RANDOM FOREST (BOSQUES ALEATORIOS)

QUÉ ES RANDOM FOREST?

- ES UN ALGORITMO DE ‘ENSEMBLE’
- MÉTODO DE CLASIFICACIÓN Y REGRESIÓN PARA APRENDIZAJE SUPERVISADO
- EXISTE UNA VARIANTE NO SUPERVISADA PARA TRANSFORMAR CARACTERÍSTICAS
- INTUITIVAMENTE, UN BOSQUE ALEATORIO SE PUEDE CONSIDERAR COMO UN CONJUNTO DE ÁRBOLES DE DECISIÓN.
- LA IDEA QUE HAY DETRÁS DE UN BOSQUE ALEATORIO ES PROMEDIAR ÁRBOLES DE DECISIÓN MÚLTIPLES (PROFUNDOS) QUE INDIVIDUALMENTE SUFREN UNA ELEVADA VARIANZA PARA CREAR UN MODELO MÁS ROBUSTO QUE TENGA UN MEJOR RENDIMIENTO DE GENERALIZACIÓN Y SEA MENOS SUSCEPTIBLE AL SOBREAJUSTE
- EL PRIMER ALGORITMO FUE DESARROLLADO EN 1995, EN 2006 UNA VERSIÓN MEJORADA FUE PATENTADA POR BREIMAN Y CUSTLER
- EXISTEN VARIANTES: RANDOM FOREST, ÁRBOLES POTENCIADOS POR DESCENSO DE GRADIENTE, EXTREMELY RANDOMIZED TREES (EXTRA TREES), KERNEL RANDOM FORESTS (KERF)...
- LIN Y JEON SEÑALARON EN 2002 UNA RELACIÓN ENTRE LOS BOSQUES ALEATORIOS Y EL ALGORITMO K-VECINOS CERCANOS (K-NN).RESULTA QUE AMBOS PUEDEN VERSE COMO LOS LLAMADOS “ESQUEMAS DE VECINDARIOS PONDERADOS” (WEIGHTED NEIGHBORHOODS SCHEMES).

QUÉ SON LOS MÉTODOS DE ENSEMBLE?

EL OBJETIVO DE LOS MÉTODOS DE CONJUNTO (ENSEMBLE) ES COMBINAR LAS PREDICCIONES DE VARIOS ESTIMADORES BASE CONSTRUIDOS CON UN ALGORITMO DE APRENDIZAJE DADO PARA MEJORAR LA GENERALIZACIÓN/ROBUSTEZ SOBRE UN SOLO ESTIMADOR.

SE SUELEN DISTINGUIR DOS FAMILIAS DE MÉTODOS DE CONJUNTO:

EN LOS MÉTODOS DE PROMEDIO, EL PRINCIPIO FUNDAMENTAL ES CONSTRUIR VARIOS ESTIMADORES DÉBILES DE FORMA INDEPENDIENTE Y LUEGO PROMEDIAR SUS PREDICCIONES.
EN GENERAL, EL ESTIMADOR COMBINADO SUELE SER MEJOR QUE CUALQUIERA DE LOS ESTIMADORES DE BASE ÚNICA PORQUE SE REDUCE SU VARIANZA.

EJEMPLOS: MÉTODOS DE BAGGING, FORESTS OF RANDOMIZED TREES

POR EL CONTRARIO, EN LOS MÉTODOS BOOSTING, LOS ESTIMADORES BASE SE CONSTRUYEN SECUENCIALMENTE Y SE INTENTA REDUCIR EL SESGO DEL ESTIMADOR COMBINADO.
LA MOTIVACIÓN ES COMBINAR VARIOS MODELOS DÉBILES PARA PRODUCIR UN CONJUNTO PODEROSO.

EJEMPLOS: ADABOOST, GRADIENT TREE BOOST

CÓMO FUNCIONA EL BOSQUE?

- EL ALGORITMO DEL BOSQUE ALEATORIO SE PUEDE RESUMIR EN CUATRO SENCILLOS PASOS:

- 1. DIBUJAR UNA MUESTRA BOOTSTRAP ALEATORIA DE TAMAÑO N (ELIGE AL AZAR 'N' MUESTRAS DEL CONJUNTO DE ENTRENAMIENTO CON REEMPLAZO).

- 2.CREAR UN ÁRBOL DE DECISIÓN A PARTIR DE LA MUESTRA BOOTSTRAP.

- PARA CADA NODO:

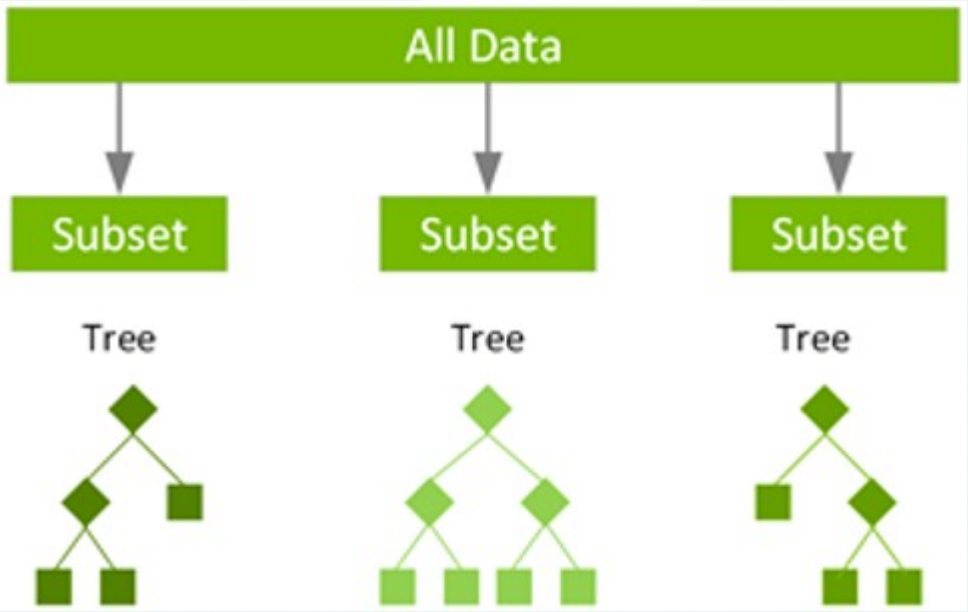
- A. SELECCIONAR AL AZAR CARACTERÍSTICAS D SIN REEMPLAZO.

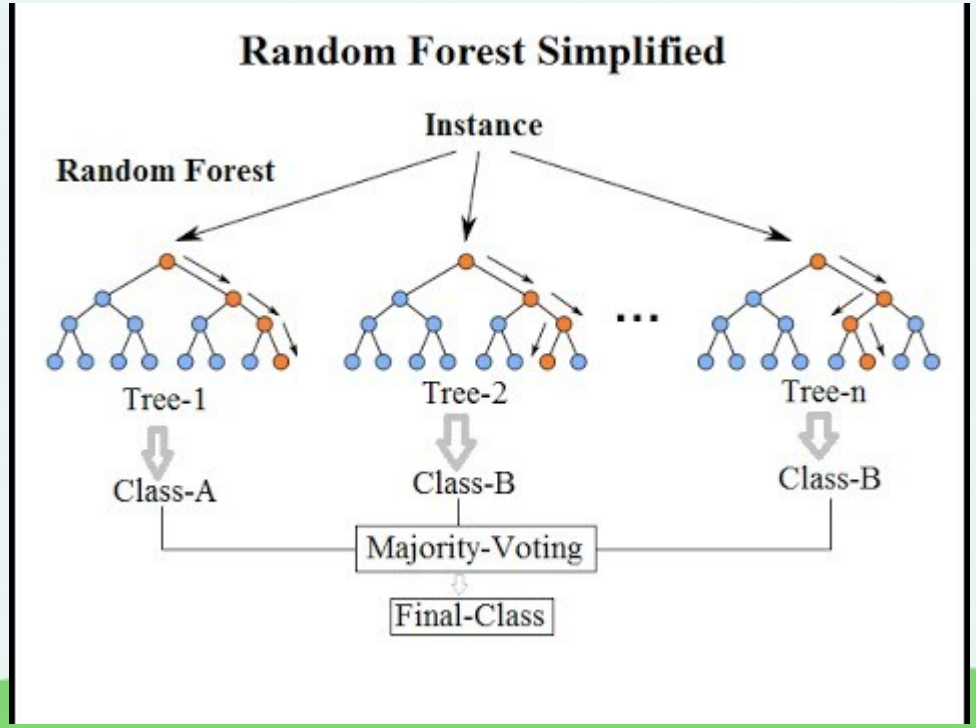
- B. DIVIDIR EL NODO UTILIZANDO LA CARACTERÍSTICA QUE PROPORCIONA LA MEJOR DIVISIÓN SEGÚN LA FUNCIÓN OBJETIVO; POR EJEMPLO, MAXIMIZANDO LA GANANCIA DE INFORMACIÓN.

- AL ENTRENAR LOS ÁRBOLES DE DECISIÓN INDIVIDUALES: EN LUGAR DE EVALUAR TODAS LAS CARACTERÍSTICAS PARA DETERMINAR LA MEJOR DIVISIÓN PARA CADA NODO, SOLO CONSIDERAR UN SUBCONJUNTO AL AZAR DE ELLAS.

- 3. REPETIR LOS PASOS 1 Y 2 'K' VECES.

- 4. AÑADIR LA PREDICCIÓN PARA CADA ÁRBOL PARA ASIGNAR LA ETIQUETA DE CLASE POR MAYORÍA DE VOTOS.



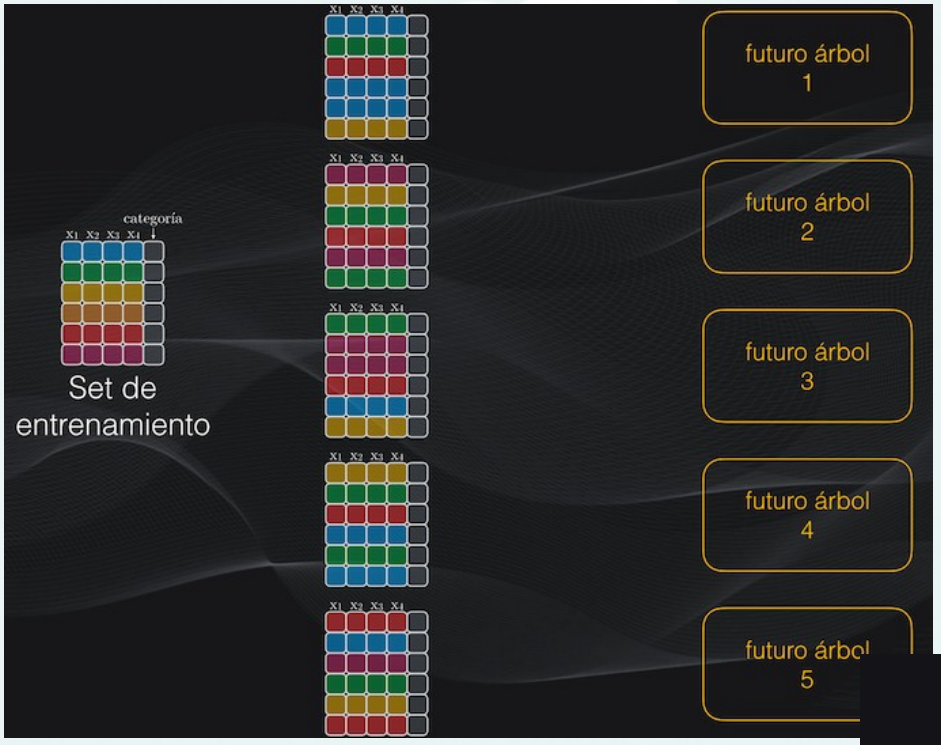


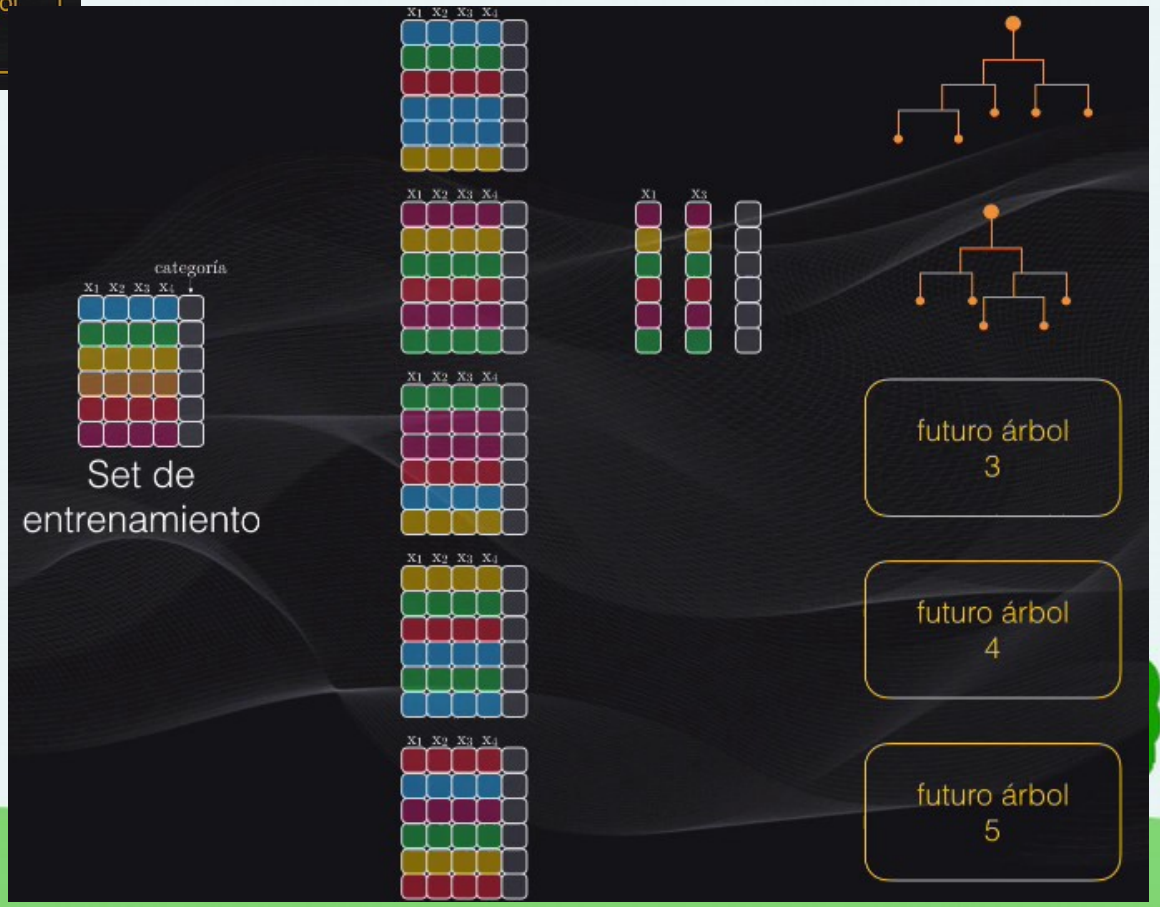
Bagging y out-of-bag samples

*2 mecanismos esenciales en el funcionamiento de los Bosques aleatorios: el bootstrapping durante el entrenamiento y la agregación de los resultados para realizar la predicción. La combinación de estos dos términos se conoce como bagging, que es precisamente el nombre genérico con que se conoce a los algoritmos como los Bosques Aleatorios y otros similares.

Si miramos en detalle el bootstrapping veremos que, en esta selección aleatoria de observaciones, al final no todas quedarán incluidas en los subsets de entrenamiento. En realidad, *aproximadamente una tercera parte de los ejemplos originales quedará por fuera* y no será usada para entrenar.

Así que, a diferencia de otros algoritmos de Machine Learning en donde al inicio se divide el set de datos en entrenamiento y validación, en el caso de los bosques aleatorios *los datos de validación serán precisamente aquellos que no fueron seleccionados durante el muestreo aleatorio*. Estos ejemplos de entrenamiento se conocen como "muestras por fuera de la bolsa", o out-of-bag-samples.


categoría
Set de entrenamiento
futuro árbol 1
futuro árbol 2
futuro árbol 3
futuro árbol 4
futuro árbol 5


categoría
Set de entrenamiento
futuro árbol 3
futuro árbol 4
futuro árbol 5

VENTAJAS

- Es adecuado tanto para problemas de regresión como de clasificación. La variable de salida en la regresión es una secuencia de números, como el precio de las casas en un barrio. La variable de salida en un problema de clasificación suele ser una respuesta única, como si es probable que una casa se venda por encima o por debajo del precio de venta.
- Maneja los valores faltantes y mantiene una alta precisión, incluso cuando faltan grandes cantidades de datos gracias al embolsado y al muestreo de reemplazo.
- El algoritmo hace que el sobreajuste del modelo sea casi imposible debido a la salida de "reglas de la mayoría".
- El modelo puede manejar conjuntos de datos muy grandes con miles de variables de entrada, lo que lo convierte en una buena herramienta para la reducción de la dimensionalidad.
- Sus algoritmos se pueden utilizar para identificar las características más importantes del conjunto de datos de entrenamiento.

DESVENTAJAS

- Los bosques aleatorios superan a los árboles de decisión, pero su precisión es inferior a la de los conjuntos de árboles potenciados por gradientes (como XGBoost)
- Con una gran cantidad de árboles, RF es más lento que XGBoost
- Se sacrifica la interpretabilidad intrínseca presente en los árboles de decisión, pues seguir los caminos de decenas o cientos de árboles es mucho más difícil. Algunas técnicas de compresión de modelos permiten transformar un bosque aleatorio en un árbol de decisión mínimo "renacido" que reproduce fielmente la misma función de decisión
- Si se establece que los atributos predictivos se correlacionan linealmente con la variable objetivo, es posible que el uso de bosques aleatorios no mejore la precisión del predictor base
- En problemas con múltiples variables categóricas, es posible que el bosque aleatorio no pueda aumentar la precisión del predictor base.

APLICACIONES

- CLASIFICACIÓN:
- Detección de fraude, Detección de correo no deseado, Análisis de sentimiento de texto, Predicción del riesgo del paciente, sepsis o cáncer
- REGRESIÓN:
- Predecir la cantidad de fraude, Predicción de ventas

- EXISTEN VARIAS BIBLIOTECAS QUE IMPLEMENTAN RANDOM FOREST EN PYTHON.
- ÉSTAS SON ALGUNAS
- H2O https://docs.h2o.ai/h2o/latest-stable/h2o-py/docs/tree.html
- https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html
- FAST https://www.fast.ai/
- random-forest-mc https://github.com/ysraell/random-forest-mc
- TENSORFLOW https://www.tensorflow.org/decision_forests?hl=es-419
- SPARK https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.mllib.tree.RandomForest.html
- SCIKIT LEARN https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees
- cuML (NVIDIA) https://developer.nvidia.com/blog/gradient-boosting-decision-trees-xgboost-cuda/

class sklearn.ensemble.**RandomForestRegressor**(n_estimators=100, *, criterion='squared_error', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=1.0, max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, ccp_alpha=0.0, max_samples=None)

class sklearn.ensemble.**RandomForestClassifier**(n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None)

EL REGRESOR RF SOPORTA, AL IGUAL QUE LOS ÁRBOLES DE DECISIÓN,

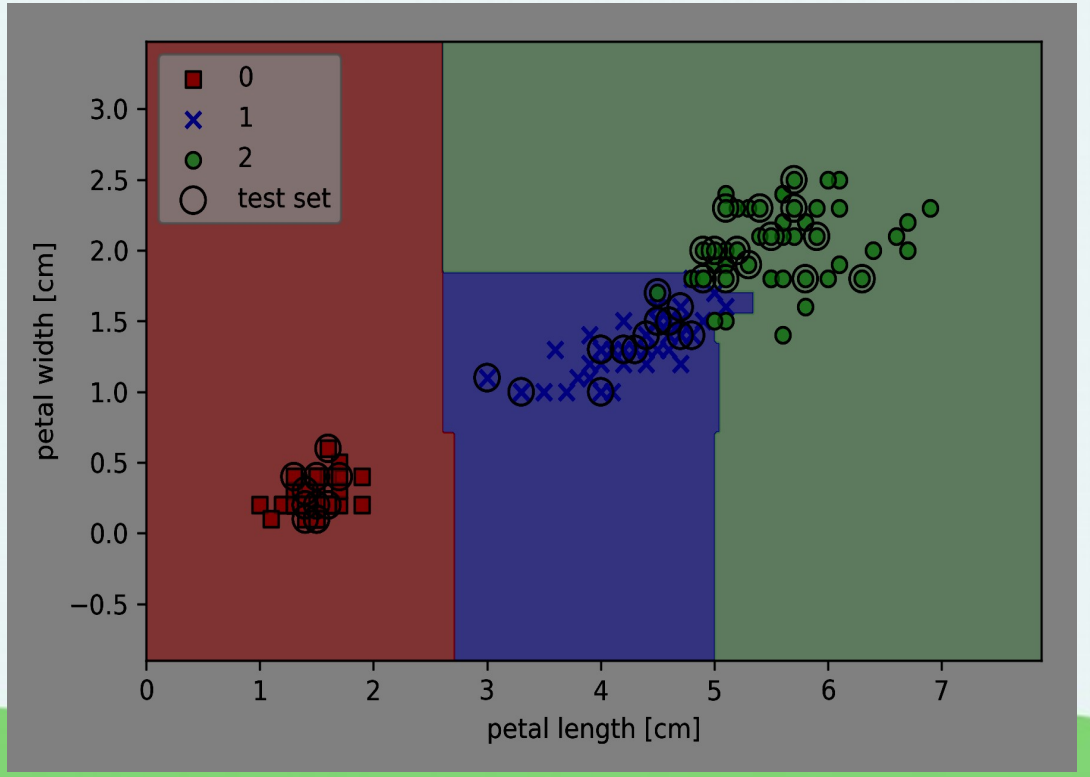
LA POSIBILIDAD DE REGRESIONES MULTIOUTPUT

class sklearn.multioutput.MultiOutputRegressor(estimator, *, n_jobs=None)

```
from sklearn import datasets
from sklearn.cross_validation import train_test_split
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap

def plot_decision_regions(X, y, classifier, test_idx=None, resolution=0.02):
   # setup marker generator and color map
   markers = ('s', 'x', 'o', '^', 'v')
   colors = ('red', 'blue', 'lightgreen', 'gray', 'cyan')
   cmap = ListedColormap(colors[:len(np.unique(y))])

   # plot the decision surface
   x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() + 1
   x2_min, x2_max = X[:, 1].min() - 1, X[:, 1].max() + 1
   xx1, xx2 = np.meshgrid(np.arange(x1_min, x1_max, resolution),
   np.arange(x2_min, x2_max, resolution))
   Z = classifier.predict(np.array([xx1.ravel(), xx2.ravel()]).T)
   Z = Z.reshape(xx1.shape)
   plt.contourf(xx1, xx2, Z, alpha=0.4, cmap=cmap)
   plt.xlim(xx1.min(), xx1.max())
   plt.ylim(xx2.min(), xx2.max())
```

```
iris = datasets.load_iris()
X = iris.data[:, [2, 3]]
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state

forest = RandomForestClassifier(criterion='entropy',
                                n_estimators=10, random_state=1, n_jobs=2)

forest.fit(X_train, y_train)

X_combined = np.vstack((X_train, X_test))
y_combined = np.hstack((y_train, y_test))

plot_decision_regions(X_combined, y_combined,
         classifier=forest, test_idx=range(105,150))

plt.xlabel('petal length [cm]')
plt.ylabel('petal width [cm]')
plt.legend(loc='upper left')
plt.show()
```

```
   # plot all samples
   X_test, y_test = X[test_idx, :], y[test_idx]
   for idx, cl in enumerate(np.unique(y)):
      plt.scatter(x=X[y == cl, 0], y=X[y == cl, 1],
                  alpha=0.8, c=cmap(idx),
                  marker=markers[idx], label=cl)
   # highlight test samples
   if test_idx:
      X_test, y_test = X[test_idx, :], y[test_idx]
      plt.scatter(X_test[:, 0], X_test[:, 1], c='',
                  alpha=1.0, linewidth=1, marker='o',
                  s=55, label='test set')
```



DUDAS, CONSULTAS?

MUCHAS GRACIAS POR ESCUCHAR!!!

AHORA VEREMOS UNOS EJEMPLOS...

FUENTES:

- “PYTHON MACHINE LEARNING” SEBASTIAN RASCHKA, VAHID MIRJALILI, MARCOMBO, 2DA EDICIÓN
- https://en.wikipedia.org/wiki/Random_forest
- SKLEARN

https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees