

## Bagging y Random Forest

Supongamos que hacemos una pregunta a miles de personas. En muchos casos encontraremos que si agregamos estos datos, la respuesta será mejor que la de una única persona. Esto se conoce como "Wisdom of the crowd". Obviamente podemos extender esto al área de Machine Learning: Si agregamos la respuesta de distintos predictores, habitualmente obtendremos una mejor predicción que al usar un único predictor. La técnica de agrupar predictores se conoce como "Ensemble Learning".

## Voting classifiers

Supongamos que tenemos un dataset y entrenamos 3 clasificadores (SVM, KNN y Regresión Logística) en base a ese dataset. Ahora al ver una instancia nueva hacemos lo siguiente:

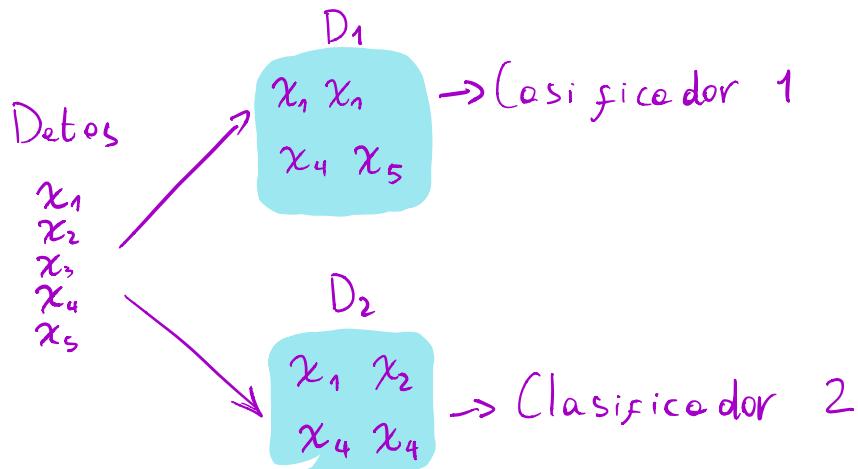


En donde vamos a agregar las respuestas sumando los votos para cada posible clase, y la predicción final es la clase con más votos. Sorprendentemente, un clasificador en base a votos tiende a tener mejor desempeño que el mejor clasificador individual.

## Bagging y Pasting

Supongamos que tenemos un dataset con 5 instancias  $\{x_1, \dots, x_5\}$  y queremos entrenar dos clasificadores, con 4 instancias cada uno.

La idea de hacer Bagging es "samplear" los 4 elementos del dataset original, pudiendo repetir observaciones en un mismo clasificador:



Ojo! el sampleo para armar D<sub>i</sub> es aleatorio.

Así, el clasificador 1 se entrena con los datos  $D_1$  y el clasificador 2 se entrena con los datos  $D_2$ . Notamos que en  $D_1$  se repite dos veces la observación  $x_1$ , mientras que en  $D_2$  se repitió dos veces  $x_4$ . Cuando no toleramos repeticiones para un dataset  $D_i$ , le llamamos **Pasting**.

### Out-of-bag evaluation

Cuando hacemos Bagging de un dataset de tamaño  $m$  cada clasificador ve o verá solo el 63% de los datos:

\_\_\_\_\_ . \_\_\_\_\_

Probabilidad de que en 1 intentos no esté presente la instancia  $x_i$

$$\hookrightarrow \left(1 - \frac{1}{m}\right)$$

Probabilidad de que en  $m$  intentos no esté presente la instancia  $x_i$

$$\hookrightarrow \left(1 - \frac{1}{m}\right)^m$$

$$\text{Y además } \lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e}$$

Entonces la probabilidad de que esté  $x_i$  es:

$$1 - \frac{1}{e} \approx 63.212\%$$

Por lo mismo, un clasificador desconoce varias instancias! Una buena forma de evaluar en este contexto es pasárselas a cada clasificador las instancias que no vio para evaluar el desempeño. Así, cada elemento del dataset es evaluado por los clasificadores que no lo vieron en su entrenamiento, y así tenemos una predicción para cada elemento del dataset original. Luego, el out-of-bag (oob) score es el accuracy obtenido de esta forma.

## Random Forests

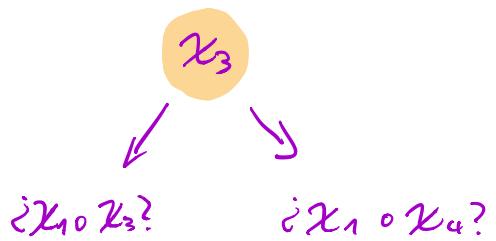
Un random forest es simplemente un método de "Ensemble", en donde hacemos Bagging de varios árboles de decisión (sí, muchos "trees" forman un "forest"), pero con una variación: cuando entramos cada árbol y decidimos hacer split, solo podemos considerar un número limitado de features para continuar. Este número se fija al inicio.

Ejemplo: Si tenemos este dataset y decidimos considerar 2 features por split:

$x_1$	$x_2$	$x_3$	$x_4$
-------	-------	-------	-------

① Supongamos que partimos con  $x_3$

② Ahora en cada branch escogemos 2 valores al azar



③ Solo podemos hacer split por esos valores

Así, todos los árboles son entrenados de ese forma aleatoria. Ahora cuando tenemos una instancia desconocida, la evaluamos en cada árbol y votamos!

¿Y cómo sabemos el número de features a considerar en el random forest? En general, vamos probando y evaluando con el oob score, y nos quedamos con la mejor configuración.