

Ejercicios

1. Medir la performance del modelo de clasificación generado para especies de árboles, de distintas maneras.
2. Para un problema genérico de clasificación binaria, definir una función `matriz_confusion_binaria`, que tome dos listas `Y_test`, `Y_pred` y devuelvas los valores (en orden) de TP, TN, FP, FN.

```
def matriz_confusion_binaria(Y_test, Y_pred):  
    # Y_test e Y_pred deben ser listas de 0 y 1  
    # completar  
    return tp, tn, fp, fn
```

3. Para un problema genérico de clasificación binaria, definir funciones para cada una de las siguientes métricas: accuracy, precision, recall, F1. Las funciones deben tomar como parámetros los TP, TN, FP, FN.

```
def accuracy_score(tp, tn, fp, fn):  
    # completar  
    return acc  
  
def precision_score(tp, tn, fp, fn):  
    # completar  
    return prec
```

4. Construir, usando sklearn, un árbol de decisión para el problema Titanic, y analizar su performance de distintas maneras.
5. Para la próxima clase: considerar al menos dos árboles de decisión (seleccionando distintas elecciones hiperparámetros, por ejemplo variando el valor de la profundidad entre `max_depth = 3` y `max_depth = 5`).
6. Anotar y traer para la próxima clase las medidas obtenidas con dichos modelos, al haberlos entrenado con los datos de train y evaluando en el conjunto de train por un lado y en el conjunto de test por el otro. Anotar en cada caso las 4 medidas definidas: accuracy, precision, recall, F1. Por ejemplo con una tabla así:

| modelo | acc train | acc test | prec train | prec test | rec train | rec test | f1 train | f1 test |
|--------|-----------|----------|------------|-----------|-----------|----------|----------|---------|
| md = 3 | | | | | | | | |
| md = 5 | | | | | | | | |