Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Matemática e Instituto de Cálculo

Métodos para interpretar ensambles de árboles

Tesis presentada para optar al título de Magister en Estadística Matemática de la Universidad de Buenos Aires.

Autor: Martín Gabriel Cargnel

Enero, 2026

Resumen

Los ensambles de árboles, en particular *Random Forest* y *Gradient Boosting Machines*, son muy utilizados por su alta capacidad predictiva. Sin embargo, son considerados modelos de çaja negra"dado que por su complejidad resulta difícil el efecto de las covariables en la variable de respuesta. En este trabajo se estudiaron técnicas que permiten entender la importancia y el impacto que tienen las covariables, con el fin de utilizar modelos este tipo de modelo no solo para predecir, sino también para interpretar.

Índice general

1. Introducción

Este capítulo introduce los conceptos fundamentales del modelado estadístico, diferenciando entre los objetivos de predicción y explicación. Se analiza el compromiso o *trade-off* existente entre ambos y se propone el uso de técnicas de aprendizaje automático interpretable, específicamente en el contexto de ensambles de árboles. Finalmente, se detalla la estructura del trabajo y la notación utilizada.

Dentro del modelado estadístico se destacan dos objetivos principales: predecir y explicar. Esta distinción puede parecer filosófica en un primer momento, pero es de interés estudiarla dado que genera diferencias metodológicas importantes que llevan a los investigadores y practicantes a optar por técnicas y modelos muy distintos para alcanzar sus objetivos.

A continuación, se estudian las dos principales corrientes descritas en [**?**].

1.1. Modelado Explicativo

El modelado explicativo se centra en la comprensión de las relaciones subyacentes entre las variables, priorizando la interpretabilidad y el sustento teórico.

Cuando el interés es interpretar o explicar, los estudios se centran en entender el efecto que tienen variables explicativas (X) en una variable de respuesta (Y). Este tipo de estudios tratan de emplear modelos más simples para que sea más sencillo entender cómo cada covariable se relaciona con la variable de respuesta. En estos casos, típicamente se utilizan regresiones lineales por su simplicidad.

Generalmente esta clase de modelado se utiliza mucho en las ciencias sociales, donde existe una teoría muy fuerte por detrás que sustenta los modelos. En definitiva, se puede considerar que los modelos se utilizan con el objetivo de testear las teorías con datos empíricos, muchas veces intentando entender relaciones causales. Finalmente, pese a que la capacidad de predicción, medida utilizando alguna métrica como error cuadrático medio o error de clasificación, pasa a un segundo plano, es de interés encontrar modelos que superen un cierto umbral de precisión.

En resumen, el objetivo principal de este enfoque es validar teorías y cuantificar relaciones, aceptando limitaciones en la capacidad predictiva a cambio de transparencia en el modelo.

1.2. Modelado Predictivo

Por otro lado, el modelado predictivo se enfoca en la precisión de las estimaciones futuras, a menudo sacrificando la simplicidad del modelo.

Cuando el objetivo es predecir, el foco cambia y se busca estimar nuevos valores de la variable dependiente (Y), dadas ciertas variables explicativas o predictoras (X). En estos casos se suelen utilizar modelos más complejos, como ensambles de árboles o redes neuronales, normalmente asociados al aprendizaje automático. El procesamiento del lenguaje natural es un caso claro donde el objetivo es predecir. Para esta disciplina se emplean redes neuronales con múltiples capas que hacen muy difícil su interpretación que, de todas formas, pasa a un segundo plano. Como puede esperarse, cuando el objetivo es predecir, validar una teoría tampoco juega un rol tan importante.

En conclusión, este paradigma prioriza la minimización del error de predicción utilizando algoritmos flexibles, donde la comprensión del mecanismo generador de datos es secundaria.

1.3. *Trade-off* entre explicar y predecir

Existe una tensión inherente entre la capacidad de un modelo para explicar la realidad y su capacidad para predecir nuevos eventos, conocida como el *trade-off* entre interpretabilidad y flexibilidad.

Según lo descrito anteriormente, parece haber un *trade-off* entre predecir y explicar. Es decir, que si el objetivo es predecir se debería optar por modelos más complejos, poco interpretables pero muy flexibles, para minimizar lo más posible una métrica de error. Por otra parte, si el objetivo es explicar, deberíamos aceptar modelos con baja capacidad predictiva, dado que para aumentar el poder predictivo necesariamente deberíamos complejizar demasiado los modelos, al punto de no poder interpretarlos.

Este *trade-off* también puede entenderse como que en aplicaciones donde se busca predecir, el modelo se adapta a los datos, mientras que cuando se busca explicar, los datos (o la realidad) se simplifican para ser explicados por el modelo. Por lo que es de interés estudiar técnicas que permitan utilizar modelos complejos para explicar el efecto de X sobre Y.

En síntesis, este compromiso obliga habitualmente a elegir entre modelos simples e interpretables o modelos complejos y precisos, lo que motiva la búsqueda de herramientas que unan ambos mundos.

1.4. Aprendizaje automático interpretable

El aprendizaje automático interpretable surge como un campo de estudio destinado a dotar de transparencia a los modelos predictivos complejos.

Generalmente los modelos de aprendizaje automático destacan por su capacidad predictiva. Sin embargo, muchas veces se consideran cajas negras dado que, debido a su complejidad, es muy difícil responder preguntas como: ¿Qué variables son más importantes? o ¿Cuál es el efecto de las variables explicativas en la variable de respuesta?

Sin embargo, existe una batería de técnicas que permiten responder estas preguntas e intentar mitigar el *trade-off* descrito anteriormente. Las técnicas que se estudiarán en este trabajo pueden ser consideradas agnósticas al modelo, es decir,

se asume que el modelo es una caja negra y se aplican luego técnicas para interpretarlo. Esto se utiliza dado que la cantidad de modelos que pueden interpretarse "directamente" es limitada. Pero al usar estas técnicas se evita tener que restringir tanto la cantidad de modelos a utilizar, pudiendo emplear, por ejemplo, modelos más flexibles.

En definitiva, estas metodologías permiten aprovechar el alto rendimiento de los modelos de caja negra sin renunciar a la comprensión de su comportamiento.

1.5. Ensambles de árboles como candidatos

Dentro de los modelos de aprendizaje automático, los ensambles de árboles presentan características ideales para balancear precisión e interpretabilidad.

En este trabajo se decidió estudiar los ensambles de árboles como modelo ya que, junto con técnicas de interpretabilidad, permiten mitigar el *trade-off* entre interpretabilidad y poder predictivo. Este modelo se eligió sobre otras alternativas debido a su alto poder predictivo, lo que lo hace muy popular para tareas de predicción. Su popularidad y buen rendimiento resultaron en la disponibilidad de numerosos paquetes en varios lenguajes de programación, lo que facilita su implementación de manera sencilla y eficiente. Además, dado que los ensambles de árboles se basan en utilizar múltiples árboles de decisión para la predicción, mantienen algunas ventajas de los árboles individuales, como la capacidad de modelar las interacciones entre las variables.

Cabe destacar que si bien estos modelos pueden utilizarse tanto para clasificación como para regresión, en este trabajo solo se estudiará la metodología para problemas de regresión. De todas formas, todos los resultados a los que se lleguen son fácilmente adaptables a problemas de clasificación. Por lo tanto, los ensambles de árboles constituyen el núcleo del estudio, ofreciendo un equilibrio óptimo para la aplicación de técnicas de interpretación.

1.6. Estructura del trabajo

El trabajo se estructura de la siguiente manera: en el Capítulo 2 se hace una revisión de la teoría detrás de los árboles de regresión y clasificación, mencionando el algoritmo que se usa para podarlos, cómo se comportan cuando las variables están correlacionadas y ante la presencia de *outliers*. Se finaliza con algunas de las limitaciones que llevaron al desarrollo de técnicas más avanzadas. En el Capítulo 3 se estudian los ensambles de árboles y cómo estos alivian algunas de las limitaciones de los árboles individuales. En particular se profundiza sobre las técnicas *Bagging* y *Boosting*. Luego, en el Capítulo 4, se revisan tres técnicas que permiten interpretar los ensambles de árboles descritos anteriormente. Finalmente, se presenta una aplicación de todo lo descrito anteriormente en el Capítulo 5, seguido por las conclusiones en el Capítulo 6.

Esta estructura secuencial guía al lector desde los fundamentos teóricos básicos hacia modelos más complejos y sus técnicas de interpretación, culminando con la aplicación práctica.

1.7. Notación

Para cerrar este capítulo introductorio, se establece la notación matemática que se utilizará de manera consistente a lo largo del documento.

Sea P_{XY} la distribución conjunta inducida por el proceso generador de datos, donde X es una variable aleatoria p-dimensional e Y es una variable aleatoria unidimensional. El conjunto de datos se denota como $D_n = \{(x(1), y(1)), \ldots, (x(n), y(n))\}$, donde $x(i) \in \mathbb{R}^p$ e $y(i) \in \mathbb{R}$ para $i \in \{1, \ldots, n\}$. Para las Secciones 2, 3 y 4 se utilizan datos simulados con distribución $\mathcal{U}(-1, 1)$, con $p = 10$ y $n = 1000$. El vector y fue creado de forma tal que no sea lineal y ocurran interacciones entre las variables; además, el término de error ϵ se asumió $\mathcal{N}(0, 1)$. Mientras que en la Sección 5 se emplean tres conjuntos de datos reales.

El mapeo verdadero entre características y variable objetivo se denota como $f(X) = E[Y|X = x]$. Para cualquier modelo, denotamos $f : X \to Y$ como el modelo teórico y $\hat{f} : X \to Y$ como el modelo ajustado sobre D_n, el cual se obtiene minimizando una función de pérdida $L : Y \times \mathbb{R}^p \to \mathbb{R}^0_+$. A lo largo del trabajo se estudiarán distintos tipos de modelos, denotados como f^{tree} para árboles de regresión, f^{rf} para *Random Forest* y f^{gb} para *Gradient Boosting*.

La adopción de esta notación unificada facilitará la comprensión precisa de las formulaciones matemáticas presentadas en los capítulos subsiguientes.

2. CART

En este capítulo se hará una revisión sobre árboles de regresión y clasificación (CART), basando la explicación en [**?**].

Estos modelos entran dentro de la categoría de análisis supervisado no paramétrico, por lo que tienen una gran flexibilidad que les permite modelar naturalmente las interacciones entre variables. Otra de las ventajas de estos algoritmos es su interpretabilidad, dado que es muy sencillo entender el efecto de cada variable explicativa en la variable dependiente siguiendo la estructura del árbol.

Con el objetivo de introducir el modelo estadístico en cuestión, en la **??** se muestra un árbol de decisión cuyo objetivo es predecir el índice de progresión de la diabetes utilizando la edad y el índice de masa corporal (BMI)[1]. El árbol se interpreta de arriba hacia abajo, lo que indica que el índice de masa corporal es la variable más relevante, ya que aparece en la primera división. Se observa que niveles más altos de BMI están asociados con un aumento en el índice de progresión de la diabetes.

Por otro lado, la edad también influye en la variable dependiente, pero su efecto se observa después de las primeras condiciones basadas en el BMI. Por ejemplo, para pacientes con un BMI menor o igual a 24.35 y una edad menor o igual a 57.5, el índice de progresión de la diabetes es 99.69. Esto destaca (1) la facilidad con la que se pueden interpretar visualmente los árboles de decisión y (2) cómo estos modelos representan de manera natural las interacciones entre diferentes variables.

Con este breve ejemplo se puede apreciar la facilidad con la que se pueden interpretar visualmente los árboles de decisión y cómo estos modelos representan de manera natural las interacciones entre diferentes variables. A continuación, se explicarán los elementos que componen un árbol de decisión y cómo se construyen.

2.1. Elementos que componen un árbol

Antes de pasar a la construcción de los árboles, es de interés enfatizar que estos modelos se leen de arriba hacia abajo y definir ciertos elementos que los componen. En primer lugar definimos los nodos como los puntos en los cuales se dividen las observaciones. A su vez pueden clasificarse como: la raíz, que es el primer nodo que contiene todas las observaciones y sobre el cual se hace la primera partición; los nodos internos, que son los puntos en los cuales se dividieron las observaciones; y los nodos terminales u hojas, que presentan las predicciones finales del modelo y no tienen descendientes inmediatos.

Luego, como se ve en la **??**, los nodos definidos anteriormente están unidos por

[1] Los datos provienen de [**?**]

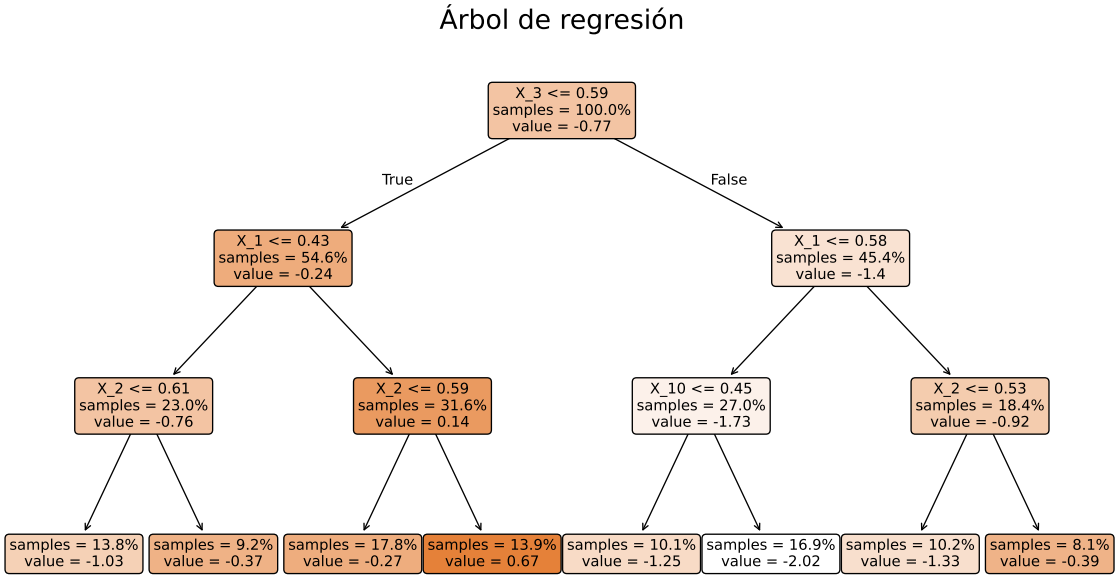

Figura 2.1: Árbol de decisión para predecir el índice de progresión de la diabetes utilizando la edad y el índice de masa corporal (BMI).

flechas que llamaremos ramas y tienen el objetivo de conectar los distintos nodos del árbol.

Habiendo definido estos elementos podemos pasar a estudiar cómo construir los árboles.

2.2. Construyendo árboles de regresión

Para construir los árboles necesitamos particionar secuencialmente el conjunto de datos X que tenemos. El objetivo final es construir un modelo de regresión que nos permita estimar el valor de una variable dependiente Y. Para ello necesitamos tres elementos: un criterio para particionar los datos, una regla para determinar cuándo un nodo es terminal y una forma de asignar un valor a las predicciones de los nodos terminales.

2.2.1. Criterio para particionar los datos

Dado que necesitamos crear $R_1, R_2, ..., R_J$ regiones distintas y disjuntas, éstas podrían tener a priori cualquier forma. Sin embargo, por simplicidad, asumiremos que tienen forma rectangular. Además, dado que se trata de un algoritmo de regresión supervisado tiene el objetivo de predecir con el menor error posible, por lo que se suele utilizar el error cuadrático medio para medir el error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.1}$$

con n la cantidad de observaciones, Y los valores reales de la variable dependiente y $\hat{Y}$ las predicciones realizadas por nuestro modelo con la muestra X.

En definitiva, si lo que buscamos son regiones que minimicen el error cuadrático medio podemos expresarlo como

$$MSE(T) = \frac{1}{n} \sum_{j=1}^{J} \sum_{i \in R_j} (Y_i - \hat{Y}_{R_j})^2 \tag{2.2}$$

con el árbol T, n el número total de observaciones en la muestra, J el número de regiones R, Y_i el valor real de las observaciones i que se encuentran en la región R_j y $\hat{Y}_{R_j}$ la predicción de nuestro modelo.

Sin embargo, calcular esta expresión es computacionalmente demasiado caro dado que implica buscar todas las posibles particiones en todas las posibles regiones que se pueden generar con la muestra. A modo de ejemplo esto significa que si tenemos p variables y cada variable tiene s puntos en los que podría ser particionada, el número total de posibles particiones de todas las variables en cada nodo sería $s \cdot p$. Dado que cada partición es binaria, el número de nodos crece exponencialmente con la profundidad del árbol d. Es decir que para calcular la cantidad total de particiones posibles (TP) para todo el árbol tendríamos que calcular

$$TP = \sum_{d=0}^{D} s \cdot p \cdot 2^d \tag{2.3}$$

que crece exponencialmente por lo que se hace computacionalmente muy intensiva.

Es por ello que se utiliza el algoritmo conocido como división binaria recursiva, este algoritmo, que va de arriba hacia abajo en el árbol, se considera *greedy* dado que en cada paso solo mira el valor óptimo en ese momento, por lo que hace menos cálculos aunque puede que no llegue a la solución óptima. En definitiva el algoritmo consiste en considerar todas las variables $X_1, X_2, ..., X_p$ y todos los posibles valores en los cuales se podría particionar (s) para quedarnos con el que minimice el error cuadrático medio en ese paso. En definitiva buscamos

$$MSE(T) = \frac{1}{n} \sum_{r=1}^{R} \sum_{i: x_i \in R_r(j,s)} (Y_i - \hat{Y}_{R_r})^2 \tag{2.4}$$

con n el número de observaciones, R el número total de regiones después de aplicar todas las divisiones, $R_r(j, s)$ es la región r creada al dividir el predictor X_j en el punto de corte s, siendo X_j el predictor que minimiza el error, Y_i el valor real de las observaciones i dentro de la región R_r y $\hat{Y}_{R_r}$ los valores predichos para todas las observaciones i en la región R_r.

Este algoritmo ya no crece de manera exponencial como el anterior, por lo que es más sencillo de calcular que el anterior.

2.2.2. Definir cuándo un nodo es terminal

Definir cuándo un nodo es terminal es equivalente a establecer una regla de parada en el algoritmo de partición recursiva. En teoría, se podría continuar dividiendo los nodos hasta que cada uno contenga una única observación, lo cual minimizaría el error en la muestra de entrenamiento a cero. No obstante, esto conduciría a un modelo sobreajustado con poca capacidad de predicción en nuevos datos.

Para evitar esto, [**?**] proponen definir un nodo como terminal cuando se cumple alguna condición restrictiva. Las condiciones más habituales incluyen que el número de observaciones en el nodo sea menor a un mínimo preestablecido (por ejemplo, $n_{min} = 5$) o que la disminución del error cuadrático medio que se obtendría con la mejor partición posible no supere un cierto umbral. También se detiene el proceso si el nodo es puro, es decir, si todas las observaciones tienen el mismo valor de respuesta.

Es importante notar que detener el crecimiento del árbol demasiado pronto puede impedir que se descubran estructuras importantes en los datos que solo aparecerían tras particiones adicionales. Por esta razón, la estrategia recomendada y más utilizada consiste en establecer criterios de parada laxos para construir un árbol inicialmente muy grande (T_0), y posteriormente aplicar técnicas de poda (*pruning*) para encontrar el subárbol óptimo, como se explicará más adelante.

2.2.3. Asignar valores a nodos terminales

Dado que la función a optimizar es el error cuadrático medio, se puede probar que la media muestral será la constante que minimice ese error.

Partimos de que queremos c que minimice

$$MSE(c) = \frac{1}{n}\sum_{i=1}^{n}(y_i - c)^2 \tag{2.5}$$

para ello primero derivamos con respecto de c

$$\begin{aligned}
\frac{d}{dc}MSE(c) &= \frac{d}{dc}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - c)^2\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{d}{dc}\left[(y_i - c)^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}2(y_i - c)(-1) \\
&= -\frac{2}{n}\sum_{i=1}^{n}(y_i - c)
\end{aligned} \tag{2.6}$$

y luego igualamos a cero:

$$\begin{aligned} -\frac{2}{n}\sum_{i=1}^{n}(y_i - c) &= 0 \\ \sum_{i=1}^{n}(y_i - c) &= 0 \\ \sum_{i=1}^{n} y_i - nc &= 0 \\ \sum_{i=1}^{n} y_i &= nc \\ \frac{1}{n}\sum_{i=1}^{n} y_i &= c \end{aligned} \tag{2.7}$$

Por lo que los valores que tendrán los nodos terminales (predicciones) serán simplemente la media de las observaciones que están en ese nodo.

2.3. Poda de árboles

Construir un árbol con demasiadas particiones puede llevar a un sobre ajuste. Básicamente porque si pensamos en el árbol más grande que se puede construir con una muestra, el mismo tendría un nodo terminal por cada observación, lo cual llevaría a un error muy bajo en la muestra de entrenamiento, pero una capacidad de generalización muy mala.

Para evitar esto, buscaremos poner un límite a cuanto puede crecer el árbol, para ello existen múltiples alternativas como fijar cotas a la profundidad del árbol, al número de observaciones que pueden quedar en la misma hoja o bien hacer crecer un árbol muy grande y luego "podarlo"para obtener árboles más pequeños que podemos comparar con validación cruzada. Esta última es una técnica que permite estimar el error de generalización dividiendo el conjunto de datos en k grupos o *folds*. El procedimiento consiste en entrenar el modelo utilizando $k-1$ grupos y evaluarlo en el grupo restante, repitiendo este proceso k veces para que cada grupo funcione como conjunto de prueba una vez. El error de validación final es el promedio de los errores obtenidos en cada iteración, lo que permite seleccionar el parámetro de complejidad que minimiza dicho error.

Antes de describir el procedimiento es importante definir T_0 como un árbol maximal y considerar que T' sea un subárbol podado de T. Para que un árbol sea considerado subárbol de otro tiene que tener $T' = T_0 - T_t$, donde T_t es una rama (subárbol) que se elimina de T_0 para obtener T'.

El problema de querer encontrar subárboles es que pueden ser demasiados, haciendo que sea computacionalmente demasiado exigente. Es por ello que se utiliza *cost complexity pruning* para trabajar un pequeño número de subárboles. A continuación vemos la ecuación:

$$\sum_{m=1}^{|T|}\sum_{i\in R_m}(Y_i - \hat{Y}_{R_m})^2 + \alpha|T| \tag{2.8}$$

con α que sirve para regular la intensidad de la regularización y $|T|$ el número de nodos terminales del árbol T. Esta expresión permite obtener una secuencia ordenada de subárboles que podemos evaluar por medio de validación cruzada. Queda claro que al aumentar α se generan árboles más pequeños.

Se puede ver en la **??** como a medida que aumentan los valores de α inicialmente el error cuadrático medio disminuye, pero luego empieza a subir debido al sobre ajuste.

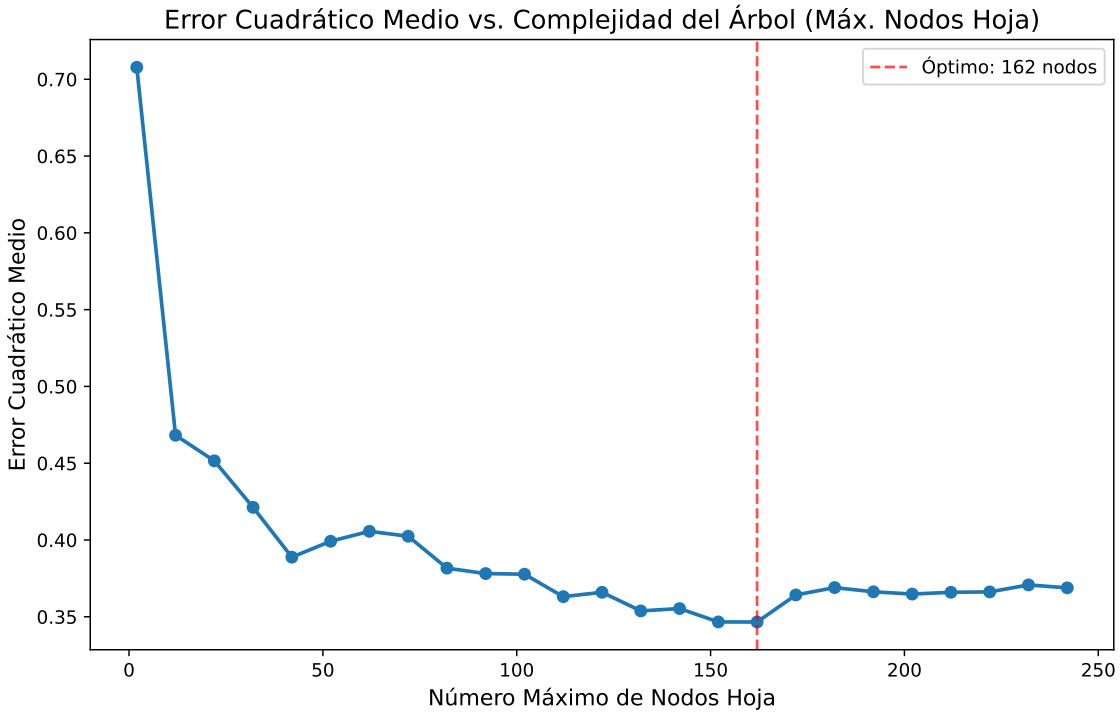

Figura 2.2: Error cuadrático medio vs complejidad del árbol para diferentes valores de α.

2.4. Outliers y Limitaciones

2.4.1. Outliers

Una característica relevante de los árboles de decisión es su manejo de los valores atípicos u *outliers*. Debido a su estructura de particionamiento recursivo del espacio de predictores, estos modelos tienden a aislar las observaciones extremas en nodos específicos. Esto contrasta con métodos que buscan ajustar una estructura global única a todos los datos, donde un valor extremo podría distorsionar la estimación general. En un árbol, una vez que el *outlier* ha sido segregado en una región terminal, su influencia queda confinada a esa predicción local, sin afectar necesariamente la estructura del resto del árbol.

Para ilustrar este comportamiento, se realizó una simulación generando $N = 100$ datos con la siguiente estructura:

$$Y = \beta_0 + \sum_{i=1}^{4} \beta_i X_i + \gamma X_5 + u \tag{2.9}$$

donde β_i son coeficientes aleatorios, y las variables explicativas $X_1, \ldots, X_4$ se distribuyen uniformemente, $X_i \sim \mathcal{U}(-5, 5)$. La variable X_5 es la encargada de introducir los valores atípicos, distribuida como una mezcla de normales. Siguiendo la notación de [**?**], su función de distribución F se define como:

$$F = (1 - \epsilon)\mathcal{N}(\mu, 1) + \epsilon\mathcal{N}(\mu, \tau^2) \tag{2.10}$$

En esta expresión, ϵ representa la probabilidad de contaminación. Para esta simulación se establecieron los parámetros $\mu = 0$, $\tau = 100$ y $\epsilon = 0{,}05$. Esto implica que el 95 % de las observaciones de X_5 provienen de una distribución normal estándar, mientras que el 5 % restante proviene de una distribución con una varianza significativamente mayor, constituyendo los *outliers*. El término de error u sigue una distribución $\mathcal{N}(0, 1)$.

La **??** muestra el histograma de la variable respuesta Y. Se puede apreciar que, si bien la mayoría de los datos se concentran en torno al cero, existen observaciones con valores extremos en ambas colas de la distribución, producto de la contaminación introducida en X_5.

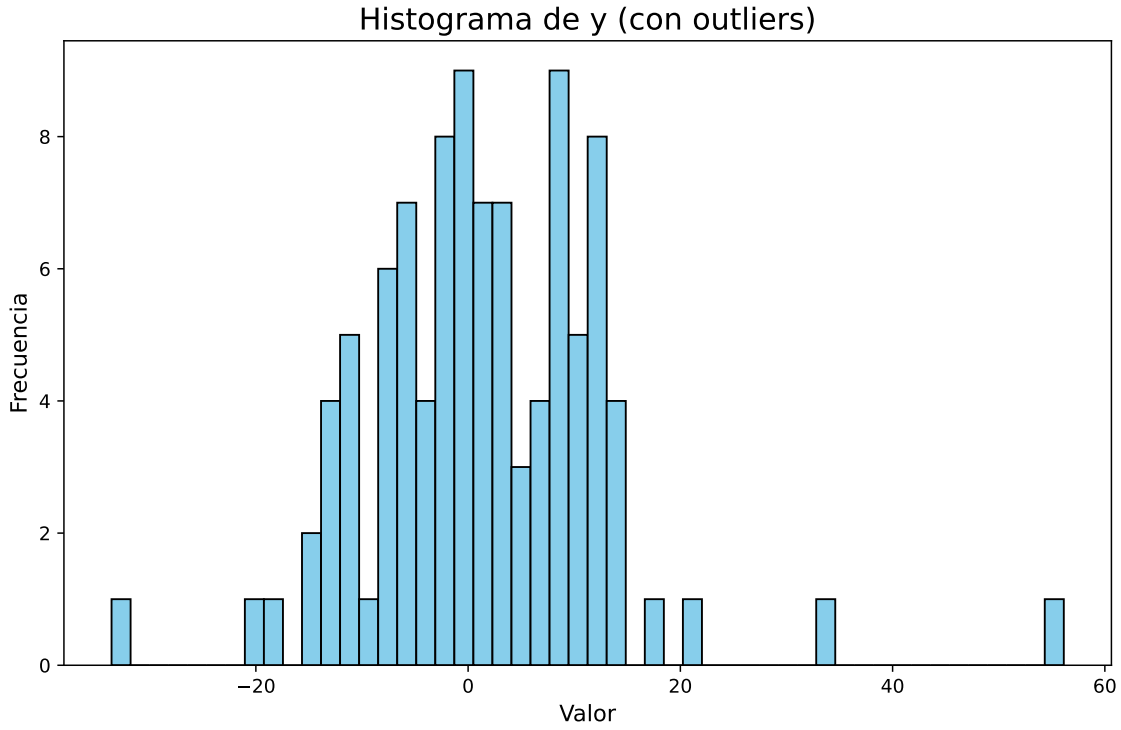

Figura 2.3: Histograma de Y evidenciando la presencia de valores atípicos.

Al ajustar un árbol de regresión a estos datos (limitando la profundidad para facilitar la visualización), el algoritmo identifica rápidamente la variable X_5 como un punto de corte crítico. Como se observa en la **??**, el árbol logra separar eficazmente las observaciones regulares de los valores anómalos. Las divisiones basadas en X_5 actúan como un filtro, agrupando los *outliers* en nodos terminales con pocos datos

pero con medias muy altas, permitiendo que los demás nodos capturen correctamente la relación subyacente para el resto de la muestra.

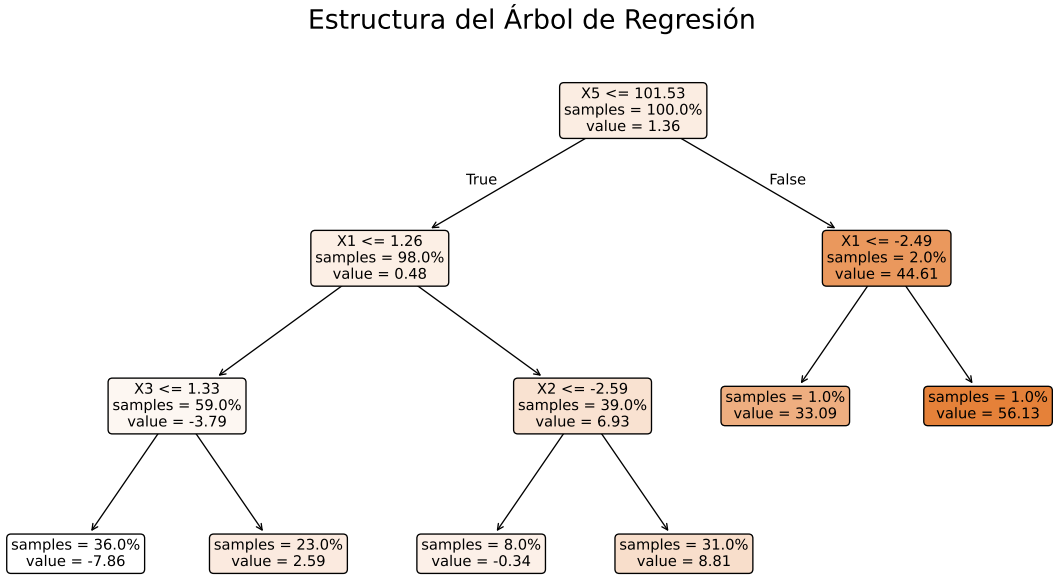

Figura 2.4: Estructura del árbol de regresión. Se observa cómo las particiones aíslan los valores extremos.

2.4.2. Limitaciones

Dado que este trabajo se centra en la interpretación de modelos, la principal limitación que se destaca es la falta de estabilidad (o alta varianza). Esto significa que pequeños cambios en los datos de entrenamiento pueden dar lugar a particiones muy diferentes y, por ende, a una estructura de árbol completamente distinta. Esta inestabilidad surge de la naturaleza jerárquica del algoritmo: un error o cambio en una división superior se propaga a todas las ramas inferiores [**?**]. Esto representa un desafío para la interpretación, ya que la importancia de las variables puede variar drásticamente con ligeras modificaciones en la muestra.

Otra limitación relevante es la dificultad que presentan los árboles para capturar estructuras aditivas simples. Mientras que una regresión lineal puede modelar eficientemente una relación del tipo $Y = \beta_0 + \sum \beta_j X_j$, un árbol de decisión necesita realizar múltiples particiones para aproximar esta estructura lineal mediante una función escalonada. Esto suele resultar en una menor capacidad predictiva en comparación con modelos paramétricos cuando la verdadera relación es lineal.

Además, las predicciones de los árboles de decisión no son continuas ni suaves, sino constantes por regiones. Esto puede ser una desventaja en problemas donde se espera que la variable respuesta cambie gradualmente con las variables explicativas.

Por último, en la práctica los árboles no suelen tener una alta capacidad predictiva, lo que limita su uso en problemas complejos y requiere el uso de técnicas como *bagging* o *boosting* para mejorar su capacidad predictiva y reducir la varianza.

3. Ensambles de árboles

Como se mencionó anteriormente, dos de los problemas principales de los árboles de decisión son la baja capacidad predictiva y la inestabilidad. Por lo que en este capítulo se estudian dos métodos para evitar estos problemas, en particular *boosting* y *bagging* con dos implementaciones clásicas como *Random Forest* y *Gradient Boosting Machines*.

Los ensambles mejoran la capacidad predictiva y reducen la variabilidad. La idea general es agrupar múltiples modelos con el objetivo de tener un único modelo con mejor capacidad predictiva.

3.1. *Bagging*

Uno de los grandes problemas de los árboles decisión es la varianza que tienen, por lo que parece intuitivo aplicar técnicas que permitan promediar métodos estadísticos para reducir la varianza manteniendo el sesgo bajo.

La idea de *Bagging* [**?**] es calcular $\hat{f}^1(x), \hat{f}^2(x), ..., \hat{f}^B(x)$ usando B muestras *bootstrap* y promediar las predicciones para reducir la varianza. Es decir

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x) \tag{3.1}$$

La ventaja de utilizar *Bagging* para predecir es que no hay que preocuparse por la profundidad de los árboles, dado que los árboles individuales tendrán muy poco sesgo y la varianza será reducida al tomar promedios. Además, si bien *Bagging* puede utilizarse en múltiples métodos estadísticos, es particularmente útil para árboles de decisión dado que permite reducir la varianza sin aumentar el sesgo.

Bagging reduce la varianza sin aumentar el sesgo. Partiendo de las predicciones

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x) \tag{3.2}$$

calcular el error viene dado por

$$MSE = E[(y - \hat{f}_{bag}(x))^2] = \text{Sesgo}^2 + \text{Varianza} + \sigma^2 \tag{3.3}$$

ahora nos concentramos en cómo la varianza afecta al error

$$Var(\hat{f}_{bag}(x)) = Var(\frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x)) = \frac{1}{B^2}Var(\sum_{b=1}^{B}\hat{f}^b(x)) \tag{3.4}$$

asumiendo que los $\hat{f}^b(x)$ están idénticamente distribuidos, se puede reescribir:

$$Var(\sum_{b=1}^{B} \hat{f}^b(x)) = \sum_{b=1}^{B} Var(\hat{f}^b(x)) + \sum_{i=1}^{B}\sum_{j\neq i}^{B} cov(\hat{f}^i(x), \hat{f}^j(x)) \tag{3.5}$$

ahora asumiendo varianzas iguales ($Var(\hat{f}^b(x)) = \sigma^2$) y correlaciones idénticas (ρ) para cada par de árboles:

$$Var(\sum_{b=1}^{B} \hat{f}^b(x)) = B\sigma^2 + B(B-1)\rho\sigma^2 \tag{3.6}$$

volviendo a la varianza total:

$$\begin{aligned} Var(\hat{f}_{bag}(x)) &= \frac{1}{B^2}(B\sigma^2 + B(B-1)\rho\sigma^2) \\ &= \sigma^2(\frac{1}{B} + \frac{\rho B(B-1)}{B^2}) \end{aligned} \tag{3.7}$$

Por lo tanto:

Por lo tanto, cuando $B \to \infty$, el término $\frac{1}{B} \to 0$ y la varianza se aproxima a $\rho\sigma^2$. Esto implica que la reducción de la varianza depende tanto de B como de ρ. Dado que la correlación entre árboles ρ es menor a 1, la varianza final ($\rho\sigma^2$) resultará menor que la varianza original (σ^2), demostrando la efectividad del método.

Bagging tiene una forma muy natural de calcular el error, básicamente se basa en el hecho de que cuando se hace una muestra *bootstrap* solamente se toma en cuenta aproximadamente el 63 % de los datos.

Para probar que *Bootstrap* solo ve aproximadamente 2/3 de los datos, partimos de la probabilidad de No ser seleccionado en una muestra:

$$P(\text{No ser seleccionado en una muestra}) = 1 - \frac{1}{n} \tag{3.8}$$

Luego, como se toman n muestras con reemplazo, la probabilidad de no ser seleccionado en ninguna muestra es

$$P(\text{No ser seleccionado en ninguna muestra}) = (1 - \frac{1}{n})^n \tag{3.9}$$

cuando $n \to \infty$:

$$\lim_{n\to\infty} (1 - \frac{1}{n})^n = \frac{1}{e} \approx 0{,}368 \tag{3.10}$$

Esto significa que la probabilidad de ser seleccionado al menos una vez es $1 - \frac{1}{e} \approx$ 0,632. Es decir, aproximadamente el 63.2 % de las observaciones serán incluidas en cada muestra *bootstrap*, mientras que el restante 36.8 % de las observaciones quedarán fuera de la muestra (*out-of-bag*).

Este resultado permite estimar el error usando las observaciones *out-of-bag* como muestra de testeo. Con un B suficientemente grande, este error es asintóticamente equivalente a *leave-one-out cross validation*, lo cual es particularmente útil cuando se cuenta con una muestra grande donde el *cross validation* tradicional sería computacionalmente costoso. Esta relación se evidencia en la **??**, que compara las estimaciones de error *out-of-bag* y *leave-one-out cross validation* utilizando el conjunto de datos *Abalone* y variando la cantidad de árboles B. Se observa claramente

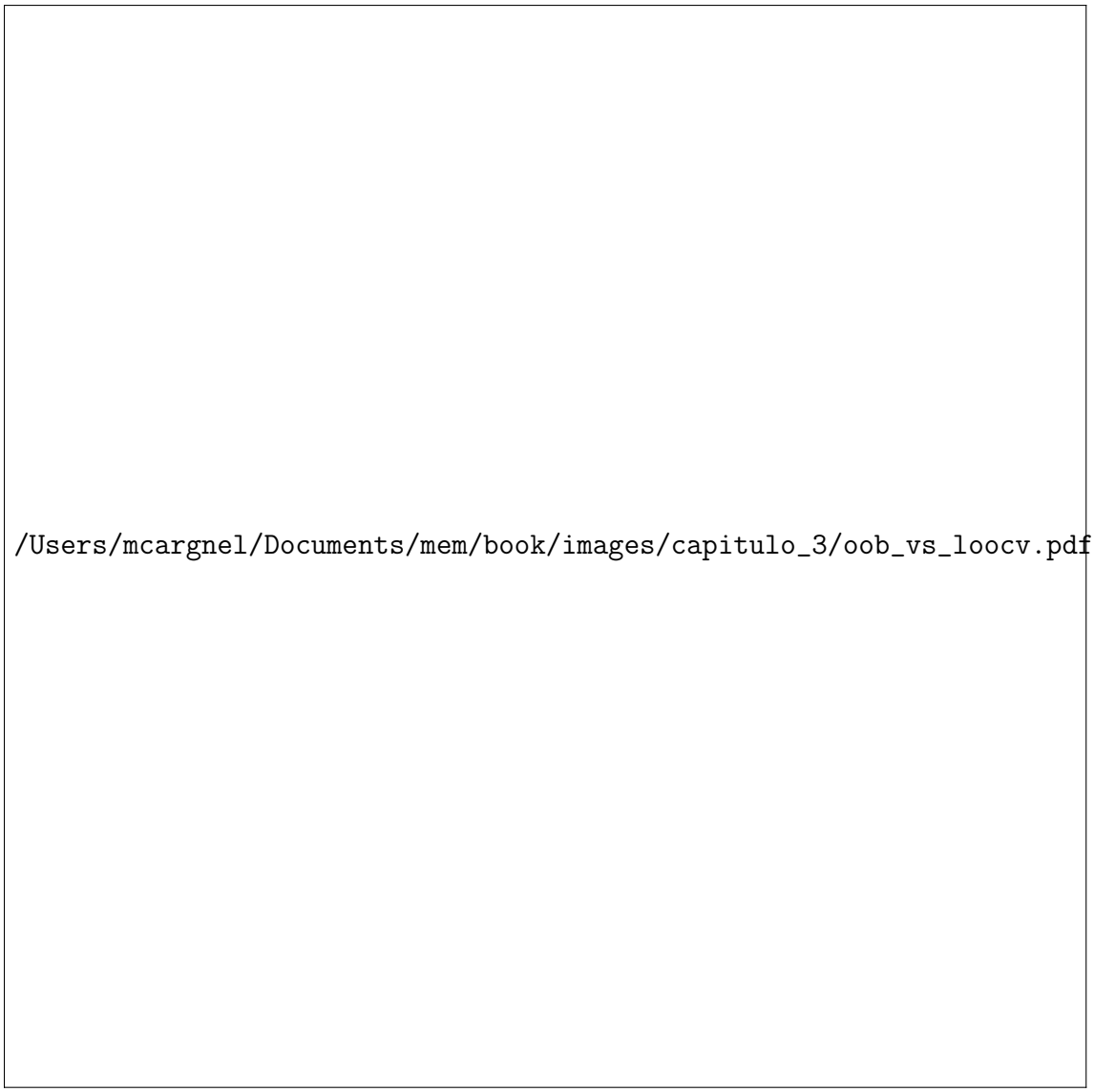

Figura 3.1: Comparación entre el error de *out-of-bag* y *leave-one-out cross validation* para modelos de *Bagging* con distintos números de árboles ajustados al conjunto de *Abalone*.

que, al incrementar el número de árboles, ambas métricas convergen hacia valores similares.

Una de las limitaciones de *Bagging* es que, al utilizar todas las variables independientes en cada árbol del modelo, puede no reducir significativamente la varianza cuando estas variables están altamente correlacionadas. Esto se debe a que los árboles generados tienden a ser similares, lo que limita el beneficio del promediado de las predicciones. Este punto puede verse en la demostración anterior, donde a menor ρ menor es el error.

3.1.1. Random Forest

Random Forest [**?**] propone una mejora con respecto a *Bagging* al tener en cuenta solo un sub-conjunto de las covariables para cada *split*. Esto permite que los árboles sean distintos entre ellos, haciendo que los árboles estén menos correlacionados entre

ellos y por lo tanto reduciendo más el error.

El parámetro que define la cantidad de variables que se tiene cuenta en cada *split* se denomina m. Cabe destacar que un *Random Forest* con $m = p$, siendo p la cantidad de covariables, es *Bagging*. Por lo que *Random Forest* puede considerarse un caso particular de *Bagging*.

Más formalmente se puede pensar el predictor *Random Forest* como

$$\hat{f}_{\text{rf}}(x) = \frac{1}{B} \sum_{b=1}^{B} T(x; \Theta_b) \tag{3.11}$$

Donde $(T(x; \Theta_b))$ representa el valor predicho por el b-ésimo árbol de decisión para una observación x, construido a partir de una muestra *bootstrap* y parámetros aleatorios Θ_b.

Es interesante notar que, si tomamos si $B \to \infty$, es decir cuando se promedian muchos árboles, por Ley de los Grandes Números se puede probar que:

$$\lim_{B \to \infty} \hat{f}_{\text{rf}}(x) = E[T(x; \Theta_b)] \tag{3.12}$$

Esto significa que, con suficientes árboles, la predicción de *Random Forest* se aproxima al valor esperado de un único árbol. La demostración formal y los detalles pueden verse en [**?**]. Sin embargo, de manera intuitiva puede verse en la **??** cómo a medida que aumenta la cantidad de árboles la variabilidad de las predicciones disminuye.

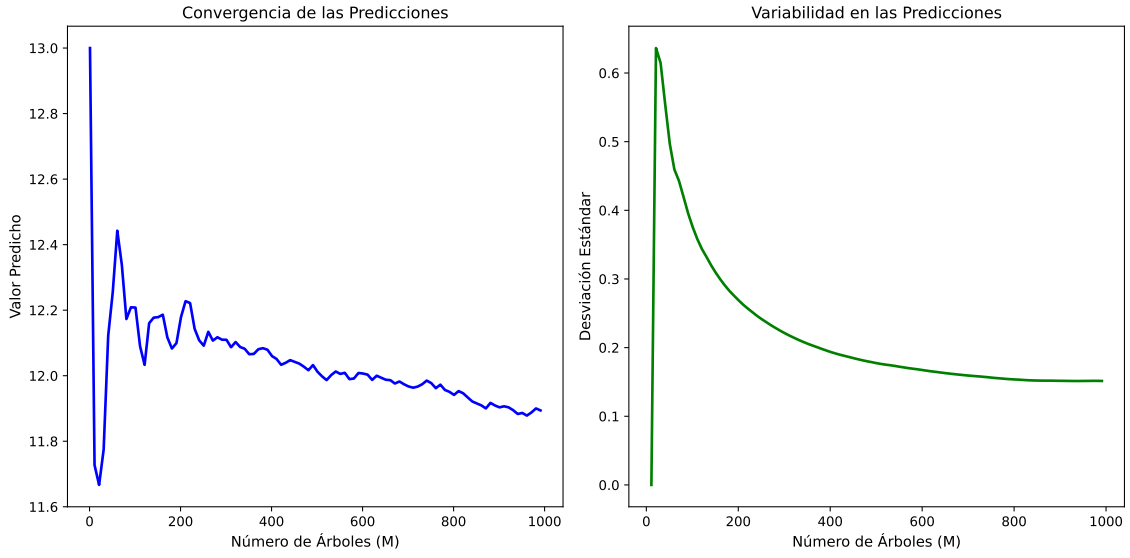

Figura 3.2: Convergencia y variabilidad de las predicciones para modelos de Random Forest con distintos números de árboles.

3.2. Boosting

A diferencia de *Bagging*, *Boosting* no promedia predicciones, sino que construye un modelo iterativamente, donde cada modelo se construye con el objetivo de corregir el error cometido por el modelo anterior. A continuación se presenta un ejemplo de

cómo funciona el algoritmo de *Gradient Boosting Machines* (*GBM*) propuesto por [**?**].

Como en cualquier problema de aprendizaje supervisado, el objetivo es estimar $\hat{f}(x)$ a partir de un conjunto de datos (X, Y). En este caso, el conjunto de datos es una muestra i.i.d. de (X, Y) de tamaño n:

$$(X, Y) = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \tag{3.13}$$

Donde x_i es un vector de covariables y y_i es la variable de respuesta, en este caso, continua. Por lo que se trata de un problema de regresión.

El algoritmo de *Gradient Boosting Machines* propone un método para encontrar un aproximación de la función $f(x)$ que minimiza una función de pérdida $L(y, f(x))$:

$$\hat{f}(x) = \arg\min_f \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \tag{3.14}$$

En definitiva, se ve que la forma de aproximar la función $f(x)$ es a través de una combinación lineal de modelos base $h(x, \theta)$:

$$\hat{f}(x) = \sum_{b=1}^{B} \gamma_b h_b(x, \theta_b) \tag{3.15}$$

3.2.1. Algoritmo

El algoritmo de *Gradient Boosting Machines* requiere un conjunto de datos (X, Y), una función de pérdida $L(y, f(x))$, un número de iteraciones B y un modelo base $h(x, \theta)$. Con esto, el algoritmo es el siguiente:

1. Inicializar $\hat{f}_0(x)$ con una constante.
2. Para cada iteración $b = 1, 2, ..., B$:
   - Calcular el gradiente de la función de pérdida y utilizar su valor negativo.
   - Ajustar el modelo base $h_b(x, \theta_b)$ a los datos $(X, g_b(X))$.
   - Encontrar el mejor tamaño de paso γ_b:

$$\gamma_b = \arg\min_\gamma \sum_{i=1}^{n} L(y_i, \hat{f}_{b-1}(x_i) + \gamma h_b(x_i, \theta_b)) \tag{3.16}$$

   - Actualizar la aproximación:

$$\hat{f}_b(x) = \hat{f}_{b-1}(x) + \gamma_b h_b(x, \theta_b) \tag{3.17}$$

3. Devolver el modelo final.

3.2.2. Funciones de pérdida

Dependiendo del problema y los datos con los que se esté trabajando, se pueden utilizar distintas funciones de pérdida. A continuación se presentan algunas de las más comunes para problemas de regresión, es decir, cuando la variable de respuesta y es continua.

Error Cuadrático Medio (MSE)

Una de las funciones de pérdida más comunes es el error cuadrático medio (MSE), que se define como:

$$L(y, f(x)) = \frac{1}{2}(y - f(x))^2 \tag{3.18}$$

Una de las ventajas que tienen este tipo de funciones de pérdida es que son diferenciables, lo cual facilita el cálculo del gradiente. Que simplemente es:

$$\begin{aligned} \frac{\partial L(y, f(x))}{\partial f(x)} &= \frac{\partial}{\partial f(x)} \frac{1}{2}(y - f(x))^2 \\ &= \frac{1}{2} 2(y - f(x)) \frac{\partial (y - f(x))}{\partial f(x)} \\ &= -(y - f(x)) = f(x) - y \end{aligned} \tag{3.19}$$

Luego, como el gradiente indica la dirección de máximo crecimiento, el negativo del gradiente indica la dirección de máximo decrecimiento. Lo cual es lo que se busca en el algoritmo de *boosting*, es decir, minimizar la función de pérdida. Por lo que el gradiente de la función de pérdida es:

$$-g_b(x) = y - f(x) \tag{3.20}$$

Lo cual es simplemente el residuo.

Alternativas robustas

Como se mencionó anteriormente, una de las ventajas de utilizar la función de pérdida MSE es que es diferenciable. Sin embargo, el hecho de que penalice los errores grandes puede ser una desventaja cuando se trabaja con *outliers*. Por lo que también se puede utilizar la función de pérdida absoluta (MAE) que se define como:

$$L(y, f(x)) = |y - f(x)| \tag{3.21}$$

ó la función de pérdida Huber que se define como:

$$L(y, f(x)) = \begin{cases} (y - f(x))^2 & \text{si } |y - f(x)| \leq \delta \\ 2\delta|y - f(x)| - \delta^2 & \text{si } |y - f(x)| > \delta \end{cases} \tag{3.22}$$

A continuación, en la **??**, se presentan las funciones de pérdida anteriores. Puede verse que MSE aumenta mucho cuando hay errores grandes, mientras que MAE y Huber son más robustas a los *outliers*. Además, Huber se comporta similar a MSE cuando el error es pequeño, pero al aumentar el error, comienza a penalizar menos.

3.2.3. Modelos base

Si bien el algoritmo de *Boosting* puede utilizar múltiples modelos base, en este trabajo se expondrán únicamente los modelos base de árboles de decisión.

Como se mencionó anteriormente, una de las ventajas de los árboles de decisión es poder capturar interacciones entre las covariables. Esta característica puede ser regulada mediante un hiperparámetro que indica la profundidad máxima de los

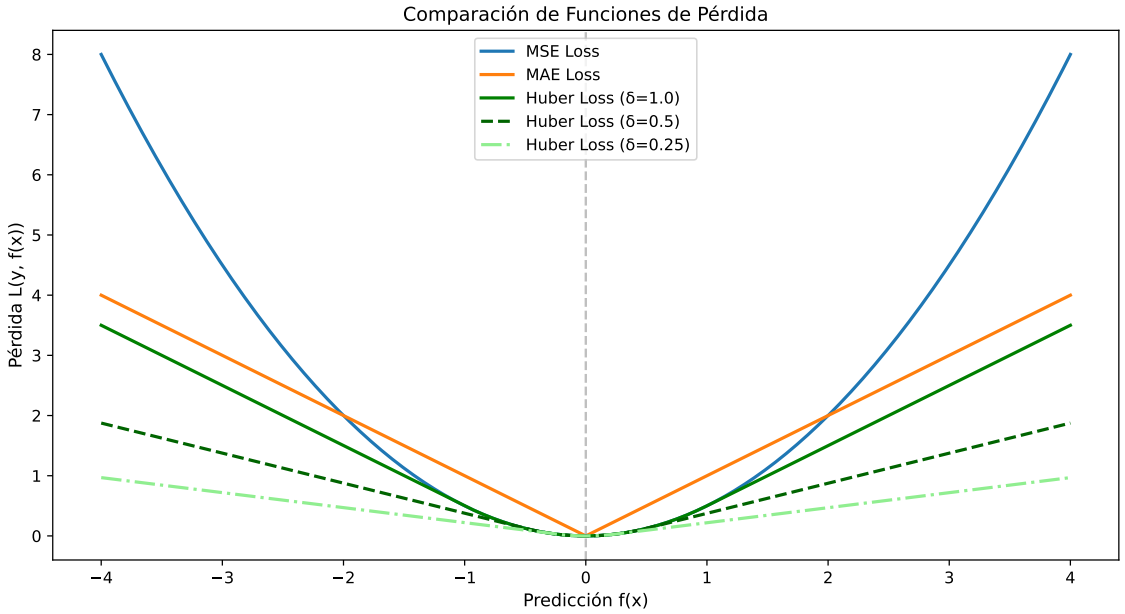

Figura 3.3: Comparación de funciones de pérdida: MSE, MAE y Huber.

árboles. En particular, la literatura sugiere utilizar árboles de poca profundidad en este tipo de modelos. Siendo los *stumps* los árboles de profundidad 1 los más utilizados.

En la **??** se puede ver el efecto de la profundidad de los árboles en el error del modelo. Puede verse que a medida que aumenta la profundidad del árbol, el error disminuye hasta llegar a aproximadamente 10, luego de lo cual el error se estabiliza.

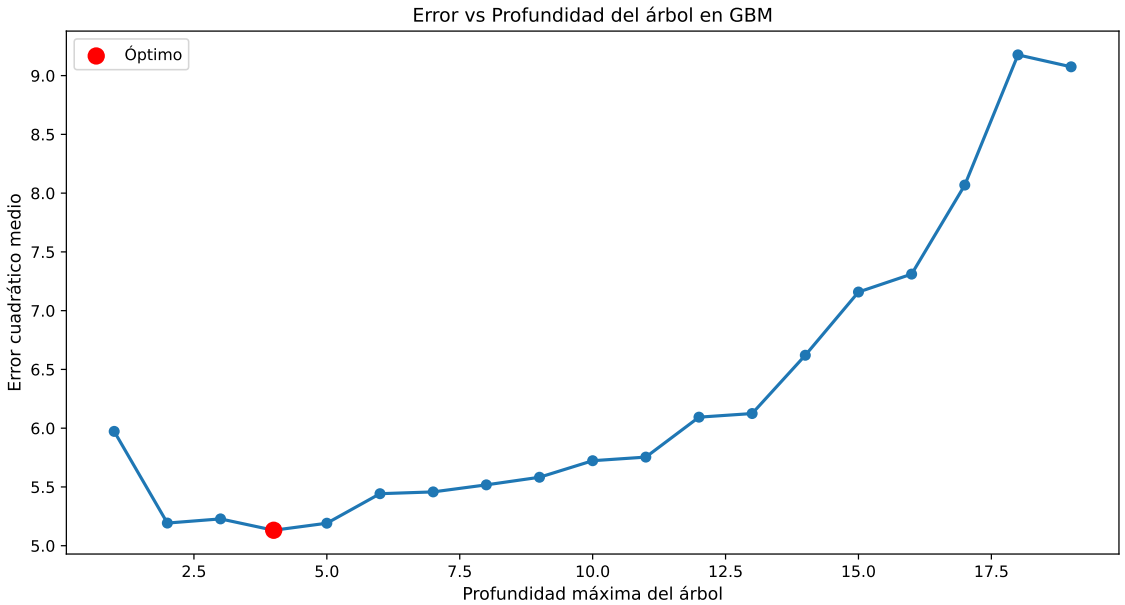

Figura 3.4: Error en función de la profundidad del árbol.

3.2.4. Regularización

Uno de los principales problemas de los modelos de aprendizaje automático es el sobreajuste. Es decir, que el modelo se ajuste demasiado a los datos de entrenamiento y por lo tanto no generalice bien a nuevos datos. Esto se da porque las funciones que se utilizan son muy flexibles y pueden llegar a ajustarse al conjunto de datos, por lo que existen determinadas técnicas para evitar esto. En este trabajo, se estudiarán tres técnicas: *subsampling*, *shrinkage* y *early stopping*.

Subsampling

La idea intuitiva detrás del *subsampling* es introducir variabilidad en el entrenamiento del modelo. Esto se logra al considerar un subconjunto de las observaciones para cada iteración del algoritmo. El sampleo puede ser con o sin reemplazo. Cabe aclarar que para definir la proporción de la muestra que se utiliza, se utiliza el hyperparámetro *bag fraction* que toma valores entre 0 y 1. Por ejemplo, si se tiene una muestra de 1000 observaciones y se utiliza un *bag fraction* de 0.1, se utilizarán 100 observaciones para cada iteración.

Pese a que esta técnica puede ayudar a en casos donde los *datasets* sean muy grandes y el costo computacional sea alto, al solo considerar una parte de las observaciones el entrenamiento puede ser menos preciso por lo que se debe balancear.

Shrinkage

Shrinkage se utiliza para reducir el efecto de los árboles individuales en el modelo. Intuitivamente, es un hiperparámetro que permite regular cuánto se toman en cuenta los errores de los árboles anteriores y se basa en que es preferible mejorar poco a poco las predicciones.

En definitiva, el shrinkage puede incluirse $\hat{f}_b(x)$ como:

$$\hat{f}_b(x) = \hat{f}_{b-1}(x) + \lambda\gamma_b h_b(x, \theta_b) \tag{3.23}$$

Donde λ es el *shrinkage* que toma valores entre 0 y 1. Lógicamente, a medida que λ se acerca a 0, los incrementos de las predicciones son cada vez más pequeños y por lo tanto el modelo es menos sensible a los errores de los árboles anteriores, alcanzando así una mejor generalización. Además, al elegir un *shrinkage* pequeño, el modelo converge más lento, por lo cual se necesita un mayor número de iteraciones B para que el modelo converja al mismo error.

En la **??** se puede ver el efecto del shrinkage en el error del modelo tanto en la muestra de entrenamiento como en la muestra de test. Puede verse que en la muestra de entrenamiento el error disminuye más rápido a medida que aumenta el shrinkage. Sin embargo, en la muestra de testeo puede verse como el *shrinkage* afecta tanto la velocidad de convergencia como la capacidad de generalización del modelo.

Early Stopping

Puede verse también en la **??** que el modelo converge a un error mínimo. Sin embargo, el modelo puede sobreajustarse a los datos de entrenamiento y por lo tanto no generalizar bien a nuevos datos. Para evitar esto, se puede utilizar el *early*

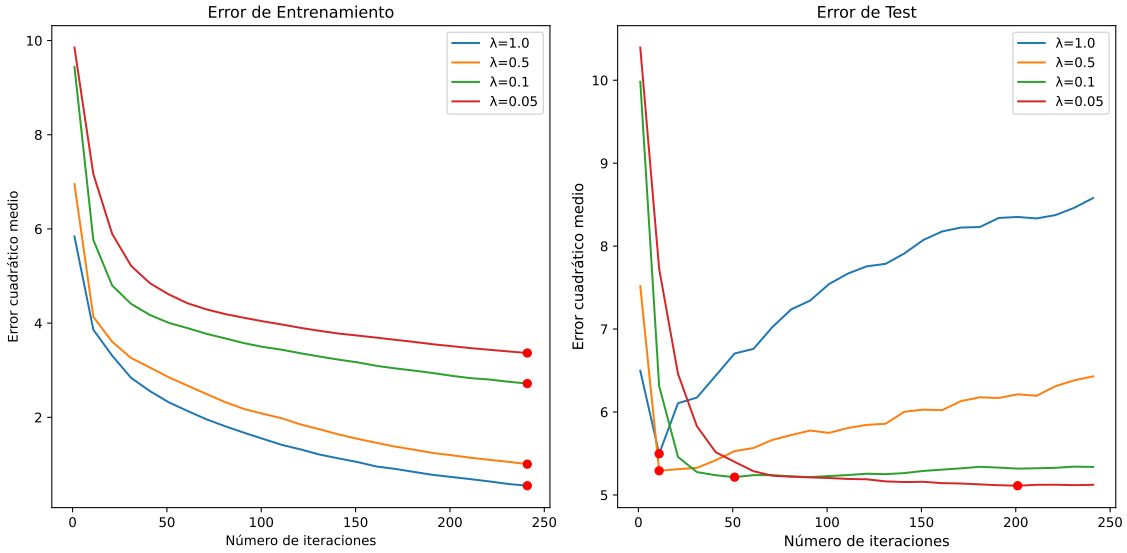

Figura 3.5: Comparación del error de testeo y entrenamiento en función de distintos valores de *shrinkage* y cantidad de árboles.

stopping que consiste en detener el entrenamiento cuando el error en la muestra de test deja de disminuir o lo hace de manera muy lenta.

De esta forma, se puede evitar el sobreajuste y reducir el tiempo de entrenamiento de estos modelos. Esto último es particularmente útil cuando se trabaja con muestras grandes.

4. Interpretabilidad

Tal como se mencionó anteriormente, uno de los *trade-off* de los ensambles de árboles es que al agrupar múltiples árboles el modelo en conjunto deja de ser interpretable. Esto puede no suponer un problema si el objetivo es predecir, pero si el objetivo es interpretar el modelo el hecho de trabajar con 'cajas negras' puede ser un problema.

Es por ello que se desarrollaron técnicas para extraer información de modelos de aprendizaje automático. Estas técnicas se aplican una vez entrenado el modelo y permiten entender cómo el mismo utiliza las variables. Pese a que existen distintas técnicas para interpretar modelos, algunas son específicas para ciertos tipos de modelos mientras que otras son agnósticas. En este trabajo se estudiará como acceder a la importancia de cada covariable y al efecto que tiene cada una de ellas sobre la variable dependiente.

4.1. Importancia de las variables

Con el acceso a información que se tiene actualmente, no es extraño querer incluir múltiples covariables en un mismo modelo. El problema, es que entender cuales de ellas realmente están aportando al mismo no es trivial. Por lo que se desarrollaron múltiples técnicas para evaluarlo, todas con sus argumentos a favor y en contra.

En este trabajo nos centramos en dos: *split gain feature importance* y *permutation feature importance*.

4.1.1. *Split Gain Feature Importance*

Esta técnica fue propuesta por [**?**] y asigna importancias en base a la cantidad de *splits* que cada variable tiene en el árbol. Esta metodología solamente se puede aplicar para árboles de decisión y más formalmente se ve para un solo árbol:

$$\hat{I}_j^2(T) = \sum_{t=1}^{N-1} \hat{i}_t^2 1(v_t = j) \tag{4.1}$$

donde I es la influencia para el nodo j del árbol T, siendo v la *splitting variable* que se suma por todos los nodos no terminales.

La ecuación **??** es calculada para un solo árbol. Por lo que cuando se trabaja con ensambles de árboles se debe generalizar a

$$\hat{\mathbf{I}}_j^2 = \frac{1}{B} \sum_{m=1}^{B} \hat{I}_j^2(T_b) \tag{4.2}$$

Intuitivamente, se puede entender como el promedio de los *splits* de cada variable. Sin embargo, esta medida esta basada en heurísticas y no contiene fundamentos estadísticos sólidos. Tal es así que fue criticada por [**?**] por dar demasiada influencia a variables correlacionadas y favorecer variables categóricas con muchas categorias.

El problema de las variables correlacionadas radica en que el modelo no puede distinguir entre aquellas variables que realmente aportan a la generación de los datos con las que simplemente correlacionan con variables influyentes, por lo que termina asignando mayor influencia a variables que no deberían tenerla.

En el caso de las variables categóricas, este método prioriza aquellas variables que son propensas a tener muchos *splits*, sin que esto signifique que las variables son importantes o estan reduciendo el error.

4.1.2. Permutation Feature Importance

Otra alternativa para calcular las importancias de las variables es utilizar *Permutation Feature Importance* (PFI), técnica desarrollada por [**?**] exclusivamente para *Random Forest* que luego fue generalizada para cualquier modelo [**?**]. Estas técnicas se basan, intuitivamente, en entender cómo cambia la función de pérdida cuando se permutan las variables. Las variables más importantes serán aquellas que más cambien el error del modelo al ser permutadas. Como se ve a continuación, si bien [**?**] propone métodos para calcular la importancia en modelos de clasificación, se presentará la adaptación a un modelo de regresión. Por lo que habiendo entrenado un modelo de *random forest* $\hat{f}$ para cada covariable $j \in \{1, 2, ..., p\}$, se construye un nuevo conjunto de datos X_j permutando las observaciones de la covariable j y dejando el resto de las observaciones fijas. Posteriormente, se entrena un nuevo modelo $\hat{f}_j$ utilizando el nuevo conjunto de datos X_j y se calcula el error *out-of-bag*:

$$E_{OOB}(X_j) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}^{(b)}(X_{i,j}) \right)^2 \tag{4.3}$$

Luego se compara el error *out-of-bag* con el error del modelo original y se calcula la diferencia mediante:

$$PFI(j) = E_{OOB}(X_j) - E_{OOB}(X) \quad \text{ó} \quad PFI(j) = \frac{E_{OOB}(X_j)}{E_{OOB}(X)} \tag{4.4}$$

Sin embargo, como se mencionó anteriormente este método para calcular importancias no es generalizable dado que utiliza los errores *out-of-bag*, por lo que un modelo que no utilice esta metodología no podría usarla. Sin embargo, es fácilmente extendible a dichos casos. A continuación se presenta la generalización propuesta en [**?**].

Los autores se basan en la propuesta de [**?**], pero deciden llamar a la medida de importancia *Model Reliance* (MR). Por lo que partiendo de un modelo entrenado $\hat{f}$, una variable de respuesta y, una función de pérdida L y una matriz de diseño X con covariables $j \in \{1, 2, ..., p\}$, para cada covariable j se permuta dicha covariable y se calcula la pérdida esperada:

$$L_{perm}(f) = \mathcal{E}[L(f, (Y, X_1^{perm}, X_2))] \tag{4.5}$$

donde se permutan los valores de X_1, dejando constantes las demás variables. Luego, se compara la pérdida con el error original:

$$MR^j(f) = \frac{L^j_{perm}(f)}{L(f)} \tag{4.6}$$

De esta forma se obtiene cuanto afecta cada covariable a la variable de respuesta.

Para calcular L_{perm} los autores proponen dos alternativas. La primera consiste en realizar una permutación por todas las observaciones:

$$\hat{e}_{perm}(F) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} L\{f, y_j, X_{1[i,.]}, X_{2[i,.]}\} \tag{4.7}$$

Sin embargo esto puede ser computacionalmente muy pesado, por lo que proponen como alternativa dividir la muestra a la mitad y reemplazar los valores de X_1 de la primera mitad con los de la segunda y viceversa:

$$\begin{aligned} \hat{e}_{\text{divide}}(f) = \frac{1}{2\lfloor n/2 \rfloor} \sum_{i=1}^{\lfloor n/2 \rfloor} \Bigg[& L\Big(f, \Big(y_{[i]}, X_{1\left[i+\lfloor n/2 \rfloor\right]}, X_{2[i]}\Big)\Big) \\ & + L\Big(f, \Big(y_{i+\lfloor n/2 \rfloor}, X_{1[i]}, X_{2\left[i+\lfloor n/2 \rfloor\right]}\Big)\Big)\Bigg]. \end{aligned} \tag{4.8}$$

4.1.3. Model Class Reliance

Un problema común a la hora de calcular importancias es que el modelo que terminamos usando se decide únicamente en función de una métrica de error. Esto no supondría inconvenientes si las importancias de los modelos fueran parecidas. Sin embargo, suele pasar que para muchos modelos con un poder predictivo similar las importancias de las variables cambien. A continuación se presenta un ejemplo en el cual se entrenaron múltiples *Gradient Boosting Machines* con diferentes combinaciones de hiperparámetros, eligiendo las combinaciones que minimicen el error cuadrático medio. Para el elegir dicha combinación se utilizó validación cruzada con $k = 5$. El conjunto de datos para este ejemplo en particular fue *Abalone* [**?**] donde se busca predecir la edad del Abalone en función de distintas características.

Específicamente, para este ejemplo se entrenaron modelos con cantidad de árboles de 100 a 900 (con incrementos de 100), tasas de aprendizaje de 0.01, 0.05 y 0.1, profundidad máxima de 2, 3, 4 y 5, y funciones de pérdida de tipo pérdida cuadrática, error absoluto y Huber. Siendo un total de 324 modelos entrenados. En la **??** se pueden ver los 15 modelos que menor error tuvieron, junto con los hiperparámetros asociados. Puede verse claramente, como hay múltiples modelos que tienen una performance predictiva similar pero distintos hiperparámetros.

Además, y posiblemente más preocupante, cuando vemos la Figura **??** vemos como las importancias de las variables (calculadas mediante *Permutation Feature Importance*) varían entre modelos con un poder predictivo similar. Este mismo fenómeno puede verse también en la Figura **??** donde para los distintos modelos las variables tienen rankings de importancia muy diferentes. El heatmap muestra claramente cómo una misma variable puede ser considerada la más importante (ranking 1) en un modelo, mientras que en otro puede tener una importancia mucho menor (ranking 5 o 6). Esta inconsistencia en la importancia de las variables entre modelos con rendimiento similar sugiere inestabilidad en la interpretación de los resultados a los que se puede llegar.

Cuadro 4.1: Tabla con resultados de los 15 modelos con menor error de predicciones, junto con los parámetros seleccionados.

Modelo	RMSE	Shrinkage	Función de Pérdida	Profundidad máx	Árboles
RF1	2.1488	0.10	Huber	3	200
RF2	2.1498	0.05	Huber	4	200
RF3	2.1498	0.01	Huber	4	900
RF4	2.1506	0.10	Huber	4	100
RF5	2.1513	0.01	Pérdida cuadrática	4	600
RF6	2.1516	0.05	Pérdida cuadrática	4	200
RF7	2.1518	0.05	Huber	3	300
RF8	2.1518	0.01	Pérdida cuadrática	4	700
RF9	2.1519	0.05	Huber	3	400
RF10	2.1522	0.01	Pérdida cuadrática	4	800
RF11	2.1522	0.01	Huber	4	800
RF12	2.1526	0.05	Pérdida cuadrática	4	100
RF13	2.1530	0.01	Pérdida cuadrática	4	500
RF14	2.1531	0.10	Pérdida cuadrática	4	100
RF15	2.1537	0.01	Pérdida cuadrática	4	900

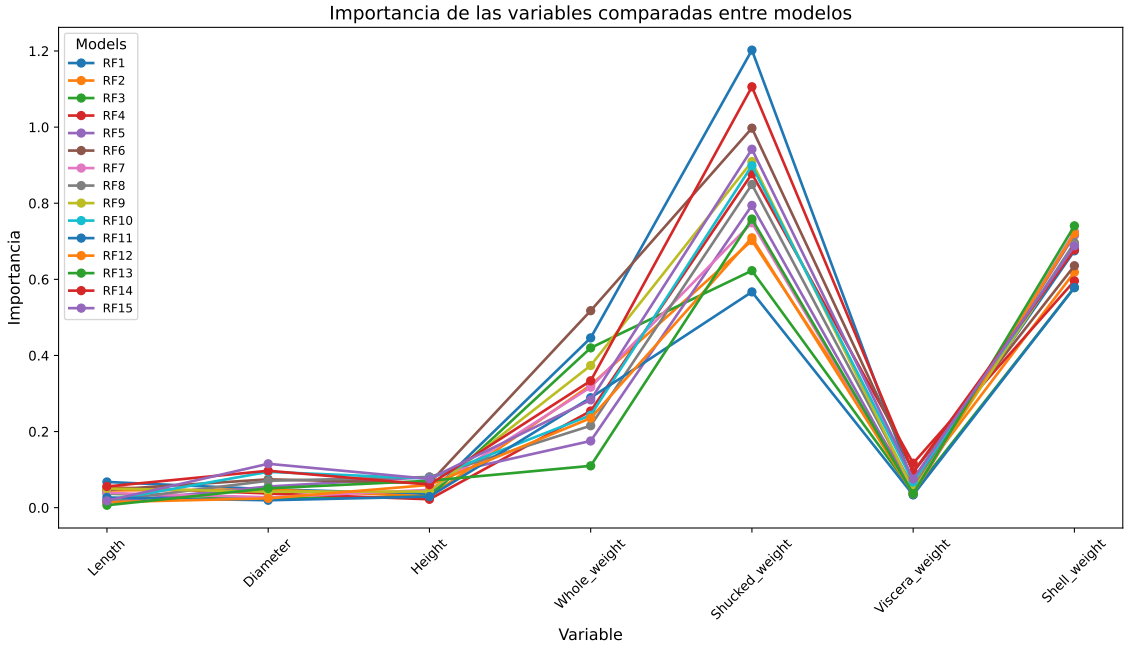

Figura 4.1: Importancias de las variables para distintos modelos de Gradient Boosting con rendimiento similar.

El fenómeno observado, donde múltiples modelos con similar capacidad predictiva asignan importancias sustancialmente diferentes a las variables, se conoce como el *Rashomon Effect* en estadística [**?**]. Este término, inspirado en la película de Kurosawa donde un mismo evento es narrado desde perspectivas contradictorias es formalizado en [**?**] como el çonjunto de Rashomon", definido como la colección de modelos con rendimiento predictivo similar pero con diferentes estructuras internas. Esta multiplicidad de explicaciones válidas representa un desafío fundamental para

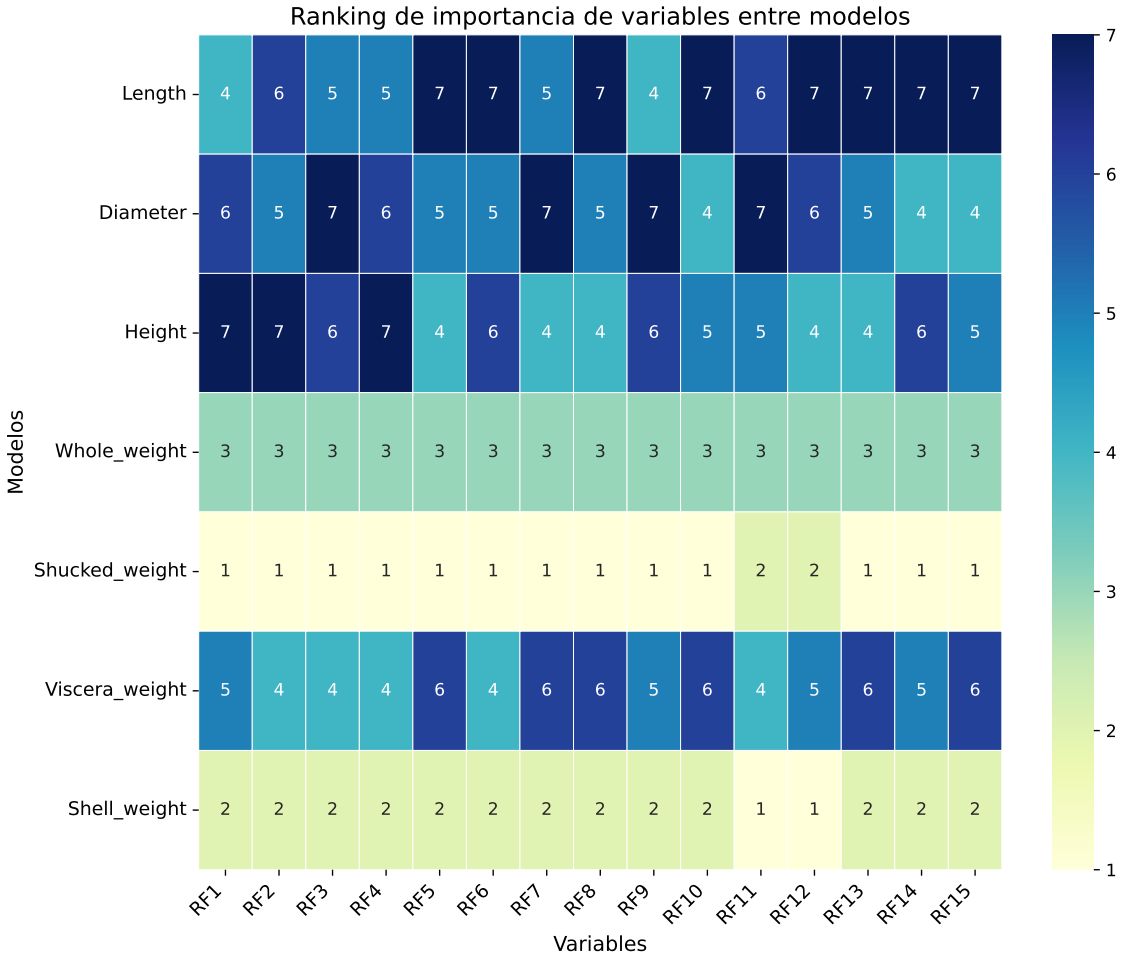

Figura 4.2: Mapa de calor de las importancias de las variables para distintos modelos de Gradient Boosting con rendimiento similar.

la interpretabilidad en *machine learning*, ya que sugiere que la importancia de una variable no es una propiedad intrínseca de los datos, sino que depende del modelo específico elegido.

Es por ello que los autores trabajaron sobre la *Model Class Reliance* (MCR), que se basa en para una clase de modelos $\mathcal{F}$, MCR es el rango de todos los posibles valores de MR sobre modelos que tengan un poder predictivo similar al óptimo. Para lo cual, primero se define el çonjunto de Rashomonçomo

$$R(\epsilon) = \{f \in \mathcal{F} \mid e_{orig}(f) \leq e_{orig}(f_{ref}) + \epsilon\} \tag{4.9}$$

Este set contiene todos los modes cuya pérdida sea como mucho ϵ peor que el modelo que minimiza el error f_{ref}

Luego se define el intervalo MCR como

$$[MCR^{-}(\epsilon), MCR^{+}(\epsilon)] = [\min_{f \in R(\epsilon)} MR(f), \max_{f \in R(\epsilon)} MR(f)] \tag{4.10}$$

Este intervalo MCR proporciona información valiosa sobre la importancia de una variable en toda la clase de modelos.

Interpretación

Si $MCR^{-}(\epsilon)$ es mayor que 1, significa que todos los modelos "buenos"(dentro del conjunto de Rashomon) dependen de la variable más que del modelo de referencia, lo que sugiere que la variable es consistentemente importante. Por otro lado, si $MCR^{+}(\epsilon)$ es cercano a 1, indica que ningún modelo "bueno"depende fuertemente de esa variable.

La interpretación del MCR nos permite evaluar si la importancia de una variable es consistente a través de diferentes modelos o si es específica de una arquitectura particular. Un intervalo MCR estrecho sugiere que la importancia de la variable es estable en todos los modelos con buen rendimiento, mientras que un intervalo amplio indica que la importancia varía significativamente dependiendo del modelo elegido. Por ejemplo, pueden ocurrir casos donde para una variable J el $MCR^{-}(\epsilon)$ sea cercano a 1 pero $MCR^{+}(\epsilon)$ no. Esto indicaría existen modelos con buena capacidad predictiva que no dependen de la variable J por lo que dicha variable no sería tan importante.

Estimación

En la práctica para estimar MCR se utiliza un estimador *plug-in*, es decir se calcula $\hat{MR}(f)$ para todos los modelos que integren el conjunto Rashomon empírico $\hat{R}(\epsilon)$.

4.2. Efecto de las variables

Otra técnica útil para interpretar modelos de árboles es mediante *Partial Dependence Plots*. Los mismos también fueron introducidos por [**?**], donde los plantea como una técnica para aislar el efecto de una covariable sobre la variable de respuesta, más formalmente lo que se busca estimar es:

$$\bar{f}(X_S) = \mathcal{E}_{X_C}[\hat{f}(x)] = \int \hat{f}(X_S, X_C) p(X_C) dX_C \tag{4.11}$$

donde $\hat{f}(x)$ es el modelo entrenado, X_S es el subset de variables a las que queremos estimar el efecto, X_C representa todas las otras predicciones (que se mantienen constantes), $p(X_C)$ es la distribución conjunta de todas las otras variables y $\mathcal{E}_{X_C}[.]$ es la esperanza sobre la distribución X_C

Sin embargo, en la práctica es raro conocer $p(X_C)$ por lo que se utiliza el siguiente procedimiento:

$$\bar{f}(X_S) \approx \frac{1}{n} \sum_{i=1}^{n} \hat{f}(X_S, x_C^i) \tag{4.12}$$

Cuyo algoritmo sería, para cada observación i en la muestra, tomar los valores originales de las variables no seleccionadas x_C^i. Posteriormente, se utilizan los valores modificados de las variables X_S en cada x_C^i dentro del modelo $\hat{f}$ para obtener las predicciones $\hat{f}(X_S, x_C^i)$. Finalmente, se promedian las predicciones para $i = 1, 2, ..., n$.

Cabe aclarar que al modificar solamente una variable y dejar el resto como estaban, se esta asumiendo que las variables X_S son independientes a todo el resto.

Algo que puede o no ser cierto dependiendo de la naturaleza de los datos. Además, al cambiar los valores de las variables a evaluar se podrían dar combinaciones poco realistas en la practica.

Con el fin de ejemplificar los gráficos de dependencia parcial se entrenó un modelo de *Gradient Boosting Machine* con 100 árboles con el fin de predecir la edad del *Abalone* en función de distintas medidas físicas. Una vez entrenado el modelo, se puede ver en la **??** el efecto del peso total en la edad.

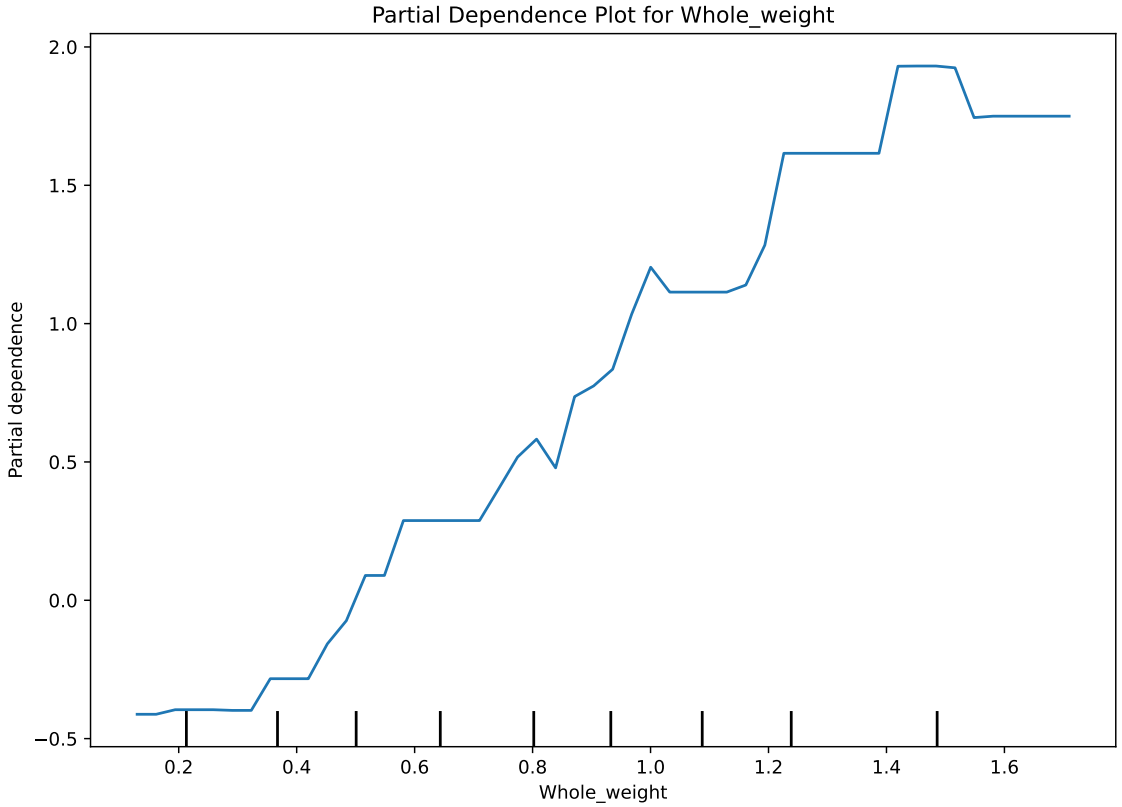

Figura 4.3: Gráfico de dependencia parcial para el peso total en la predicción de la edad del Abalone.

Sin embargo, como se mencionó anteriormente, los PDPs sufren problemas cuando las variables no son independientes. Para solventar esto, se puede utilizar el método de *Individual Conditional Expectation* (ICE) que consiste en calcular el PDP para cada observación en la muestra.

4.2.1. *Individual Conditional Expectation* (ICE)

Los ICE fueron introducidos por [**?**] como una técnica para estimar el efecto de una covariable sobre la variable de respuesta, más formalmente lo que se busca estimar es:

$$f_i(X_S) = E[Y|X_S = x_S, X_C = x_C^i] \tag{4.13}$$

Es decir, en lugar de calcular el efecto promedio de X_S sobre y calculando el promedio de $f(X_S, X_C)$ para todas las observaciones, se calcula el efecto de X_S para cada observación i manteniendo el resto de las variables constantes. De esta manera

se puede obtener una estimación del efecto de X_S para cada observación y no solo un promedio, teniendo así una visión más detallada.

De esta forma, se podrían entender los PDPs como el promedio de los ICE. Dado que mientras el primero se calcula mediante la distribución marginal de las variables, los segundos lo hacen utilizando la distribución condicional.

Cabe destacar que PDP y ICE deberían ser similares cuando las variables son independientes, pero cuando las variables no son independientes, los ICE pueden variar mucho por lo que se recomienda utilizar las dos técnicas.

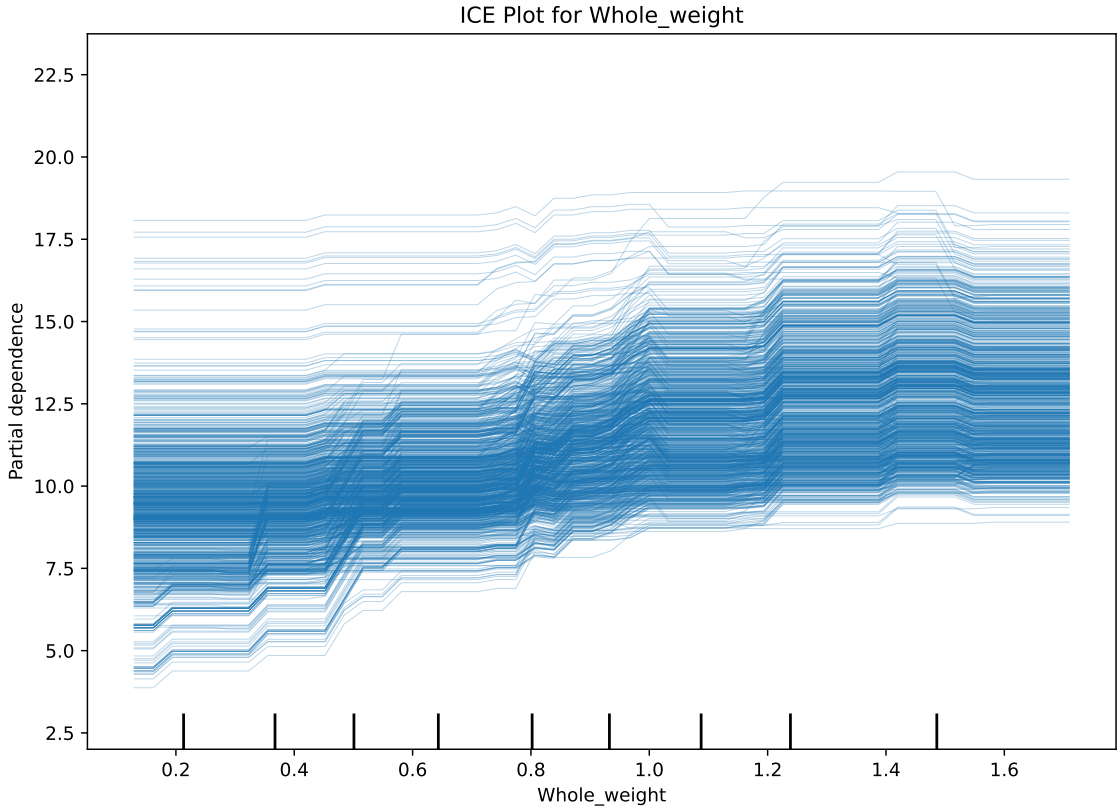

Figura 4.4: Gráfico de *Individual Conditional Expectation* (ICE) para el peso total en la predicción de la edad del Abalone.

Como se puede ver en la **??**, los ICE son mucho más variables que los PDP, lo que permite tener una visión más detallada del efecto de la variable en cuestión. Sin embargo, se puede ver que los ICE tienen distintas ordenadas al origen, lo que dificulta la comparación entre distintas variables y la identificación de efectos heterogéneos.

Para solventar este problema, se puede utilizar el método de *Centered ICE* que consiste en centrar los ICE en algún punto de la distribución de la variable (los autores de este método sugieren el valor mínimo o máximo de la variable para tener gráficos sencillos de interpretar). Más formalmente se busca centrar cada curva $\hat{f}_i$ en un valor de referencia (x^*, x_{Ci}) tal que:

$$\hat{f}_{cent}^{(i)}(x_S) = \hat{f}(x_S, x_{Ci}) - \hat{f}(x^*, x_{Ci}) \tag{4.14}$$

A modo de ejemplo, si x^* es el valor mínimo de x_S todas las curvas empiezan en cero, por otro lado si es el máximo se puede ver el efecto acumulado de x_S sobre $\hat{f}$

relativo al caso base.

A modo de ejemplo, en la **??** se puede ver el *Centered ICE* para la misma variable de antes. En este caso x^* es el valor mínimo por lo que todas las curvas empiezan en cero. También, se puede ver en rojo la curva de PDP centrada.

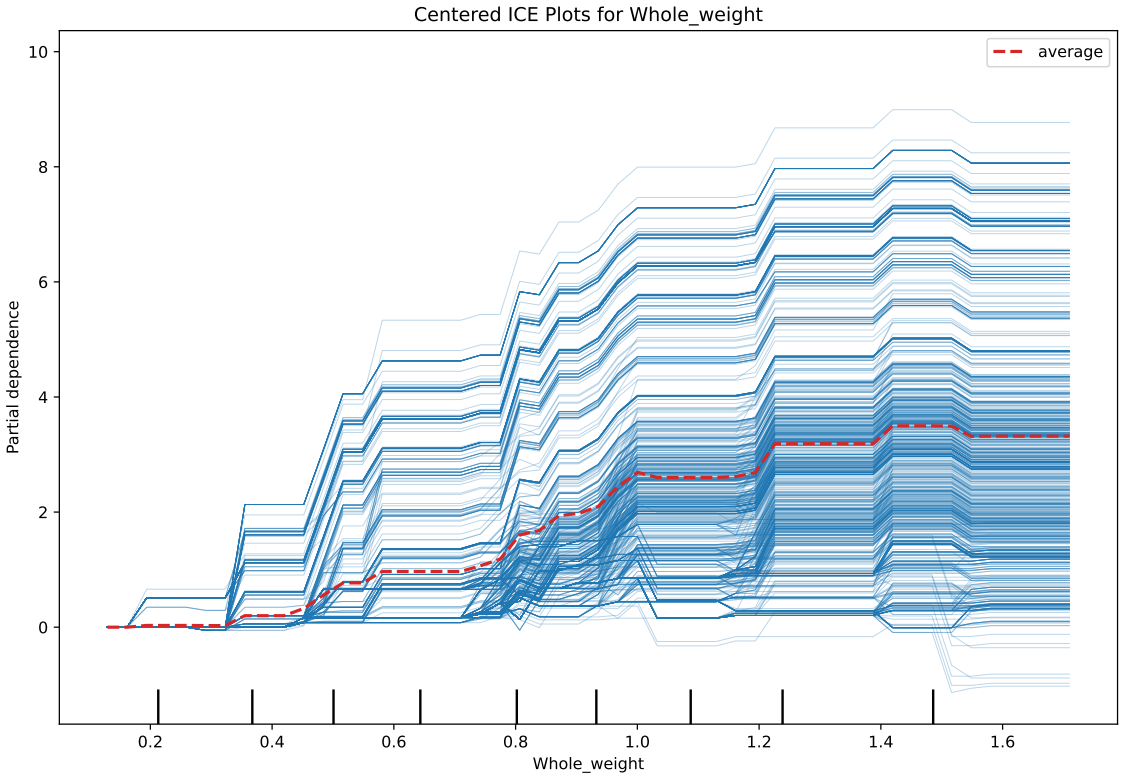

Figura 4.5: Gráfico de *Individual Conditional Expectation* (ICE) centrado para el peso total en la predicción de la edad del Abalone.

4.2.2. *Derivative ICE*

Con el fin de seguir identificando las interacciones entre las variables los autores proponen calcular los gráficos de la derivada parcial de $\hat{f}$ con respecto a x_S. A modo de formalizarlo, se asume independencia entre x_S y el resto de las covariables, por lo que $\hat{f}$ puede ser escrita como:

$$\hat{f}(x) = \hat{f}(x_S, x_C) = g(x_S) + h(x_C) \Rightarrow \frac{\partial \hat{f}(x)}{\partial x_S} = g'(x_S) \tag{4.15}$$

lo cual significa que la relación entre x_S y $\hat{f}$ no depende de x_C. Cuando esto se cumple todas las líneas deberían ser equivalentes.

4.3. Extensiones

4.3.1. Causalidad

Tal como señala [**?**], los modelos de aprendizaje automático convencionales se basan en asociaciones estadísticas y no suelen emplearse para la inferencia causal,

ya que no contemplan intervenciones ni escenarios contrafactuales. No obstante, resulta de gran interés analizar cómo, bajo ciertos supuestos, algunas técnicas de interpretabilidad pueden admitir una lectura causal. Este es el eje central del trabajo de [**?**], donde se demuestra que los PDPs y los ICEs poseen una interpretación causal bajo condiciones específicas.

Los autores se basan en que **??** puede expresarse como el *backdoor adjustment* definido por [**?**] como

$$P(Y|do(X_S = x_S)) = \int P(Y|X_S = x_S, X_C = x_C)dP(x_C) \tag{4.16}$$

donde $P(Y|do(X_S = x_S))$ se refiere a la distribución de Y luego de hacer un intervención en X_S. Esta formula permite identificar el efecto causal de X_S en Y siempre y cuando se cumpla cuando (1) ningún nodo X_C sea dependiente descendiente de X_S y (2) que X_C 'bloquee' cualquier camino "back-door.entre X_S e Y que se puede pensar intuitivamente como una causa común entre X_S e Y que impide entender el efecto causal de X_S.

$$E[Y|do(X_S = x_S)] = \int E[Y|X_S = x_S, X_C = x_C]dP(x_C) \tag{4.17}$$

Ahora si partimos de $\hat{f}(x)$ la función aprendida por el modelo, que aproxima la expectativa condicional, es decir,

$$\hat{f}(x) \approx E[Y \mid X = x]. \tag{4.18}$$

asumiendo que el modelo está bien calibrado, es decir,

$$\hat{f}(x_S, x_C) = E[Y \mid X_S = x_S, X_C = x_C], \tag{4.19}$$

podemos vincular ambas expresiones de la siguiente forma:

$$\begin{aligned} \text{PDP}(x_S) &= \int \hat{f}(x_S, x_C)\, dP(x_C) \\ &= \int E[Y \mid X_S = x_S, X_C = x_C]\, dP(x_C) \\ &= E[Y \mid do(X_S = x_S)] \end{aligned} \tag{4.20}$$

por lo que puede verse que **??** y **??** son equivalentes si condicionando en el set C es complementario al set S.

Sin embargo, existen otros *frameworks* que se centran expresamente en la inferencia causal, como el *Double Machine Learning* (DML) [**?**] y los *Causal Forests* [**?**]. El DML utiliza modelos de aprendizaje automático para estimar los *nuisance parameters* que relacionan tanto el tratamiento como el resultado con los *confounders*. Mediante el uso de *cross-fitting*, este enfoque permite obtener estimadores del efecto causal con propiedades estadísticas deseables, como la normalidad asintótica, incluso cuando se utilizan modelos de alta complejidad como ensambles de arboles o redes neuronales. Por otro lado, los *Causal Forests* extienden los *Random Forests* para la estimación de efectos de tratamiento heterogéneos (CATE). En lugar de centrarse en la predicción del resultado observado, este método adapta la función de pérdida para maximizar la varianza de los efectos del tratamiento estimados a través de las

particiones. Esto permite aislar subgrupos con respuestas diferenciales a la intervención, capturando así la estructura local de la heterogeneidad causal. Como puede verse, estos enfoques no incluyen las técnicas de interpretación de variables clásicas de los modelos de aprendizaje automático, por lo que su estudio en profundidad queda fuera del alcance de este trabajo.

4.3.2. Relación con el proceso generador de los datos

Es interesante considerar que, aunque los modelos de aprendizaje automático suelen superar en rendimiento predictivo a los métodos estadísticos convencionales, estos últimos permiten establecer un vínculo directo entre los parámetros del modelo y las propiedades del proceso generador de datos (DGP). Esta capacidad resulta fundamental para comprender en profundidad la naturaleza subyacente de la población.

En [**?**] se propone un marco estadístico que relaciona formalmente las técnicas de interpretación basadas en *Partial Dependence Plots* (PDP) y *Permutation Feature Importance* (PFI) con el DGP. En este contexto, se introducen los conceptos de DGP-PD y DGP-PFI, definidos respectivamente como

$$DGP\text{-}PD(x) = E_{X_C}\left[f(x, X_C)\right] \tag{4.21}$$

y

$$DGP\text{-}PFI = E_{\bar{X}_S, X_C, Y}\left[L\left(Y, f(\bar{X}_S, X_C)\right)\right] - E_{X,Y}\left[L\left(Y, f(X)\right)\right] \tag{4.22}$$

donde $f(x) = E[Y|X = x]$ es la función de esperanza condicional. Los autores realizan una descomposición del error de estos estimadores en términos de sesgo y varianza, identificando fuentes de error tales como la aproximación de Monte Carlo para el cálculo de las esperanzas, la posible especificación incorrecta del modelo y la variabilidad inherente al proceso de entrenamiento.

Para capturar la incertidumbre debida a la variabilidad en el entrenamiento, se introduce la metodología *learner-PD/PFI*, la cual promedia los resultados obtenidos a partir de múltiples modelos reajustados. Esto permite incorporar la variabilidad del aprendizaje en la estimación y, en consecuencia, obtener intervalos de confianza que reflejen de manera más realista la incertidumbre de las medidas de interpretación.

No obstante, esta metodología asume condiciones fuertes, como la insesgadez del modelo. Además, los intervalos de confianza pueden resultar demasiado estrechos debido al solapamiento en el remuestreo utilizado para reentrenar los modelos, y la dependencia entre variables puede afectar los estimadores.

5. Aplicaciones

En esta sección se presentan tres casos de estudio para poner a prueba las técnicas descriptas: *Airfoil Self-Noise*, *Concrete Comprehensive Strength* y *Wine Quality*. El análisis de cada conjunto de datos comienza con una breve exploración y limpieza, seguida de una búsqueda de hiperparámetros (*grid search*) para minimizar el error de validación cruzada en el subconjunto de entrenamiento (80 % de los datos). Una vez optimizados, los modelos se evalúan en el conjunto de prueba restante para finalmente interpretar sus resultados mediante *Permutation Feature Importance* (PFI) y *Partial Dependence Plots* (PDP).

Se comparan cuatro tipos de modelos: regresión lineal, árboles de decisión, *Random Forest* y *Gradient Boosting Machines*. Su rendimiento se mide a través del Error Cuadrático Medio (RMSE), el Error Absoluto Medio (MAE) y el Coeficiente de Determinación (R^2).

La configuración de hiperparámetros explorada para los árboles de decisión incluye variaciones en la profundidad máxima (10, 20, 30) y los criterios de división (mínimo de observaciones para dividir: 2, 5, 10; mínimo por hoja: 1, 2, 4). Para los métodos de ensamble, *Random Forest* y *Gradient Boosting*, se ajusta el número de estimadores (de 100 a 2000), sumando en este último la tasa de aprendizaje o *learning rate* (0.01, 0.1, 0.2).

Por último, se analiza la estabilidad de la importancia de variables (PFI) entre los cinco mejores modelos de cada caso. La consistencia en estos resultados sugiere robustez en la interpretación; de lo contrario, podríamos estar ante la presencia de *Rashomon Sets*, donde modelos con capacidades predictivas similares ofrecen explicaciones contradictorias del fenómeno. El objetivo no es necesariamente obtener el modelo perfecto, sino ilustrar la aplicación práctica de las metodologías de interpretabilidad discutidas.

5.1. *Airfoil Self-Noise*

El primer conjunto de datos proviene de la NASA y describe pruebas aerodinámicas y acústicas realizadas en un túnel de viento anecoico sobre diferentes secciones de palas [**?**]. El objetivo es predecir la presión sonora escalada en decibelios (*scaled-sound-pressure*) a partir de características físicas y operativas. La muestra consta de 1503 observaciones completas (sin valores faltantes) que incluyen perfiles NACA 0012 de distintos tamaños sometidos a diversas velocidades y ángulos de ataque. Las variables disponibles se detallan en la **??**.

El análisis de correlación inicial (**??**) reveló una fuerte asociación entre el espesor de desplazamiento (*suction-side-displacement-thickness*) y el ángulo de ataque (*attack-angle*). Dado que este último también presentaba una correlación conside-

Cuadro 5.1: Variables del conjunto de datos *Airfoil Self-Noise* y su significado.

Variable	**Significado**
frequency	Frecuencia en hertzios
attack-angle	Ángulo de ataque en grados
chord-length	Longitud de cuerda en metros
free-stream-velocity	Velocidad de flujo libre en metros por segundo
suction-side-displacement-thickness	Espesor de desplazamiento del lado de succión en metros
scaled-sound-pressure	Presión sonora escalada en decibelios (var dep)

rable con la longitud de cuerda (*chord-length*) pero menor influencia lineal sobre la variable objetivo, se optó por excluirlo del modelado para reducir la multicolinealidad.

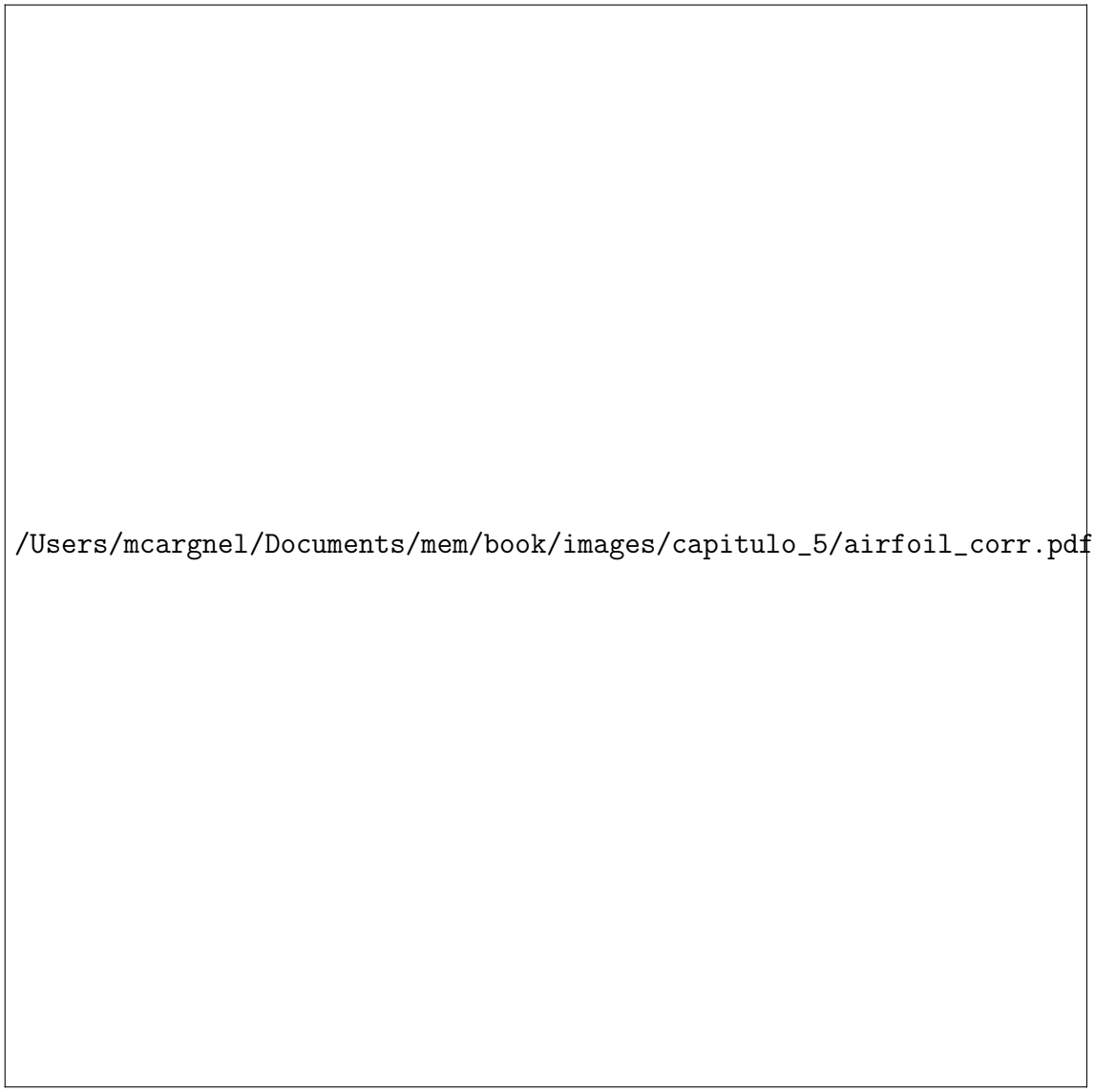

Figura 5.1: Correlación de las variables del conjunto de datos *Airfoil*.

Tras el entrenamiento, los resultados en el conjunto de prueba (**??**) muestran que *Gradient Boosting* supera significativamente a los demás modelos en todas las

métricas evaluadas. En contraste, la regresión lineal presenta el peor ajuste, un patrón que, como se verá más adelante, se repite en los tres casos de estudio, lo cual es esperable dada la complejidad y posibles no linealidades de los fenómenos físicos estudiados.

Cuadro 5.2: Resultados del mejor modelo de cada tipo evaluados en el conjunto de testeo para *Airfoil*.

Modelo	RMSE	MAE	R^2
Regresión Lineal	4.9766	3.9174	0.5056
Árbol de decisión	2.2708	1.7131	0.8971
Random Forest	1.8258	1.3177	0.9335
Gradient Boosting	1.4761	1.0337	0.9565

Para comprender qué factores impulsan las predicciones del modelo de *Gradient Boosting*, se recurre a la importancia de variables por permutación (PFI). La **??** señala a la frecuencia y el espesor de desplazamiento como los predictores dominantes, seguidos por la longitud de cuerda y la velocidad de flujo libre. Es relevante notar que los cinco modelos con mejor desempeño coincidieron en este orden de importancia, otorgando robustez a esta conclusión (**??**).

Profundizando en la relación entre las variables más importantes y la presión sonora, los gráficos de dependencia parcial (**??**) ofrecen una visión detallada. En el caso de la frecuencia, se observa una relación inversa clara: los niveles de presión sonora son máximos (130–140 dB) en el rango de bajas frecuencias (0–2500 Hz) y descienden progresivamente hasta los 100–110 dB cerca de los 20000 Hz. El análisis centrado corrobora este efecto, mostrando variaciones de hasta ±20 dB respecto a la media. No obstante, la escasez de observaciones por encima de los 7500 Hz sugiere interpretar con cautela las predicciones en ese extremo del espectro.

Por su parte, el espesor de desplazamiento muestra un comportamiento no lineal más complejo. Aunque la presión sonora fluctúa principalmente entre 120 y 130 dB, la curva de respuesta presenta múltiples discontinuidades y una leve tendencia descendente general. Al igual que con la frecuencia, la distribución de los datos es asimétrica, concentrándose la gran mayoría de las observaciones por debajo de 0.015 metros.

En síntesis, el análisis de este conjunto de datos demuestra la superioridad predictiva de los métodos de ensamble y permite identificar claramente a la frecuencia y el espesor de desplazamiento como los factores acústicos determinantes.

5.2. *Concrete Compressive Strength*

El segundo caso de estudio utiliza el conjunto de datos *Concrete Compressive Strength* [**?**]. Este consta de 1030 observaciones sobre la composición del concreto y su edad, con el fin de determinar su resistencia a la compresión. En la **??** se describen las 9 variables involucradas:

A diferencia del caso anterior, el análisis de correlación (**??**) no mostró coeficientes superiores a 0.8 en valor absoluto, por lo que se decidió conservar todas las variables predictoras para el modelado.

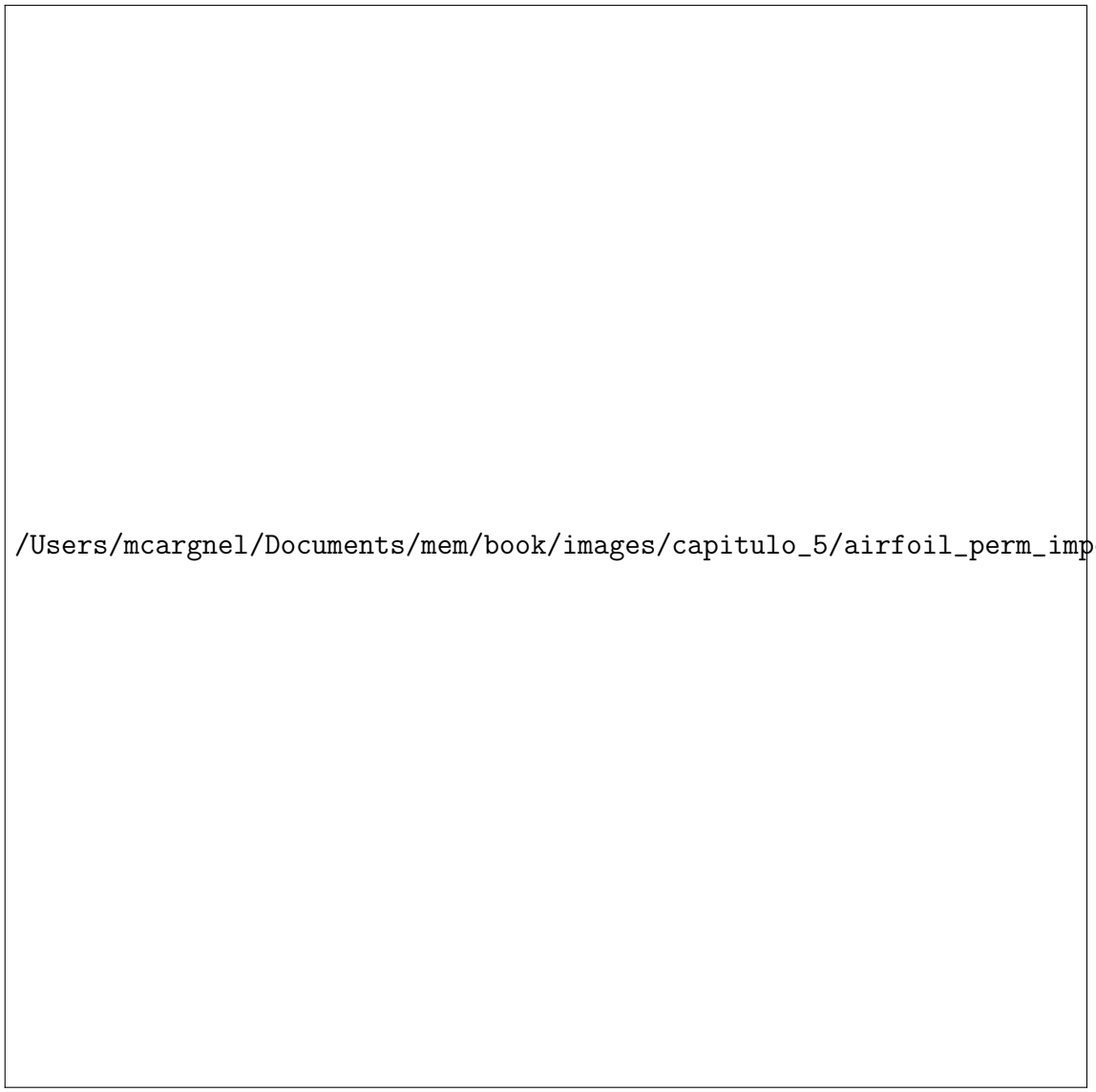

Figura 5.2: Importancia de variables (PFI) para el modelo de *Gradient Boosting* en el conjunto *Airfoil*.

Cuadro 5.3: Variables del conjunto de datos *Concrete Compressive Strength* y su significado.

Variable	**Significado**
Cemento	Cantidad de cemento en la mezcla (kg/m^3)
Escoria de alto horno	Cantidad de escoria de alto horno (kg/m^3)
Cenizas volantes	Cantidad de cenizas volantes (kg/m^3)
Agua	Cantidad de agua en la mezcla (kg/m^3)
Superplastificante	Cantidad de superplastificante (kg/m^3)
Agregado grueso	Cantidad de agregado grueso (kg/m^3)
Agregado fino	Cantidad de agregado fino (kg/m^3)
Edad	Edad del concreto en días
Resistencia	Resistencia a la compresión del concreto (var dep)

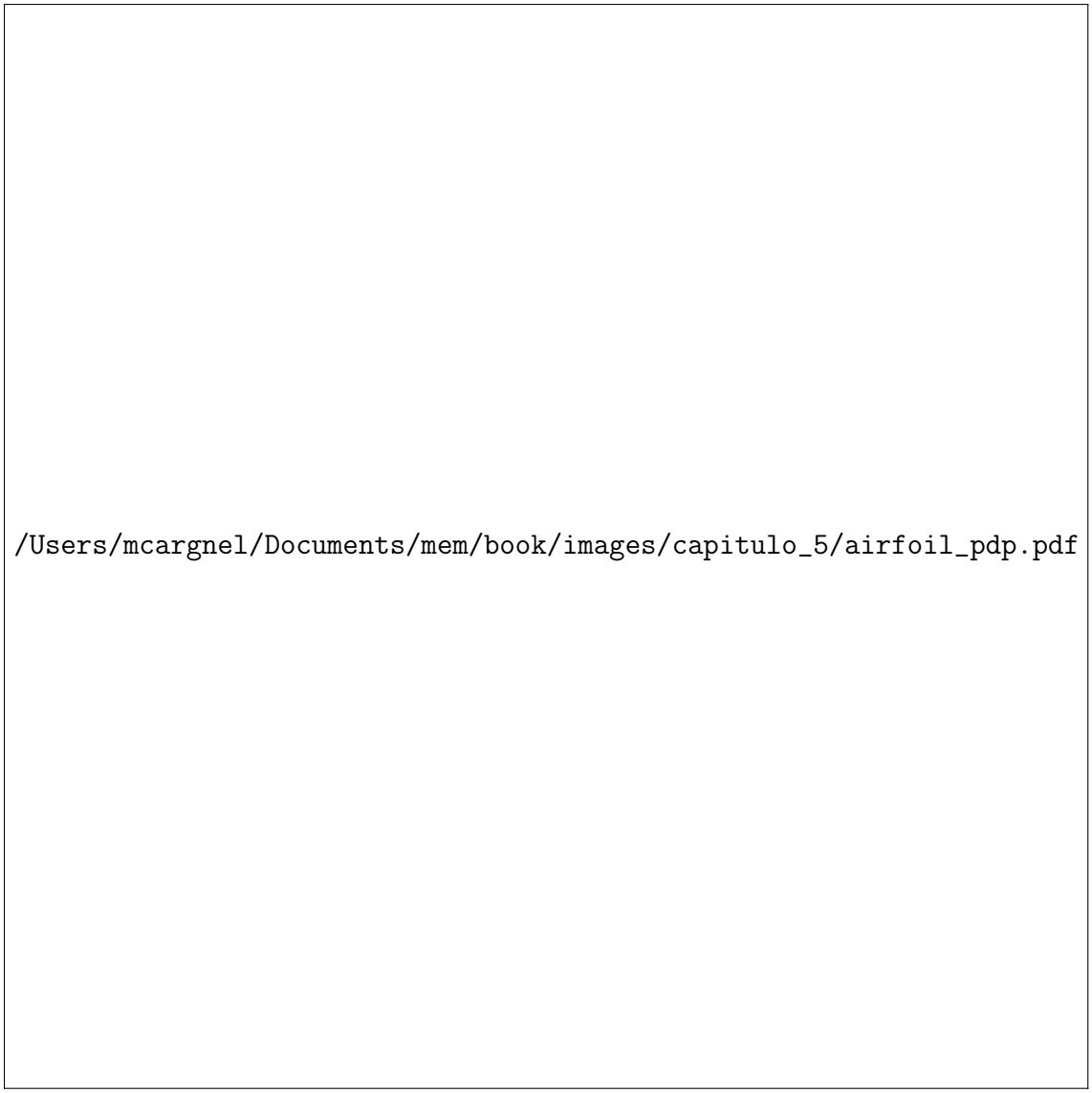

Figura 5.3: Gráficos de Dependencia Parcial (PDP) para *frequency* y *suction-side-displacement-thickness* en el modelo de *Gradient Boosting* (*Airfoil*).

Al evaluar el rendimiento predictivo en el conjunto de prueba (**??**), *Gradient Boosting* emerge nuevamente como el modelo más preciso, consolidando la tendencia observada previamente sobre la eficacia de los métodos de ensamble en estos contextos.

Cuadro 5.4: Resultados de los modelos en el conjunto de testeo para *Concrete Compressive Strength*.

Modelo	RMSE	MAE	R^2
Regresión Lineal	9.7965	7.7456	0.6276
Árbol de decisión	7.0675	4.8364	0.8062
Random Forest	5.5196	3.7957	0.8818
Gradient Boosting	4.3500	2.8629	0.9266

En cuanto a la interpretabilidad, el análisis de importancia de variables (**??**)

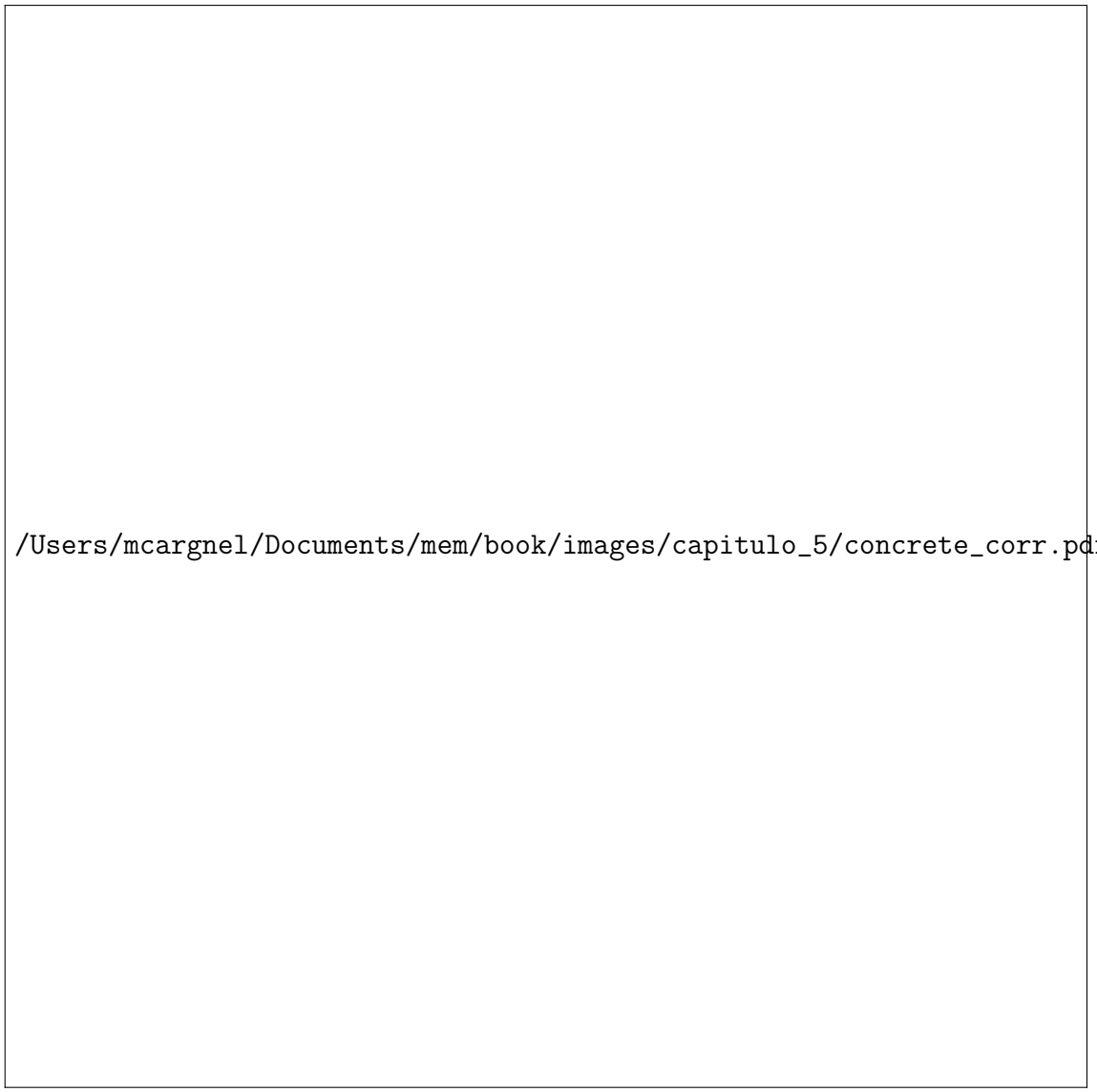

Figura 5.4: Matriz de correlación entre las variables del conjunto de datos *Concrete Compressive Strength*.

identifica a la Edad y la Cantidad de Cemento como los factores más influyentes en la resistencia del concreto. Esta conclusión es consistente entre los cinco mejores modelos ajustados (**??**).

Los gráficos de dependencia parcial (**??**) permiten examinar en detalle el efecto de estas variables clave. Para la Edad, se aprecia un crecimiento pronunciado de la resistencia durante los primeros 50 días, tras lo cual la curva se estabiliza, indicando ganancias marginales más modestas con el paso del tiempo. Por su parte, la relación con el Cemento presenta un comportamiento escalonado con incrementos notables en umbrales específicos (250, 300 y 350 kg/m^3). Un hallazgo interesante de los gráficos ICE es el paralelismo de las curvas individuales, lo que sugiere que el efecto de estas variables es aditivo y relativamente independiente de las interacciones con otros componentes de la mezcla.

En resumen, este análisis confirma que la maduración del concreto (edad) y su contenido de cemento son los determinantes críticos de su resistencia, siendo *Gradient Boosting* el algoritmo más capaz de capturar estas relaciones no lineales.

Figura 5.5: Importancia de las variables (PFI) del modelo *Gradient Boosting* para *Concrete Compressive Strength*.

5.3. *Wine Quality*

Finalmente, se analiza el conjunto de datos *Wine Quality* [**?**], que contiene 4898 muestras de vinos blancos de una región específica de Italia. Se dispone de 11 variables fisicoquímicas (**??**) para predecir la calidad sensorial del vino, medida en una escala discreta de 0 a 10.

Antes del modelado, se detectó una alta colinealidad entre el dióxido de azufre total y el libre (**??**), procediéndose a eliminar este último para simplificar el modelo sin perder información significativa.

Los resultados en el conjunto de prueba (**??**) presentan un cambio notable respecto a los casos anteriores: aquí, *Random Forest* obtiene la mejor, superando incluso a *Gradient Boosting*. Aunque los valores de R^2 son en general más bajos que en los otros conjuntos de datos, lo cual es típico en problemas de calidad subjetiva con alta varianza, el bosque aleatorio logra explicar casi el 50 % de la variabilidad.

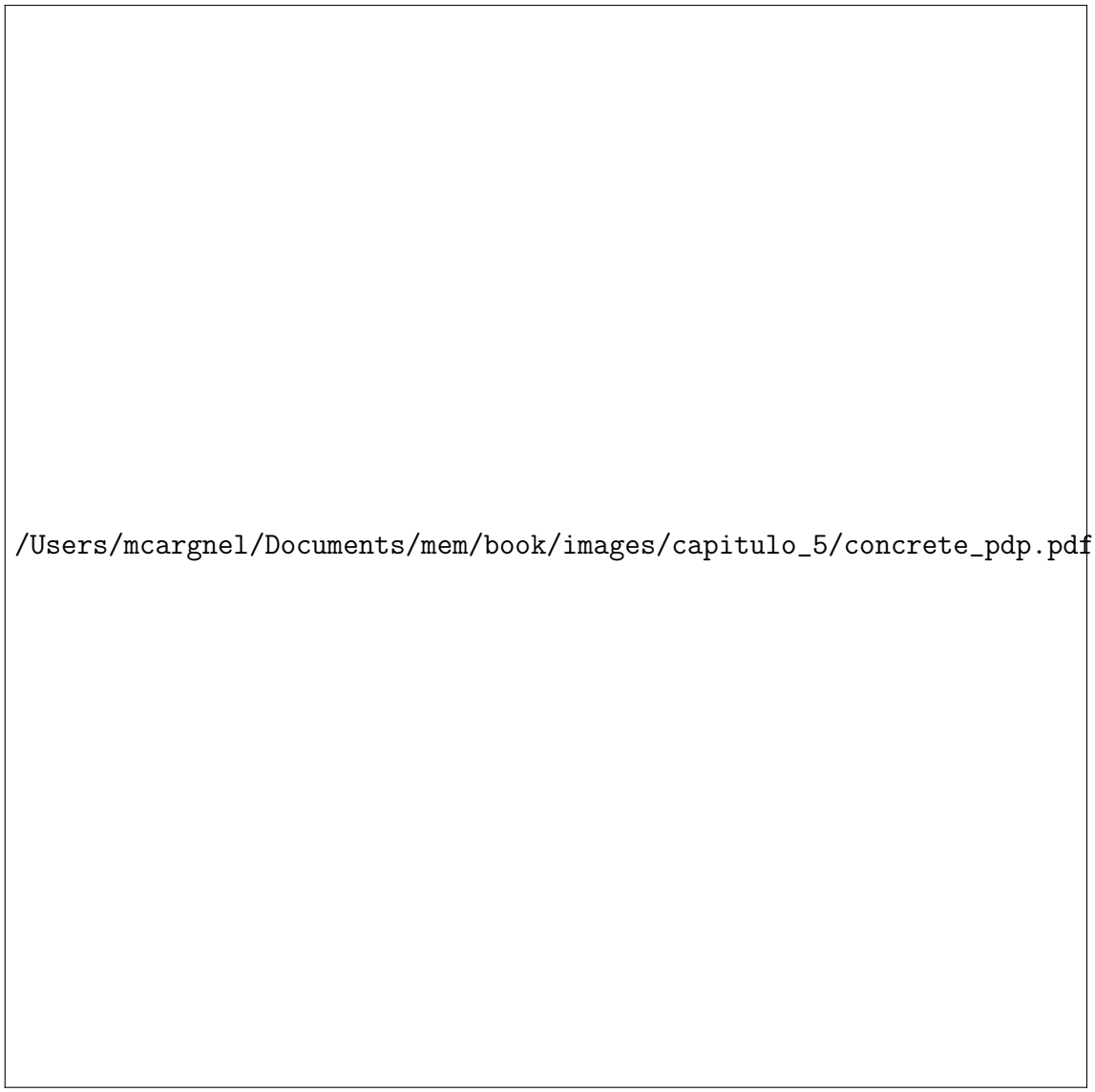

Figura 5.6: Partial dependence plots (PDPs) e individual conditional expectation (ICE) para edad y cemento en el conjunto de datos *Concrete Compressive Strength*.

La importancia de variables (**??**) destaca rotundamente al contenido de alcohol como el principal predictor de la calidad, seguido por la acidez volátil. Esta jerarquía se mantiene estable a través de los diversos modelos ajustados (**??**).

El análisis de dependencia parcial (**??**) arroja luz sobre la naturaleza de estas influencias. Para el alcohol, la relación con la calidad es positiva y no lineal, con un impacto mayor en el rango medio (11 %–13 %). Por el contrario, la acidez volátil penaliza la calidad de forma casi lineal: a medida que aumenta su concentración, la puntuación predicha disminuye consistentemente.

En conclusión, la aplicación de estas técnicas en tres dominios dispares demuestra su versatilidad. Más allá de la precisión predictiva, donde los ensambles dominaron, herramientas como PFI y PDP permitieron traducir cajas negras matemáticas en conocimientos de dominio tangibles: la importancia de las frecuencias bajas en el ruido aerodinámico, el rol crítico del curado en el concreto, y la preferencia por vinos con mayor grado alcohólico y menor acidez volátil.

Cuadro 5.5: Variables del conjunto de datos *Wine Quality* y su significado.

Variable	**Significado**
fixed_acidity	Acidez fija en el vino (principalmente ácido tartárico)
volatile_acidity	Acidez volátil en el vino (principalmente ácido acético)
citric_acid	Ácido cítrico presente en el vino
residual_sugar	Cantidad de azúcar residual en el vino
chlorides	Contenido de cloruros (sal) en el vino
free_sulfur_dioxide	Dióxido de azufre libre en el vino
total_sulfur_dioxide	Dióxido de azufre total en el vino (libre + combinado)
density	Densidad del vino
pH	Nivel de acidez/alcalinidad del vino (escala pH)
sulphates	Contenido de sulfatos en el vino
alcohol	Contenido de alcohol en el vino (% por volumen)
quality	Calidad del vino, puntuación entre 0 y 10 (var. dep.)

Cuadro 5.6: Resultados de los modelos en el conjunto de testeo para *Wine*.

Modelo	RMSE	MAE	R^2
Regresión Lineal	0.7378	0.5684	0.2630
Árbol de decisión	0.7697	0.5869	0.1979
Random Forest	0.6103	0.4379	0.4957
Gradient Boosting	0.6483	0.4818	0.4308

Figura 5.7: Matriz de correlación para las variables del conjunto de datos *Wine Quality*.

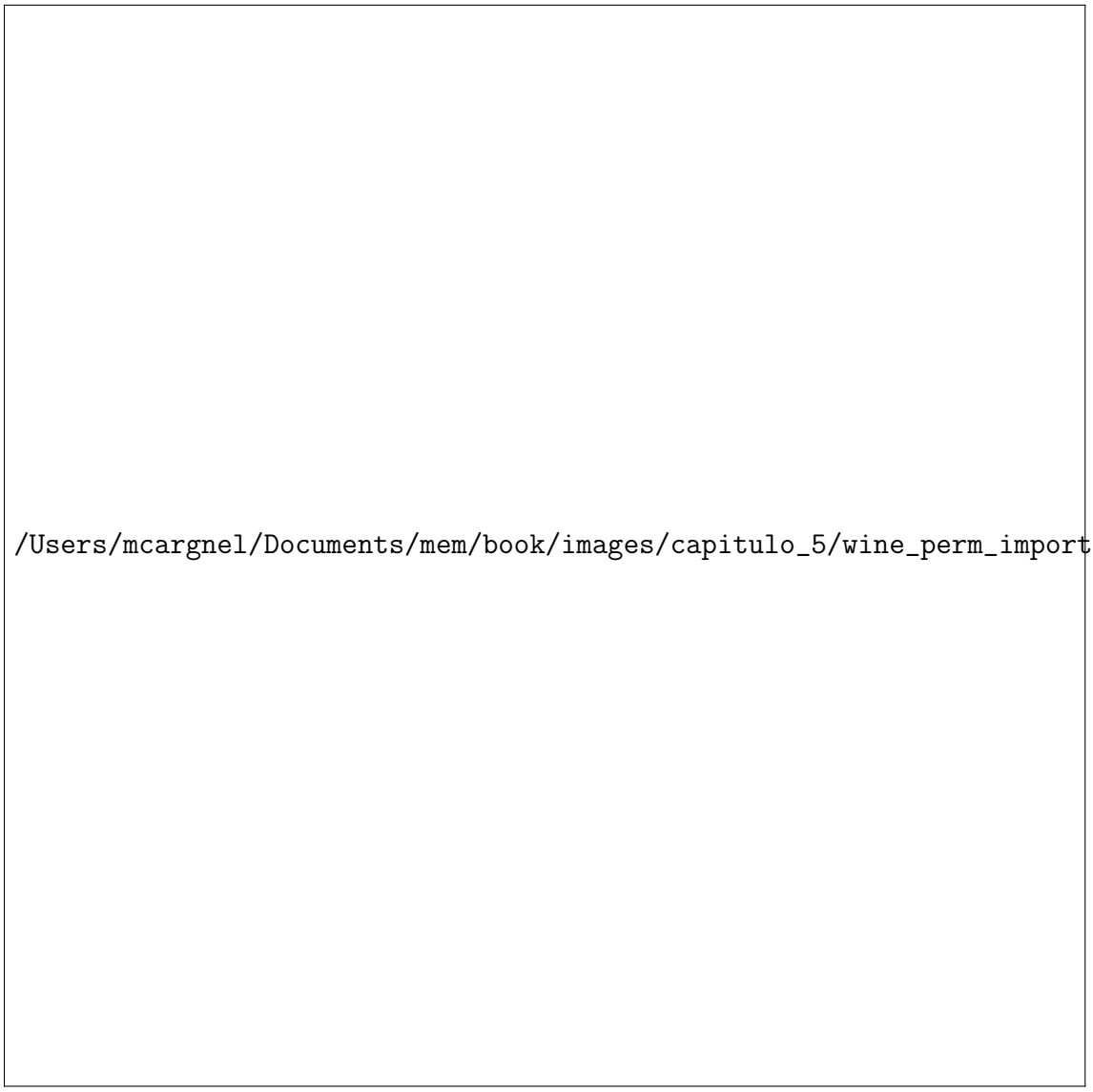

Figura 5.8: Importancia de variables (PFI) en el modelo de *Random Forest* para *Wine Quality*.

Figura 5.9: Gráficos de Dependencia Parcial (PDP) e ICE para alcohol y acidez volátil en el modelo de *Random Forest* (*Wine Quality*).

6. Conclusiones

Este trabajo se propuso abordar el compromiso o *trade-off* entre la capacidad predictiva y la capacidad explicativa. Se partió de la premisa de que, tradicionalmente, se debía elegir entre modelos simples e interpretables, como la regresión lineal, o modelos complejos de alta capacidad predictiva pero difícil interpretación, conocidos como cajas negras o *black boxes*.

A lo largo del desarrollo, se mostró que los ensambles de árboles, específicamente *random forest* y *gradient boosting machines*, representan una buena solución para problemas de predicción, superando consistentemente a los modelos base y a las regresiones lineales en los casos de estudio analizados. Sin embargo, su complejidad inherente planteaba el desafío de la interpretabilidad.

Para superar esta barrera, se estudiaron y aplicaron técnicas de aprendizaje automático interpretable agnósticas al modelo. Se vio que la (*permutation feature importance*) permite identificar qué covariables son fundamentales para el modelo, mientras que los (*partial dependence plots*) e *individual conditional expectation* (ICE) permiten visualizar cómo se relacionan estas variables con la variable dependiente. Además, se abordó el problema de la estabilidad y los *Rashomon Sets*, destacando que la interpretación debe ser cautelosa ante la existencia de múltiples modelos con rendimiento similar pero interpretaciones diferentes.

La aplicación de estas metodologías en conjuntos de datos reales de naturaleza diversa (aerodinámica, ingeniería civil y enología) demostró como llevar a la practica estas tecnicas e interpretar las variables que se incluyeron.

Es fundamental concluir este trabajo remarcando ciertas limitaciones. En primer lugar, la interpretación del modelo no es causal. Las técnicas presentadas, como los gráficos de dependencia parcial, describen cómo el modelo utiliza las variables para predecir, basándose en las relaciones capturadas durante el entrenamiento.

Si bien se discutió teóricamente, siguiendo a [**?**], que supuestos como la ausencia de confusores no observados y la correcta especificación del DAG estas interpretaciones pueden tener una lectura causal, en la práctica, y especialmente con datos observacionales, estos supuestos raramente se pueden verificar en su totalidad. Por lo tanto, las conclusiones extraídas de estas herramientas deben entenderse principalmente en términos de asociación y poder predictivo. Para realizar inferencia causal rigurosa, sería necesario complementar estas técnicas con marcos de trabajo específicos, como el *double machine learning* o el diseño experimental adecuado, que exceden el alcance de este estudio.

Otra limitación importante es la falta de una teoría de inferencia formal robusta similar a la que existe para la regresión lineal o la estadística clásica. Mientras que en los modelos lineales tradicionales se cuenta con intervalos de confianza, pruebas de hipótesis y propiedades asintóticas conocidas para los estimadores, en los modelos de aprendizaje automático estas garantías teóricas son mucho más difíciles de esta-

blecer. Si bien existen aproximaciones basadas en remuestreo, como el *bootstrapping*, para estimar la varianza y construir intervalos de confianza, estas técnicas suelen carecer del mismo rigor matemático y formalismo que caracteriza a la inferencia estadística clásica. Esto implica que las conclusiones sobre la importancia de una variable o los efectos deben tomarse como direccionales. Sin embargo, es relevante el desarrollo de *Conformal Prediction* [**?**]. Este enfoque permite pasar de predicciones puntuales a intervalos de predicción válidos en muestras finitas. De este modo, se dota a los modelos de caja negra de garantías estadísticas, un avance que ha cobrado gran interés reciente [**?**].

En definitiva, este trabajo evidencia que el aprendizaje automático interpretable es una herramienta poderosa para dotar de transparencia a los modelos predictivos, permitiendo a los usuarios confiar en sus decisiones, aunque siempre manteniendo una postura crítica respecto a sus limitaciones.

A. Apéndice

Cuadro A.1: Metricas de los 5 modelos con menor error en la muestra de validación cruzada - Airfoil Self Noise

Rank	*Learning Rate*	*N Estimators*	RMSE	MAE	R^2
1	0.2	1900	1.4761	1.0337	0.9565
2	0.2	1800	1.4838	1.0420	0.9561
3	0.2	1700	1.4943	1.0508	0.9554
4	0.2	1600	1.5117	1.0663	0.9544
5	0.2	1500	1.5173	1.0773	0.9540

Cuadro A.2: Comparación de importancia de características para los 5 modelos con menor error en la muestra de validación cruzada - Airfoil Self Noise

Feature	Top 1	Top 2	Top 3	Top 4	Top 5
suction-side-displacement-thickness	0.4134	0.4133	0.4133	0.4131	0.4129
frequency	0.4084	0.4084	0.4085	0.4086	0.4087
chord-length	0.1385	0.1385	0.1386	0.1386	0.1386
free-stream-velocity	0.0397	0.0397	0.0397	0.0397	0.0397

Cuadro A.3: Metricas de los 5 modelos con menor error en la muestra de validación cruzada - Concrete Strength

Rank	*Learning Rate*	*N Estimators*	RMSE	MAE	R^2
1	0.1	800	4.3500	2.8629	0.9266
2	0.1	700	4.3934	2.9138	0.9251
3	0.1	900	4.3479	2.8536	0.9266
4	0.1	1000	4.3259	2.8221	0.9274
5	0.1	600	4.4356	2.9745	0.9236

Cuadro A.4: Comparación de importancia de características para los 5 modelos con menor error en la muestra de validación cruzada - Concrete Strength

Feature	Top 1	Top 2	Top 3	Top 4	Top 5
Age	0.3578	0.3578	0.3578	0.3579	0.3579
Cement	0.3021	0.3023	0.3020	0.3018	0.3024
Water	0.1139	0.1138	0.1138	0.1138	0.1139
Blast Furnace Slag	0.0811	0.0812	0.0811	0.0811	0.0812
Superplasticizer	0.0747	0.0747	0.0747	0.0747	0.0746
Fine Aggregate	0.0369	0.0368	0.0369	0.0369	0.0367
Coarse Aggregate	0.0189	0.0188	0.0190	0.0191	0.0187
Fly Ash	0.0146	0.0146	0.0146	0.0147	0.0146

Cuadro A.5: Metricas de los 5 modelos con menor error en la muestra de validación cruzada - Wine Quality

Rank	*Learning Rate*	*N Estimators*	RMSE	MAE	R^2
1	0.2	1600	0.6103	0.4379	0.4957
2	0.2	1300	0.6102	0.4379	0.4959
3	0.2	1400	0.6100	0.4378	0.4961
4	0.2	1900	0.6102	0.4378	0.4958
5	0.2	1800	0.6103	0.4379	0.4957

Cuadro A.6: Comparación de importancia de características para los 5 modelos con menor error en la muestra de validación cruzada - Wine Quality

Feature	Top 1	Top 2	Top 3	Top 4	Top 5
alcohol	0.2602	0.2602	0.2603	0.2601	0.2601
volatile_acidity	0.1397	0.1395	0.1395	0.1396	0.1396
total_sulfur_dioxide	0.0885	0.0885	0.0885	0.0887	0.0886
sulphates	0.0832	0.0833	0.0833	0.0831	0.0831
pH	0.0807	0.0806	0.0806	0.0808	0.0807
residual_sugar	0.0785	0.0787	0.0785	0.0786	0.0786
chlorides	0.0712	0.0710	0.0712	0.0712	0.0712
citric_acid	0.0681	0.0683	0.0682	0.0680	0.0681
density	0.0656	0.0656	0.0656	0.0655	0.0655
fixed_acidity	0.0644	0.0643	0.0643	0.0644	0.0643

Bibliografía

[Aeberhard and Forina, 1992] Aeberhard, S. and Forina, M. (1992). Wine. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC7J.

[Angelopoulos and Bates, 2023] Angelopoulos, A. and Bates, S. (2023). *Conformal Prediction: A Gentle Introduction*.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[Breiman, 2001a] Breiman, L. (2001a). Random forests. *Machine Learning*, 45(1):5–32.

[Breiman, 2001b] Breiman, L. (2001b). Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199 – 231.

[Breiman et al., 1984] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Routledge.

[Brooks et al., 1989] Brooks, T., Pope, D., and Marcolini, M. (1989). Airfoil Self-Noise. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5VW2C.

[Chernozhukov et al., 2018] Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68.

[Fisher et al., 2019] Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of machine learning research : JMLR*, 20.

[Freiesleben et al., 2024] Freiesleben, T., König, G., Molnar, C., and Tejero-Cantero, Á. (2024). Scientific inference with interpretable machine learning: Analyzing models to learn about real-world phenomena. *Minds and Machines*, 34.

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232.

[Goldstein et al., 2015] Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65.

[James et al., 2023] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2023). *An Introduction to Statistical Learning: with Applications in R*. Springer.

[Kahn,] Kahn, M. Diabetes. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5T59G.

[Maronna et al., 2019] Maronna, R., Martin, R., Yohai, V., and Salibián-Barrera, M. (2019). *Robust Statistics: Theory and Methods (with R)*. Wiley Series in Probability and Statistics. Wiley.

[Nash et al., 1994] Nash, W., Sellers, T., Talbot, S., Cawthorn, A., and Ford, W. (1994). Abalone. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55C7W.

[Pearl, 1993] Pearl, J. (1993). Comment: Graphical models, causality and intervention. *Statistical Science*, 8(3):266–269.

[Pearl, 2018] Pearl, J. (2018). Theoretical impediments to machine learning with seven sparks from the causal revolution. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 3, New York, NY, USA. Association for Computing Machinery.

[Shmueli, 2010] Shmueli, G. (2010). To Explain or to Predict? *Statistical Science*, 25(3):289 – 310.

[Strobl et al., 2008] Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC bioinformatics*, 9:307.

[Vovk et al., 2005] Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*.

[Wager and Athey, 2018] Wager, S. and Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242.

[Yeh, 1998] Yeh, I.-C. (1998). Concrete Compressive Strength. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PK67.

[Zhao and Hastie, 2021] Zhao, Q. and Hastie, T. (2021). Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, 39(1):272–281.