7. Ensambles

Ejercicio 7.1. Dar una explicación general del algoritmo Bagging.

Ejercicio 7.2. Dar una explicación general del algoritmo Random Forest. Basar la explicación en el algoritmo original presentado en el artículo: Breiman, Leo. "Random Forests." Machine learning 45.1 (2001). Incluir:

- ¿Cuál es su principal diferencia con Bagging?
- ¿Cómo funciona la estimación de error "out-of-bag"? [1]
- ¿Cómo propone Breiman medir la importancia de features? [2] ¿Qué diferencia hay con la manera en que la importancia se mide en el paquete sckit-learn de python? [3]

Ejercicio 7.3. Sea un clasificador binario de tipo Bagging en donde los submodelos predicen decisiones duras (Positivo o Negativo), entrenado sobre un conjunto de datos de entrenamiento . Al evaluar 5 instancias el clasificador devuelve las siguientes probabilidades de pertenencia a la clase positiva: `[0.75, 0.50, 0.25, 0.75, 1.0]`. Determinar la cantidad de sub-modelos utilizados en el ensamble. Justificar.

Ejercicio 7.4. Verdadero o Falso (justificar)

(a) En Bagging, cada subconjunto tiene la misma cantidad de instancias que el dataset original.

(b) En RF, cada subconjunto tiene la misma cantidad de instancias que el dataset original.

(c) En RF, cada árbol es entrenado sólo con un subconjunto de los atributos.

(d) En Random Forest, tomar m = 1 significa que cada árbol tendría a lo sumo un nivel.

(e) En Random Forest, tomar m = 1 significa que cada árbol tendría a lo sumo un atributo en todo el árbol.

(f) En Random Forest, la importancia de atributos puede medirse como la suma (entre todos los árboles) de la ganancia obtenida en cada corte por cada atributo

(g) En Random Forest, la importancia de atributos puede medirse como la suma (entre todos los árboles) de la ganancia obtenida en cada corte por cada atributo dividido B.

(h) La varianza que se reduce utilizando Bagging debería ser mayor que la que se reduce utilizando Random Forest.

(i) En Bagging, se puede estimar el error de generalización sólo utilizando un train set (sin necesidad de utilizar CrossVal) y los resultados seguramente estén sub-estimando el error real.

(j) En Bagging, se puede estimar el error de generalización sólo utilizando un train set (sin necesidad de utilizar CrossVal) y los resultados seguramente estén sobre-estimando el error real.

Ejercicio 7.5. (Opcional) Explicar la idea conceptual del meta-algoritmo AdaBoost. Incluir:

(a) ¿Qué significa "weak-learners"?

(b) ¿Cómo se calculan los pesos para las instancias en cada iteración?

(c) ¿Cuál es el criterio para determinar la cantidad de modelos en el ensamble?

Ejercicio 7.6. (Opcional) Explicar la diferencia entre AdaBoost y GradientBoosting. Además, analizar cómo está implementado XGBoost para entender mejor el funcionamiento de GradientBoosting.

[1] ver Sección 3 del paper.
[2] ver Sección 10 del paper.
[3] ver la sección *1.11.2.5. Feature importance evaluation* en la documentación de scikit-learn sobre ensambles