizar el EDA, nos encontramos con 4748 instancias y 23 o 24 columnas, dependiendo de si se trata del dataset `wind_comp` o `wind_ava`, ya que uno cuenta con la columna `energy` y el otro no. Originalmente teníamos 552 columnas pero hemos eliminado aquellas que no eran para la cuadrícula 13.
 - b. De las variables de los 2 datasets, son todas numéricas excepto `datetime` que es categórica (es una fecha). La diferencia entre `wind_ava` y `wind_comp` es que `wind_comp` contiene una variable menos, `energy`, ya que es el resultado del modelo de regresión y este dataset será utilizado para testear el modelo que creemos.
 - c. No hemos encontrado ningún valor faltante en ninguna de las columnas de ninguno de los 2 datasets.
 - d. Tampoco hemos encontrado ninguna columna constante que eliminar en ninguno de los 2 datasets.
 - e. Se trata de un problema de regresión ya que lo que queremos hacer es predecir qué cantidad de energía se producirá en función de una serie de variables, es decir, un resultado numérico. Si el objetivo de la práctica fuera determinar si a raíz de la cantidad de energía predicha fuese catalogada como energía alta o baja, como tratamos más adelante, sería un problema de clasificación.
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.
 - a. La evaluación outer la hemos llevado a cabo de diferentes maneras. Por un lado, hemos evaluado el rendimiento de los modelos finales mediante la precisión (accuracy) en los modelos iniciales y el `balanced_accuracy` en el

modelo de clasificación, con el que obtenemos una matriz de confusión normalizada. Otro método de evaluación del rendimiento de los modelos ha sido el RMSE, ya que consideramos que es una métrica más precisa de error frente a la misma MSE y otros tipos de error. Esta métrica ha sido bastante utilizada a lo largo de toda la práctica. Del mismo modo hemos utilizado el R^2 junto al RMSE en algunas ocasiones como otro evaluador del rendimiento del modelo.

- b. La evaluación inner la hemos llevado a cabo mediante la búsqueda y el ajuste de hiperparámetros para los distintos modelos. Esta búsqueda ha sido por rangos de valores (en los casos numéricos) o listas (en los casos de opciones textuales). La hemos ido ajustando para algunos modelos como el SVM ya que veíamos que los mejores hiperparámetros nos daban en el límite y nos hacían ampliar los rangos de valores. Hemos utilizado también un cross-validation de 5 puesto que creemos que es un valor adecuado.

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.

- a. Usando KNN, el método de escalado más apropiado para este problema es el *StandardScaler* ya que tiene menos error (RMSE) en comparación con otros escaladores como *MinMaxScaler* o *RobustScaler*. Los datos de error son los siguientes: *StandardScaler*: RMSE = 408.7986938049903; *MinMaxScaler*: RMSE = 415.5635070483056; *RobustScaler*: RMSE = 414.31840748065605.

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.** En este apartado 4a), hemos seleccionado 5 modelos (KNN, árboles de decisión, Regresión lineal, regresión Lasso y SVM) y los hemos evaluado con sus hiperparámetros por defecto (la regresión lineal simple no tiene hiperparámetros). El KNN ha sido el modelo que menos RMSE y más R^2 tiene (408.798694 y 0.613193), además de haber sido el segundo más rápido en tiempo de entrenamiento (0.049020 segundos). El peor modelo ha sido el SVM en todo.
- b. **Después, se ajustarán los hiperparámetros más importantes de cada método y se obtendrá su evaluación. Medir tiempos del entrenamiento, ahora con HPO.** Una vez ajustados los hiperparámetros para los 4 modelos que tienen, hemos notado un cambio pues ha salido como mejor modelo el SVM con los siguientes hiperparámetros (`{'C': 1000, 'gamma': 'auto', 'kernel': 'rbf'}`). El SVM ha sido el modelo que menos RMSE y más R^2 tiene, aunque ha sido el más lento en tiempos de entrenamiento (367.486189 y 0.687423, 2.490927 segundos). El segundo mejor modelo ha sido el KNN.
- 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. Por lo que hemos visto, el KNN es un buen modelo porque ha sido el mejor en rendimiento con los parámetros por defecto y el segundo después de ajustar los hiperparámetros. Además, es muy rápido. Por otro lado, el SVM es un modelo que es bueno si encuentras los hiperparámetros adecuados, si no, es malo; aunque es notablemente más lento que los otros.

Si comparamos los resultados que nos han dado estos modelos con los dummy iniciales, hay varios puntos a comentar. El dummy tiene un RMSE de 657.3001647415589, ligeramente peor que el SVM por defecto (641.516761). El árbol de decisión básico genera un RMSE de 508.9337404533689, que no está nada mal para ser un modelo tan simple. Se puede apreciar que el dummy no es el mejor modelo que existirá pero el árbol de decisión puede llegar a usarse si tienes problemas al tratar con problemas sencillos y fáciles de comprender. Las regresiones lineales (normal y Lasso) son bastante malas, o al menos para este problema en concreto.

En conclusión, el KNN es un buen modelo en todo (tiempo, sencillez, parámetros por defecto e hiperparámetros) que se puede utilizar sin problemas en un problema de regresión. Pero si quieres ser más preciso, debes utilizar encontrar los hiperparámetros adecuados para el SVM, aunque tarde más tiempo.

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.** La mejor alternativa ha sido la comentada anteriormente (SVM con los siguientes hiperparámetros ({'C': 1000, 'gamma': 'auto', 'kernel': 'rbf'})).
 - b. **Estimar el rendimiento / desempeño futuro del modelo (evaluación outer).** Esta es una estimación de cómo se desempeñaría el modelo en la competición. En este apartado hemos entrenado el modelo SVM con los datos de entrenamiento del dataset wind_ava y hemos utilizado los datos de competición (el dataset wind_comp) para predecir su desempeño real.
 - 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.**

- a. Para el apartado a, hemos utilizado el modelo de regresión que mejores resultados nos ha dado (el KNN). Tras dividir el dataset `wind_ava` entre valores altos y bajos de energía en función del tercer cuantil, lo hemos aplicado para los valores altos y los valores bajos respectivamente. El resultado ha sido un error bajo para valores bajos de energía (RMSE para baja energía: 254.04428405257823) y un error alto para valores altos de energía (RMSE para alta energía: 504.51120912501847). Cogiendo el RMSE inicial del SVM sin dividir el dataset (367.486189) podemos sacar como conclusiones que hace un muy buen trabajo con los valores bajos, uno malo con los altos y uno aceptable al juntar ambos. Al tener el triple de valores bajos que altos, permite generar un modelo más adaptado a los valores bajos y además “maquilla” los valores altos de energía cuando se ejecuta el dataset junto.
- b. Para el apartado b, se nos pide adaptar el problema de regresión en uno de clasificación con 2 clases: clase baja de energía y clase alta de energía a partir de lo calculado en el apartado 6a). Inicialmente habíamos seleccionado como modelo un regresor logarítmico que nos daba entorno a un 0'82 de precisión y un valor AUC de 0'79, pero decidimos cambiar a un gradient boosting que nos devuelve un AUC de 0'93. Finalmente cambiamos la métrica del score de `ror_auc` a `balanced_accuracy` para obtener mejores resultados en términos globales. En adición a esto último, como teníamos que balancear el peso de las clases, hemos añadido un gráfico para representar la importancia de los atributos en el modelo XGBoost, siendo la característica más importante el *Instantaneous Eastward Turbulent Surface Stress* con un 14% de peso.

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.

ChatGPT ha sido utilizado como una herramienta para asistirnos y consultar en diversas etapas del proceso de modelado, análisis de datos y evaluación las diferentes dudas que nos surgían.

Algunos ejemplos de prompts y respuestas han sido:

Para la carga de datos

Prompt: "Necesito un script de Python para cargar un dataset de producción de energía eólica y seleccionar solo las columnas relevantes para mi análisis. El archivo se llama 'wind_ava.csv.gz' y está comprimido en formato gzip. Quiero conservar todas las columnas que terminan en '.13' porque corresponden a la localización de Sotavento, además de la columna 'energy' que es mi variable objetivo. ¿Cómo hago esto?"

Respuesta de ChatGPT:

import pandas as pd # Cargar los datos

```
wind_ava = pd.read_csv('wind_ava.csv.gz', compression="gzip") # Definir las columnas de interés: todas las que terminan en '.13' y la columna 'energy'
```

```
columns_of_interest = [col for col in wind_ava.columns if col.endswith('.13') or col == 'energy'] # Filtrar el DataFrame para conservar solo las columnas de interés
```

```
wind_ava_filtered = wind_ava[columns_of_interest] print(wind_ava_filtered.head())
```

Este script usa pandas para cargar tu archivo comprimido **wind_ava.csv.gz**, identifica las columnas de interés basándose en tu criterio específico (aquellas que terminan en '.13' y la columna 'energy'), y filtra el DataFrame para incluir solo esas columnas. El resultado es un DataFrame **wind_ava_filtered** que contiene solo la información relevante para tu análisis posterior.

Para la evaluación del Modelo

Prompt: "¿Cómo evalúo mi modelo de clasificación para un conjunto de datos desbalanceado?"

Respuesta de ChatGPT: "Para conjuntos de datos desbalanceados, la métrica AUC-ROC es una opción popular. También considera utilizar la precisión, recall, y F1-score para tener una visión más completa del rendimiento del modelo en cada clase."

Contexto: Aunque la métrica AUC-ROC es útil, la recomendación inicial omitió la importancia de la **balanced_accuracy**. Tras una revisión y con la sugerencia de ChatGPT en un prompt de seguimiento, incorporamos **balanced_accuracy** como métrica principal para evaluar el modelo.

Para la selección del Modelo

Prompt: "¿Qué modelo de machine learning me recomiendas para predecir la producción de energía eólica a partir de datos meteorológicos?"

Respuesta de ChatGPT: "XGBoost puede ser una buena opción debido a su eficiencia, efectividad y capacidad para manejar múltiples tipos de características. Además, XGBoost maneja bien los datos faltantes y es robusto frente a datos desbalanceados, aunque te recomiendo explorar técnicas de balanceo de clases para mejorar el rendimiento."

Contexto: La recomendación de utilizar XGBoost fue valiosa y se implementó en la práctica. Sin embargo, la sugerencia de que XGBoost maneja bien por sí solo los datos desbalanceados llevaron a una exploración adicional, concluyendo que técnicas como el ajuste del parámetro **scale_pos_weight** y el uso de SMOTE eran necesarias para manejar efectivamente el desbalance de clases.

Matriz de Confusión y Curva ROC

Prompt Original: "He usado la matriz de confusión para evaluar mi modelo de clasificación. ¿Qué otra visualización puedo utilizar?"

Respuesta de ChatGPT: "Además de la matriz de confusión, la curva ROC es una herramienta poderosa para evaluar la capacidad discriminativa de tu modelo en diferentes umbrales de clasificación."

Ajuste Necesario: Aunque la curva ROC fue una recomendación válida, para el contexto específico de un problema desbalanceado, se debatió posteriormente la pertinencia de centrarse en la **balanced_accuracy** y la importancia de las características para una evaluación más holística del modelo.