Árboles de decisión

Tres productores de vino, de la misma región de Italia, sometieron su producto a un análisis químico. Se desea clasificar el productor correspondiente a partir del análisis mencionado.

(**a**) *Exploración de datos:* Cargar la base de datos utilizando `load_wine` (sklearn).

(**b**) *Árbol de decisión:*

- Utilizando el comando `tree.DecisionTreeClassifier` (sklearn), entrenar un clasificador con entropía como función de impureza.
- Utilizando `plot_tree` (sklearn) graficar el diagrama de árbol. Indicar la cantidad de nodos y hojas.
- Encontrar los 3 *features* más relevantes según la *Feature Importance*.
- Clasificar el siguiente análisis químico:

$$13.0,\ 2.33,\ 2.37,\ 19.5,\ 99.7,\ 2.29,\ 2.03,\ 0.36,\ 1.59,\ 5.06,\ 0.96,\ 2.61,\ 747$$

(**c**) *Podado:* Repetir el inciso (**b**) para un árbol podado con una complejidad $\alpha = 0.05$. Calcular el *costo-complejidad* asociado.

(**d**) *Bosques aleatorios:* Implementar un bosque aleatorio de 30 árboles (sin poda) con entropía como función de impureza. Puede utilizar el comando `tree.DecisionTreeClassifier` (sklearn), pero la combinación de estos árboles debe ser implementación propia. El código debe estar estructurado de la siguiente manera:

```
class RandomForest:

    # Inicializar atributos y declarar hiperparámetros
    def __init__(self,...

    # Etapa de entrenamiento
    def fit(self,X,y):

    # Etapa de testeo hard
    def predict(self,X):
```

Clasificar el análisis químico del inciso (**b**).