Trabajo Práctico Integrador

Cantidad de integrantes máxima por grupo: 2
Escala de Calificaciones: 1 - 10, siendo 10 la máxima calificación
Formato de entrega: Jupyter Notebook o archivo .py son igualmente válidos.
Cada uno de los pasos que observamos será una o un conjunto de funciones que debemos desarrollar.

Parte 1: Arboles de Decision para Regresión

- A partir del archivo `Life Expectancy.csv` entrenar un modelo de árboles de decisión que permita predecir la esperanza de vida en un determinado país a partir las variables explicativas del dataset.
- Revisar en la documentación de la librería el valor por defecto que toma el hiperparámetro `max_depth` . Graficar la variación del score para el set de entrenamiento y para el de prueba en función de la variación de este parámetro (puede inicialmente tomar un valor 1 e incrementarse hasta `max_depth = 10`).
- En función del gráfico sugerir el `max_depth` óptimo para el modelo definitivo.
- Finalmente, mostrar gráficamente el árbol de decisión. Recordar que puede llegar a ser necesario aclarar por parámetro la dimension del gráfico para que sea observable.

Parte 2: Transformación del dataset

- Transformar la columna Life Expectancy de manera tal que considere desarrollado (valor 1) a aquellos países cuya esperanza de vida sea mayor o igual a 72 años. Caso contrario, en desarrollo (valor 0). Renombrar la columna de manera que refleje esta situación.

- Visualizar la información obtenida de manera que considere más conveniente y de forma que añada valor en la comprensión de la información obtenida. Realizar al menos 3 gráficos.

Parte 3: Modelos de Clasificación

Bootstrap Aggregation

- Entrenar un modelo tipo Bagging Classifier de manera tal que se empleen N muestras bootstrap con reposición, cada una del 50% del tamaño del dataset original. Configurar los hiperparámetros de manera tal que cumpla las condiciones especificadas.

- Graficar como varía el accuracy de cada modelo en función de la cantidad de muestras bootstrap con el que se lo entrenó. Realizarlo para valores entre 10 y 100 con saltos de 10 unidades.

- Presentar la matriz de confusión correspondiente para un umbral neutro para el modelo que emplea 100 muestras bootstrap.

- Graficar, para distintos umbrales (desde 0 a 1 con saltos de 0.1) la variación de VP y FP

AdaBoost y GradientBoosting

- Entrenar los dos modelos de boosting del enunciado con la siguiente configuración para ambos casos:
    - `learning_rate = 1`
    - `n_estimators = 10`

Recordar que en GradientBoosting tomaremos un `max_depth = 1`

- Indicar el score sobre el set de prueba para cada uno de los casos

- Realizar la matriz de confusión correspondiente para ambos modelos.

- Retornar para cada caso los clasificadores débiles que utilizó el algoritmo. Mostrar gráficamente un árbol a elección.

- Indicar para cada caso si para la configuración dada nos encontramos con un caso de overfitting. En caso afirmativo, indicar qué parámetros modificaría para evitar esto.