CLASIFICACION 2- METODOS BASADOS EN ARBOLES

Los métodos basados en árboles son técnicas de aprendizaje supervisado que utilizan divisiones jerárquicas del espacio de predictores para realizar tareas de predicción o clasificación. Son intuitivos, fáciles de interpretar y pueden ser usados tanto para variables respuesta continuas como categóricas.

**1. Árboles de Decisión**

**1.1 Árboles de Regresión**

Un árbol de regresión divide los datos en regiones disjuntas y utiliza la media de la variable respuesta en cada región para hacer predicciones.

* **División del espacio predictor:** Dado un conjunto de datos, el árbol busca dividir el espacio predictor X en J regiones R1,R2,…,RJ, de manera que se minimice la suma de cuadrados dentro de cada región:

A black and white math symbol

Description automatically generated with medium confidence

Donde y^Rj​​ es la media de las observaciones yi​ en la región Rj​.

* **Criterio de división:** En cada nodo, se busca el predictor Xj​ y un punto de división s que minimicen el error dentro de las nuevas regiones:



**1.2 Árboles de Clasificación**

Los árboles de clasificación se construyen de forma similar, pero su objetivo es predecir clases en lugar de valores continuos.

* **Medida de impureza:** La división en nodos utiliza métricas como:
  + **Índice de Gini:** A black and white math equation

    Description automatically generated with medium confidence
  + Donde pk​ es la proporción de observaciones en la clase kkk.
  + **Entropía:** A black and white symbol

    Description automatically generated
* **Regla de decisión:** Asignar la clase mayoritaria en el nodo final.

**2. Métodos de Ensamble.**

**Los métodos de agregación y ensamble (como Bagging, Random Forests y Boosting) se desarrollaron para abordar las limitaciones intrínsecas de los árboles de decisión individuales. Estas limitaciones motivan el uso de estos enfoques avanzados para mejorar la precisión y la robustez de los modelos basados en árboles.**

**Limitaciones de los Árboles de Decisión Individuales**

Alta varianza

* Sensibilidad a los datos de entrenamiento: Los árboles de decisión son modelos de alta varianza, lo que significa que un pequeño cambio en los datos de entrenamiento puede producir un árbol completamente diferente.
* Problema: Esto lleva a modelos que tienden a sobreajustarse a los datos, perdiendo capacidad de generalización.

Baja precisión

* Aunque los árboles de decisión pueden capturar patrones complejos, suelen ser menos precisos que otros modelos más sofisticados, como regresión logística o modelos lineales regulares.
* En problemas reales, un solo árbol puede tener un desempeño deficiente porque no captura bien las relaciones complejas entre las variables predictoras y la respuesta.

Rigidez en las predicciones

* Los árboles dividen el espacio predictor en regiones rígidas y no interpolan suavemente entre valores. Esto puede ser un problema para datos continuos o relaciones no lineales complicadas.

**¿Cómo solucionan esto los métodos de agregación y ensamble?**

**Bagging (Bootstrap Aggregating): Reducir la varianza**

Bagging aborda la alta varianza de los árboles individuales al entrenar múltiples árboles en diferentes subconjuntos bootstrap de los datos originales y luego promediar sus predicciones (regresión) o usar un voto mayoritario (clasificación).

* Ventaja:
  + Reduce la varianza del modelo promedio porque los errores individuales de los árboles tienden a cancelarse entre sí.
* Ejemplo:
  + Si un árbol se ajusta en exceso a un patrón específico del subconjunto de datos, otros árboles corregirán este error al no incluir el mismo patrón.

**Random Forests: Reducir la correlación entre árboles**

Random Forests extienden el Bagging introduciendo aleatoriedad adicional en la selección de predictores para dividir en cada nodo del árbol.

* Por qué es necesario:
  + En Bagging, los árboles suelen estar correlacionados si algunos predictores dominan las divisiones iniciales. Esto limita la reducción de varianza.
* Cómo lo resuelve Random Forests:
  + En cada división, selecciona un subconjunto aleatorio de predictores, forzando al modelo a considerar diferentes combinaciones de variables.
* Ventaja:
  + Reduce aún más la varianza y mejora la precisión general del modelo.

**Boosting: Mejorar la precisión**

Boosting aborda tanto la alta varianza como la baja precisión, pero de una manera diferente. Los árboles en Boosting se entrenan secuencialmente, donde cada árbol intenta corregir los errores del árbol anterior.

* Por qué es necesario:
  + En los métodos como Bagging y Random Forests, los árboles se entrenan de manera independiente. Esto no aprovecha la posibilidad de que los errores de un modelo sean corregidos progresivamente.
* Cómo funciona Boosting:
  + Ajusta árboles simples (generalmente de poca profundidad, llamados "weak learners").
  + Cada árbol se entrena en los residuos (errores) de los árboles anteriores, enfocándose en las observaciones más difíciles de predecir.
* Ventaja:
  + Boosting mejora la precisión al crear un modelo combinado que es más fuerte que cualquier árbol individual.

A screenshot of a computer screen

Description automatically generated

**¿Por qué seguimos usando árboles individuales?**

Aunque los métodos de ensamble son más potentes, los árboles individuales tienen ciertas ventajas

**Interpretabilidad:** Son intuitivos y fáciles de entender. Útiles cuando la claridad del modelo es crucial (p. ej., en problemas médicos).

**Eficiencia:** Rápidos de ajustar y predicen rápidamente en comparación con los métodos de ensamble más complejos.

**Base para otros métodos:** Bagging, Random Forests y Boosting se construyen a partir de árboles individuales como bloques básicos.

* + Los métodos de agregación y ensamble como Bagging, Random Forests y Boosting existen porque los árboles individuales tienen alta varianza, menor precisión y tienden a sobreajustarse. Estos métodos combinan múltiples árboles para aprovechar sus fortalezas mientras mitigan sus debilidades, logrando modelos más precisos, estables y robustos en una amplia variedad de problemas.

**2.1 Bagging (Bootstrap Aggregating)**

Bagging reduce la varianza combinando múltiples árboles de decisión generados a partir de diferentes subconjuntos bootstrap del conjunto de datos original.

* **Predicción:** Para regresión, se promedian las predicciones: A number and mathematical symbols

  Description automatically generated with medium confidence
* Para clasificación, se toma la mayoría de votos.

**2.2 Bosques Aleatorios (Random Forests)**

Extiende el bagging al introducir aleatoriedad adicional en el crecimiento de los árboles. En cada división, se selecciona aleatoriamente un subconjunto de predictores.

* **Ventaja:** Reduce la correlación entre los árboles, mejorando la precisión.

**2.3 Boosting**

Boosting ajusta árboles secuencialmente, donde cada árbol intenta corregir los errores de los anteriores.

* **Modelo:** Las predicciones se construyen como una suma ponderada de los árboles ajustados: A black and white math symbols

  Description automatically generated
* Donde λ es una tasa de aprendizaje pequeña para evitar sobreajuste.

**3. Ventajas y Desventajas de los Árboles**

**Ventajas:**

* Intuitivos y fáciles de interpretar.
* No requieren normalización de datos ni supuestos lineales.
* Capaces de capturar interacciones entre variables.

**Desventajas:**

* Pueden sobreajustarse a los datos si no se podan.
* Menor precisión en comparación con métodos como random forests o boosting.

**4. Comparación de Métodos Basados en Árboles**

1. **Bagging:** Reduce la varianza pero puede perder interpretabilidad.
2. **Random Forests:** Similar a bagging, pero más robusto al introducir aleatoriedad.
3. **Boosting:** Muy preciso, pero propenso al sobreajuste si no se ajustan los parámetros cuidadosamente.

**Cómo se construye un Árbol de Decisión**

Un árbol de decisión, ya sea para clasificación o regresión, se construye dividiendo recursivamente el espacio predictor en regiones más pequeñas, con el objetivo de minimizar un criterio de error.

**1. Comenzando con el nodo raíz**

El nodo raíz contiene todas las observaciones del conjunto de datos de entrenamiento.

1. Se evalúa cada predictor Xj​ para encontrar un punto de división s.
2. Se calcula un criterio de error en las dos regiones resultantes:
   * Para **regresión**, minimizamos la suma de los cuadrados dentro de las regiones: A black and white math symbols

     Description automatically generated with medium confidence
   * Para **clasificación**, se mide la **impureza del nodo** usando índices como:
     + **Índice de Gini**
     + **Entropía**
3. Se selecciona el predictor Xj y el punto s que minimicen el criterio de error. El conjunto de datos se divide en dos regiones:
   * R1={X ∣ Xj ≤ s}
   * R2={X ∣ Xj > s}

**2. División Recursiva**

Cada región se trata como un nuevo nodo en el árbol. El proceso de división se repite para cada nodo:

1. Se evalúan todas las posibles divisiones en los datos del nodo.
2. Se elige la división que minimice el error dentro de las nuevas subregiones.
3. Esto continúa hasta alcanzar uno de los siguientes criterios de parada:
   * Se alcanza un tamaño mínimo de nodo (por ejemplo, menos de 5 observaciones).
   * La reducción en el error es insignificante.
   * La profundidad máxima del árbol ha sido alcanzada.

**3. Poda del Árbol (Opcional)**

Un árbol sin restricciones tiende a sobre ajustarse a los datos de entrenamiento. La **poda** consiste en eliminar ramas que no aportan mejoras significativas al rendimiento general del modelo.

* **Poda de complejidad de costo:** Se define un parámetro α\alpha que penaliza la complejidad del árbolA black and white symbol

  Description automatically generated
* Donde:
  + RSSm: Error residual de las hojas.
  + ∣T∣: Número de hojas en el árbol.

Se selecciona α utilizando validación cruzada para optimizar el equilibrio entre ajuste y simplicidad.

**4. Predicción con el Árbol**

Una vez que el árbol está construido:

1. **Para clasificación:** Cada nodo terminal predice la clase mayoritaria en esa región.
2. **Para regresión:** Cada nodo terminal predice la media de los valores respuesta en esa región.

**Ejemplo de Construcción**

Supongamos que queremos predecir la variable respuesta Y usando dos predictores X1​ y X2​:

1. En el nodo raíz, probamos divisiones en X1​ y X2​ para encontrar la división que minimice el error.
2. Dividimos los datos en dos regiones y repetimos el proceso dentro de cada región.
3. Continuamos dividiendo hasta que se alcance el criterio de parada.
4. Podamos el árbol, si es necesario, para evitar sobreajuste.

**Resumen del Proceso**

1. **Nodo raíz:** Incluye todos los datos.
2. **Primer nivel:** Se divide en función del predictor y punto de división seleccionados.
3. **Niveles subsiguientes:** Cada región se subdivide más.
4. **Nodos terminales (hojas):** Estas representan las predicciones finales.