03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

1 MACHINE LEARNING – RANDOM FOREST

Tabla de contenido

El **Random Forest** es un algoritmo de aprendizaje automático supervisado que se utiliza ampliamente en problemas de clasificación y regresión. Construye **árboles de decisión** sobre diferentes muestras y toma su voto mayoritario para la clasificación y la media en el caso de la regresión.

Una de las características más importantes del algoritmo Random Forest es que puede manejar el conjunto de datos que contiene variables continuas como en el caso de la regresión y variables categóricas como en el caso de la clasificación. **Obtiene mejores resultados en los problemas de clasificación**.

1.1 Analogía de la vida real

Veamos una analogía de la vida real para entender mejor este concepto. Un estudiante llamado X quiere elegir un grado después de su bachillerato, y está confundido sobre la elección del grado basado en su conjunto de habilidades. Así que decide consultar a varias personas, como sus primos, sus profesores, sus padres... Les hace varias preguntas como por qué debería elegir, las oportunidades de trabajo con ese grado, el precio del grado, etc. Finalmente, después de consultar a varias personas sobre el grado, decide tomar el grado sugerido por la mayoría de las personas.

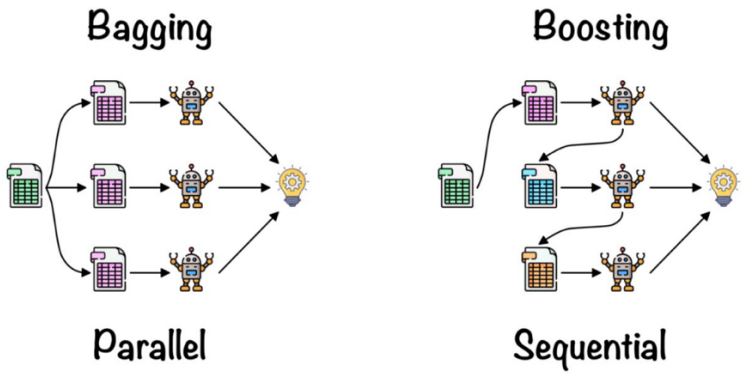
03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa Tel 966870140 Fax 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

1.2 Funcionamiento del algoritmo Random Forest

Antes de entender el funcionamiento del algoritmo Random Forest en el aprendizaje automático, debemos examinar la técnica de **ensemble**. Ensemble significa simplemente la combinación de múltiples modelos. Así, se utiliza una colección de modelos para hacer predicciones en lugar de un modelo individual.

Ensemble usa dos tipos de métodos:

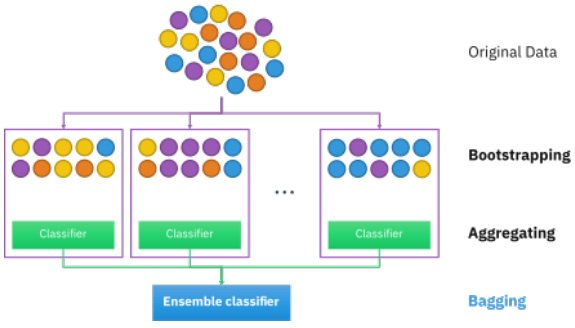
1. **Bagging** Crea un subconjunto de entrenamiento diferente a partir de los datos de entrenamiento de la muestra con reemplazo y el resultado final se basa en la votación por mayoría. Por ejemplo, Random Forest.
2. **Boost** Combina los aprendices débiles en aprendices fuertes creando modelos secuenciales de manera que el modelo final tenga la mayor precisión. Por ejemplo, ADA BOOST, XG BOOST



Como se ha mencionado anteriormente, el **Random Forest** funciona según el principio de **Bagging**. Ahora vamos a entender en detalle este proceso.

1.2.1 Bagging

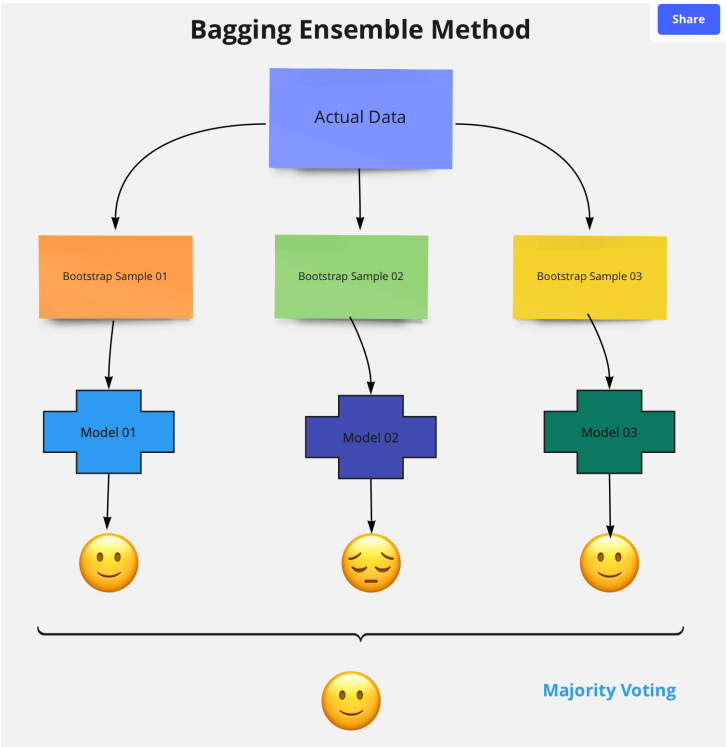
El Bagging, también conocido como Bootstrap Aggregation, es la técnica de ensamblaje utilizada por Random Forest. El bagging **elige una muestra aleatoria del conjunto de datos.** Por lo tanto, cada modelo se genera a partir de las muestras (Muestras Bootstrap) proporcionadas por los Datos Originales con reemplazo conocido como muestreo de filas. Este paso de muestreo de filas con reemplazo se llama bootstrap. Ahora **cada modelo se entrena de forma independiente,** lo que genera resultados. **El resultado final se basa en la votación por mayoría** tras combinar los resultados de todos los modelos. Este paso, que implica la combinación de todos los resultados y la generación de resultados basados en la votación por mayoría, se conoce como agregación.



03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

Ahora veamos un ejemplo desglosándolo con la ayuda de la siguiente figura. Aquí la muestra bootstrap se toma de los datos reales (*muestra bootstrap 01, muestra bootstrap 02 y muestra bootstrap 03*) con un reemplazo, **lo que significa que hay una alta posibilidad de que cada muestra no contenga datos únicos**. Ahora el modelo (Modelo 01, Modelo 02 y Modelo 03) obtenido de esta muestra bootstrap se entrena de forma independiente. Cada modelo genera los resultados que se muestran.

El emoji feliz tiene una mayoría en comparación con el emoji triste. Por lo tanto, basándose en la votación por mayoría, el resultado final es emoji feliz.



03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

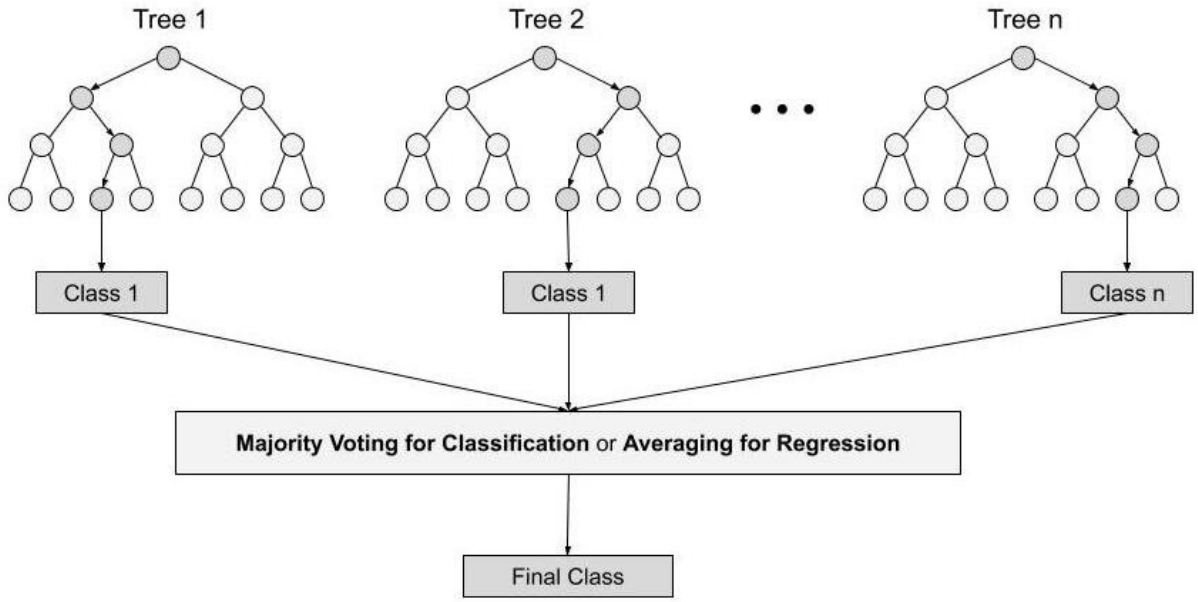
1.3 Pasos del algoritmo Random Forest:

Paso 1: En el Random Forest se toma un número n de registros aleatorios del conjunto de datos que tiene un número k de registros.

Paso 2: Se construyen árboles de decisión individuales para cada muestra.

Paso 3: Cada árbol de decisión genera un resultado.

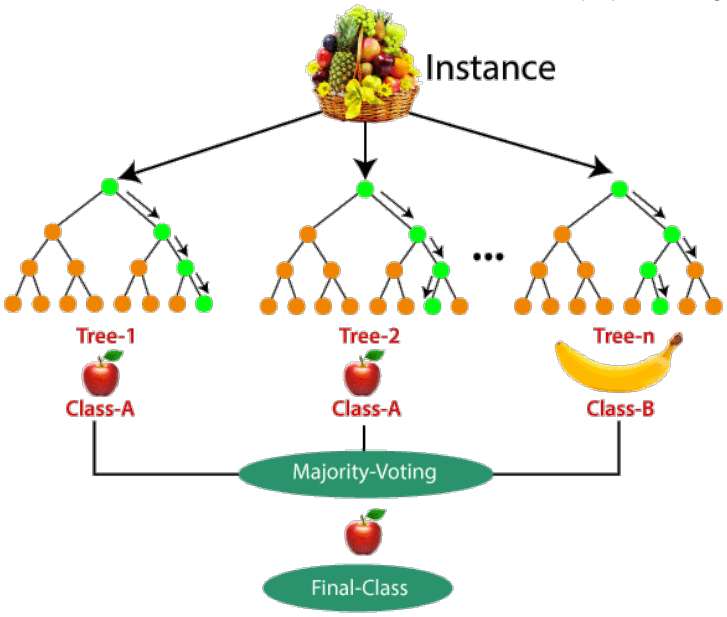
Paso 4: El resultado final se considera basado en el voto mayoritario o el promedio para la clasificación y la regresión respectivamente.



Por ejemplo: considerar la cesta de fruta como los datos que se muestran en la figura siguiente. Ahora se toman **n muestras** de la cesta de fruta y se construye un árbol de decisión individual para cada muestra. **Cada árbol de decisión generará una salida** como se muestra en la figura. **El resultado final se considera basado en la votación por mayoría**.

En la figura siguiente se puede ver que el árbol de decisión mayoritario da como resultado una manzana en comparación con un plátano, por lo que el resultado final se toma como una manzana.

03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza



1.4 Características importantes de Random Forest

Diversidad- No se consideran todos los atributos/variables/características mientras se hace un árbol individual, cada árbol es diferente.

Inmune a la maldición de la dimensionalidad - Dado que cada árbol no considera todas las características, el espacio de características se reduce.

Paralelización-Cada árbol se crea de forma independiente a partir de diferentes datos y atributos. Esto significa que podemos hacer un uso completo de la CPU para construir Random Forest.

Estabilidad - La estabilidad se debe a que el resultado se basa en el voto mayoritario/el promedio.

03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

1.5 Ventajas y desventajas de RandomForest

Ventajas	Desventajas
1. Puede utilizarse en problemas de clasificación y regresión. **2. Resuelve el problema del sobreajuste ya que la salida se basa en la votación por mayoría o en el promedio.** **3. Tiene un buen rendimiento incluso si los datos contienen valores nulos/faltantes (*cosa que no es aconsejable*).** **4. Cada árbol de decisión creado es independiente del otro, por lo que muestra la propiedad de paralelización.** **5. Es altamente estable ya que se toman las respuestas medias dadas por un gran número de árboles.** **6. Mantiene la diversidad ya que no se consideran todos los atributos al hacer cada árbol de decisión, aunque no es cierto en todos los casos.** **7. Es inmune a la maldición de la dimensionalidad. Como cada árbol no considera todos los atributos, el espacio de características se reduce.**	1. El Random Forest es muy complejo si lo comparamos con los árboles de decisión en los que las decisiones se pueden tomar siguiendo la trayectoria del árbol. 2. El tiempo de entrenamiento es mayor en comparación con otros modelos debido a su complejidad. Cada vez que tiene que hacer una predicción, cada árbol de decisión tiene que generar una salida para los datos de entrada dados.

03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

1.6 HIPERPARÁMETROS

- Propios del Bosque Aleatorio:
  - **n_estimators**: número de árboles que va a tener el bosque aleatorio. Normalmente cuantos más mejor, pero a partir de cierto punto deja de mejorar y sólo hace que vaya más lento. Un buen valor por defecto puede ser el uso de 100 árboles.
  - **n_jobs**: número de cores que se pueden usar para entrenar los árboles. Cada árbol es independiente del resto, así que entrenar un bosque aleatorio es una tarea muy paralelizable. Por defecto sólo utiliza 1 core de la CPU. Para mejorar el rendimiento puedes usar tantos cores como estimes necesario. Si usas n_jobs = -1, estás indicando que quieres usar tantos cores como tenga tu máquina.
  - **max_features**: usa forma de garantizar que los árboles son diferentes, es que todos se entrenan con una muestra aleatoria de los datos. Si queremos que todavía sean más diferentes, podemos hacer que distintos árboles usen distintos atributos. Esto puede ser útil especialmente cuando algunos atributos están relacionados entre sí.

- Regularización (también disponibles para Decision Trees):
  - **max_depth**: la profundidad máxima del árbol.
  - **min_samples_split**: número mínimo de muestras necesarias antes de dividir este nodo. También se puede expresar en porcentaje.
  - **min_samples_leaf**: número mínimo de muestras que debe haber en un nodo final (hoja). También se puede expresar en porcentaje.
  - **max_leaf_nodes**: número máximo de nodos finales