Desafío - Detección de cardiopatía

En el campo de la salud, los modelos de Machine Learning pueden ayudar a la detección temprana de posibles enfermedades pudiendo salvar vidas.

En este desafío te enfrentarás a una base de datos real de pacientes con y sin cardiopatía, con el objetivo de poner en práctica los conceptos aprendidos durante las clases de ensambles secuenciales, para la detección de esta enfermedad.

Lee todo el documento antes de comenzar el desarrollo individual, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos. Asegúrate de seguir las instrucciones específicas en cada ejercicio y de completar los requerimientos adicionales, si los hubiera.

Tiempo asociado: 4 horas cronológicas

Descripción

En este proyecto deberás construir tres modelos de Machine Learning de ensambles secuenciales para determinar si una persona padece o no una cardiopatía, a partir de los datos.

La base de datos se compone de tres archivos: **processed_cleveland.csv, processed_hungarian.csv** y **processed_switzerland.csv**, cada uno representa muestras adquiridas de diferentes clínicas. Los atributos de la base de datos son:

- **age:** Edad del sujeto en años
- **sex:** Sexo (1 Hombre, 0 Mujer)
- **cp:** Tipo de dolor torácico
    - **a.** Valor 1: angina típica
    - **b.** Valor 2: angina atípica
    - **c.** Valor 3: dolor no anginoso
    - **d.** Valor 4: asintomático
- **trestbps:** Presión arterial en reposo (en mm Hg al ingreso en el hospital)
- **chol:** Colesterol sérico en mg/dl
- **fbs:** (glucemia en ayunas > 120 mg/dl) (1 = verdadero; 0 = falso)
- **restecg:** Medición electrocardiográfica en reposo (0 = normal, 1 = presenta anomalía de la onda ST-T, 2 = muestra hipertrofia ventricular izquierda probable o definida según los criterios de Estes)
- **thalach:** Frecuencia cardiaca máxima alcanzada
- **exang:** Angina inducida por ejercicio (1 = sí; 0 = no)
- **oldpeak:** Depresión del ST inducida por el ejercicio en relación con el reposo

- **slope:** La pendiente del segmento ST de ejercicio máximo
  - **a.** Valor 1: pendiente ascendente
  - **b.** Valor 2: plano
  - **c.** Valor 3: pendiente descendente
- **ca:** Número de vasos mayores (0-3) coloreados por la flouroscopia
- **thal:** 3 = normal; 6 = defecto fijo; 7 = defecto reversible
- **num:** Diagnóstico de cardiopatía (estado de la enfermedad angiográfica)
  - **a.** Valor 0: < 50% estrechamiento del diámetro
  - **b.** Valor 1: > 50% de estrechamiento del diámetro

El atributo **num** será la variable objetivo (0 sin enfermedad, 1 con enfermedad)

Para construir los modelos y resolver el problema, deberás aplicar los siguientes pasos:

1. Carga los datos de los tres archivos unidos en un DataFrame, y prepáralos considerando las siguientes etapas:
   a. asigna a la variable objetivo **num** un cero cuando su valor es cero, y un 1 en otro caso. Muestra la cantidad de valores ausente por atributo. Procésalos considerando los siguientes criterios:
      i. si el atributo presenta un porcentaje de valores ausentes mayor a 25%, entonces descartamos ese atributo.
      ii. para los atributos con valores ausentes menores al 25% se deben rellenar con el valor promedio del atributo de acuerdo a la clase, es decir, los valores ausentes para los cuales **num=0** se deben rellenar con el promedio de los valores presentes para los cuales **num=0**.
      iii. Para las variables categóricas, los valores a asignar (promedio) deben ser aproximados al valor entero más cercano.
   b. Construye variables dummies para las variables discretas con tres o más categorías.
   c. Realiza un análisis descriptivo por variable visualizando histogramas y gráficos de barra según corresponda. Construye una matriz de correlaciones con un heatmap, para las variables continuas. Plantea tus observaciones y conclusiones

2. Divide la muestra en entrenamiento y test (33%), y con ello:
   a. entrena un modelo de ensamble secuencial **AdaBoost**
   b. entrena un modelo de ensamble secuencial **Gradient Boosting**
   c. entrena un modelo de ensamble secuencial **XGBoosting**

   Para cada uno, utiliza hiper parámetros por defecto y muestra su desempeño usando las métricas adecuadas. Comenta.

3. Realiza una búsqueda del hiper parámetro para el modelo que resulte tener mejor desempeño (utiliza f1-score para determinarlo). Los valores a buscar para la grilla son:
    a. n_estimators: 20 a 200 con 15 valores
    b. learning_rate: 0.004 a 1.0 con 40 valores
    c. sub_sample: 0.1 a 1.0 con 18 valores

    Muestra los mejores hiper parámetros encontrados y entrena un modelo con estos, presentando sus métricas.

4. Elabora un gráfico con las curvas ROC para cada modelo entrenado, y a partir de ello recomienda alguno de ellos. Justifica la decisión

5. Muestra las variables por nivel de importancia para el mejor modelo, luego de la búsqueda de hiper parámetro. Comenta.

Requerimientos

1. Carga, analiza y prepara los datos **(1 punto)**
2. Plantea, entrena, muestra y compara el desempeño de modelos secuenciales AdaBoost, Gradient Boosting y XGBoosting, considerando curvas ROC y seleccionando las variables más importantes para el mejor modelo. **(5 puntos)**
3. Realiza búsquedas de hiper parámetros utilizando grilla, y entrena modelos utilizándolos. **(4 puntos)**

 ¡Mucho éxito!

Consideraciones y recomendaciones

Debes entregar tu solución en un archivo Jupyter, con los comentarios y observaciones necesarias para comprender tu procedimiento.