/tikz/,/tikz/graphs/

conversions/canvas coordinate/.code=1 ,
conversions/coordinate/.code=1

Universidad Externado **Departamento de Matemáticas**

Modelos Supervisados

Certificación en Ciencia de Datos

Andrés Martínez

28 Junio 2024

Contenido

Métodos de Ensamble

- Bagging.
- Random forest.

Características

- Votación.
- Sobreajuste.

Métodos de Ensamble

Los métodos de ensamble son técnicas de aprendizaje automático que combinan múltiples modelos para mejorar la precisión y generalización. Estos métodos se basan en la premisa de que la combinación de múltiples modelos puede superar las limitaciones de un solo modelo.

Objetivo: Mejorar la precisión, reducir la varianza y controlar el sobreajuste.

Principales Métodos de Ensamble:

- Bagging (Bootstrap Aggregating)
- Random Forest
- Boosting
- Adaboost
- Gradient Boosting.

Proceso de Voting y Control de Sobreajuste

Proceso de Voting en Métodos de Ensamble

Los métodos de ensamble, como Bagging, Boosting y Random Forest, combinan múltiples modelos para obtener una predicción final. El proceso de voting es común en estos métodos y se refiere a la combinación de las predicciones individuales de los modelos base para tomar una decisión.

Problema de Sobreajuste: Al combinar múltiples modelos, existe el riesgo de sobreajuste a los datos de entrenamiento, lo que puede afectar la generalización a datos no vistos.

Control de Sobreajuste: Los métodos de ensamble controlan el sobreajuste a través de estrategias como la diversidad introducida por modelos base, ponderación de instancias o selección aleatoria de características.

Voting en Clasificación y Regresión

Voting (Votación)

El proceso de Voting en Machine Learning combina las predicciones de varios modelos para tomar una decisión final.

Clasificación por Votación:

$$\hat{y}(x) = \text{Moda}\{f_1(x), f_2(x), \ldots, f_B(x)\}$$

Regresión por Votación:

$$\hat{y}(x) = \frac{1}{B} \sum_{i=1}^{B} f_i(x)$$

Donde:

- $\hat{y}(x)$ es la predicción final.
- $f_i(x)$ son las predicciones de los modelos base i para el punto de datos x.
- Moda es la función que devuelve la etiqueta de clase más común en clasificación.

Proceso de voto

Bootstrap Aggregating (Bagging)

Bagging (Bootstrap Aggregating)

El Bagging implica la creación de múltiples conjuntos de datos de entrenamiento mediante muestreo con reemplazo. Cada conjunto se utiliza para entrenar un modelo base. Las predicciones de los modelos se combinan, por ejemplo, a través de votación (en clasificación) o promedio (en regresión).

Para crear subconjuntos de datos de entrenamiento, se utiliza la técnica de Bootstrap. Dado un conjunto de entrenamiento $\mathcal{D}$ con N puntos de datos, generamos B subconjuntos $\mathcal{D}_i$ con reemplazo.

$$\mathcal{D}_i = \text{Muestra Bootstrap}(\mathcal{D}, N)$$

Bagging (Bootstrap Aggregating)

El Bagging es un método de ensamble que combina múltiples modelos base a través de muestreo con reemplazo.

Proceso de Bagging (Clasificación):

$$\hat{y}(x) = \text{Moda}\{f_1(x), f_2(x), \ldots, f_B(x)\}$$

Proceso de Bagging (Regresión):

$$\hat{y}(x) = \frac{1}{B} \sum_{i=1}^{B} f_i(x)$$

Donde:

- $\hat{y}(x)$ es la predicción final.
- $f_i(x)$ es la predicción del modelo base i para el punto de datos x.
- Moda es la función que devuelve la etiqueta de clase más común en clasificación.

Bagging: Proceso de Votación

Bagging (Bootstrap Aggregating)

El proceso de votación en Bagging implica la combinación de múltiples modelos base para obtener una predicción final. En clasificación, se utiliza un enfoque de votación.

Proceso de Votación en Bagging (Clasificación):

$$\hat{y}(\mathrm{x}) = \mathrm{argmax}_j \left(\sum_{i=1}^{B} 1\{f_i(\mathrm{x}) = C_j\} \right)$$

Donde:

- $\hat{y}(\mathrm{x})$ es la etiqueta de clase final predicha para el punto de datos x.
- j es una etiqueta de clase en el conjunto de clases $\{C_1, C_2, \ldots, C_k\}$.
- $f_i(\mathrm{x})$ es la predicción del modelo base i para el punto de datos x.
- $1\{A\}$ es la función indicadora que devuelve 1 si la condición A es verdadera y 0 en caso contrario.

Control de Sobreajuste en Bagging

Bagging (Bootstrap Aggregating)

El Bagging reduce el sobreajuste al combinar múltiples modelos entrenados en conjuntos de datos de entrenamiento generados mediante muestreo con reemplazo.

Control de Sobreajuste:

- La diversidad introducida por los conjuntos de datos de entrenamiento bootstrap y modelos base reduce el riesgo de sobreajuste.
- La agregación de predicciones mediante votación o promedio suaviza el impacto de predicciones individuales erróneas.

Random Forest

Random Forest combina múltiples árboles de decisión construidos con muestreo y selección de características aleatorias.

Proceso de Combinación (Clasificación):

$$\hat{y}(x) = \text{Moda}\{f_1(x), f_2(x), \ldots, f_B(x)\}$$

Donde:

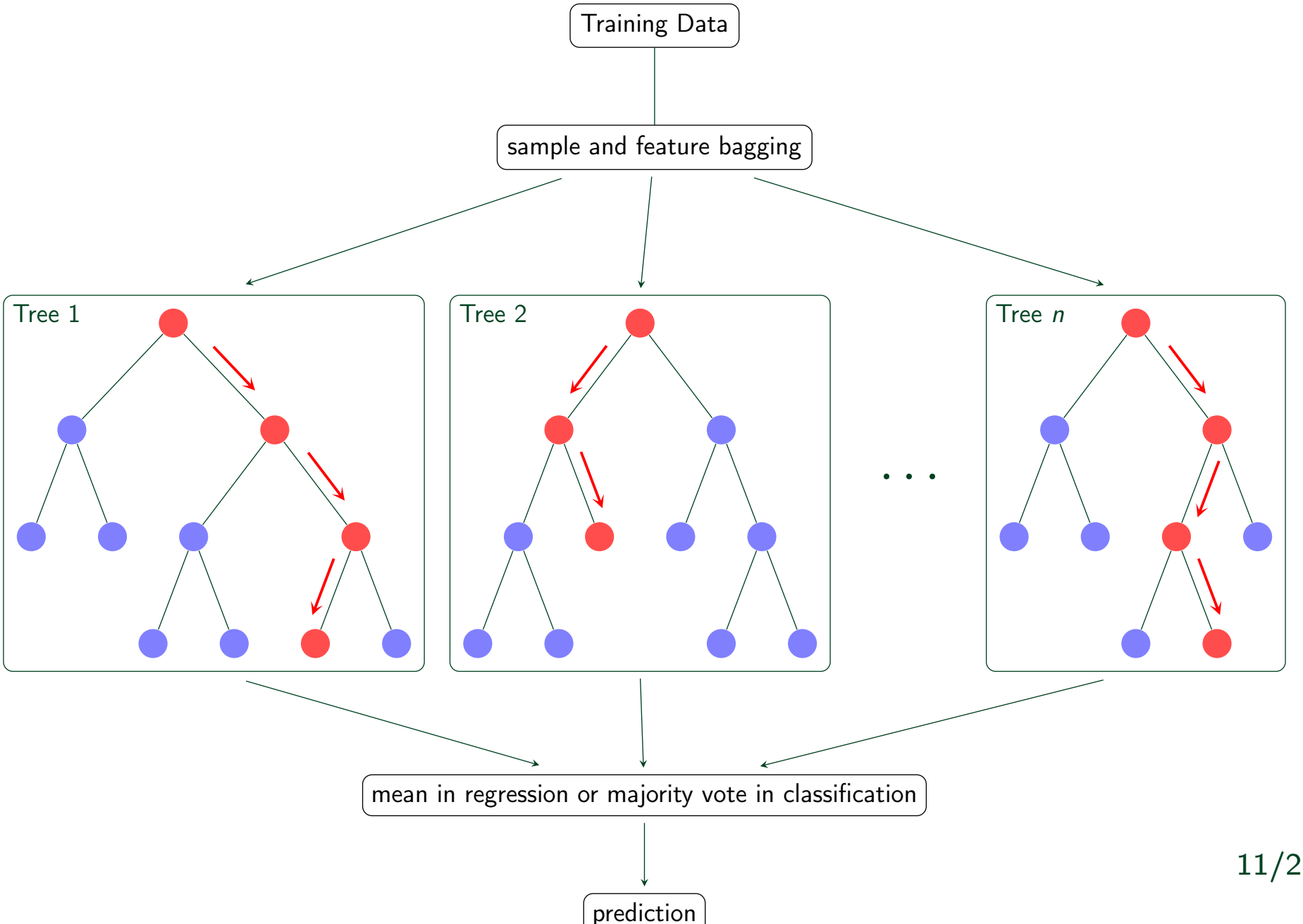
- $\hat{y}(x)$ es la predicción final.
- $f_i(x)$ es la predicción del árbol de decisión i para el punto de datos x.
- Moda es la función que devuelve la etiqueta de clase más común en clasificación.

Random Forest en Regresión

Funcionamiento:

- Un Random Forest consta de múltiples árboles de decisión.
- Cada árbol de decisión se entrena con una muestra aleatoria de datos de entrenamiento y características.
- Durante la predicción, los valores numéricos predichos por cada árbol se combinan para obtener una predicción final.
- La predicción final puede ser un promedio de las predicciones de los árboles.

Ejemplo


Training Data
sample and feature bagging
Tree 1
Tree 2
Tree n
mean in regression or majority vote in classification
prediction

Random Forest: Proceso de Votación

Random Forest

El proceso de votación en Random Forest combina las predicciones de múltiples árboles de decisión construidos a partir de conjuntos de datos de entrenamiento y características seleccionadas aleatoriamente.

Proceso de Votación en Random Forest (Clasificación):

$$\hat{y}(\mathrm{x}) = \mathrm{argmax}_j \left(\sum_{i=1}^{B} 1\{f_i(\mathrm{x}) = C_j\} \right)$$

Donde:

- $\hat{y}(\mathrm{x})$ es la etiqueta de clase final predicha para el punto de datos x.
- j es una etiqueta de clase en el conjunto de clases $\{C_1, C_2, \ldots, C_k\}$.
- $f_i(\mathrm{x})$ es la predicción del árbol de decisión i para el punto de datos x.
- $1\{A\}$ es la función indicadora que devuelve 1 si la condición A es verdadera y 0 en caso contrario.

Control de Sobreajuste en Random Forest

Random Forest

Random Forest aborda el sobreajuste al combinar múltiples árboles de decisión y características seleccionadas aleatoriamente.

Control de Sobreajuste:

- El muestreo con reemplazo y la selección aleatoria de características introducen diversidad en los modelos base.
- Cada árbol se ajusta a un subconjunto diferente de datos y características, lo que reduce la tendencia al sobreajuste.
- La combinación de múltiples árboles reduce la varianza y mejora la generalización.

Métodos de Boosting

Los métodos de Boosting son técnicas de aprendizaje automático que combinan múltiples modelos débiles para mejorar la precisión y el rendimiento del modelo final.

Principales Métodos de Boosting:

- AdaBoost (Adaptive Boosting)
- Gradient Boosting
- XGBoost (Extreme Gradient Boosting)
- LightGBM
- CatBoost.

Objetivo: Reducir el sesgo y la varianza del modelo, mejorando así la capacidad de generalización.

Estos métodos se basan en la premisa de que la combinación de modelos débiles puede generar un modelo fuerte y robusto.

Regularización en Boosting

La regularización en Boosting es una técnica utilizada para controlar el sobreajuste (overfitting) y mejorar la generalización del modelo. Esto es especialmente importante en Boosting, ya que se centra en la construcción de modelos complejos a partir de modelos débiles.

Métodos de Regularización:

- **Profundidad del Árbol:** Limitar la profundidad máxima de los árboles base para evitar modelos demasiado complejos.
- **Tasa de Aprendizaje:** Reducir la tasa de aprendizaje para disminuir la contribución de cada modelo base al modelo final.
- **Regularización L1 y L2:** Aplicar términos de regularización para penalizar coeficientes excesivamente grandes en los modelos base.

Importancia de la Regularización: La regularización es esencial para prevenir el sobreajuste y garantizar que el modelo Boosting sea robusto y generalice bien en datos no vistos.

Control de Sobreajuste en Boosting

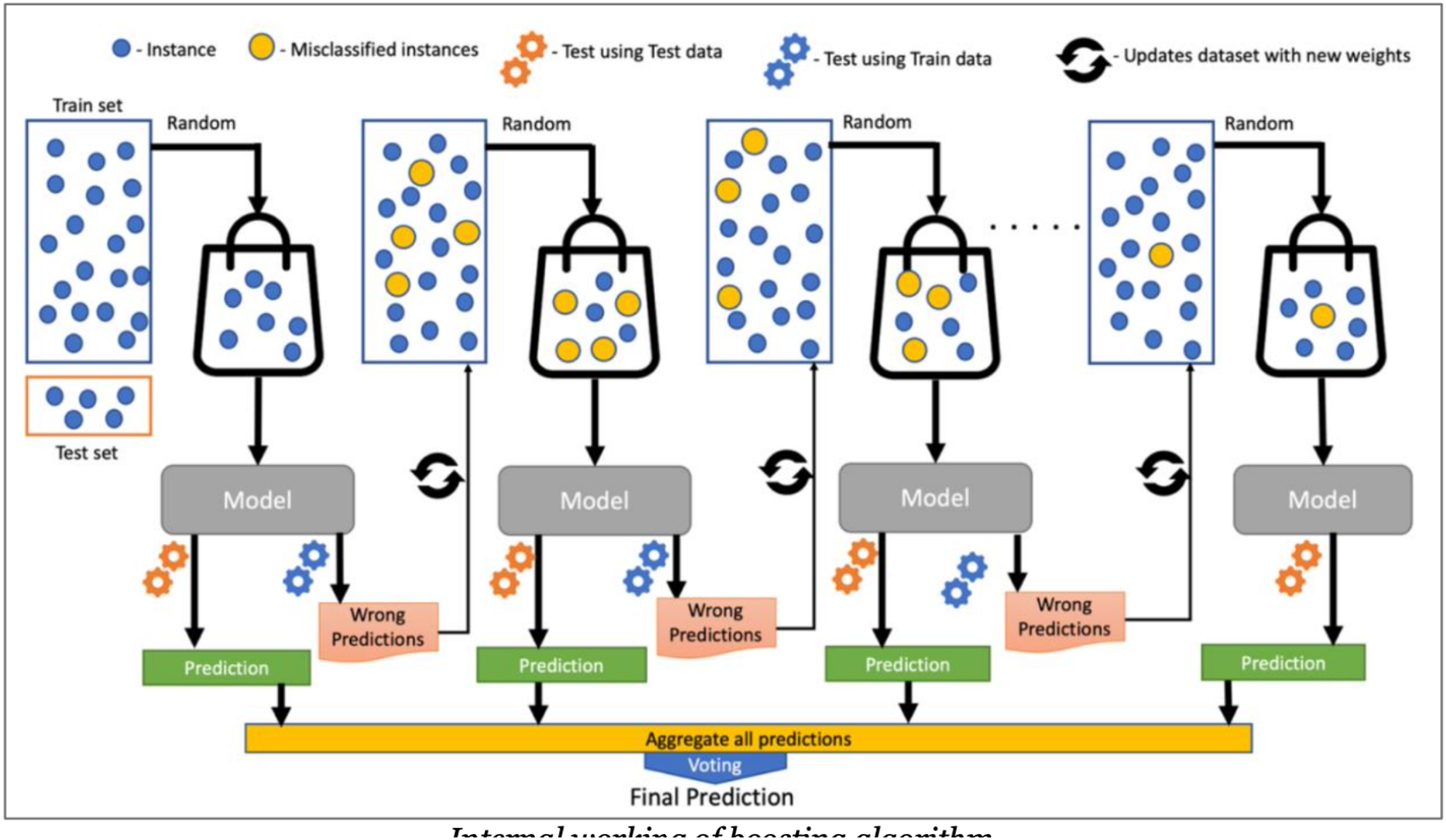
Boosting

Boosting controla el sobreajuste al dar más peso a las instancias mal clasificadas en cada iteración.

Control de Sobreajuste:

- El proceso secuencial de construcción de modelos da prioridad a las instancias difíciles.
- Los modelos posteriores se centran en corregir los errores de los modelos anteriores.
- Se ajusta la importancia de las instancias en función de su dificultad, reduciendo el impacto de los datos ruidosos.

Boosting



Internal working of boosting algorithm

AdaBoost: Aspectos Relevantes

AdaBoost

- **Enfoque:** Combina múltiples modelos débiles para formar un modelo fuerte.
- **Ponderación de Instancias:** Asigna pesos a las instancias de entrenamiento, dando más importancia a las instancias mal clasificadas.
- **Ajuste a Errores:** Iterativamente, se enfoca en corregir los errores del modelo anterior.
- **Tasa de Aprendizaje:** Controla la contribución de cada modelo débil al modelo final.

Gradient Boosting: Aspectos Relevantes

Gradient Boosting

- **Enfoque:** Construye un modelo fuerte al mejorar gradualmente los errores de los modelos anteriores.
- **Árboles de Decisión:** A menudo utiliza árboles de decisión como modelos base.
- **Hiperparámetros Importantes:** Incluye la profundidad del árbol, la tasa de aprendizaje y el número de estimadores.
- **Regularización:** Puede aplicar regularización para evitar el sobreajuste.

XGBoost: Aspectos Relevantes

XGBoost

- **Enfoque:** Extreme Gradient Boosting (XGBoost) es una implementación eficiente y escalable de Gradient Boosting.
- **Optimización:** Utiliza técnicas de optimización como la poda de árboles y muestreo para mejorar la eficiencia.
- **Hiperparámetros:** Ofrece una amplia gama de hiperparámetros, incluyendo profundidad del árbol, tasa de aprendizaje y regularización.

LightGBM: Aspectos Relevantes

LightGBM

- **Enfoque:** Algoritmo de Gradient Boosting de Microsoft con un fuerte enfoque en eficiencia.
- **Binning y Hoja Óptima:** Utiliza la técnica de binning y encuentra la hoja óptima para mejorar la velocidad.
- **Paralelización:** Aprovecha la paralelización en CPU para entrenamiento rápido.
- **Hiperparámetros:** Incluye hiperparámetros para controlar la profundidad del árbol y la tasa de aprendizaje.

CatBoost: Aspectos Relevantes

CatBoost

- **Enfoque:** Algoritmo de Gradient Boosting optimizado para datos categóricos.
- **Manejo de Categorías:** Capacidad incorporada para manejar características categóricas sin preprocesamiento.
- **Tasa de Aprendizaje Adaptativa:** Ajusta automáticamente la tasa de aprendizaje durante el entrenamiento.
- **Regularización:** Ofrece opciones de regularización para evitar el sobreajuste.

Diferencias en Métodos

AdaBoost

- **Número de Estimadores (n_estimators):** Controla el número de estimadores débiles utilizados en el proceso de boosting. Más estimadores pueden llevar al sobreajuste.
- **Tasa de Aprendizaje (learning_rate):** Controla la contribución de cada estimador al modelo final. Valores bajos requieren más estimadores.

Gradient Boosting

- **Número de Estimadores (n_estimators):** Similar a AdaBoost, controla el número de estimadores utilizados.
- **Profundidad del Árbol (max_depth):** Controla la profundidad máxima de los árboles base. A mayor profundidad, mayor riesgo de sobreajuste.

Diferencias en Métodos

XGBoost

- **Número de Estimadores (n_estimators):** Similar a AdaBoost y Gradient Boosting.
- **Profundidad del Árbol (max_depth):** Controla la profundidad máxima de los árboles. También es crítico para controlar el sobreajuste.
- **Tasa de Aprendizaje (learning_rate):** Similar a AdaBoost, controla la contribución de cada estimador.

LightGBM

- **Número de Estimadores (n_estimators):** Similar a otros métodos.
- **Profundidad Máxima (max_depth):** Controla la profundidad máxima de los árboles. Crítico para el rendimiento y el riesgo de sobreajuste.
- **Tasa de Aprendizaje (learning_rate):** Controla la contribución de cada estimador.

Diferencias en Módelos

CatBoost

- **Número de Estimadores (n_estimators):** Controla el número de estimadores utilizados en el ensamble.
- **Profundidad Máxima (depth):** Define la profundidad máxima de los árboles utilizados en el modelo. Ayuda a controlar el sobreajuste.
- **Tasa de Aprendizaje (learning_rate):** Regula la contribución de cada estimador en el modelo final, similar a otros métodos de Boosting.
- **Categorías (cat_features):** Permite especificar las características categóricas para un manejo especializado.