uc3m
2024-25. APRENDIZAJE AUTOMÁTICO

PRIMERA PRÁCTICA: PREDICCIÓN DEL ABANDONO (BURNOUT) DE EMPLEADOS (3.5 PUNTOS)

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN

El propósito de esta primera práctica es practicar con diferentes métodos de aprendizaje automático. Además, se trata de tratar todo el proceso: determinar el mejor método para un conjunto de datos (**selección de modelo**, incluido el preproceso y 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**).

La práctica se centra en la predicción del abandono de empleados: una empresa está preocupada por el abandono/agotamiento de los empleados y le gustaría crear un modelo que prediga si un empleado va a renunciar en función de un conjunto de datos recopilado por el departamento de recursos humanos.

CONSIDERACIONES GENERALES

1. Los resultados **deben ser reproducibles**. Por lo tanto, hay que fijar la semilla (el NIA de uno de los miembros del grupo) en los lugares apropiados.

2. Se valorará presentar y analizar los resultados de **forma clara**, utilizando principalmente tablas para presentar múltiples resultados y comparar alternativas.

3. 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.

4. **El preprocesamiento** debe llevarse a cabo mediante **pipelines,** siempre que sea posible, y utilizando los pasos de preprocesamiento adecuados para cada uno de los métodos elegidos.

5. Cada grupo (de dos personas máximo) utilizará **dos ficheros**: "attrition_availabledata_xx.csv" y "attrition_competition_xx.csv". xx = a + b, donde "ab" son los dos últimos dígitos del NIA de uno de los miembros del grupo. "available_data" contiene los datos para realizar la mayoría de las tareas de la práctica (entrenamiento, optimización de hiperparámetros, estimación de rendimiento futuro, etc.). "competition_data" contiene datos que simulan una competición y, por lo tanto, no contiene la variable de respuesta ("Attrition"). El modelo final se utilizará para hacer predicciones para el "competition_data".

PASOS A SEGUIR

1. GITHUB (0,5 PUNTOS)

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**. Además, también habrá que entregar el cuaderno (notebook) final a través de Aula Global.

2. EDA SIMPLIFICADO (0,3 PUNTOS)

Realice un **EDA simplificado**, principalmente para determinar cúantas variables e instancias hay, qué variables son categóricas/ordinales/numéricas, si hay variables categóricas con alta cardinalidad, qué variables tienen valores faltantes y cuántos, si hay columnas constantes o columnas de ID, y si se trata de un problema de regresión o clasificación. Si es esto último, ¿está desbalanceado?
Este EDA se utilizará como guía cuando haya que llevar a cabo el preproceso de los datos.

3. DECIDIR CÓMO SE VA A REALIZAR LA EVALUACIÓN (0,2 PUNTOS)

1. La estimación del rendimiento futuro (o evaluación *outer*) se realizará con **Holdout** (*train (2/3) / test (1/3)*). Las métricas principales será **balanced accuracy**, aunque se debe informar también del **TPR** / **TNR** (*accuracy* de la clase positiva y negativa, respectivamente) y **accuracy**. También se pueden utilizar **matrices de confusión** para informar de los resultados.
2. Dividir los datos en *train* y *test*. Importante: la mayor parte de la práctica se llevará a cabo utilizando sólo *train*. La partición de *test* sólo se usará una vez que se haya decidido cúal es la mejor manera de obtener el modelo, sólo entonces se usará *test* para calcular la estimación de rendimiento/desempeño futuro.
3. Decide cómo se va a llevar a cabo la evaluación interna (*inner*). *Inner* se utiliza cuando se realiza la optimización de hiperparámetros (HPO), pero también se utiliza para evaluar y comparar diferentes alternativas. Por lo tanto, la mayor parte de la práctica utilizará la evaluación *inner*, excepto al final, donde se utilizará el conjunto de *test* (*outer*) para evaluar el modelo final.

4. MÉTODOS BÁSICOS: KNN Y TREES (1 PUNTO)

1. Decidir, usando KNN los métodos de escalado y de imputación más apropiados para este problema y usarlos de aquí en adelante cuando sea necesario. Se considerarán tres métodos de escalado (minmax, standard, robust) y dos de imputación (media y mediana).
2. A continuación, se considerarán estos métodos: KNN y árboles:
    a. Se evaluarán 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. Se explicará mediante plots el efecto de los distintos valores de los hiper-parámetros en el resultado final (por ejemplo, en este problema, ¿los árboles necesitan de valores altos o bajos de *max_depth*? ¿qué ocurre con el número de vecinos en KNN?, etc.)
3. Obtener algunas conclusiones para esta sección, entre otras: ¿cuál es el mejor método? ¿A qué coste computacional? ¿Los resultados son mejores que los modelos triviales/naive/dummy? ¿El ajuste de hiperparámetros mejora con respecto a los valores por omisión? ¿Si hay mejora, es el coste computacional elevado?

5. AVANZADOS: MODELOS LINEALES Y SVMS (0.8 PUNTOS)

En esta sección, se considerarán dos tipos de métodos: lineales (sin y con regularización L1) y SVMs:

1. Se evaluarán con sus hiperparámetros por omisión (medir tiempos).
2. Después, se ajustarán los hiperparámetros más importantes de cada método y se obtendrá su evaluación (medir también tiempos).
3. ¿Es posible extraer de alguna técnica qué atributos son más relevantes? ¿Cuáles son?.

4. RESULTADOS Y MODELO FINAL (0.3 PUNTOS)

1. Seleccionar la mejor alternativa de las evaluadas en los puntos anteriores (usando la evaluación *inner*).
2. Estimar el rendimiento / desempeño futuro del modelo (evaluación *outer*). Esta es una estimación de cómo se desempeñaría el modelo en la competición.
3. Entrenar el modelo final. Guardarlo en un fichero (llamado «modelo_final.pkl»).
4. Utilizar el modelo final para obtener predicciones para el conjunto de datos de la competición. Guardar estas predicciones en un fichero (llamado «predicciones.csv»).

5. TAREA DE ELECCIÓN ABIERTA (0.4 PUNTOS)

Decidir alguna tarea adicional, ya sea porque podría mejorar los resultados o porque te parece especialmente interesante. Justificar la elección**.**

¿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.
  - Recordar escribir los nombres de los miembros del equipo al principio de los notebooks.
- 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 en Aula Global en un .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).
- Si se decide usar cualquier chatbot de IA, explicar brevemente en las celdas apropiadas del notebook para qué se usó y cómo (breve resumen).

Column Name	Column Description	Data Ty
hrs	The number of hours worked by the employee	float64
absences	The number of absences taken by the employee	float64
JobInvolvement	The level of involvement the employee has in their job	float64
PerformanceRating	The employee's performance rating	float64
EnvironmentSatisfaction	The level of satisfaction the employee has with their work environment	float64
JobSatisfaction	The level of satisfaction the employee has with their job	float64
WorkLifeBalance	The balance between work and personal life for the employee	float64
Age	The age of the employee	float64
Attrition	Whether the employee has left the company or not	object
BusinessTravel	The frequency of the employee's business travel	object
Department	The department the employee works in	object
DistanceFromHome	The distance from the employee's home to their workplace	float64
Education	The highest level of education attained by the employee	int64
EducationField	The field of study the employee specialized in	object
EmployeeCount	The number of employees in the company	float64
EmployeeID	A unique identifier for each employee	int64
Gender	The gender of the employee	object
JobLevel	The employee's job level in the company hierarchy	float64
JobRole	The specific role the employee has in their department	object
MaritalStatus	The employee's marital status	object
MonthlyIncome	The employee's monthly income	float64
NumCompaniesWorked	The number of companies the employee has worked for before joining the current company	float64
Over18	Whether the employee is over 18 years old (presumably all employees are)	object
PercentSalaryHike	The percentage of salary increase the employee received in their last salary hike	float64
StandardHours	The standard number of working hours in the company	float64
StockOptionLevel	The level of stock option the employee has	float64
TotalWorkingYears	The total number of years the employee has worked	float64
TrainingTimesLastYear	The number of times the employee received training in the last year	float64
YearsAtCompany	The number of years the employee has been with the company	float64