USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

MODELADO ESTADÍSTICO DE DATOS DEPORTIVOS

Álvaro García Areñas

2024/2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRADO DE MATEMÁTICAS

Traballo Fin de Grado

MODELADO ESTADÍSTICO DE DATOS DEPORTIVOS

Álvaro García Areñas

Julio, 2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Conocimiento: Estadística, Análisis Matemático y Optimización
Título: Modelado Estadístico de Datos Deportivos
Breve descripción del contenido
El objetivo de este TFG es doble. En una primera fase se construirá una base de datos basada en estadísticas deportivas con el objetivo de construir una métrica que permita cuantificar el rendimiento de un jugador o equipo. En una segunda fase, se pretenden utilizar técnicas de aprendizaje estadístico para estimar y/o hacer predicción de futuros resultados deportivos basándose en estos indicadores de rendimiento.
Recomendaciones
Otras observaciones

Índice

Resumen

A lo largo de este trabajo se presenta una aplicación del modelo de aprendizaje supervisado *Random Forest* a datos deportivos. En concreto, a datos asociados a los equipos de la NBA en las últimas temporadas.

En el primer capítulo se realiza una breve introducción a los algoritmos de aprendizaje supervisado haciendo especial énfasis en el dilema sesgo-varianza, problema fundamental en este tipo de modelos.

A continuación, se realiza una descripción sistemática de los árboles de decisión. Estos son unos de los modelos más sencillos de aprendizaje supervisado, pero son piezas fundamentales en otros modelos más complejos como el *Random Forest*.

En el Capítulo 3 se introduce el modelo *Random Forest* tal y como lo definió Leo Breiman en 2001. Además, se presentan unos resultados fundamentales relacionados con la reducción de su error relativo y su varianza.

Finalmente, en el último capítulo se aplica el modelo *Random Forest* a datos de estadística avanzada de los equipos de la NBA. Se analizará tanto un caso de clasificación como uno de regresión. En ambos casos, se estudiará la dependencia de los modelos con sus hiperparámetros y se compararán los resultados con otros modelos habituales en este tipo de problemas.

Abstract

Throughout this work, an application of the supervised learning model Random Forest to sports data is presented. Specifically, data associated with NBA teams from recent seasons.

In the first chapter, a brief introduction to supervised learning algorithms is provided, with a particular emphasis on the bias-variance tradeoff, a fundamental problem in this type of model.

Next, a systematic description of decision trees is given. These are among the simplest supervised learning models but serve as essential components in more complex models such as Random Forest.

In Chapter 3, the Random Forest model is introduced as defined by Leo Breiman in 2001. Additionally, key results related to its relative error reduction and variance are presented.

Finally, in the last chapter, the Random Forest model is applied to advanced statistics of NBA teams. Both a classification case and a regression case will be analyzed. In each scenario, the dependence of the models on their hyperparameters will be studied, and the results will be compared with other commonly used models for this type of problem.

Introducción

Con el reciente aumento de la capacidad de computación, los datos están cobrando cada vez más importancia en todos los ámbitos de nuestras vidas, llegando a considerarse como "el oro del siglo XXI". Siendo capaces de recoger datos de forma consistente y aplicar modelos estadísticos de forma correcta, podemos ser capaces de predecir sucesos futuros con gran precisión.

Dentro de los principales modelos que se emplean en la actualidad se encuentran los algoritmos de aprendizaje supervisado, cuya idea principal es entrenar el modelo corrigiendo los errores que comete a partir de resultados conocidos. Estos tipos de modelos pueden ser utilizados para una gran gama de problemas, desde tratar de predecir la supervivencia de un paciente a partir de sus síntomas (ejemplo de problema de clasificación) hasta estimar el número de litros que tiene que producir una empresa de refrescos en función de las características de los consumidores (ejemplo de problema de regresión).

El ámbito deportivo no se queda atrás en la tendencia de tratar de predecir resultados a partir de grandes cantidades de datos. En las últimas décadas ha sido notable la mejora exponencial en el perfeccionamiento de muchos deportes, especialmente en el ámbito profesional, gracias a esta nueva visión analítica de la disciplina. Es por ello que los mejores equipos del mundo de la mayoría de los deportes más populares están contratando matemáticos e informáticos para recopilar datos y predecir estrategias que les lleven a mejoras del rendimiento deportivo.

La NBA, la liga nacional de baloncesto americana, ha sido pionera en la recolección de grandes bases de datos de sus partidos. De esta forma, los equipos pueden aprovecharlos para mejorar su rendimiento en los partidos. A lo largo de las siguientes páginas se presenta un ejemplo del tratamiento de estos datos a partir de algoritmos de aprendizaje supervisado, no sin antes asentar la base teórica de estos; en particular, del modelo conocido como *Random Forest*.

Capítulo 1

Aprendizaje Supervisado

Los algoritmos de aprendizaje supervisado son aquellos mediante los cuales se trata de predecir una respuesta a partir de una muestra de observaciones conocidas. Para ello se parte de un modelo inicial que se entrena a partir de las observaciones, es decir, el modelo aprende de la información conocida para poder predecir la respuesta en nuevas observaciones. En función del modelo que se escoja, existen diferentes algoritmos de aprendizaje supervisado, aunque en este trabajo nos centraremos en uno de los más empleados: *Random Forest* (ver Capítulo 3), que se trata de un ensamblado de árboles de decisión (ver Capítulo 2).

Un tipo de modelo de aprendizaje supervisado muy conocido, y que se ve en el grado, es el modelo de regresión lineal. Al emplearlo, se quiere predecir el valor que toma una variable respuesta Y en función de ciertas variables explicativas $X = (X_1, \ldots, X_p)$ (también conocidas como características) suponiendo que la dependencia con la respuesta es lineal. Para ello, se parte de un conjunto de observaciones $\{(x_i, y_i)\}_{i=1}^n$ que cuentan con valores para las variables explicativas y la variable respuesta. El modelo es el siguiente:

$$Y = \beta_0 + \beta_1 \cdot X_1 + \cdots + \beta_p \cdot X_p + \varepsilon\,, \tag{1.1}$$

donde $\beta = (\beta_0, \beta_1, \ldots, \beta_p)'$ es un vector[1] de parámetros a estimar y ε es el término de error, que se supone que sigue una distribución $N(0, \sigma)$. En este caso, el entrenamiento del modelo consiste en calcular un estimador del vector β, o incluso para el parámetro σ, a partir de los datos originales $\{(x_i, y_i)\}_{i=1}^n$. De esta forma, se podrá predecir el valor para una nueva observación $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_p)$ a partir de la expresión

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \tilde{x}_1 + \cdots + \hat{\beta}_p \cdot \tilde{x}_p\,, \tag{1.2}$$

[1] Se denota al vector traspuesto de X como X'.

siendo $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)'$ el estimador obtenido para el parámetro β.

Por supuesto, no siempre la dependencia de Y con las variables $X_1, \ldots, X_p$ será lineal o el modelo normal para el error es adecuado, incluso puede que no exista una forma funcional clara para la dependencia entre estas variables. Es por ello que entran en juego una gran cantidad de modelos de aprendizaje supervisado, entre los que se encuentran los *Random Forest*, cuyas características se explicarán en el Capítulo 3.

Dentro de los problemas en los que se emplean algoritmos de aprendizaje supervisado se pueden distinguir dos grandes grupos: problemas de clasificación y problemas de regresión. La diferencia entre ellos radica en la forma de la variable respuesta que se quiera predecir. Por ejemplo, supongamos que queremos predecir el color de una flor en función de ciertas características biológicas de esta. Los valores de Y serán en este caso *blanca*, *rosa* y *azul*, sabiendo de antemano que la flor solo puede ser de estos tres colores. Como la variable es discreta y toma un conjunto finito de valores, nos encontramos en un problema de clasificación. Por otro lado, si lo que queremos es predecir la temperatura que hará en cierta localidad conociendo factores meteorológicos como la humedad o la presión nos hallamos en un problema de regresión, ya que en este caso la variable respuesta toma un continuo de valores. Las diferencias entre los algoritmos a emplear en cada caso residen en la forma de calcular los errores que se querrán minimizar para conseguir una mejor predicción. De todas formas, los *Random Forest* son un conjunto de modelos que cubre ambos tipos de problemas, tanto regresores como clasificadores.

Un problema central en los modelos de aprendizaje supervisado es el balance entre el sesgo y la varianza. Consiste, a grandes rasgos, en la incapacidad de conseguir estos dos objetivos fundamentales a la vez a la hora de usar un modelo:

- Minimizar la diferencia entre las predicciones del modelo y los valores reales.
- Emplear un modelo que no sea muy sensible a ligeros cambios en los datos de entrenamiento, proporcionando estimaciones estables.

Si se intenta reducir el error cometido por las predicciones el modelo gana "complejidad", por lo que será mucho más sensible a cualquier cambio en los datos. Análogamente, si tratamos de reducir la sensibilidad de un modelo para realizar el ajuste, las predicciones se alejarán en promedio de los valores reales. Esto es lo que se conoce como el dilema del sesgo y la varianza. Los siguientes ejemplos aclararán el problema y ayudarán a comprenderlo.

Ejemplo 1.1. Generamos con R una muestra de $n = 20$ observaciones. Los valores de la variable Y_1, dependientes de la variable X, vienen dados de la siguiente forma:

$$Y_1 = 2 + 30 \cdot X + \varepsilon_1 \,, \tag{1.3}$$

siguiendo el error una distribución normal con media 0 y desviación típica creciente con X, $\varepsilon_1 \sim N(0, 1.3 \cdot X^2)$. Las observaciones aparecen representadas en la Figura 1.1, de verde.

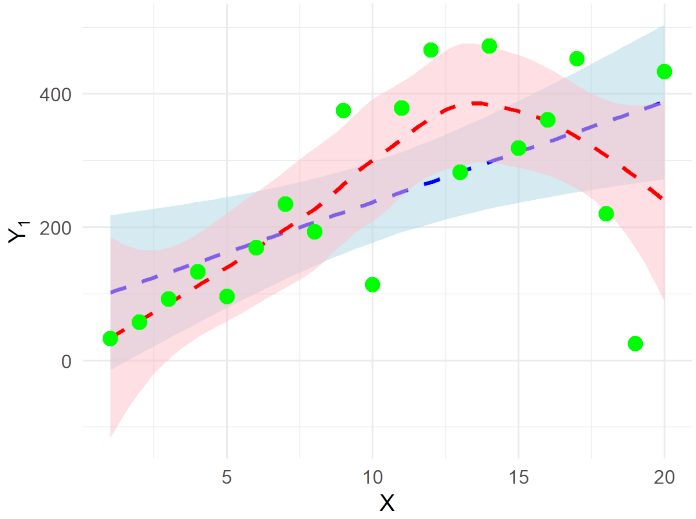


Figura 1.1: Ajustes lineal y polinómico a los datos de la muestra con tendencia lineal.

Junto con las observaciones aparecen también dos ajustes a dos modelos distintos: uno lineal (en azul) y otro polinómico[2] de grado 15 (en rojo). Sabemos, por construcción, que la tendencia de los datos es lineal; sin embargo, el aumento de la desviación típica del error con el aumento de la variable X enmascara ligeramente esta tendencia. Es por ello que, *a priori*, el ajuste polinómico se ajusta mejor a las observaciones que el modelo lineal, ya que cuenta con mayor complejidad al realizar el ajuste para poder reducir el error entre predicciones y observaciones. Por otro lado, si el objetivo de estos modelos es predecir nuevos datos de la muestra, el modelo lineal recoge mejor la tendencia de los datos y generarán mejores predicciones que el modelo polinómico. Cualitativamente, decimos que el modelo polinómico está "sesgado" a los datos, porque tiene poco sesgo (Definición 1.3) y mucha varianza (Definición 1.4).

Hacemos uso de otro ejemplo sencillo para aclarar aún más las ideas.

Ejemplo 1.2. Contamos de nuevo con $n = 20$ observaciones generadas en R. En este caso, la tendencia será parabólica:

$$Y_2 = 10 \cdot (X - 10)^2 + \varepsilon_2 \,, \tag{1.4}$$

[2] Un ajuste polinómico es de la forma $\hat{f}(X) = \sum_{i=0}^{r} c_i \cdot X^i$, donde los c_i son valores reales y r es el grado del polinomio de ajuste.

donde el error no depende de la variable X: $\varepsilon_2 \sim N(0, 200)$. Las observaciones aparecen representadas en la Figura 1.2, de verde.

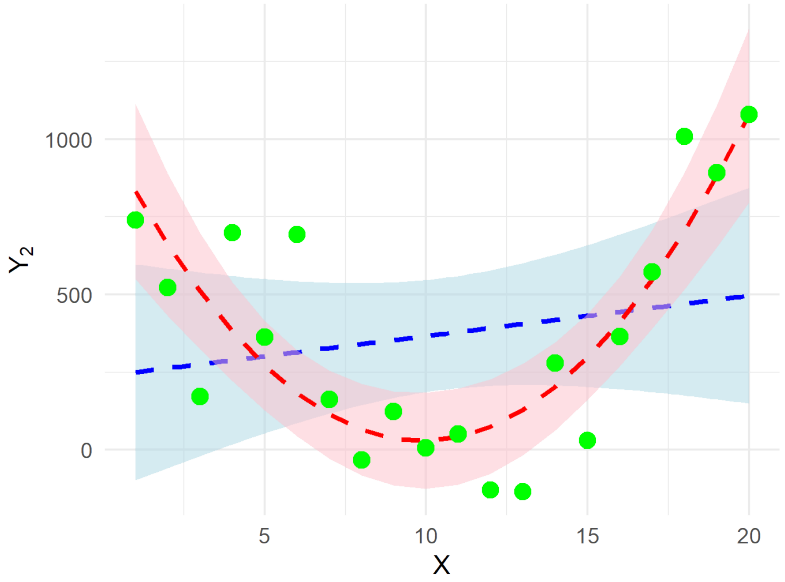


Figura 1.2: Ajustes lineal y polinómico a los datos de la muestra con tendencia parabólica.

En este caso, se realizaron un ajuste lineal y otro parabólico (polinómico de grado 2). Ahora el ajuste más "complejo", el parabólico, no tiene problemas por estar especialmente "sesgado", ya que sabemos que el modelo se ajusta perfectamente al origen de los datos de entrenamiento. Por otro lado, el modelo lineal no es capaz de ajustarse de forma correcta a las observaciones, ya que no es capaz de recoger la tendencia de los datos. Decimos que el modelo lineal tiene "poca variabilidad", porque tiene mucho sesgo y poca varianza.

El estudio del dilema sesgo-varianza es fundamental a la hora de construir un modelo para ajustar a una muestra. De hecho, muchos de los modelos más empleados en la actualidad cuentan con modificaciones mediante la definición de hiperparámetros[3] para reducir su complejidad, y así conseguir que sean menos "sesgados". Por ejemplo, en el caso de los árboles de decisión, veremos en el Capítulo 2 la existencia de distintos criterios de nodo terminal. De esta forma, consiguiendo árboles menos profundos el modelo pierde complejidad. En el caso de modelos simples, una forma habitual de reducir el sesgo y aumentar la complejidad, consiste en combinar varios de estos estimadores, conocidos como ensamblados. Un ejemplo habitual de ensamblado es el modelo *Random Forest*, en el que se profundizará en el Capítulo 3.

Es muy importante también introducir la nomenclatura habitual empleada una vez ajustado

[3]Características modificables de un modelo para mejorar su rendimiento.

el modelo; esta tiene especial relación con el balance sesgo-varianza. Cuando parece que el modelo se ajusta en exceso a los datos de entrenamiento, perdiendo capacidad de predicción para nuevas observaciones, hablamos de sobreajuste (*overfitting*). Por el contrario, cuando el modelo no es capaz de captar la tendencia de los datos por no ser lo suficientemente complejo, hablamos subajuste (*underfitting*).

Pasamos ahora a formular de manera rigurosa los conceptos de sesgo y varianza de un modelo (ver [3]).

1.1. Dilema Sesgo-Varianza

Presentamos en primer lugar las hipótesis y la notación con las que vamos a trabajar. Para simplificar la explicación de los conceptos y su desarrollo subsecuente supondremos que la variable para la que queremos predecir, Y, es unidimensional y continua. De forma general, supondremos una dependencia de las variables $X = (X_1, \ldots, X_p)$ de la forma

$$Y = f(X) + \varepsilon\,, \tag{1.5}$$

donde f es una función que refleja la dependencia entre las variables y ε es una variable (independiente del resto) que constituye un error aleatorio, verificando que $\mathbb{E}[\varepsilon] = 0$ y $\mathrm{Var}[\varepsilon] = \sigma^2$, siendo este último un valor constante. Al ajustar un modelo a los datos de la muestra se obtiene una función $\hat{f}$ estimadora de f. Esta función dependerá evidentemente de las observaciones.

Vamos a suponer que podemos simular varias muestras distintas de valores $\{(x_i, y_i)\}_{i=1}^n$, de forma que $Y_i \equiv Y|_{X=x_i}$ (la variable Y sabiendo que la variable X vale x_i) será una nueva variable aleatoria para cada i. Como $\hat{f}$ depende de los valores que tome Y para cada observación, esta también será una variable aleatoria. Con este conjunto de muestras, podremos estudiar el comportamiento de un modelo en general, independientemente de los valores que tome cada muestra.

Definición 1.3. Sea Y una variable aleatoria dependiente del conjunto de variables aleatorias X según la expresión (1.5). Sea un modelo que produce una función estimadora de f, $\hat{f}$, en función de la muestra aleatoria de observaciones $\{(x_i, y_i)\}_{i=1}^n$. Se define el **sesgo** del modelo en $X = x$ tal que

$$\mathrm{Sesgo}(x) := f(x) - \mathbb{E}_Y\left[\hat{f}(x)\right]\,. \tag{1.6}$$

También se define el **sesgo cuadrático global** de la siguiente forma:

$$\mathrm{Sesgo}^2 := \mathbb{E}_X\left[\left(f(X) - \mathbb{E}_Y\left[\hat{f}(X)\right]\right)^2\right] = \mathbb{E}_X\left[\left(\mathrm{Sesgo}\,(X)\right)^2\right]\,, \tag{1.7}$$

donde es necesario añadir el cuadrado para que no se anulen los valores de signo opuesto.

Vemos entonces que el sesgo no es otra cosa que la diferencia entre el valor de la función f (que *a priori* no conocemos) en $X = x$ y el valor esperado de la función del modelo para ese mismo valor. Por lo tanto, para valores elevados de sesgo no se espera que la función $\hat{f}$ y la función f tomen valores parecidos, lo que llevaría a subajuste. Por otra parte, valores muy bajos de sesgo llevan a esperar que los valores $\hat{f}$ y f en $X = x$ sean prácticamente idénticos, lo que en principio es bueno para las aspiraciones del modelo. Sin embargo, esta casuística implica aumentar la varianza, lo que puede llevar a un problema de sobreajuste.

Definición 1.4. En las mismas condiciones que para la Definición 1.3, se define la **varianza** del modelo en $X = x$ tal que

$$\mathrm{Var}\left(\hat{f}(x)\right) := \mathbb{E}_Y\left[\left(\hat{f}(x) - \mathbb{E}_Y\left[\hat{f}(x)\right]\right)^2\right]. \tag{1.8}$$

También se define la **varianza global** de la siguiente forma:

$$\mathrm{Var}\left(\hat{f}\right) := \mathbb{E}_X\left[\mathbb{E}_Y\left[\left(\hat{f}(X) - \mathbb{E}_Y\left[\hat{f}(X)\right]\right)^2\right]\right] = \mathbb{E}_X\left[\mathrm{Var}\left(\hat{f}\,(X)\right)\right]. \tag{1.9}$$

Por lo tanto, la varianza de un modelo representa la dispersión que se espera que tengan los valores $\hat{f}(x)$ si se realizan ajustes a varias muestras aleatorias $\{(x_i, y_i)\}_{i=1}^n$. De esa forma, si la varianza es muy grande significará que el modelo depende mucho de ligeras variaciones en los datos. Esto no nos interesa ya que un modelo con tanta sensibilidad a los datos no genera predicciones "estables". Debemos pensar que el modelo a estimar f es fijo y no es deseable que $\hat{f}$, estimación de f, varíe demasiado. Por otro lado, varianzas bajas implican modelos poco dependientes de las variaciones de las observaciones, lo que significa conseguir predicciones más "estables". Sin embargo, reducir excesivamente la varianza lleva a aumentar demasiado el sesgo, ya que implicaría tener que emplear modelos demasiado "sencillos" para la muestra que se pretende ajustar.

En general, a la hora de escoger el modelo lo que nos interesaría es minimizar, para cada valor $X = x$, la diferencia entre el valor que tome la variable Y y el valor que predice la estimación de f, $\hat{f}(x)$. Para ello, definimos el siguiente error:

Definición 1.5. En las mismas condiciones que para la Definición 1.3, se define el **error de predicción esperado (EPE)** en $x = X$ tal que

$$EPE(x) := \mathbb{E}_Y\left[\left(Y|_{X=x} - \hat{f}(x)\right)^2\right]. \tag{1.10}$$

También se define el **error de predicción esperado global** de la siguiente forma:

$$EPE := \mathbb{E}_X \left[\mathbb{E}_Y \left[\left(Y - \hat{f}(X) \right)^2 \right] \right] = \mathbb{E}_X \left[EPE\,(X) \right] . \tag{1.11}$$

De esta forma, el modelo con el menor error total esperado en las predicciones será el que minimice el EPE. Sin embargo, al no conocer por lo general la forma exacta de la función f, no se puede obtener dicho modelo de forma analítica. A pesar de ello, podemos usar una estimación del EPE para comparar distintos modelos, como podemos ver con el siguiente ejemplo.

Ejemplo 1.6. Trabajaremos con una variable Y que depende de otra variable X de forma sinusoidal tal que así:

$$Y = \sin(4\pi \cdot x) + \varepsilon\,, \tag{1.12}$$

donde $\varepsilon \sim N(0, 0.4)$. Fijando $n = 100$ valores de la variable X, $\{x_i\}_{i=1}^{100}$, distribuidos de forma uniforme entre el 0 y el 1, generamos con $m = 50$ conjuntos de valores aleatorios para Y, por lo que contaremos con 50 muestras $\{(x_i, y_{ij})\}_{i=1}^{100}$ (con $j \in \{1, \ldots, 50\}$). Para cada muestra, realizamos ajustes a modelos polinómicos desde grado 1 (recta) hasta grado 25, como se puede ver en la Figura 1.3.

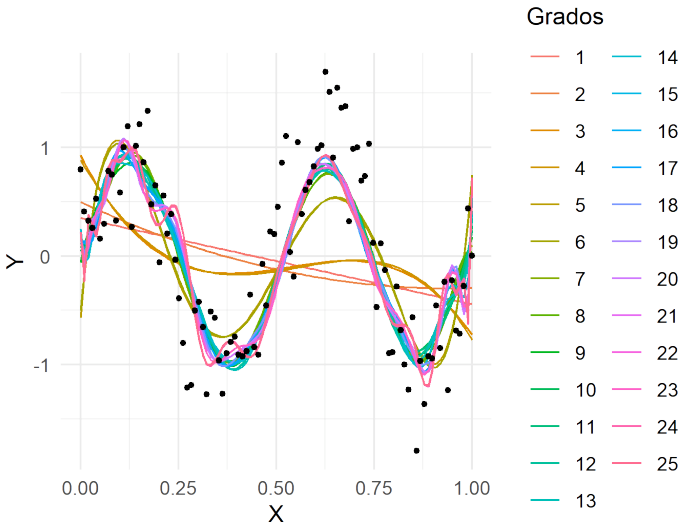


Figura 1.3: Ajustes realizados para la muestra $\{(x_i, y_{i47})\}_{i=1}^{100}$ a distintos modelos polinómicos.

En la Figura 1.3 vemos cómo a grados muy bajos los modelos no consiguen captar la tendencia de las observaciones y a grados muy altos los modelos se sobreajustan a los datos. Vamos a estudiar este comportamiento estimando los valores de sesgo cuadrático y varianza global para

cada modelo, así como el valor de EPE global. Los estimadores empelados fueron los siguientes:

$$\widehat{\text{Sesgo}^2} = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(x_i\right) - \frac{1}{m}\sum_{j=1}^{m}\hat{f}_j\left(x_i\right)\right)^2, \tag{1.13}$$

$$\widehat{\text{Var}(\hat{f})} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{m}\sum_{j=1}^{m}\left(\hat{f}_j\left(x_i\right) - \frac{1}{m}\sum_{j=1}^{m}\hat{f}_j\left(x_i\right)\right)^2\right], \tag{1.14}$$

$$\widehat{EPE} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{m}\sum_{j=1}^{m}\left(y_{ij} - \hat{f}_j\left(x_i\right)\right)^2\right], \tag{1.15}$$

donde $n = 100$ es el número de observaciones, $m = 50$ es el número de muestras y $f(x) = \sin(4\pi \cdot x)$ la función que recoge la dependencia entre X e Y. En la Figura 1.4 se muestran los resultados ordenados en función del grado del polinomio.

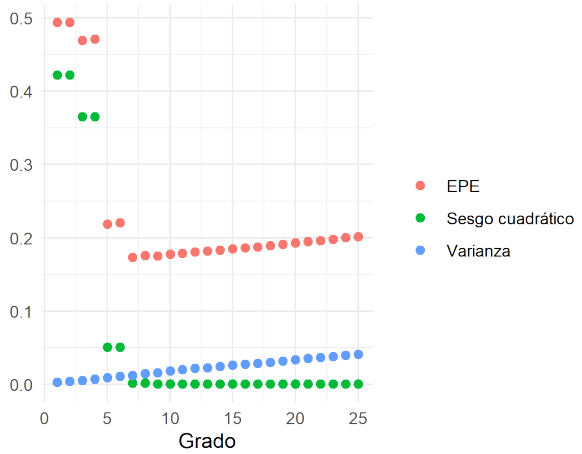


Figura 1.4: Resultados de la simulación en función del grado del polinomio

Vemos cómo el sesgo cuadrático se reduce a medida que aumenta el grado del polinomio, y por lo tanto la complejidad. La varianza, por otro lado, aumenta (en este caso parece que de forma lineal) con el grado del polinomio, acorde con el aumento de la complejidad. Ahora bien, vemos que la estimación del EPE alcanza un mínimo para grado 7. Este es el grado para el que termina la caída tan pronunciada del sesgo. Además, parece que la tendencia del EPE está altamente correlacionada con las del sesgo y la varianza. Para verificar este hecho, demostramos la siguiente proposición que relaciona los tres valores.

Proposición 1.7. *En las condiciones de la Definición 1.3, se cumple la siguiente igualdad:*

$$EPE(x) = (Sesgo(x))^2 + Var\left(\hat{f}(x)\right) + \sigma^2. \tag{1.16}$$

Demostración. Aplicamos en primer lugar la Definición 1.5 y descomponemos el cuadrado:

$$\begin{aligned} EPE(x) &= \mathbb{E}_Y\left[\left(Y-\hat{f}(x)\right)^2\right] = \mathbb{E}_Y\left[\left(f(x)+\varepsilon-\hat{f}(x)\right)^2\right] = \mathbb{E}_Y\left[\left(\left(f(x)-\hat{f}(x)\right)+\varepsilon\right)^2\right] \\ &= \mathbb{E}_Y\left[\left(f(x)-\hat{f}(x)\right)^2 + 2\cdot\left(f(x)-\hat{f}(x)\right)\cdot\varepsilon+\varepsilon^2\right] \\ &= \mathbb{E}_Y\left[\left(f(x)-\hat{f}(x)\right)^2\right] + 2\cdot\mathbb{E}_Y\left[\left(f(x)-\hat{f}(x)\right)\cdot\varepsilon\right] + \mathbb{E}_Y\left[\varepsilon^2\right]. \end{aligned}$$

Ahora bien, sabemos que ε no depende de ninguna variable; en particular, es independiente de $f(x)$ y de $\hat{f}(x)$ para cualquier $x \in X$. Por lo tanto:

$$\mathbb{E}_Y\left[\left(f(x)-\hat{f}(x)\right)\cdot\varepsilon\right] = \mathbb{E}_Y\left[f(x)-\hat{f}(x)\right]\cdot\mathbb{E}_Y[\varepsilon] = 0\,,$$

donde la última igualdad se debe a que $\mathbb{E}_Y[\varepsilon]=0$ por construcción. Por otro lado:

$$\mathbb{E}_Y\left[\varepsilon^2\right] = \mathrm{Var}(\varepsilon) + \mathbb{E}_Y[\varepsilon]^2 = \mathrm{Var}(\varepsilon) = \sigma^2\,.$$

Sumando y restando $\mathbb{E}_Y\left[\hat{f}(x)\right]$ dentro del primer término y descomponiendo el cuadrado:

$$\begin{aligned} &\mathbb{E}_Y\left[\left(f(x)-\hat{f}(x)\right)^2\right] = \mathbb{E}_Y\left[\left(\left(f(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)-\left(\hat{f}(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)\right)^2\right] \\ =&\mathbb{E}_Y\left[\left(f(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)^2 - 2\cdot\left(f(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)\cdot\left(\hat{f}(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right) + \left(\hat{f}(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)^2\right]. \end{aligned}$$

Como $f(x)$ y $\mathbb{E}_Y\left[\hat{f}(x)\right]$ son constantes:

$$\begin{aligned} \mathbb{E}_Y\left[\left(f(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)\cdot\left(\hat{f}(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)\right] &= \left(f(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)\cdot\mathbb{E}_Y\left[\left(\hat{f}(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)\right] \\ &= \left(f(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)\cdot\left(\mathbb{E}_Y\left[\hat{f}(x)\right]-\mathbb{E}_Y\left[\hat{f}(x)\right]\right) \\ &= 0\,. \end{aligned}$$

Por lo tanto:

$$\begin{aligned} \mathbb{E}_Y\left[\left(f(x)-\hat{f}(x)\right)^2\right] &= \left(f(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)^2 + \mathbb{E}_Y\left[\left(\hat{f}(x)-\mathbb{E}_Y\left[\hat{f}(x)\right]\right)^2\right] \\ &= \left(\mathrm{Sesgo}(x)\right)^2 + \mathrm{Var}(\hat{f}(x))\,. \end{aligned}$$

Agrupando todos los términos:

$$EPE(x) = \left(\mathrm{Sesgo}(x)\right)^2 + \mathrm{Var}\left(\hat{f}(x)\right) + \sigma^2\,.$$

□

Corolario 1.8. *En las condiciones de la Definición 1.3, se cumple la siguiente igualdad:*

$$EPE = Sesgo^2 + Var\left(\hat{f}\right) + \sigma^2. \tag{1.17}$$

Demostración. Teniendo en cuenta la proposición anterior, las expresiones (1.7) y (1.9) y que σ^2 es una constante:

$$\begin{aligned} EPE = \mathbb{E}_X\left[EPE(X)\right] &= \mathbb{E}_X\left[\left(\text{Sesgo}\left(X\right)\right)^2 + \text{Var}\left(\hat{f}\left(X\right)\right) + \sigma^2\right] \\ &= \text{Sesgo}^2 + \text{Var}(\hat{f}) + \sigma^2\,. \end{aligned}$$

□

Esto explica también que cuando el sesgo cuadrático y la varianza eran cercanos a 0, como ocurre en nuestro ejemplo, el EPE alcance un mínimo no tan reducido. Se debe al valor σ^2, inherente al error y que no podemos evitar. Es decir, σ^2 es una cota inferior del error que se espera cometer ajustando los datos al modelo; es la variabilidad no explicada por el modelo. En todo caso, σ^2 no depende de los parámetros del modelo ajustado y no influye en la elección del modelo óptimo. Otra conclusión que podemos sacar del ejemplo es que, por lo general, la reducción del sesgo implica el aumento de la varianza, y viceversa. Sin embargo, se han estudiado a lo largo de los años varios métodos para intentar reducir la varianza en modelos complejos con poco sesgo.

1.2. Métodos de reducción de varianza

Se presentan a continuación de forma resumida los principales métodos para reducir la varianza del modelo empleado. Los dos primeros, *bagging* y *boosting*, entran dentro del grupo de técnicas de ensamblado de modelos (*ensemble learning*, ver [21]). Estas tienen como objetivo optimizar el rendimiento y la estabilidad de los modelos combinando varios modelos sencillos, conocidos como modelos base simples (*weak learners*).

1.2.1. *Bagging*

El término *bagging* (ver [2]) proviene de la combinación de *bootstrap* y *aggregating*. Para comprender este método, es necesario entender cada una de estas estrategias por separado.

El *bootstrap* (ver [4]) es una técnica empleada para generar nuevas muestras a partir de una muestra original. Esto se hace seleccionando observaciones de la muestra original con reemplazamiento, de forma que una observación pueda aparecer varias veces en una misma muestra. De esta forma, si creamos muestras del mismo tamaño que la original (práctica habitual) se espera que más del 60 % de las observaciones originales se encuentren en cada muestra. En el siguiente resultado presentamos la proporción exacta que se espera encontrar en cada muestra.

Proposición 1.9. *Sea $O = \{x_1, \ldots, x_n\}$ una muestra de n observaciones distintas a partir de la que vamos a generar una nueva muestra $\tilde{O} = \{\tilde{X}_1, \ldots, \tilde{X}_n\}$ del mismo tamaño escogiendo observaciones de forma equiprobable y con reemplazamiento. Entonces,*

$$\frac{\mathbb{E}\left[\left|\widehat{O \cap \tilde{O}}\right|\right]}{n} = 1 - \left(1 - \frac{1}{n}\right)^n , \tag{1.18}$$

donde la parte izquierda de la expresión[4] *es la esperanza del número de observaciones que se encuentran en las dos muestras entre el número total de observaciones. Es decir, la proporción de observaciones de O que aparecerán en $\tilde{O}$.*

Demostración. Sea x_j, $j \in \{1, \ldots, n\}$ una observación cualquiera de O. La probabilidad de que no sea la primera observación seleccionada para $\tilde{O}$ es

$$\mathbb{P}\left[\tilde{X}_1 \neq x_j\right] = 1 - \mathbb{P}\left[\tilde{X}_1 = x_j\right] = 1 - \frac{1}{n}.$$

Como la muestra se genera con reemplazamiento en la muestra original, la probabilidad será la misma para todas las observaciones de $\tilde{O}$. Por lo tanto:

$$\mathbb{P}\left[x_j \notin \tilde{O}\right] = \mathbb{P}\left[\tilde{X}_1 \neq x_j\right] \cdot \cdots \cdot \mathbb{P}\left[\tilde{X}_n \neq x_j\right] = \left(1 - \frac{1}{n}\right)^n .$$

Y se concluye que

$$\mathbb{P}\left[x_j \in \tilde{O}\right] = 1 - \mathbb{P}\left[x_j \notin \tilde{O}\right] = 1 - \left(1 - \frac{1}{n}\right)^n .$$

Como x_j se seleccionó de forma arbitraria, se tiene que el resultado anterior es la esperanza de la proporción de observaciones que acaben en la nueva muestra. □

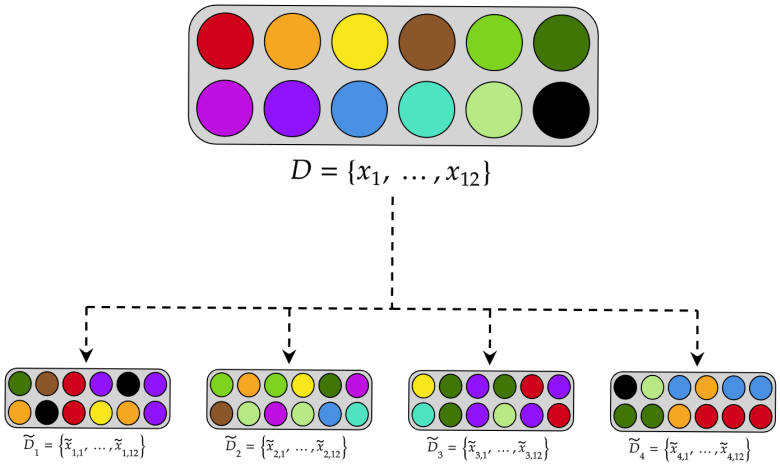


Figura 1.5: Creación de muestras *bootstrap*.

[4] Denotamos al cardinal de un conjunto X como $|X|$.

Del resultado anterior deducimos que para muestras muy grandes se tiene que

$$\frac{\mathbb{E}\left[\left|\widehat{O \cap \tilde{O}}\right|\right]}{n} = 1 - \left(1 - \frac{1}{n}\right)^n \overset{n\to\infty}{\longrightarrow} 1 - \frac{1}{e} \simeq 0.6321\,. \tag{1.19}$$

Esta es la cota inferior para la proporción esperada de observaciones en cada nueva muestra. En la Figura 1.5 se puede ver un ejemplo de cómo se generan estas muestras *bootstrap*.

El objetivo de generar estas nuevas muestras se entiende por el siguiente paso, la agregación de los resultados de los modelos (*aggregating*). Esto consiste en aplicar un modelo sencillo (*weak learner*) sobre cada una de las muestras. Así, la respuesta del modelo final será la media de las predicciones en el caso de regresión y la moda en el caso de clasificación. De esta forma conseguimos varias predicciones para las que no se necesitaron todas las observaciones originales (modelo menos sesgado) que se combinan para conseguir un resultado más "plural" (reducción de varianza). La idea es análoga a cuando se trata de estimar la media de una población, ya que la distribución del estimador tiene una varianza más reducida cuantas más observaciones de la muestra tengamos.

El ejemplo más representantivo de los algoritmos que emplean la técnica de *bagging* es el *Random Forest*. A grandes rasgos, este algoritmo emplea varios árboles de decisión como *weak learners* utilizando muestras *bootstrap*. Además, emplea otro tipo de estrategias para reducir el sobreajuste como que para cada nodo de cada árbol se escoja de entre un subconjunto de las variables explicativas X, como veremos en el Capítulo 3.

1.2.2. *Boosting*

Esta técnica consiste en asignar pesos a las observaciones en función de cómo haya sido la predicción de estas. De esta forma, aplicando de nuevo el modelo, se le dará más importancia a mejorar la predicción de los valores con más error. Para comprender mejor su funcionamiento, nos apoyamos en el ejemplo de la Figura 1.6.

Ejemplo 1.10. Se pretende clasificar correctamente una muestra de 12 observaciones, representadas en las 12 bolas de distintos colores. Suponemos que se aplica la función del modelo $\hat{f}$, y se clasifican de forma correcta 6 observaciones y de forma incorrecta las otras 6. Tras esta primera clasificación, las bolas menos coloreadas, correspondientes a las observaciones clasificadas correctamente, tendrán menos peso que las observaciones mal clasificadas. Tras asignar estos pesos, W, se vuelve a aplicar la función $\hat{f}$. En este caso, 8 observaciones fueron clasificadas de forma

acertada. Se vuelven a asignar pesos acorde a los resultados obtenidos, y se sigue con el proceso. El número de iteraciones puede estar prefijado o puede existir un test de parada. En el caso del ejemplo, solo se aplicaría la función $\hat{f}$ 3 veces.

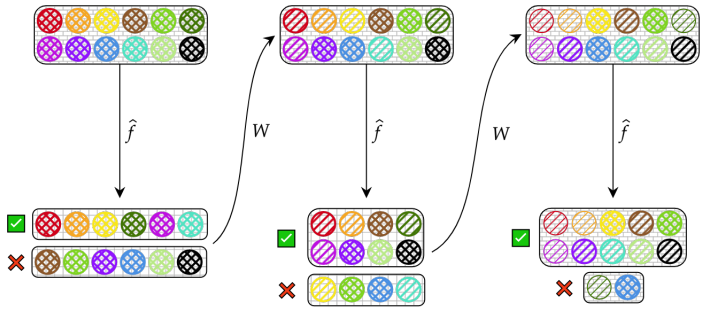


Figura 1.6: Aplicación de un modelo con *boosting*.

A diferencia del *bagging*, que genera modelos en "paralelo", la técnica del *boosting* es "secuencial", porque cada aplicación de la función del modelo $\hat{f}$ depende de la anterior. Otra diferencia es que en esta técnica siempre se emplea la muestra original, mientras que en la anterior se construyen las muestras *bootstrap*.

El empleo del *boosting* se debe principalmente a la reducción de la varianza y el sesgo. Sin embargo, mediante el uso de pesos en la muestra, puede también generar modelos demasiado sesgados, aumentando el riesgo de *overfitting*. Ejemplos habituales de algoritmos que emplean esta técnica son *AdaBoost* (*Adaptative Boosting*, ver [5]), *Gradient Boosting* (ver [20]) y *XGBoost* (*Extreme Gradient Boosting*, ver [22]).

1.2.3. Validación cruzada

Antes de centrarnos en describir la técnica de validación cruzada, es necesario presentar el procedimiento canónico de partición de los datos de la muestra. Lo habitual es seleccionar entre el 70 y el 80 % de los datos como datos de entrenamiento, entre el 10 y el 20 % como datos de validación y entre el 10 y el 20 % como datos de test. Los porcentajes fluctúan en función de las características de la muestra.

Los datos de entrenamiento son los que se aplican al modelo para optimizar los parámetros, que son las variables propias del modelo (en un *Random Forest* serían los criterios de división y de nodo terminal de los árboles de decisión). Los datos de validación entran en juego una

vez finalizado el entrenamiento y sirven para ajustar los hiperparámetros, que son las variables externas al modelo (en un *Random Forest* son el número de árboles o su profundidad máxima). Este paso sirve para evaluar el rendimiento del modelo empleado y prevenir el sobreajuste. Finalmente, los datos de test proporcionan la estimación final de lo bien que se comporta el modelo con nuevos datos que no han sido empleados en el proceso de ajuste. Sirven para medir la generalización del modelo.

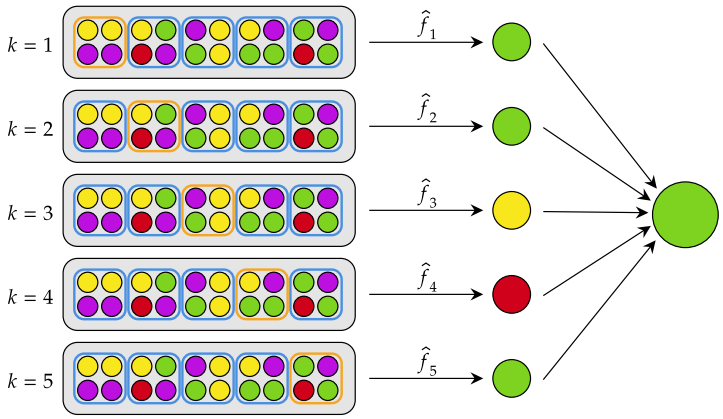


Figura 1.7: Validación cruzada de $k = 5$ iteraciones.

El problema que existe con esta división de los datos es que los subconjuntos son fijos, por lo que no es capaz de resolver el problema del sesgo de forma muy exitosa. Es por ello que entra en juego la validación cruzada. Esta consiste realizar varias particiones distintas para la muestra, de forma que se entrenen distintos modelos, y promediar los resultados (de forma similar al *bagging*). El ejemplo más habitual es el de k iteraciones, que se presenta a través del ejemplo de la Figura 1.7.

Ejemplo 1.11. Se quiere predecir para una observación a partir de una muestra de $n = 20$ observaciones. Se divide la muestra en $k = 5$ subconjuntos y se utiliza en cada caso el subconjunto k como la muestra de validación (naranja). El resto de subconjuntos conforman la muestra de entrenamiento (azul). Se recogen k predicciones y se escoge la moda de estas (en un caso de regresión sería la media), que será la predicción final.

Otros tipos de validación cruzada habituales son el aleatorio y el *leave-one-out* (ver [23]), que consiste en dejar una sola observación como validación de cada vez. Cabe destacar que aunque estas técnicas sirvan para reemplazar a la partición fija presentada antes, lo más habitual es realizar una combinación de ambas técnicas, de forma que la validación cruzada se aplique al subconjunto de la muestra sin las observaciones para el test.

Capítulo 2

Árboles de Decisión

Para comprender el funcionamiento de un modelo *Random Forest* es fundamental conocer el procedimiento de construcción de un árbol de decisión. Estos ya pueden ser considerados un modelo de por sí, pero varios de ellos pueden formar parte de un todo para crear un *bosque de decisión* o *bosque aleatorio*, que es el objetivo al que se queremos llegar. La idea principal de los árboles de decisión es muy intuitiva, y es fácil de entender a través del siguiente ejemplo.

Ejemplo 2.1. Supongamos que queremos predecir el valor que toma una variable Y dicotómica ($Y = 0$ ó $Y = 1$) en función de 3 variables explicativas $\{X_1, X_2, X_3\}$ cuyo dominio es el intervalo $[0, 5]$. Un posible árbol de decisión para este caso se presenta en la Figura 2.1. En ella se pueden ver distintos círculos que representan los nodos del árbol. Debajo de cada círculo se indican las "coordenadas" de cada nodo dentro del árbol, que serán útiles a la hora de describir su funcionamiento. Las flechas indican los caminos que puede seguir una nueva observación para la que se quiere predecir, que están determinadas por los valores de las variables explicativas. Para saber el camino que sigue cada observación, sobre cada nodo está representada una condición; si se cumple se seguirá el camino hacia la izquierda, si no hacia la derecha. Una vez que se llega a un punto en el que el camino se termina, el árbol devolverá la predicción indicada mediante el valor que aparece en el nodo. Evidentemente, los caminos que seguiría cada nueva observación se construyen a partir de los datos originales, en este caso $\{(x_{1i}, x_{2i}, x_{3i}, y_i)\}_{i=1}^{n}$.

Supongamos que queremos predecir para la observación $\tilde{x} = (1, 2, 1)$. El procedimiento empieza en el nodo $(0, 1)$. Como se cumple la condición del nodo $X_2 < 3$, se seguirá el camino de la izquierda hasta el nodo $(1, 1)$. Como también se cumple la condición $X_3 < 2$, se sigue el camino hasta el nodo $(2, 1)$. Finalmente, como la condición $X_2 \leq 1$ no se verifica, se seguirá el camino de la derecha hasta llegar al nodo $(3, 2)$, donde se acaba el recorrido. Por lo tanto, la predicción

para la observación que genera este árbol de decisión es $\hat{y} = 1$. Como se puede ver en el ejemplo, no es necesario que intervengan los valores de todas las variables explicativas para llegar a la predicción.

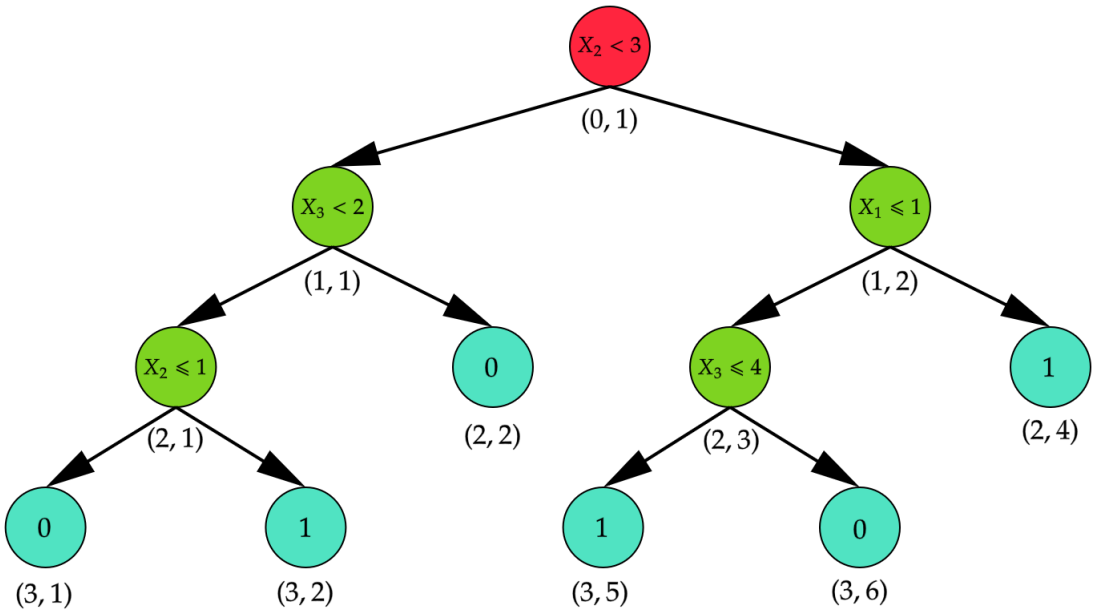


Figura 2.1: Árbol de decisión clasificador.

Podemos también estudiar el problema de forma geométrica. Trasladando el enfoque al espacio en el que se encuentran los vectores de valores que toman las variables explicativas[1], se puede obtener una interpretación más visual de lo que hace el árbol de decisión. Como para cada condición solo entra en juego una variable explicativa, las decisiones de cada nodo no son otra cosa que divisiones del espacio en dos subregiones separadas por un hiperplano. En el ejemplo anterior, el primer nodo divide el espacio con el hiperplano de $\mathbb{R}^3$ $X_2 = 3$, haciendo que las observaciones para las que $X_2 < 3$ vayan por un camino y el resto de observaciones por otro. Entendamos esta interpretación con un ejemplo más fácil de visualizar.

Ejemplo 2.2. Supongamos que ahora tenemos solo dos variables explicativas $\{X_1, X_2\}$ que toman valores en $[-6, 6]$ para predecir una variable Y dicotómica, codificada como $Y = 0$ ó $Y = 1$. La Figura 2.2 muestra cómo actuaría un árbol de decisión fragmentando el espacio generado por los posibles valores que pueden tomar X_1 y X_2.

En primer lugar, se puede apreciar que al tratarse de divisiones de la forma $X_j = c$, siendo

[1] En el caso del ejemplo anterior, estos vectores son de la forma $x = (x_1, x_2, x_3) \in [0, 5]^3 \subset \mathbb{R}^3$.

c una constante, las rectas generadas en el plano solo pueden ser horizontales o verticales; es decir, los cortes solo dependerán de una de las variables[2]. Cada división viene dada por un nodo. De esta forma, los distintos nodos van encasillando las observaciones en distintas regiones, de forma que en función de la región en que se encuentren la predicción será distinta. En el ejemplo podemos ver cómo las predicciones de la variable Y son 0 o 1, pero dentro de cada región la predicción siempre es la misma.

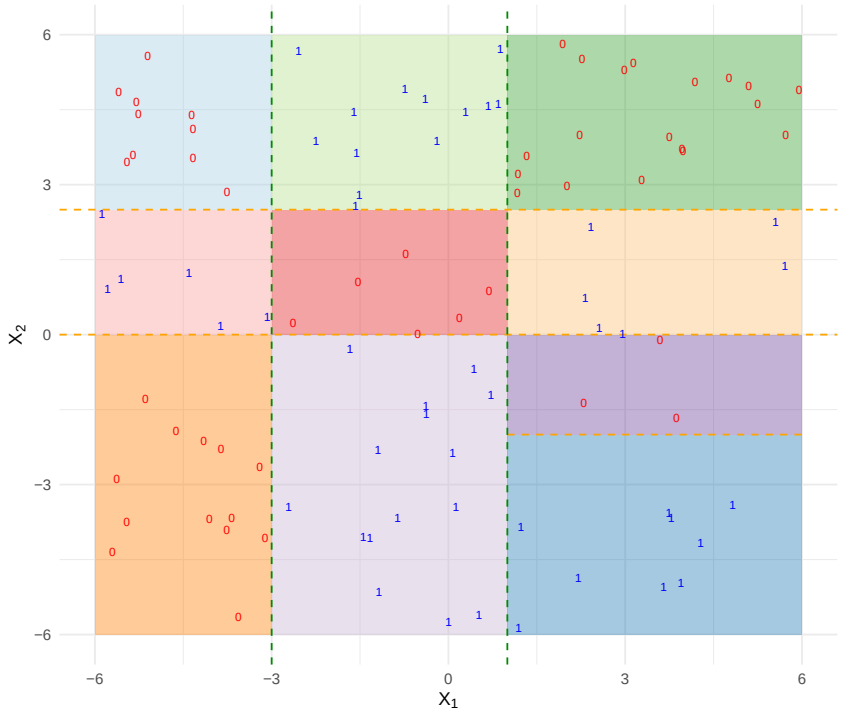


Figura 2.2: Fragmentación generada por un árbol de decisión.

Una vez introducido el funcionamiento básico de un árbol de decisión, es necesaria cierta notación para poder profundizar más en el algoritmo de aprendizaje con conceptos como los criterios de división o los criterios de nodo terminal.

2.1. Nodos del árbol

Empezamos introduciendo la nomenclatura para los distintos tipos de nodos (ver [26]). Estos vienen representados cada uno con un color en la Figura 2.1. Todo lo que se va a presentar a

[2]No tienen por qué realizar la división en todo el espacio. Por ejemplo, si la división está asociada a un nodo que viene de una división anterior, solo se divide la región que corresponde al nodo.

continuación es válido tanto para el caso de clasificación como para el caso de regresión.

- **Nodo raíz (*Root node*)**: Primer nodo del árbol. En este se realiza la primera partición de los datos. Representa la característica más relevante según el criterio de selección que se haya empleado, ya que el resto de divisiones se hace a partir de esta. En la Figura 2.1 se trata del nodo de color rojo.
- **Ramas o nodos internos (*Internal nodes*)**: Puntos de decisión dentro del árbol. Cada nodo impone una nueva condición sobre una de las variables generando un nuevo hiperplano. La diferencia con el nodo raíz radica en el hecho de que los nodos internos dependen de los nodos anteriores (el espacio con el que se trabaja ya ha sido dividido por distintos hiperplanos). En el caso de nodos consecutivos, se denomina **nodo padre** a aquel que se divide en otros dos nodos, que se conocen como **nodos hijos**. En la 2.1, los internos son los nodos de color verde. Tanto el nodo raíz como los nodos internos conforman la categoría de **nodos de decisión (*decision nodes*)**.
- **Hojas o nodos terminales (*Leaf nodes*)**: Nodos finales del árbol. Contienen la clase predicha en el caso de clasificación o el valor de salida en el caso de regresión. En la 2.1 se representan de color azul.

Para entender mejor lo que ocurre en cada nodo deberíamos conocer el criterio de división empleado para el árbol. Este consistirá en calcular el valor de una métrica para cada una de las divisiones posibles en cada variable de forma que se pueda escoger la división más relevante para cada nodo. Dicho valor se calcula a partir de la información conocida de las observaciones $\{(x_i, y_i)\}_{i=1}^{n}$, donde n es el número de observaciones con las que se entrena el modelo.

Los distintos criterios de selección de variable serán explicados la Sección 2.3. Sin embargo, presentaremos ahora un ejemplo general que, aunque no sea útil para la práctica por la escasez de datos, sirve para comprender el marco general del funcionamiento de estos criterios.

Ejemplo 2.3. Supongamos que tenemos dos variables explicativas discretas X_1 y X_2 que toman valores en $\{1, 2, 3, 4, 5\}$ y una variable respuesta dicotómica Y, que toma los valores 0 y 1. Queremos entrenar el modelo con tres observaciones $\{(x_{1i}, x_{2i}, y_i)\}_{i=1}^{3} = \{(2, 2, 0), (4, 3, 1), (1, 5, 0)\}$, de forma que en el nodo raíz tenemos que escoger la mejor división del plano según el criterio que se haya seleccionado. Las cuatro posibles divisiones se pueden ver en la Figura 2.3. Véase, por ejemplo, que la división 1.1 que genera la recta $X_1 = 1.5$ es la misma que la división 2.2, o la que generaría la recta $X_1 = 1.6$, $X_1 = 1.4$ o $X_2 = 4$, pues todas dejan repartidas las observaciones de la misma forma. Ahora supongamos que hemos escogido un criterio de selección que, tras calcular el valor numérico de una métrica para cada división, se queda con la división con el valor

mínimo. Si en este caso fuera, por ejemplo, la división 2.2 (análogamente la 1.1) la que obtuviera el valor mínimo, esta sería la división que se quedaría en el nodo raíz. En el resto de nodos el procedimiento es igual, pero con unas ligeras diferencias que se explicarán a continuación.

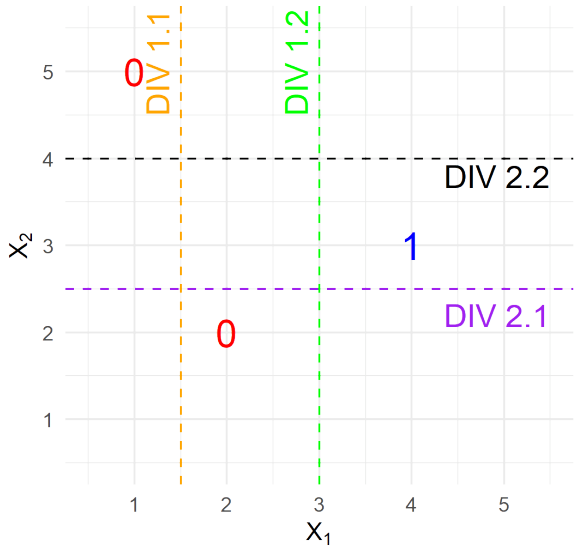


Figura 2.3: Posibles divisiones del plano para el nodo raíz.

Del ejemplo anterior se puede deducir también que para expresar la división de un nodo tan solo necesitamos conocer la variable con la que se va a hacer la división, X_d y el valor de esta para el que se hace el corte, x_d. Este valor no es único, como ya se explicó en el ejemplo anterior, y podemos escogerlo de forma que ninguna observación caiga justo en el hiperplano divisor. De esta forma, se podrá hacer uso siempre del operador $<$, sin necesidad de emplear $\leq$ en ningún caso.

Teniendo lo anterior en cuenta, si queremos definir una función que devuelva la división seleccionada en función del criterio de división y de las observaciones $\{(x_i, y_i)\}_{i=1}^n$ empleadas, bastará con que devuelva un par $(X_d, x_d) \in X \times \mathbb{R}$, con $X = \{X_1, \ldots, X_p\}$ el conjunto de todas las variables explicativas, $1 \leq d \leq p$ y $x_d \in \mathbb{R}$.

Vamos a centrarnos un momento en el conjunto de observaciones para el entrenamiento, que denominaremos $O = \{(x_i, y_i)\}_{i=1}^n$. Estas son las empleadas para obtener la división del nodo raíz. Sin embargo, no todas las observaciones afectan a la división empleada en el resto de nodos, para los que el espacio ya fue previamente segmentado. Solo influirán aquellas que pertenezcan a la región asociada al nodo. Por ejemplo, si la división del nodo raíz resulta ser $X_1 < 2$, las siguientes divisiones dependerán en un nodo de las observaciones que cumplen la condición y en otro nodo las que no la cumplen. Otra forma de verlo es que las observaciones que influyen

en cada nodo son las que, si se les aplicara el árbol para predecir sobre ellas, alcanzarían dicho nodo. Equivalentemente, todos los nodos tienen una región del espacio asociada en la que se encuentran las observaciones que influyen en la selección de la división del nodo (ver Figura 2.4). A continuación se presenta cierta notación que será útil para posteriores desarrollos.

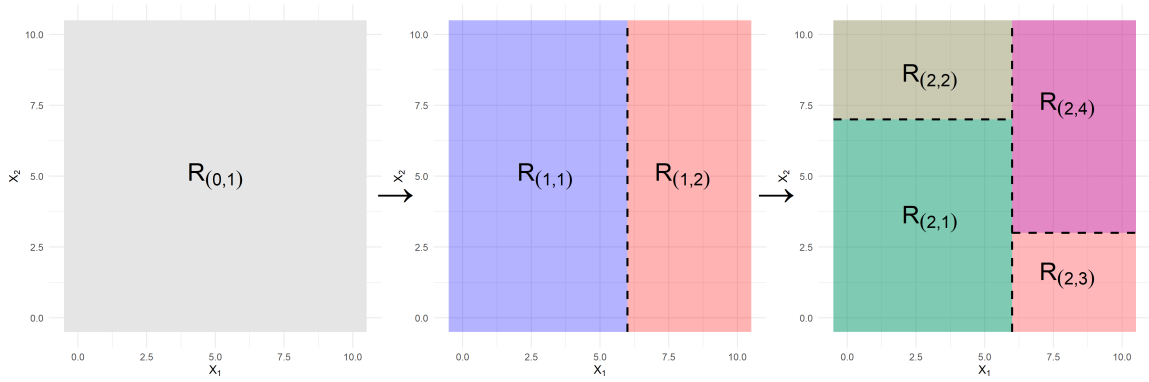


Figura 2.4: Proceso de fragmentación generado por los primeros nodos de un árbol.

Notación 2.4. Para referirnos a un nodo concreto, emplearemos coordenadas de la forma (h, t), donde h se refiere al nivel al que se encuentra el nodo y t a una numeración de los nodos en el propio nivel. Por ejemplo, el nodo raíz tendrá las coordenadas $(0, 1)$, tal y como se vio en la Figura 2.1. Los valores de t vienen determinados de forma que los dos nodos hijos de un nodo (h, t) serán $(h + 1, 2t - 1)$ y $(h + 1, 2t)$.

Véase que según el patrón que seguimos, la existencia del nodo (h, t) no implica la existencia del nodo $(h, t - 1)$ (como en el árbol de la Figura 2.1, donde existe el nodo $(3, 5)$ sin que exista el $(3, 4)$). Esta numeración facilitará la definición de las funciones de los nodos.

Notación 2.5. Sean (h, t) las coordenadas de un nodo. Denotamos como $R_{(h,t)} \subset \mathbb{R}^n$ a la región del espacio generado por las variables $X_1, \ldots, X_n$ asociada al nodo, donde se encuentran las observaciones de las que depende la división seleccionada en dicho nodo.

Nótese que si (h, t) es un nodo por el que hay que pasar[3] para poder llegar al nodo (h', t'), siempre se cumple que $R_{(h',t')} \subset R_{(h,t)}$. Ahora ya podemos pasar a definir los subconjuntos de las observaciones asociadas a cada nodo.

Definición 2.6. Sean (h, t) las coordenadas de un nodo. Denotamos como $O_{(h,t)}$ al subconjunto de las observaciones asociadas al entrenamiento que determinan la división del nodo, es decir:

$$O_{(h,t)} \equiv \{(x_i, y_i) \in O = \{(x_i, y_i)\}_{i=1}^{n} \,|\, x_i = (x_{1i}, \ldots, x_{pi}) \in R_{(h,t)}\} \subset O\,.$$

[3] Aunque cada nodo tiene asociada una condición, se puede también interpretar que van acumulando las condiciones de los nodos anteriores. De esta forma, los nodos que "preceden" a otro y por los que hay que pasar para alcanzarlo son los asociados a sus condiciones acumuladas.

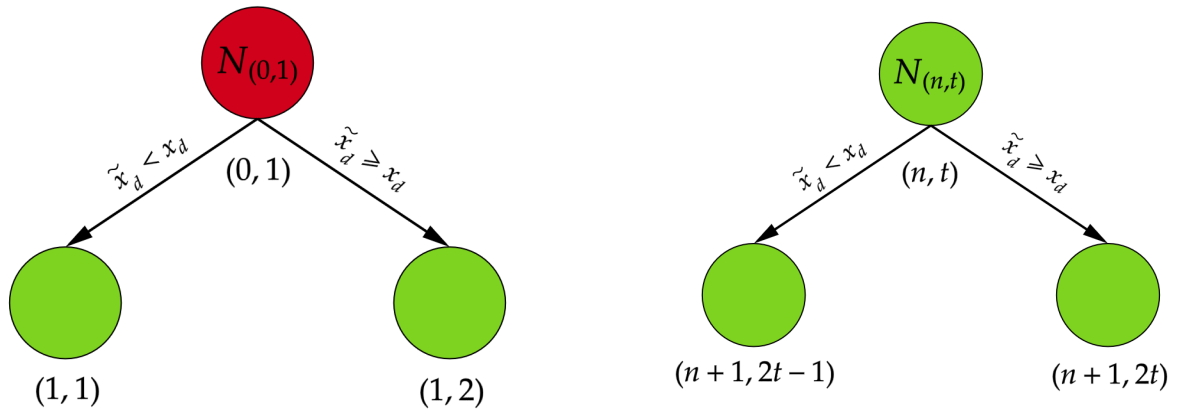
Con toda la información anterior, ya podemos describir cómo actúa de forma general la **función de selección de división** D empleada sobre los nodos. Dadas las observaciones asociadas al nodo (h,t), $O_{(h,t)}$, esta devuelve la variable divisora $X_{d,(h,t)}$ y el valor del corte $x_{d,(h,t)}$.

Notación 2.7. Al par $(X_{d,(h,t)}, x_{d,(h,t)})$ que devuelve la función de selección de división D al aplicarla al nodo (h,t) lo denotaremos como $D_{(h,t)}$:

$$D\left(O_{(h,t)}\right) = (X_{d,(h,t)}, x_{d,(h,t)}) \equiv D_{(h,t)}\,.$$

Una vez que ya ha sido explicado el procedimiento de selección de división (cuyos casos particulares se presentarán más adelante) a través de una función, ya se puede describir de forma sistemática el paso de una nueva observación por un nodo de decisión. Para ello, se definen las funciones de decisión de cada nodo.

Las funciones de decisión de cada nodo son las encargadas de escoger qué camino va a seguir la observación para la que se quiere predecir en función de la división que produce el nodo. Por lo tanto, estas devolverán las coordenadas del siguiente nodo al que saltará la observación. Definimos primero la función de decisión del nodo raíz.



(a) Esquema de nodo raíz.

(b) Esquema de nodo interno.

Figura 2.5: Esquemas de los distintos nodos de decisión.

Definición 2.8. Sea $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_p)$ una observación de las variables explicativas para la que se quiere predecir. Sea D la función de selección de división empleada en el árbol de decisión. Se define la **función de decisión del nodo raíz** como sigue:

$$N_{(0,1)}(\tilde{x}) = \begin{cases} (1,1) & \text{si se cumple la condición dada por } D_{(0,1)}\,, \\ (1,2) & \text{si no se cumple la condición dada por } D_{(0,1)}\,. \end{cases} \tag{2.1}$$

El efecto de la función está representado en la Figura 2.5a. La condición dada por el par $(X_{d,(0,1)}, x_{d,(0,1)}) \equiv (X_d, x_d) \equiv D_{(0,1)}$ se cumple si la componente de $\tilde{x}$ asociada a la variable X_d es menor que x_d. En caso de que se verifique, se avanza al nodo $(1,1)$. En caso de que no se verifique, se avanza al nodo $(1,2)$. Cabe recordar que este nodo y su división es la más importante del árbol, pues todas las demás divisiones dependen de esta.

Pasamos ahora a la definición de la función de decisión de los nodos internos. La principal diferencia con la función anterior es que para obtener $D_{(0,1)}$ se utilizan todas las observaciones del entrenamiento, O, mientras que para conseguir $D_{(h,t)}$ en general solo se trabaja con las observaciones que caigan en la región asociada al nodo, $O_{(h,t)}$.

Definición 2.9. Sea $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_p) \in R_{(h,t)}$ una observación de las variables explicativas para la que se quiere predecir. Sea D la función de selección de división empleada en el árbol de decisión. Se define la **función de decisión del nodo interno** (h,t) como sigue:

$$N_{(h,t)}(\tilde{x}) = \begin{cases} (h+1, 2t-1) & \text{si se cumple la condición dada por } D_{(h,t)}\,, \\ (h+1, 2t) & \text{si no se cumple la condición dada por } D_{(h,t)}\,. \end{cases} \tag{2.2}$$

De nuevo, podemos apoyarnos en la Figura 2.5b para entenderlo mejor. Si se cumple que $\tilde{x}_d < x_d$, el camino sigue por la izquierda. En caso contrario, por la derecha.

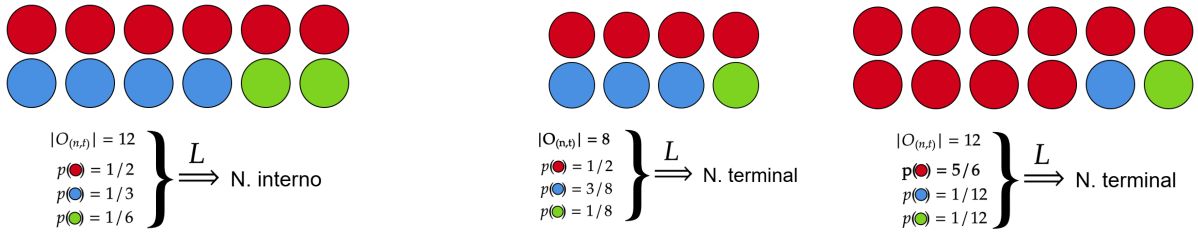
Finalmente, falta describir las funciones de los nodos terminales. Estas deberán devolver el valor que predecirá el árbol de decisión para la observación $\tilde{x}$. En este caso, sí que es necesario distinguir entre problemas de clasificación y problemas de regresión, puesto que la forma de predecir es distinta en cada caso.

Supongamos que ya se ha pasado el nodo raíz en el proceso de construcción del árbol. En cada nodo del árbol habrá que seguir una pauta para decidir si es un nodo interno o un nodo terminal. La elección del tipo de nodo vendrá dada por un criterio de nodo terminal. Por ejemplo, se puede escoger que el nodo (h,t) sea terminal cuando el número de observaciones de entrenamiento en $O_{(h,t)}$ sea menor o igual que 10. Los criterios pueden variar en función del tipo de problema al que nos enfrentemos. Por lo general, denotaremos al criterio escogido como L.

Nótese que L representa una condición lógica[4], es decir, se verifica o no se verifica (0 ó 1). Sin embargo, también puede representar una combinación de condiciones. Por ejemplo, en el caso

[4]Por lo general, aplicada a los conjuntos $O_{(h,t)}$.

de un problema de clasificación, L puede consistir en comprobar que el número de elementos de $O_{(h,t)}$ sea menor o igual que 10 y, en caso de no cumplirse, verificar si para alguno de los valores respuesta la proporción de observaciones asociadas supera el 75 %. La Figura 2.6 representa esta casuística. Cada bola representa una observación de $O_{(h,t)}$ y los colores son los valores que toma la variable respuesta.



(a) $|O_{(h,t)}| > 10$ y $\frac{1}{2} < \frac{3}{4}$. (b) $|O_{(h,t)}| \leq 10$. (c) $|O_{(h,t)}| > 10$ pero $\frac{5}{6} > \frac{3}{4}$.

Figura 2.6: Ejemplo de aplicación del criterio L.

Una vez fijado el criterio de nodo terminal para el árbol, ya es posible definir la función de nodo terminal. Esta ha de ser aplicada a cada nodo del árbol que no sea el nodo raíz, y funciona de forma un poco distinta a las anteriores. En este caso, la función devolverá las coordenadas del propio nodo[5] si no se cumple la condición L o la predicción en el caso de que se cumpla L. Su definición varía dependiendo de si se trata de un problema de regresión o de clasificación.

Para los problemas de clasificación la función de nodo terminal tendrá que devolver una de las posibles clases. Habitualmente, la clase escogida será la que más se repita entre las observaciones de $O_{(h,t)}$, siendo (h,t) las coordenadas del nodo terminal. Por lo tanto, lo que devolverá la función en caso de que se cumpla L será la moda de las clases de las observaciones.

Notación 2.10. Si se tiene una muestra de n observaciones $y = (y_1, \ldots, y_n)$ de la variable discreta Y, se denota a la moda de dicha muestra, es decir, el valor que más se repite en la muestra, como $\text{mod}(y)$.

Definición 2.11. Sean $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n) \in R_{(h,t)}$ una observación de las variables explicativas para la que se quiere predecir, L el criterio de nodo terminal fijado para el árbol de decisión e $y_{(h,t)} = (y_{1_{(h,t)}}, \ldots, y_{n_{(h,t)}})$ el vector de respuestas correspondiente a las observaciones $O_{(h,t)}$, donde $n_{(h,t)} = \left|O_{(h,t)}\right|$. Se define la **función de nodo terminal (para clasificación)** de (h,t)

[5]De esta forma, se indica que el siguiente nodo al que hay que aplicarle la función de decisión de nodo interno será el mismo al que se le ha aplicado la función de nodo terminal.

como sigue:

$$F_{(h,t)}(\tilde{x}) = \begin{cases} \hat{y} \equiv \text{mod}(y_{(h,t)}) & \text{si se cumple la condición } L \text{ sobre } O_{(h,t)}\,, \\ (h,t) & \text{si no se cumple la condición } L \text{ sobre } O_{(h,t)}\,. \end{cases} \tag{2.3}$$

De esta forma, la función anterior ha de ser la primera en aplicarse al alcanzar un nodo en el camino. Una vez obtenida su respuesta, si se cumplió L se finaliza el camino con la respuesta obtenida y si no se cumplió se sigue con la función de decisión de nodo interno una vez conocida la división dada por $D_{(h,t)}$. El esquema de la Figura 2.7 facilita la comprensión de lo descrito. Para la observación $\tilde{x}$ para la que se quiere predecir, denotaremos como $\hat{y}$ el valor que resulta de aplicar el árbol de decisión. De esta manera, en el caso de clasificación, se concluye que $\hat{y} \equiv \text{mod}(y_{(h,t)})$, siendo (h,t) el nodo terminal del recorrido que siguió $\tilde{x}$.

En el caso de que el problema sea de regresión, la función de nodo terminal no difiere mucho de la anterior. Hay que tener en cuenta que ahora se tiene un continuo de valores posibles, por lo que escoger una moda no tendría mucho sentido. El valor que se empleará como predicción no será otro que la media aritmética de las respuestas de las observaciones de $O_{(h,t)}$.

Notación 2.12. Si se tiene una muestra de n observaciones $y = (y_1, \ldots, y_n)$ de la variable continua Y, se denota a la media aritmética de dicha muestra como $\bar{y}$, es decir,

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i\,. \tag{2.4}$$

Definición 2.13. Sean $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_p) \in R_{(h,t)}$ una observación de las variables explicativas para la que se quiere predecir, L el criterio de nodo terminal fijado para el árbol de decisión e $y_{(h,t)} = (y_1, \ldots, y_{n_{(h,t)}})$ el vector de respuestas correspondiente a las observaciones $O_{(h,t)}$, donde $n_{(h,t)} = \left|O_{(h,t)}\right|$. Se define la **función de nodo terminal (para regresión)** de (h,t) como sigue:

$$F_{(h,t)}(\tilde{x}) = \begin{cases} \hat{y} \equiv \bar{y}_{(h,t)} & \text{si se cumple la condición } L \text{ sobre } O_{(h,t)}\,, \\ (h,t) & \text{si no se cumple la condición } L \text{ sobre } O_{(h,t)}\,. \end{cases} \tag{2.5}$$

Todo el proceso descrito para el caso de clasificación es análogo para el caso de regresión. Ahora la predicción para $\tilde{x}$ es $\hat{y} = \bar{y}_{(h,t)}$, siendo (h,t) el nodo terminal del recorrido que siguió la observación. Es bueno observar que la predicción dependerá en gran medida de la condición L que se escoja, por lo que esta cobra mucha importancia dentro del modelo, y es un hiperparámetro que se tiene que tener en cuenta a la hora de mejorar el desempeño del árbol.

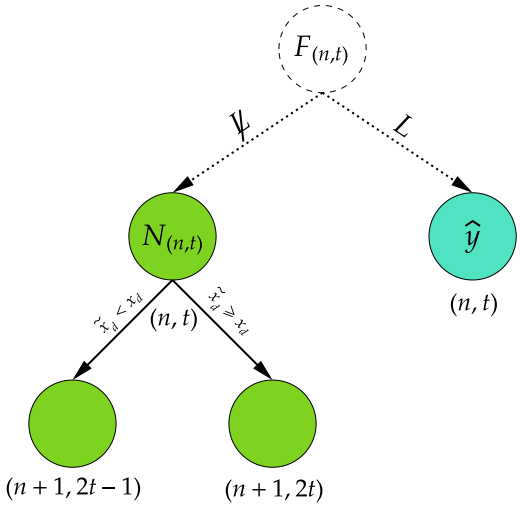


Figura 2.7: Esquema de aplicación de la función de nodo terminal.

2.2. Algoritmo del árbol de decisión

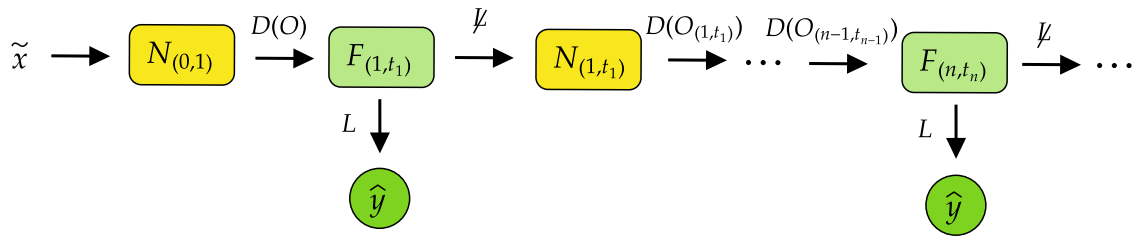


Figura 2.8: Esquema del algoritmo del árbol de decisión.

Una vez definida la estructura de nodos del árbol con sus respectivas funciones, se puede resumir el funcionamiento del árbol como predictor para una observación $\tilde{x}$ con el siguiente esquema (simplificado en la Figura 2.8):

Algoritmo 2.14. *Árbol de decisión.*

1. *Se fijan la función de selección de división D y el criterio de nodo terminal L.*
2. *Se aplica la función de decisión del nodo raíz $N_{(0,1)}$.*
3. *Se aplica la función de nodo terminal para el nodo cuyas coordenadas sean el resultado de la operación anterior: $F_{(1,1)}$ (ó $F_{(1,2)}$).*

3.1. Si se cumple L para $F_{(1,1)}$ (ó $F_{(1,2)}$), este nodo será el terminal y devuelve el valor de la predicción.

3.2. Si no se cumple L, se utiliza la función de decisión del nodo interno $N_{(1,1)}$ (ó $N_{(1,2)}$).

4. En caso ocurrir 3.2. *para el nodo (h,t), se emplea la función de nodo terminal al nodo cuyas coordenadas sean el resultado de la función de decisión de (h,t): $F_{(h+1,2t-1)}$ (ó $F_{(h+1,2t)}$).*

4.1. Caso análogo al 3.1.

4.2. Si no se cumple, se aplica la función de decisión del nodo interno ($N_{(h+1,2t-1)}$ ó $N_{(h+1,2t)}$)

5. En caso de alcanzar 4.2., *repetir el paso* 4 *hasta conseguir llegar al caso* 4.1.

Por supuesto, sabemos que el bucle del algoritmo es finito porque el número de observaciones de entrenamiento son finitas, por lo que las divisiones que se realicen y el número de nodos lo serán también. La cantidad de divisiones y de nodos dependerá de las restricciones del criterio de nodo terminal.

2.3. Criterios de división

Una vez entendido el funcionamiento del algoritmo, ya es posible extender el estudio a los hiperparámetros de los que depende el desempeño del árbol. Uno de ellos es el criterio de división, que denotamos como D, mediante el que se escoge la variable y el valor de corte que generarán la división del espacio en el que se encuentran las observaciones.

Los criterios de división clasifican a los cortes posibles calculando el valor de una métrica para cada uno a partir de los errores cometidos en las predicciones. Las métricas serán distintas para el caso de clasificación y para el caso de regresión. A continuación, se presentan los criterios de división más empleados históricamente tanto para los problemas de clasificación como para los de regresión.

2.3.1. Criterios de división para clasificación

- **Índice de Gini**
  Mide la impureza de cada una de las ramas que generaría el nodo con la división, es decir, la falta de homogeneidad de las clases de la variable respuesta en las observaciones que llegan a cada rama. Veamos cómo calcularlo apoyándonos en el ejemplo de la Figura 2.9.

Ejemplo 2.15. Contamos con una muestra de 100 observaciones generadas con para dos variables explicativas X_1 y X_2 y una respuesta Y con tres clases: rojo, azul o verde. El corte para el que se va a calcular el índice se realiza en $X_1 = 0.5$.

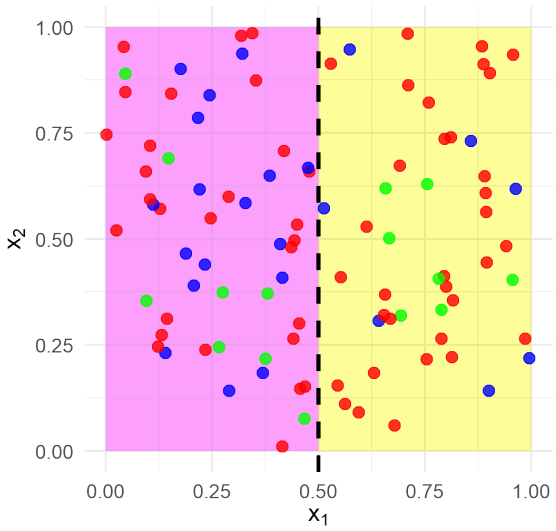


Figura 2.9: División $X_1 = 0.5$ para la que se calcula el índice de Gini.

El número de observaciones de cada clase que caen a cada lado de la recta $X_1 = 0.5$ se presentan en el Cuadro 2.1. Con estos valores es suficiente para poder calcular el índice de Gini de cada rama, que se define de la siguiente forma:

Definición 2.16. Sea $\{1, \ldots, N\}$ el conjunto de índices asociados a cada una de las clases de la variable respuesta Y. Sea p_i la proporción de observaciones de la clase i en una de las regiones en las que divide el nodo el espacio. Se define el **índice de Gini de la rama** como

$$\text{Gini} = 1 - \sum_{i=1}^{N} p_i^2 \,. \tag{2.6}$$

De la definición anterior se deduce fácilmente que cuanto mayor sea la discrepancia entre las clases de las observaciones, mayor será el índice de Gini. Es por esto que el criterio busca minimizar el valor.

	Rojo	Azul	Verde
$X_1 < 0.5$	28	17	8
$X_1 > 0.5$	33	7	7

Cuadro 2.1: Frecuencias de clases para la división $X_1 = 0.5$.

Para el ejemplo, obtenemos un valor de 0.595 para la región de la izquierda y 0.463 para la

región de la derecha. Sin embargo, contamos con 53 valores en la izquierda y 47 valores en la derecha, por lo que la región de la izquierda debería tener un peso ligeramente mayor a la hora de dar un valor final del índice de Gini para la división. Es por ello que se usa una media ponderada de los dos valores, tal y como se define a continuación:

Definición 2.17. Sea una división de nodo interno en la que n_1 observaciones caen en una región y n_2 en la otra. Sean Gini_1 y Gini_2 los índices de Gini calculados en cada rama. Se define el *índice de Gini de la división* como:

$$\overline{\text{Gini}} = \frac{n_1}{n_1+n_2} \cdot \text{Gini}_1 + \frac{n_2}{n_1+n_2} \cdot \text{Gini}_2 \,. \tag{2.7}$$

Este valor se calculará para todas las divisiones posibles. La división con menor índice de Gini será la seleccionada para el nodo. Para nuestro ejemplo, el valor calculado para la división es 0.533.

- **Entropía y ganancia de información**

  Este criterio está basado en la teoría de la información de Shannon, siendo la expresión de la entropía empleada en otras ramas de la ciencia como la *Mecánica Estadística* (ver [16]). En este caso, se calculará la entropía en el nodo en el que se va a hacer la división (nodo padre) además de en los nodos resultado de la división (nodos hijo). La entropía de define de la siguiente forma:

  Definición 2.18. Sea $\{1, \ldots, N\}$ el conjunto de índices asociados a cada una de las clases de la variable respuesta Y. Sea p_i la proporción de observaciones de la clase i en la región asociada al nodo. Se define la **entropía** del nodo como

$$S = -\sum_{i=1}^{N} p_i \log_2(p_i) \,. \tag{2.8}$$

  Como el logaritmo en base 2 se evalúa en valores entre 0 y 1, siempre resultará negativo. Por lo tanto, la entropía de un nodo siempre será un valor positivo. Además, esta será mayor cuando más dispersas estén las clases en el nodo.

  En el ejemplo anterior, en el nodo padre tenemos 100 observaciones: 61 rojas, 24 azules y 15 verdes. Estas proporciones resultan en un valor de la entropía de 1.340. Para los nodos hijo, resulta en 1.424 para el de la izquierda y 1.177 para el de la derecha.

  Ahora bien, para comparar entre las posibles divisiones se busca la que aporte mayor información. Como la entropía es un reflejo de la cantidad de información de un problema, siendo esta menor cuanto mayor información, lo que se buscará es aumentar la diferencia entre la entropía del nodo padre y una ponderación de las entropías de los nodos hijo. Es decir, el objetivo consiste en reducir la entropía del problema.

Definición 2.19. Sea una división de un nodo interno en la que n_1 observaciones van a una rama y n_2 a la otra. Sean S_p la entropía del nodo padre y S_1, S_2 las entropías de los nodos hijo calculadas. Se define la **ganancia de información** de la división como

$$\text{Ganancia} = S_p - \left(\frac{n_1}{n_1 + n_2} \cdot S_1 + \frac{n_2}{n_1 + n_2} \cdot S_2 \right) . \tag{2.9}$$

Se seleccionará la ramificación que tenga la mayor ganancia de información. En el ejemplo, la ganancia de información es de 0.032.

2.3.2. Criterios de división para regresión

En el caso de los problemas de regresión el enfoque es ligeramente diferente al de los problemas de clasificación. En este caso, se asociará una predicción a cada una de las ramificaciones de un nodo. El objetivo será encontrar la división que minimice los errores obtenidos con esas predicciones. Los criterios se diferenciarán en la función de error que se quiera minimizar. La predicción para cada ramificación será evidentemente la media ponderada de respuestas de las observaciones. Para ilustrar de forma clara el criterio más empleado, nos apoyaremos en el ejemplo de la Figura 2.10.

Ejemplo 2.20. Contamos con una muestra de 25 observaciones generadas con de una variable X tomando valores en $[0, 10]$ y una variable respuesta Y continua. El corte que se va a estudiar viene dado por $X = 6$, dividiendo el espacio en dos regiones. En cada una de las regiones se ha pintado una recta horizontal que representa el valor medio de las observaciones dentro de cada región. Estas serán importantes a la hora de visualizar los errores. Para la región de la izquierda el valor es 4.06 y para la derecha 66.52.

- **Error cuadrático medio (MSE)**

  Al igual que en un modelo de regresión lineal como el descrito con la expresión (1.1), el objetivo será minimizar la suma ponderada de los residuos. Estos no son otra cosa que la distancia vertical entre los puntos de las observaciones y la media de cada región representados en la gráfica. Sin embargo, para que la suma de residuos positivos y negativos no se cancele, se trata de minimizar la suma de los cuadrados de dichas distancias.

  Definición 2.21. Sea $y = (y_1, \ldots, y_n)$ una muestra de n observaciones. Definimos el **error cuadrático medio** de la muestra tal que

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2 . \tag{2.10}$$

Por lo tanto, para un nodo (h,t) se escogerá la división que minimice la suma de los errores cuadráticos medios de cada ramificación, es decir:

$$MSE_{(h,t)} = \sum_{j=1}^{2} \frac{1}{n_j} \sum_{i=1}^{n_j} (y_{ji} - \bar{y}_j)^2 \ , \tag{2.11}$$

donde el subíndice j es indicador de la ramificación. Es decir, n_1 y n_2 será el número de observaciones en cada ramificación; y_{1i}, $i \in \{1, \ldots, n_1\}$ las observaciones de la ramificación 1; y_{2i}, $i \in \{1, \ldots, n_2\}$ las observaciones de la ramificación 2 e $\bar{y}_1$, $\bar{y}_2$ sus respectivas medias.

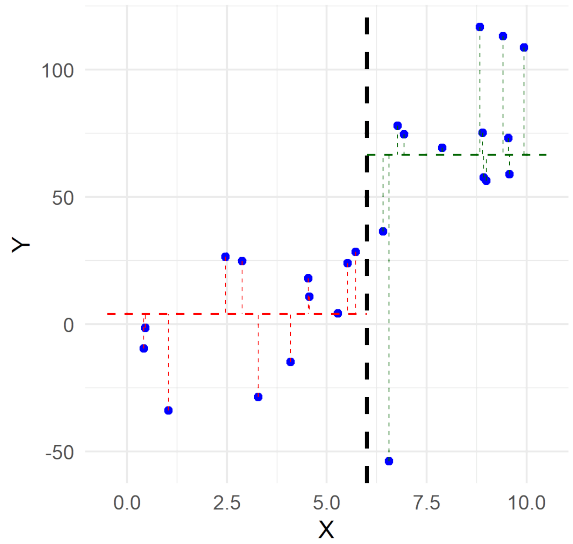


Figura 2.10: División $X = 6$ para la que se calcula el MSE.

En el ejemplo de la Figura 2.10, el error para la ramificación del lado izquierdo es 330.14 y para el lado derecho 1170.83. Por lo tanto, el error cuadrático medio asociado a la división será 1500.97.

2.4. Criterios de nodo terminal

El criterio de nodo terminal de un árbol consiste en una especie de test de parada del algoritmo. Este viene dado por una condición L, como se explicó en la Sección 2.1, deteniéndose el proceso al llegar a un nodo en el que se cumpla la condición establecida. En el Anexo I se presentan los criterios de parada más habituales dentro de los árboles de decisión. Estos tienen el objetivo de controlar el número de ramificaciones de los árboles, acorde con lo visto en el Capítulo 1 sobre la necesidad de reducir la variabilidad en modelos demasiado complejos, con tendencia al *overfitting*.

Capítulo 3

Random Forest

Presentados ya los métodos de reducción de varianza más importantes y los árboles de decisión, procedemos a introducir el modelo *Random Forest* presentado por Breiman en [8]. Los árboles de decisión de forma individual son modelos con sesgo reducido (sobre todo si los criterios de nodo terminal son poco restrictivos), pero con una varianza muy elevada, ya que un ligero cambio en las variables de la observación puede llevar a escoger una predicción asociada a otro nodo terminal. Es por ello que este tipo de modelos son candidatos ideales para aplicar una técnica de reducción de varianza, como es el *bagging*.

Un modelo de *Random Forest* consiste en un ensamblado de árboles de decisión de los que se escoge una predicción global (media en el caso de regresión y moda en el caso de clasificación) a partir de las predicciones de cada uno de los *weak learners*, entrenados a partir de muestras *bootstrap*. Además, veremos a continuación que nos interesa reducir la correlación de los árboles para obtener mejores cotas de error, por lo que añadiremos una propiedad al ensamblado de árboles que ayudará con este objetivo.

3.1. Descripción del algoritmo

Empezamos introduciendo la notación que emplearemos a lo largo de la sección. La función asociada a cada árbol $b \in \{1, \ldots, B\}$, cuyas características presentaremos ahora, vendrá dada por T. Esta dependerá tanto de la muestra *bootstrap* empleada para entrenar al árbol, $\tilde{O}_b$, como del valor de las variables explicativas para la observación sobre la que se quiere predecir, $x = (x_1, \ldots, x_p)$. También dependerá, lógicamente, de los criterios de división D y de nodo terminal L seleccionados de entre los presentados en la Sección 2.3 y el Anexo I.

A diferencia del caso individual, ya vimos que para los árboles del *Random Forest* entra en juego cierta aleatoriedad dada por la muestra bootstrap. Sin embargo, esta no será la única fuente de aleatoriedad, ya que en cada nodo la selección de la división se realizará con un subconjunto de las p variables explicativas, seleccionando las variables de forma equiprobable. El algoritmo es el siguiente (ver [30]):

Algoritmo 3.1. *Random Forest.*
Sea B el número de árboles que conforman el ensamblado. Sean $X = (X_1, \ldots, X_p)$ las variables explicativas e Y la variable respuesta (ya sea un problema de clasificación o de regresión). Sea $O = \{(x_i, y_i)\}_{i=1}^n$ la muestra de entrenamiento. Suponemos que ya se han seleccionado el criterio de división D y el criterio de nodo términal L para los árboles, y que se ha fijado el número de variables a seleccionar en cada nodo de forma equiprobable, $m \leq p$.

1. *Para b desde 1 hasta B:*

   - *Se crea una muestra bootstrap $\tilde{O}_b$ con n observaciones a partir de la muestra de entrenamiento O (ver Sección 1.2.1).*
   - *Construimos el árbol, $T(X, \Theta_b)$, a partir de la muestra bootstrap, teniendo en cuenta los criterios de división y de nodo terminal. Para cada nodo, se escogen de X m variables de forma aleatoria y equiprobable para aplicar el criterio de división. La variable Θ_b de la que depende la predicción del árbol b recoge la siguiente información:*
     - *Aleatoriedad de la muestra bootstrap $\tilde{O}_b$.*
     - *Aleatoriedad de las m variables seleccionadas para cada nodo.*
     - *Criterio de división D y criterio de nodo terminal L (ver Sección 2.1).*

     De esta forma, $\Theta_b \equiv (\tilde{O}_b, m, D, L)$ recopila todas las variables distintas de X de las que depende la predicción del árbol b.

2. *Realizamos el ensamblado de los B árboles, $\{T(X, \Theta_b)\}_{b=1}^B$. De esta forma, fijando x, la predicción será:*

   - *Para regresión:*
   $$\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^{B} T(x, \Theta_b)\,. \tag{3.1}$$
   - *Para clasificación:*
   $$\hat{f}_{rf}^B(x) = mod\left(\{T(x, \Theta_b)\}_{b=1}^B\right)\,. \tag{3.2}$$

Cabe destacar que los valores habituales para el número de variables son $m = \lfloor \sqrt{p} \rfloor$ en el caso de clasificación y $m = \lfloor \frac{p}{3} \rfloor$ en el caso de regresión (ver [1]).

Emplear muestras *bootstrap* y generar subconjuntos aleatorios de las variables es necesario para reducir la correlación entre los árboles sin aumentar demasiado su sesgo individual. De estos dos conceptos dependen los errores a la hora de predecir, como estudiaremos a continuación.

3.2. Convergencia, correlación y varianza

Por simplicidad, empezamos presentando los resultados para el caso de clasificación, que se podrán extrapolar sin mayor dificultad al caso de regresión (ver Breiman [8]). Para comenzar a enfocar el problema, presentamos la siguiente definición formal para el modelo (clasificador):

Definición 3.2. Un clasificador *Random Forest* consiste en una colección de árboles de decisión $\{T(x, \Theta_b)\}_{b=1}^{B}$ donde los $\{\Theta_b\}_{b=1}^{B}$ son vectores aleatorios independientes e idénticamente distribuidos (i.i.d.) y cada árbol emite un voto unitario para la clase más popular para x.

En primer lugar, definiremos un error y enunciaremos un resultado mediante el cual se asegura su convergencia casi segura[1] (c.s.). De esta forma, podremos pasar del estudio de medias muestrales al estudio de probabilidades para el caso asintótico ($B \to \infty$). Así, para dicho error en forma asintótica, encontraremos una cota superior que lo limita. Comenzamos definiendo la función de margen:

Definición 3.3. Sea $\{T(X, \Theta_b)\}_{b=1}^{B}$ un ensamblado de árboles de decisión clasificadores. Suponiendo que la muestra de entrenamiento fue generada de forma aleatoria a partir de la distribución de Y en función de X, se define la **función de margen** tal que

$$mg(X, Y) := \frac{1}{B} \sum_{b=1}^{B} I\left(T(X, \Theta_b) = Y\right) - \max_{j \neq Y} \left[\frac{1}{B} \sum_{b=1}^{B} I\left(T(X, \Theta_b) = j\right) \right], \tag{3.3}$$

donde I es la función indicadora, que devuelve 1 si se cumple la condición y 0 si no se cumple. Esta variable mide la diferencia entre la proporción de aciertos y la proporción de selecciones de la clase errónea más votada. Nos interesa que este valor sea positivo y cuanto más alto mejor.

Definición 3.4. Definimos el **error de generalización** como la probabilidad de que la función de margen sea negativa:

$$PE^* := \mathbb{P}_{X,Y}\left[mg(X, Y) < 0\right]. \tag{3.4}$$

[1] La sucesión de variables aleatorias $\{X_n\}$ converge de forma **casi segura** a X si

$$\mathbb{P}\left[\lim_{n \to \infty} X_n = X\right] = 1.$$

Este valor refleja la probabilidad de realizar una predicción errónea dada la distribución de Y respecto de las variables de X y el modelo empleado. Nuestra intención es, evidentemente, mantenerlo reducido. Ahora bien, si supiéramos que convergiera al aumentar B, podríamos afirmar que el modelo no se sobreajusta al añadir árboles, porque el error se estabilizaría a dicho valor. La idea se recoge en el Teorema 3.5, que se puede demostrar aplicando la *Ley fuerte de los grandes números* (ver [15]):

Teorema 3.5. *Si el número de árboles B tiende a infinito, PE^* converge de forma casi segura:*

$$PE^* \xrightarrow[B\to\infty]{c.s.} \mathbb{P}_{X,Y}\left[\mathbb{P}_\Theta\left[T(X,\Theta)=Y\right] - \max_{j\neq Y}\left\{\mathbb{P}_\Theta\left[T(X,\Theta)=j\right]\right\}\right]. \tag{3.5}$$

Gracias a este resultado podemos definir los conceptos anteriores en forma "asintótica":

Definición 3.6. Sea $T(X,\Theta)$ una variable aleatoria que representa la predicción de un árbol del modelo. Se define la **función de margen asintótico** tal que

$$mr(X,Y) := \mathbb{P}_\Theta\left[T(X,\Theta)=Y\right] - \max_{j\neq Y}\left\{\mathbb{P}_\Theta\left[T(X,\Theta)=j\right]\right\}. \tag{3.6}$$

Definición 3.7. Definimos el **error de generalización asintótico** como la probabilidad de que la función de margen asintótica sea negativa:

$$PE_a^* := \mathbb{P}_{X,Y}\left[mr(X,Y)<0\right]. \tag{3.7}$$

Para el resultado que queremos probar, también nos será útil definir la fuerza de un conjunto de clasificadores, que plasma la capacidad de cada clasificador por separado para predecir de forma correcta. Para el desarrollo necesitamos definir también la función de margen bruto, que es análoga a la función de margen pero para un solo clasificador.

Definición 3.8. La **fuerza** de un conjunto de clasificadores $\{T(X,\Theta)\}$ viene dada por

$$s := \mathbb{E}_{X,Y}\left[mr(X,Y)\right]. \tag{3.8}$$

Definición 3.9. Denotando a la clase errónea con más probabilidad de ser predicha de la siguiente forma:

$$\hat{\jmath}(X,Y) := \arg\left[\max_{j\neq Y}\left\{\mathbb{P}_\Theta\left[T(X,\Theta)=j\right]\right\}\right],$$

se define la **función de margen bruto** como

$$rmg(\Theta,X,Y) := I(T(X,\Theta)=Y) - I(T(X,\Theta)=\hat{\jmath}(X,Y)), \tag{3.9}$$

de forma que

$$mr(X,Y) = \mathbb{E}_\Theta\left[rmg(\Theta,X,Y)\right]. \tag{3.10}$$

Ya contamos con toda la notación necesaria para demostrar un resultado (presentado en [8]) que da una cota del error generalizado asintótico. Para ello, haremos uso de la *desigualdad de Chébishev* (ver [14]) y de la *desigualdad de Jensen* (ver [18]), que presentamos como lemas.

Lema 3.10. *(Desigualdad de Chébishev) Sea X una variable aleatoria con varianza finita: $Var(X) < \infty$. Entonces*

$$\mathbb{P}\left[|X - \mathbb{E}\left[X\right]| \geq \varepsilon\right] \leq \frac{Var(X)}{\varepsilon^2}\,,\ \forall \varepsilon > 0\,. \tag{3.11}$$

Lema 3.11. *(Desigualdad de Jensen) Sea X una variable aleatoria y f una función convexa definida en el dominio de X. Entonces*

$$f\left(\mathbb{E}\left[X\right]\right) \leq \mathbb{E}\left[f\left(X\right)\right]\,. \tag{3.12}$$

Teorema 3.12. *Suponiendo $s \geq 0$, una cota superior del error de generalización asintótico viene dada por*

$$PE_a^* \leq \frac{\bar{\rho}}{s^2} \cdot \left(1 - s^2\right), \tag{3.13}$$

donde

$$\bar{\rho} = \frac{\mathbb{E}_{\Theta,\Theta'}\left[\rho_{X,Y}\left(rmg\left(\Theta, X, Y\right), rmg\left(\Theta', X, Y\right)\right) \cdot sd_{X,Y}\left(rmg\left(\Theta, X, Y\right)\right) \cdot sd_{X,Y}\left(rmg\left(\Theta', X, Y\right)\right)\right]}{\mathbb{E}_{\Theta,\Theta'}\left[sd_{X,Y}\left(rmg\left(\Theta, X, Y\right)\right) \cdot sd_{X,Y}\left(rmg\left(\Theta', X, Y\right)\right)\right]} \tag{3.14}$$

es el valor medio de las correlaciones entre las funciones de margen bruto.

Demostración. Aplicando la *desigualdad de Chébishev* a $mr(X,Y) \equiv mr$ y $s = \mathbb{E}_{X,Y}\left[mr\right] \geq 0$:

$$\mathbb{P}_{X,Y}\left[|mr - \mathbb{E}_{X,Y}\left[mr\right]| \geq s\right] = \mathbb{P}_{X,Y}\left[|mr - s| \geq s\right] \leq \frac{\mathrm{Var}(mr)}{s^2}\,.$$

De esta forma:

$$\begin{aligned} PE_a^* &= \mathbb{P}_{X,Y}\left[mr < 0\right] \leq \mathbb{P}_{X,Y}\left[mr \leq 0\right] \leq \mathbb{P}_{X,Y}\left[mr \leq 0\right] + \mathbb{P}_{X,Y}\left[mr \geq 2s\right] \\ &= \mathbb{P}_{X,Y}\left[|mr - s| \geq s\right] \leq \frac{\mathrm{Var}_{X,Y}(mr)}{s^2}\,. \end{aligned}$$

Teniendo en cuenta que la siguiente identidad

$$\left[\mathbb{E}_{\Theta}\left[f\left(\Theta\right)\right]\right]^2 = \mathbb{E}_{\Theta,\Theta'}\left[f\left(\Theta\right) \cdot f\left(\Theta'\right)\right]\,, \tag{3.15}$$

se cumple para cualquier f si Θ y Θ' son i.i.d. (simplificamos $rmg(\Theta, X, Y) \equiv rmg(\Theta)$):

$$\left(mr(X,Y)\right)^2 = \mathbb{E}_{\Theta,\Theta'}\left[rmg(\Theta) \cdot rmg(\Theta')\right]\,.$$

Usando este resultado y sabiendo que estamos en condiciones de permutar el orden del cálculo de las esperanzas de Θ y Θ' con respecto de las de (X,Y), calculamos $\mathrm{Var}_{X,Y}(mr(X,Y)) \equiv \mathrm{Var}(mr)$:

$$\begin{aligned} \mathrm{Var}(mr) &= \mathbb{E}_{X,Y}\left[\left(mr\right)^2\right] - \left(\mathbb{E}_{X,Y}\left[mr\right]\right)^2 \\ &= \mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta,\Theta'}\left[rmg(\Theta) \cdot rmg(\Theta')\right]\right] - \left(\mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta}\left[rmg(\Theta)\right]\right]\right)^2 \\ &= \mathbb{E}_{\Theta,\Theta'}\left[\mathbb{E}_{X,Y}\left[rmg(\Theta) \cdot rmg(\Theta')\right]\right] - \left(\mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta}\left[rmg(\Theta)\right]\right]\right)^2\,. \end{aligned}$$

Aplicando la relación entre la covarianza de dos variables y sus esperanzas:

$$\mathrm{Cov}(\tilde{X},\tilde{Y}) = \mathbb{E}\left[\tilde{X}\cdot\tilde{Y}\right] - \mathbb{E}\left[\tilde{X}\right]\cdot\mathbb{E}\left[\tilde{Y}\right], \tag{3.16}$$

podemos transformar el primer término:

$$\begin{aligned}
\mathrm{Var}(mr) &= \mathbb{E}_{\Theta,\Theta'}\left[\mathrm{Cov}_{X,Y}\left(rmg(\Theta), rmg(\Theta')\right)\right] + \mathbb{E}_{\Theta}\left[\mathbb{E}_{X,Y}\left[rmg(\Theta)\right]\right]\cdot\mathbb{E}_{\Theta'}\left[\mathbb{E}_{X,Y}\left[rmg(\Theta')\right]\right] \\
&\quad - \left(\mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta}\left[rmg(\Theta)\right]\right]\right)^2 \\
&= \mathbb{E}_{\Theta,\Theta'}\left[\mathrm{Cov}_{X,Y}\left(rmg(\Theta), rmg(\Theta')\right)\right] + \left(\mathbb{E}_{\Theta}\left[\mathbb{E}_{X,Y}\left[rmg(\Theta)\right]\right]\right)^2 \\
&\quad - \left(\mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta}\left[rmg(\Theta)\right]\right]\right)^2 \\
&= \mathbb{E}_{\Theta,\Theta'}\left[\mathrm{Cov}_{X,Y}\left(rmg(\Theta), rmg(\Theta')\right)\right] + \left(\mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta}\left[rmg(\Theta)\right]\right]\right)^2 \\
&\quad - \left(\mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta}\left[rmg(\Theta)\right]\right]\right)^2 \\
&= \mathbb{E}_{\Theta,\Theta'}\left[\mathrm{Cov}_{X,Y}\left(rmg(\Theta), rmg(\Theta')\right)\right] = \mathbb{E}_{\Theta,\Theta'}\left(\rho(\Theta,\Theta')\cdot\mathrm{sd}(\Theta)\cdot\mathrm{sd}(\Theta')\right),
\end{aligned}$$

donde en la última igualdad se ha empleado la relación entre covarianza y correlación a través de las desviaciones típicas:

$$\rho\left(\tilde{X},\tilde{Y}\right) = \frac{\mathrm{Cov}\left(\tilde{X},\tilde{Y}\right)}{\mathrm{sd}(\tilde{X})\cdot\mathrm{sd}(\tilde{Y})}, \tag{3.17}$$

y se ha simplificado la notación de forma que $\rho\left(\Theta,\Theta'\right) \equiv \rho_{X,Y}\left(rmg(\Theta), rmg(\Theta')\right)$ y $\mathrm{sd}(\Theta) \equiv \mathrm{sd}_{X,Y}(rmg(\Theta))$. Empleando la expresión (3.14) y la (*desigualdad de Jensen*):

$$\mathrm{Var}(mr) = \bar{\rho}\cdot\mathbb{E}_{\Theta}\left[\mathrm{sd}\left(\Theta\right)\right]\cdot\mathbb{E}_{\Theta'}\left[\mathrm{sd}\left(\Theta'\right)\right] = \bar{\rho}\cdot\left(\mathbb{E}_{\Theta}\left[\mathrm{sd}\left(\Theta\right)\right]\right)^2 \leq \bar{\rho}\cdot\mathbb{E}_{\Theta}\left[\mathrm{Var}\left(\Theta\right)\right],$$

siendo $\mathrm{Var}\left(\Theta\right) \equiv \mathrm{Var}_{X,Y}\left(rmg(\Theta)\right)$. Ahora bien,

$$\begin{aligned}
\mathbb{E}_{\Theta}\left[\mathrm{Var}\left(\Theta\right)\right] &= \mathbb{E}_{\Theta}\left[\mathbb{E}_{X,Y}\left[\left(rmg(\Theta)\right)^2\right] - \left(\mathbb{E}_{X,Y}\left[rmg\left(\Theta\right)\right]\right)^2\right] \\
&= \mathbb{E}_{\Theta}\left[\mathbb{E}_{X,Y}\left[\left(rmg\left(\Theta\right)\right)^2\right]\right] - \mathbb{E}_{\Theta}\left[\left(\mathbb{E}_{X,Y}\left[rmg\left(\Theta\right)\right]\right)^2\right].
\end{aligned}$$

Aplicando Jensen al segundo término de la derecha y permutando esperanzas:

$$\begin{aligned}
\mathbb{E}_{\Theta}\left[\left(\mathbb{E}_{X,Y}\left[rmg\left(\Theta\right)\right]\right)^2\right] &\geq \left(\mathbb{E}_{\Theta}\left[\mathbb{E}_{X,Y}\left[rmg\left(\Theta\right)\right]\right]\right)^2 = \left(\mathbb{E}_{X,Y}\left[\mathbb{E}_{\Theta}\left[rmg\left(\Theta\right)\right]\right]\right)^2 \\
&= \left(\mathbb{E}_{X,Y}\left[mr\right]\right)^2 = s^2.
\end{aligned}$$

Y por lo tanto:

$$\mathbb{E}_{\Theta}\left[\mathrm{Var}\left(\Theta\right)\right] \leq \mathbb{E}_{\Theta}\left[\mathbb{E}_{X,Y}\left[\left(rmg\left(\Theta\right)\right)^2\right]\right] - s^2 \leq 1 - s^2.$$

Finalmente, poniendo todos los resultados en común:

$$PE_a^* \leq \frac{\mathrm{Var}(mr)}{s^2} \leq \frac{\bar{\rho}\cdot\mathbb{E}_{\Theta}\left[\mathrm{Var}\left(\Theta\right)\right]}{s^2} \leq \frac{\bar{\rho}}{s^2}\cdot\left(1-s^2\right).$$

□

Con este resultado, queda demostrado que el error puede ser reducido de dos formas: aumentando la capacidad de predicción de los árboles individuales o reduciendo la correlación entre ellos. Es por ello que se añade la aleatoriedad con las muestras *bootstrap* y la selección de un subconjunto de las variables para cada nodo; reducen la correlación entre árboles sin que se vea muy mermada su capacidad de predicción (si escogemos correctamente el número $m \leq p$ de variables).

Vamos a estudiar también cómo se comportan el sesgo y la varianza una vez se ensamblan los predictores. Por simplicidad, realizaremos los cálculos para el caso de regresión, donde la predicción total es la media de las predicciones de cada árbol. El caso de clasificación es análogo.

Empezamos estudiando el comportamiento del sesgo. Al ser el *bagging* una técnica de reducción de varianza que no debería influir en el sesgo, esperamos que el sesgo de un árbol sea similar al del ensamblado. De hecho, al ser $\{T(X,\Theta_b)\}_{b=1}^{B}$ un conjunto de vectores aleatorios idénticamente distribuidos:

$$\mathbb{E}\left[\hat{f}_{rf}^{B}(X)\right] = \mathbb{E}\left[\frac{1}{B}\sum_{b=1}^{B} T(X,\Theta_b)\right] = \frac{1}{B}\cdot\mathbb{E}\left[\sum_{b=1}^{B} T(X,\Theta_b)\right] = \mathbb{E}\left[T(X,\Theta)\right], \tag{3.18}$$

por lo que tal y como se define en la expresión (1.6), el sesgo del modelo ensamblado es el mismo. Este resultado va acorde con que aumentar el número de árboles no lleva a sobreajuste. El caso de la varianza es algo distinto porque entra en juego de nuevo la correlación entre los árboles. Probamos a continuación un resultado que nos dará un valor mínimo y un valor máximo de la varianza en función del número de árboles con los que cuente el modelo (ver [30]):

Proposición 3.13. *Denotando como σ^2 a la varianza de un solo árbol y como ρ a la correlación entre dos árboles del modelo, podemos expresar la varianza del modelo de la siguiente forma:*

$$Var\left(\hat{f}_{rf}^{B}(X)\right) = \rho\cdot\sigma^2 + \frac{1-\rho}{B}\cdot\sigma^2\,. \tag{3.19}$$

Demostración. Emplearemos la expresión general de la varianza de una suma:

$$\mathrm{Var}\left(\sum_{i=1}^{N} X_i\right) = \sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{Cov}(X_i,X_j)\,, \tag{3.20}$$

de forma que

$$\begin{aligned}\mathrm{Var}\left(\hat{f}_{rf}^{B}(X)\right) &= \mathrm{Var}\left(\frac{1}{B}\sum_{b=1}^{B} T(X,\Theta_b)\right) = \frac{1}{B^2}\mathrm{Var}\left(\sum_{b=1}^{B} T(X,\Theta_b)\right)\\ &= \frac{1}{B^2}\sum_{b=1}^{B}\sum_{b'=1}^{B}\mathrm{Cov}\left(T(X,\Theta_b),T(X,\Theta_{b'})\right)\end{aligned}$$

Dentro de los sumatorios, si $b = b'$:

$$\mathrm{Cov}\left(T(X,\Theta_b), T(X,\Theta_{b'})\right) = \mathrm{Cov}\left(T(X,\Theta_b), T(X,\Theta_b)\right) = \mathrm{Var}\left(T(X,\Theta_b)\right) = \sigma^2\,,$$

y si $b \neq b'$, aplicando (3.17):

$$\begin{aligned}\mathrm{Cov}\left(T(X,\Theta_b), T(X,\Theta_{b'})\right) &= \rho\left(T(X,\Theta_b), T(X,\Theta_{b'})\right) \cdot \sqrt{\mathrm{Var}\left(T(X,\Theta_b)\right) \cdot \mathrm{Var}\left(T(X,\Theta_{b'})\right)} \\ &= \rho \cdot \sigma^2\,.\end{aligned}$$

Como hay B términos con $b = b'$ y $B \cdot (B-1)$ con $b \neq b'$:

$$\begin{aligned}\mathrm{Var}\left(\hat{f}_{rf}^B(X)\right) &= \frac{1}{B^2} \cdot \left(B \cdot \sigma^2 + B \cdot (B-1) \cdot \rho \cdot \sigma^2\right) = \frac{1}{B}\left(\sigma^2 + (B-1) \cdot \rho \cdot \sigma^2\right) \\ &= \rho \cdot \sigma^2 + \frac{\sigma^2 - \rho \cdot \sigma^2}{B} = \rho \cdot \sigma^2 + \frac{1-\rho}{B} \cdot \sigma^2\,.\end{aligned}$$

□

Vemos que, efectivamente, la varianza depende tanto de la correlación entre los árboles como de la varianza de los árboles individuales, de forma análoga a lo que ocurre con el error generalizado asintótico. Sin embargo, la expresión (3.19) es una igualdad, de la que podemos sacar un máximo y un mínimo en función del número de árboles B. En el caso de contar con un solo árbol, $B = 1$, se recupera la expresión de la varianza individual, $\mathrm{Var}\left(\hat{f}_{rf}^1(X)\right) = \sigma^2$. Este será el valor máximo. Aumentando la cantidad de árboles se va reduciendo la varianza hasta un valor mínimo, cuando $B \to \infty$, de forma que $\mathrm{Var}\left(\hat{f}_{rf}^B(X)\right) \overset{B\to\infty}{\longrightarrow} \rho \cdot \sigma^2$, valor controlado por la correlación entre los árboles.

3.3. Error *out of bag*

Volvemos al problema de la partición de la muestra original, donde el procedimiento general es la división en datos de entrenamiento, validación y test. Ya vimos que también se podía añadir una técnica de validación cruzada, generando varias particiones validación-entrenamiento y calculando un promedio de los modelos (apartado 1.2.3). Sin embargo, en el caso del *Random Forest*, existe un procedimiento propio del modelo para poder realizar esta validación a partir de las observaciones *out of bag*.

Estas consisten en las observaciones que no formaron parte de la muestra de entrenamiento para un árbol, debido al uso de muestras *bootstrap*. De esta forma, para cada árbol habrá una cantidad considerable de observaciones (que se espera que sea algo superior al 35 %, como vimos con (1.19)) que podrían servir como validación del árbol. Por lo tanto, podemos construir un

error de forma que para cada observación se aplique el modelo pero solo con los árboles en los que no influyó en el entrenamiento. Esto es lo que se conoce como error *out of bag* (OOB).

El uso del error OOB es muy común a la hora de emplear este tipo de modelos. Además, los valores experimentales suelen ser superiores a los obtenidos para los errores habitualmente empleados, ya que el valor del sesgo aumenta por no pertenecer las observaciones a la muestra de entrenamiento del modelo. Además, aunque este procedimiento pueda reemplazar a la partición entrenamiento-validación (con o sin validación cruzada) para escoger los hiperparámetros óptimos, también se pueden emplear los dos métodos simultáneamente y comparar los resultados.

3.4. Importancia de variables

A la hora de trabajar con árboles de decisión, es posible medir la importancia de cada una de las variables explicativas mediante su relevancia en la división de los nodos. Para ello, podemos medir cuánto varía la impureza de un nodo al dividirse mediante la variable que se haya seleccionado (tal y como se explicó en la Sección 2.2). Este se conoce como el método de pureza en nodos (ver [27]).

Para comprenderlo mejor, supongamos que estamos en un problema de clasificación y para el criterio de división de los árboles empleamos el índice de Gini (Definición 2.17). Para cada nodo de cada árbol en el que se realiza una división, se contará con un valor del índice para el nodo y un valor del índice para la división (una suma ponderada de los valores para cada nuevo nodo, tal y como se presenta en la expresión (2.7)). De esta forma, calculamos la diferencia entre ambos valores y la sumamos a la medida de importancia acumulada para la variable mediante la que se realizó la división. Procediendo de igual forma con todos los nodos de todos los árboles, obtendremos un valor de importancia para cada una de las variables, que será más elevado cuanta mayor reducción de impureza haya conseguido la variable. En el caso de regresión se puede proceder de igual forma, por ejemplo, con el MSE.

Existe otra forma de medir la importancia de las variables mediante las muestras OOB. Esta consiste en permutar los valores en una variable de las observaciones OOB de cada árbol y medir cuánto se modifica la precisión del modelo, repitiendo para cada variable. De esta forma, las variables que produzcan una diferencia mayor al ser modificadas serán las más relevantes para el modelo. En este caso, la influencia será positiva si reduce el error y negativa si lo aumenta. Este se conoce como el método de permutación (ver [28]).

Capítulo 4

Aplicación del modelo

A lo largo de este capítulo se presentarán los resultados de aplicar el modelo *Random Forest* a datos reales del ámbito deportivo. En paralelo a la exposición de los resultados, se hará también un estudio general del comportamiento del modelo; desde la dependencia del modelo con los hiperparámetros hasta la importancia que se le da a las variables. Además, se trabajará tanto con el caso de clasificación como el de regresión, por lo que también se presentarán las principales diferencias entre ambos casos. Finalmente, se realizará una comparación de resultados con otros modelos habituales en el ámbito del aprendizaje supervisado.

4.1. Base de datos

La base de datos con la que se trabajará para aplicar el modelo consta con datos relativos a la NBA (*National Basketball Association*), la liga de baloncesto de Estados Unidos. En particular, cuenta con datos de todos los equipos que forman parte de la liga[1] entre la temporada 2004/2005 y la temporada 2022/2023. Por lo tanto, contando con datos de 19 temporadas para los 30 equipos participantes, el número total de observaciones es de 570.

Con el avance exponencial que se está consiguiendo en capacidad de computación durante las últimas décadas, la rama de la estadística tiene cada día más relevancia en el ámbito deportivo. Así ocurre, en particular, en el baloncesto. Sobre todo en la NBA, en la que se lleva tiempo haciendo esfuerzos en conseguir recopilar cada vez más datos de los partidos para optimizar el juego. Por otro lado, se han buscado formas de combinar estos datos de forma que las variables resultantes sirvan de mejores indicadores para el rendimiento de un jugador o de un equipo;

[1] Aunque durante el período al que corresponden los datos ha habido cambios de nombre y de localización de algunos equipos, las franquicias han sido siempre las mismas.

más allá de las habituales como puntos, rebotes y asistencias. A esto se le conoce en el ámbito deportivo como *estadística avanzada*.

La base de datos empleada cuenta con variables que se consideran pertenecientes a la estadística avanzada. Esta fue extraída de *Kaggle* [9], una plataforma abierta en la que se recoge una inmensa cantidad de bases de datos. En el Anexo II se describe brevemente cada una de las variables que se emplearán para los modelos.

Las variables que vamos a predecir fueron añadidas a la base de datos original. Una es la variable *Odds* (ver expresión (II.19)), que es continua. Por lo tanto, para hacer predicciones se utilizará un modelo de regresión. La otra es una variable dicotómica que vale 1 cuando el equipo entra en *Playoffs* (última fase de la competición en la que entran 16 de los 30 equipos, 8 por cada conferencia) y 0 cuando no. Al ser dicotómica, el modelo que se empleará será de clasificación. En ambos casos, se tratará de predecir la variable respuesta a partir de datos de la temporada anterior, no de la misma. Es decir, si las variables explicativas corresponden a la temporada 2007/2008, las predicciones de *Odds* y de *Playoffs* serán para la temporada 2008/2009. En total, para ambos modelos se emplearán 20 variables explicativas y 1 variable respuesta: *Playoffs_n* para clasificación y *Odds_n* para regresión, donde n refleja que corresponden a la temporada siguiente. En el Cuadro 4.1 se refleja una observación con los valores para todas estas variables más la variable *Equipo* y *Temp*, que sirven de etiquetas. Por comodidad, la variable *Temp* solo indica el año en el que acabó la temporada.

Equipo	Temp	Edad	PW	PL	MOV	SOS	SRS	ORtg	DRtg	Pace	FTr
MIL	2019	26.9	61	21	8.87	−0.82	8.04	113.8	105.2	103.3	0.255
X3PAr	**TS %**	**FT/FGA**	**FT/FGAa**	**eFG %**	**eFG %a**	**TOV %**	**TOV %a**	**ORB %**	**DRB %**	**Playoffs_n**	**Odds_n**
0.419	0.583	0.197	0.162	0.550	0.503	12.0	11.5	20.8	80.3	1	3.2941176

Cuadro 4.1: Valores de las variables de los Milwaukee Bucks (MIL) en la temporada 2018/2019.

Antes de aplicar los modelos a los datos, es importante observar de forma cualitativa cómo es la asociación lineal entre las distintas variables. Para ello, se presenta en la Figura 4.1 la matriz de correlación en forma de mapa de calor[2]. En este se incluyen tanto las variables explicativas como las variables respuesta. Por un lado vemos que, a excepción de la zona superior izquierda, la correlación entre las variables explicativas es muy baja, información que será relevante a la hora de interpretar la correlación entre los árboles. Por otro lado, vemos que las correlaciones de las variables explicativas con las variables respuesta no alcanzan valores muy altos en valor absoluto. Esto supondrá un problema para los modelos, puesto que las variables respuesta no dependen

[2]Esta gráfica se generó con la librería *corrplot* de R, mientras que el resto de gráficas del capítulo se generaron con la librería *ggplot2*.

fuertemente de las variables explicativas, lo que llevará a predicciones con errores elevados.

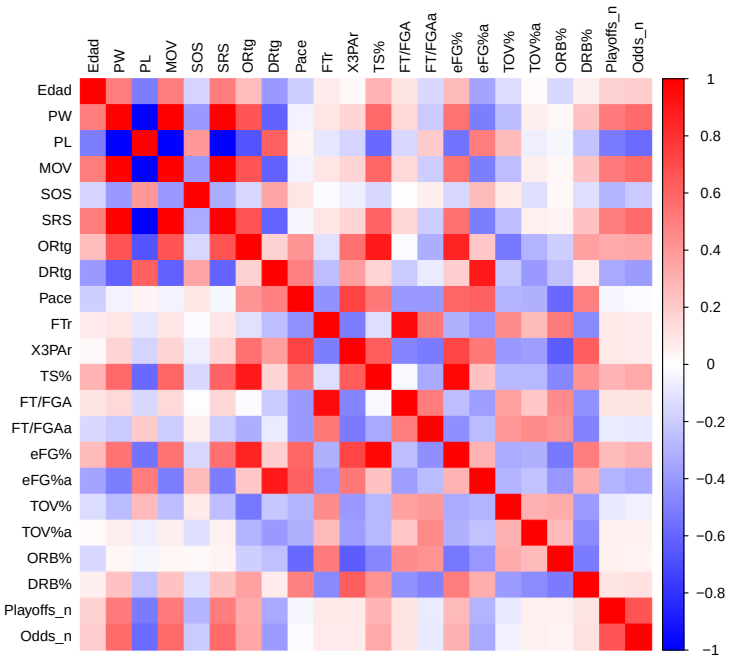


Figura 4.1: Correlación entre las variables del modelo.

4.2. Modelos óptimos

Pasamos ya a la aplicación del modelo *Random Forest* a los datos, tanto en el caso de clasificación como el de regresión. Empezamos describiendo el procedimiento de partición de la muestra, que no fue el entrenamiento-validación-test habitual. En primer lugar, se dividió la muestra en entrenamiento y test, siendo la muestra de test las observaciones relativas a la temporada 2022/2023, la última registrada en la muestra. Por lo tanto, los modelos se entrenarán con los datos de todas las temporadas anteriores para predecir los resultados de dicha temporada. De esta forma, se emplearon 540 observaciones para entrenar los modelos y predecir la variable respuesta para 30 observaciones. Con la muestra de entrenamiento no se realizó una partición entrenamiento-validación, sino que se emplearon las observaciones *out of bag* (OOB) para validar el modelo optimizando los hiperparámetros (ver Sección 3.3).

De esta forma, para escoger los hiperparámetros, se entrena el modelo construyendo los árboles con las muestras OOB y se calcula el error que se comete con las observaciones OOB. Variando los hiperparámetros, el modelo óptimo no es otro que el que minimice dichos errores.

Este proceso, como el de todos los modelos que se presentan en este capítulo, se realizó con el software libre R, en este caso con la librería *randomForest*. Tanto para el caso de clasificación como el de regresión los hiperparámetros que se optimizaron con sus posibles valores fueron los siguientes (siguiendo la nomenclatura de R):

$$\begin{aligned} mtry &\in \{i \in \mathbb{N} \mid i \leq 20\}, \\ ntree &\in \{100, 200, 300, 400, 500\}, \\ nodesize &\in \{1, 5, 10, 15, 20, 50, 75, 100, 125, 150, 175, 200\}, \\ maxnodes &\in \{25, 50, 100\}. \end{aligned}$$

El hiperparámetro *mtry* ajusta el número de variables m a seleccionar en cada nodo (tal y como se explica en la Sección 3.1), *ntree* fija el número de árboles que se crean, *nodesize* indica el número mínimo de observaciones en un nodo n_{min} para poder dividirse (ver Definición I.2) y *maxnodes* el número máximo de nodos terminales que puede tener cada árbol. Los valores óptimos, tanto para clasificación como para regresión, se recogen en el Cuadro 4.2.

	mtry	*ntree*	*nodesize*	*maxnodes*
Clasificación	1	200	50	50
Regresión	4	100	1	25

Cuadro 4.2: Hiperparámetros óptimos para los modelos.

Vemos que el modelo óptimo para clasificación solo escoge una variable de forma aleatoria para cada nodo, por lo que sus árboles van a estar poco correlacionados. Sin embargo, cuando menor sea el número de variables a escoger, menor será la capacidad de predicción de árboles individuales, pero en este caso parece que el balance entre la correlación de árboles y la capacidad de árboles (ver Teorema 3.12) se optimiza minimizando la correlación. Por otro lado, el número de árboles para el modelo de clasificación resulta mayor que para el de regresión, hecho que va de la mano de la necesidad de compensar la reducción de la capacidad de predicción de los árboles de forma individual. Cabe destacar también que para el caso de regresión el modelo óptimo no pone restricciones para el número mínimo de observaciones en un nodo, $n_{min} = 1$.

En cuanto a los criterios de división de los nodos (ver Sección 2.3), la librería *randomForest* de R emplea de forma predeterminada el índice de Gini en el caso de clasificación (Definiciones 2.16 y 2.17) y el error cuadrático medio en el caso de regresión (expresión (2.11)).

El error que se minimizó para el modelo de clasificación para las observaciones OOB consiste en la proporción de predicciones equivocadas. El modelo óptimo obtenido se equivocó el 26.85 %

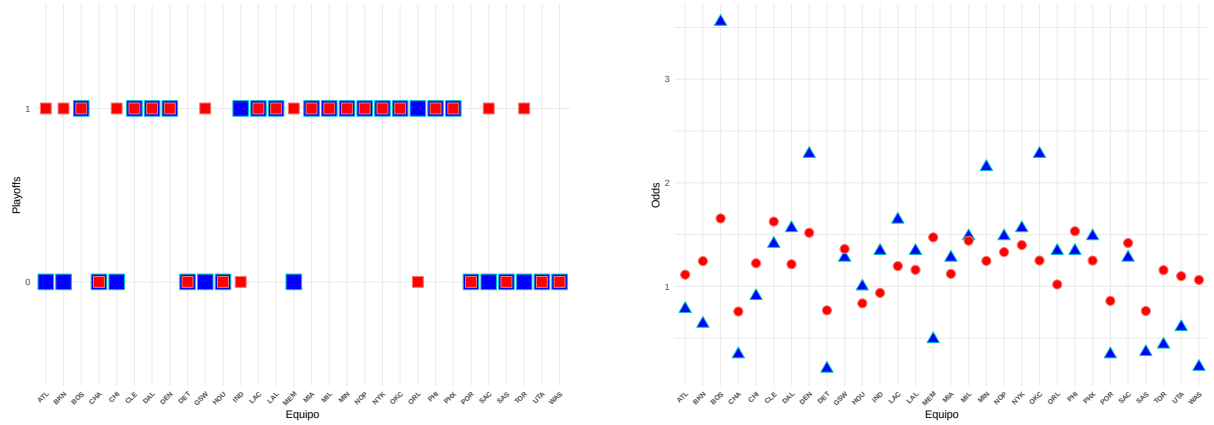
de las veces. Para regresión, el error a minimizar fue el MSE[3] (ver Definición 4.1), para el que se obtuvo un valor mínimo para el modelo óptimo de 0.4463. Si hubiéramos tomado como predictor para todos los equipos la media los valores reales, habríamos obtenido un MSE de 0.6876. Estos valores se emplearán para la comparación con otros modelos en los apartados 4.3.2 y 4.4.2.

En la Figura 4.2 se presentan las predicciones (rojo) de los modelos en comparación con los valores reales (azul) para los 30 equipos en la temporada 2022/2023. En el modelo de clasificación la tendencia del modelo es a predecir que entran más equipos en *playoffs* de lo debido. De esta forma, acierta para casi todos los equipos que realmente accedieron a esta fase de la competición y tiende a cometer más fallos para los equipos que no entraron. El porcentaje de error de predicción para estos datos es del 33.33 %, que es un resultado un poco pobre para nuestros intereses.

Con respecto al número de equipos que debería predecir el modelo, debemos tener en cuenta que los $\{y_i\}_{i=1}^{30}$ (valores de *Playoff_n* para cada equipo) no son independientes entre sí. Esto se debe a que para cada temporada se clasifican siempre 16 equipos a *Playoffs*, de forma que los 16 valores han de sumar 16. De hecho, la NBA se divide en 2 conferencias de 15 equipos, clasificándose 8 equipos de cada una. Por lo tanto, existen las dos siguientes restricciones

$$\sum_{j=1}^{15} y_{j,O} = 8\,,\ \sum_{j=1}^{15} y_{j,E} = 8\,; \tag{4.1}$$

siendo O y E los valores que puede tomar la variable *Conf* (ver Anexo II), indicando si el equipo esta en la conferencia oeste o este.



(a) Clasificación. (b) Regresión.

Figura 4.2: Comparaciones de predicciones con los valores reales para cada equipo.

[3]En las gráficas se representa el RMSE (Definición 4.3), la raíz cuadrada del MSE, habitual en estos casos.

Una forma de tener en cuenta esta esta dependencia entre los valores de la respuesta sería tener en cuenta la proporción de árboles que predicen el acceso a *playoffs* para cada equipo y seleccionar solo los 8 con mayor proporción de cada conferencia; esta posibilidad se estudia en el apartado 4.3.3. De la misma forma, en el caso de regresión también existe cierta dependencia entre las respuestas, pero no es tan notable como en el caso de clasificación.

En el modelo de regresión se puede ver claramente una tendencia a "aplanar" los datos, es decir, la gran mayoría de las predicciones se encuentran entre 0.5 y 1.5 mientras que para los valores reales hay varios valores inferiores a 0.5 o superiores a 2. Sin embargo, parece que el modelo es capaz de discriminar qué equipos van a tener mejores resultados, aunque "subestime" de cierta forma la proporción entre victorias y derrotas. El MSE obtenido fue de 0.362590, mientras que si se hubiera usado la media como predictor para todos los equipos sería de 0.5260. Otro valor que se suele calcular en estos casos es el del R^2, que refleja la proporción de variabilidad que es capaz de explicar el modelo para los datos. El valor de R^2 obtenido para el modelo a partir de las observaciones *OOB* es de 0.3107; un valor bajo que nos lleva a pensar que la capacidad de predicción del modelo será bastante reducida. La forma de calcular estos valores, así como el MAE que se empleará para comparar modelos, se presenta a continuación.

Definición 4.1. Sea $\{(x_i, y_i)\}_{i=1}^n$ un conjunto de n observaciones y sea $\{\hat{y}_i\}_{i=1}^n$ el conjunto de predicciones que produce un modelo para dichas observaciones. Se define el MSE (*Mean Square Error*) asociado a las predicciones como

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \ . \tag{4.2}$$

De esta forma, se define el RMSE (*Root Mean Square Error*) asociado a las predicciones como

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}\,. \tag{4.3}$$

Se define también el MAE (*Mean Absolute Error*) asociado a las predicciones como

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \ . \tag{4.4}$$

Finalmente, se define el R^2 asociado a las predicciones como

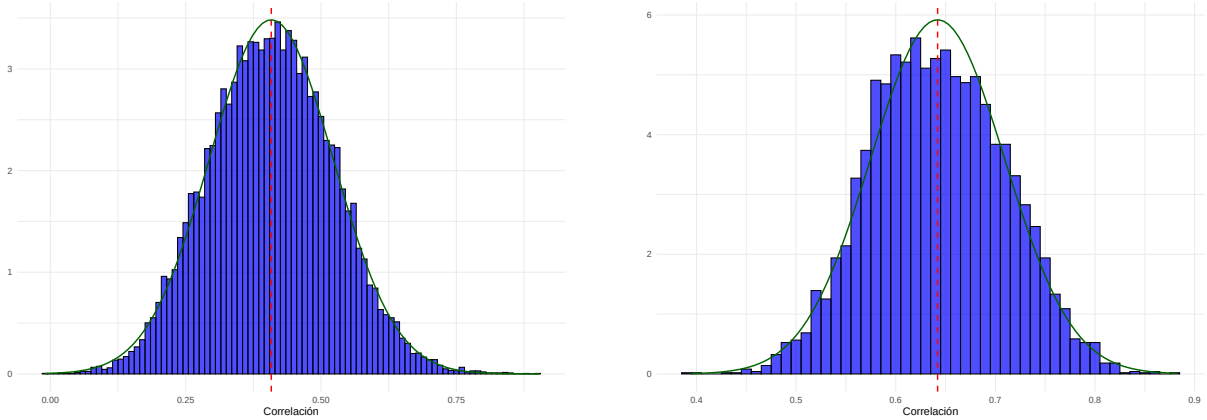
$$R^2 = 1 - \frac{RSS}{TSS}\,, \tag{4.5}$$

donde RSS (*Residual Sum of Squares*) y TSS (*Total Sum of Squares*) vienen dados por

$$RSS = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \ , \tag{4.6}$$

$$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2 \ . \tag{4.7}$$

Para ahondar un poco más en las características de los modelos, se recogen en la Figura 4.3 los histogramas relativos a la correlación entre los árboles de cada uno de los modelos. Para ambos casos, podemos ver una distribución gaussiana (verde) de la muestra, más clara en el caso de clasificación al contar con más observaciones (mayor número de árboles). Por otro lado, vemos que la media de las correlaciones para el caso de clasificación, 0.408079, es menor que para el caso de regresión, 0.642006. Como ya se comentó antes, esto es debido a que cuanto menor es el número de observaciones entre las que se escoge en cada nodo, más se reduce la correlación en los árboles. Además, como para ambos modelos se trabaja con la misma muestra, no existen otros factores que influyan tanto en esta diferencia significativa. Es destacable también que la dispersión de los datos de las correlaciones es mayor en el caso de clasificación que en el de regresión, a pesar de la diferencia en el número de árboles. La justificación consiste en que en el caso de clasificación solo existen dos posibles predicciones, por lo que lleva a correlaciones menores entre los árboles por no existir valores intermedios como ocurre para el modelo de regresión.



(a) Clasificación. (b) Regresión.

Figura 4.3: Histogramas de correlación entre los árboles de los modelos.

Para finalizar con este análisis inicial de los modelos óptimos conseguidos, interpretamos a continuación la importancia de las variables en dichos modelos. Para ello, ya vimos en la Sección 3.4 que existen dos métodos: permutación y pureza de nodos. En la Figura 4.4 se presentan los resultados aplicando estos dos métodos a los modelos óptimos para clasificación y para regresión. Para poder comparar los resultados de los dos métodos, las importancias están escaladas de forma que sumen 1 para cada método. Podemos ver que las diferencias entre métodos no son especialmente significativas, salvo para la variable *Edad* en el caso de regresión. Vemos también que para algunas de las variables con menos importancia esta es negativa, lo que significa que el modelo obtuvo mejores predicciones tras permutar los valores de dichas variables en las observaciones OOB.

Ahora bien, ambos modelos le dan especial importancia a estas cuatro variables: *MOV*, *SRS*, *PW* y *PL*. Esto no es casualidad, ya que como vimos en la Figura 4.1, son las variables explicativas que más correlación (en valor absoluto) tenían con las variables respuesta. Por lo tanto, los modelos recogen de forma adecuada la importancia de las variables, siendo las más relevantes las que más correlación tengan con la variable respuesta en cada uno de los casos. En cuanto a las diferencias entre los dos modelos, vemos que la importancia está más distribuida en el caso de clasificación, mientras que en el caso de regresión la mayor parte de la importancia total se reparte entre pocas variables. Esto hace pensar que el modelo *Random Forest* tiende a tener en cuenta todas las variables que le des al modelo en el caso de clasificación (al menos para el caso dicotómico) y a omitir las variables menos importantes en el caso de regresión.

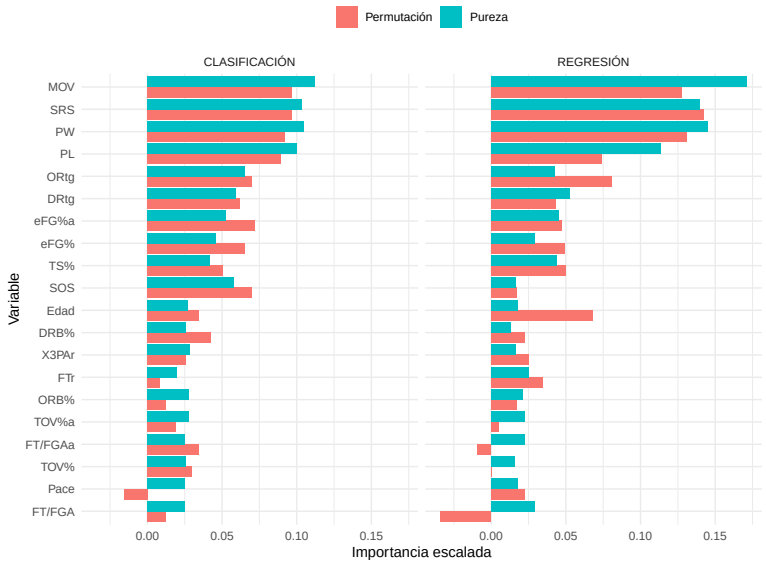


Figura 4.4: Importancias de las variables en los modelos de clasificación y regresión.

A continuación, se estudiará la dependencia de cada modelo con los valores de sus hiperparámetros y se compararán los resultados con los obtenidos para otros modelos.

4.3. Modelo de clasificación

Con el modelo *Random Forest* de clasificación estamos tratando de predecir si los equipos acceden a la ronda de *playoffs* la temporada siguiente a la que se recogieron los datos. Como ya vimos, optimizando los hiperparámetros se consiguió minimizar el error de predicción hasta un 33.33 %. Se puede estudiar cómo influye cada hiperparámetro al modelo dejando fijos el resto de valores óptimos de hiperparámetros y variando el que nos interesa.

4.3.1. Dependencia con hiperparámetros

Empezamos viendo cómo afecta el número de variables a escoger por nodo m a la correlación entre árboles. En la Figura 4.5 se presentan para cada valor de m del 1 a 20 los valores de correlación entre árboles como diagramas de cajas. En ellos, la caja representa el rango intercuartílico (IQR), es decir, el espacio entre el primer cuartil (percentil 25) y el tercer cuartil (percentil 75). La recta dentro de la caja representa el valor de la mediana, las líneas que se extienden desde la caja (bigotes) tienen una longitud de 1.5 veces el IQR y los puntos que caen fuera de este rango son considerados valores atípicos.

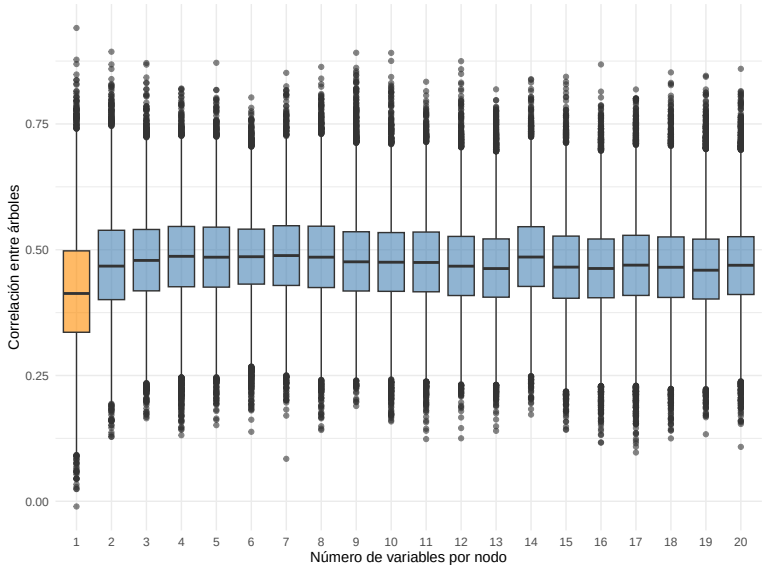
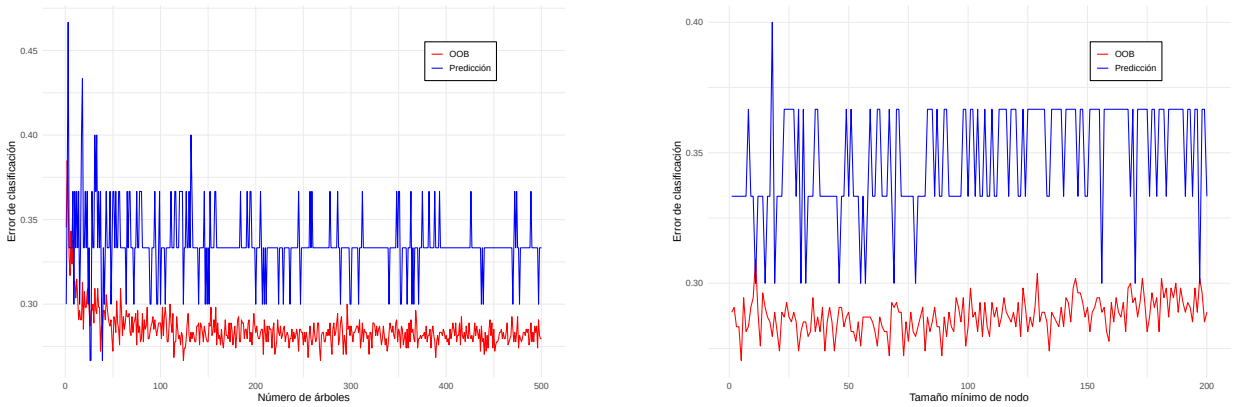


Figura 4.5: Correlación de los árboles frente a número de variables para cada nodo.

Como se puede apreciar, para todos los valores menos el 1 (marcado de otro color por ser el valor óptimo calculado) no existe dependencia perceptible de la correlación con el hiperparámetro. A pesar de ello, como cabría esperar, es el menor valor de m el que lleva a una correlación menor. Esto es debido a que al escoger entre menos variables, no van a predominar tanto las variables con mayor correlación con la respuesta, ya que se podrán escoger menos veces. Podemos deducir de la gráfica que el valor óptimo es $m = 1$ ya que es el único valor que consigue reducir la correlación y no debe disminuir lo suficiente la capacidad de predicción de cada árbol individual como para que no salga rentable seleccionarlo. Este razonamiento se apoya en el Teorema 3.12, del que se deduce que la capacidad de predicción de un modelo *Random Forest* es directamente proporcional a la capacidad de predicción de cada árbol individual e inversamente proporcional a la correlación entre los árboles del modelo.

También podemos variar el número de árboles del modelo para ver cómo evoluciona el error, tanto para las observaciones OOB como para la muestra para la que tratamos de predecir. Podemos ver en la Figura 4.6a que en este caso el error en las predicciones es siempre superior al error OOB. También se puede ver que es entre 100 y 200 árboles cuando se empieza a estabilizar el error OOB, siendo 200 el valor óptimo obtenido. Cabe destacar que el error de predicción varía entre un número reducido de valores posibles al contar tan solo con 30 observaciones.

En la Figura 4.6b el procedimiento es análogo pero variando el número mínimo de observaciones en un nodo, n_{min}. A diferencia del caso anterior, donde el error decrecía para los primeros valores, este hiperperámetro no parece influir mucho en la capacidad de predicción del modelo. Para valores muy altos sí que parece que el error OOB tiende a crecer ligeramente, debido principalmente a la reducción de la complejidad de los árboles. El valor óptimo para este hiperparámetro, como se vio antes, es $n_{min} = 50$.



(a) Error - Número de árboles.

(b) Error - n_{min}.

Figura 4.6: Error de clasificación en función de los hiperparámetros.

4.3.2. Comparativa con otros modelos

Para comprobar la utilidad de un modelo, es necesario comparar sus resultados con los de otros modelos generalmente utilizados en este tipo de problemas. En este caso, compararemos el modelo óptimo de clasificación obtenido con los modelos LDA, regresión logística y GBM.

El Análisis de Factorial Discriminante, conocido como LDA, es un modelo presentado en la asignatura *Regresión y Análisis Multivariante* [24] que consiste en construir un nuevo conjunto de variables incorreladas combinación lineal de las originales que mejor separe las observaciones en grupos en función de los distintos valores de la variable respuesta. Como el número de variables

que genera es el mínimo entre $K-1$ y p, siendo K el número de valores que toma la variable respuesta y p el número de variables explicativas, en nuestro caso el modelo solo generará una variable.

El modelo de regresión logística se trata de un modelo lineal generalizado (GLM) que usa como enlace la función logística (ver [10]). Este modelo es presentado en la asignatura optativa *Modelos de Regresión y Análisis Multivariante* y, al emplear la función logística que consiste en aplicar un logaritmo a la función *odds* (Definición II.18), parece natural utilizarlo en el caso de regresión. Sin embargo, también es útil en el caso de clasificación, en el que usaremos en particular el modelo logístico binario. Por comodidad, nos referiremos a él como GLM.

El *Gradient Boosting Machine*, o GBM, consiste en resumidas cuentas en un modelo de *boosting* (ver apartado 1.2.2) que usa como *weak learners* árboles de decisión. Este método, al igual que el modelo de *Random Forest* que estamos utilizando, se debe a [17]. Para este modelo también es posible optimizar ciertos hiperparámetros para minimizar el error de predicción. En este caso, los hiperparámetros y sus posibles valores (presentados con la nomenclatura de ) fueron los siguientes:

$$\begin{aligned} n.trees &\in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}, \\ interaction.depth &\in \{2, 4, 6\}, \\ shrinkage &\in \{0.01, 0.05, 0.1\}. \end{aligned}$$

Estos se calcularon haciendo uso de la librería *gbm* en  . El primer hiperparámetro indica el número de árboles, mientras que *interaction.depth* es la profundidad máxima de cada árbol h_{max} (ver Anexo I) y *shrinkage* es el tamaño del paso de cada iteración de entrenamiento, propio de los modelos de *boosting*. Los valores óptimos se presentan en el Cuadro 4.3 junto con los del modelo de regresión, que también se comentará en la Sección 4.4.

	n.trees	*interaction.depth*	*shrinkage*
Clasificación	600	4	0.1
Regresión	300	6	0.01

Cuadro 4.3: Hiperparámetros óptimos para los modelos GBM.

Los errores de clasificación de los 4 modelos se recogen en el Cuadro 4.4. En este podemos ver que de los 4 modelos el que predice mejor el acceso a *playoffs* para los datos de la temporada 2022/2023 es el *RandomForest*. Sin embargo, no se pueden sacar grandes conclusiones ya que, al estar trabajando con una muestra de test muy pequeña para un caso dicotómico, está claro

que hay fluctuaciones estadísticas que influyen en la precisión que obtuvieron los modelos. Es decir, las diferencias de acierto entre los modelos son de muy pocos equipos, por lo que con otra muestra es posible que fuera otro modelo el que rindiera mejor.

Como ya se comentó antes, estos resultados son bastante pobres como para que ninguno de los modelos pueda ser de utilidad real. Esto es debido a la escasez de datos pero, sobre todo, a la escasa correlación de las variables explicativas con la respuesta. De hecho, un modelo tan sencillo como predecir siempre que la respuesta sea 1 cuenta siempre con un error del 46.67 % (14 fallos), que no se aleja mucho de los resultados obtenidos. Cabe destacar de todas formas que para su reducido coste, tanto *LDA* como *GLM* resultan bastante competitivos, obteniendo resultados similares a los otros modelos, que son más complejos y costosos computacionalmente.

	Random Forest	**LDA**	**GLM**	**GBM**
Error de clasificación	33.33 %	36.67 %	40.00 %	43.33 %
Aciertos	20	19	18	17

Cuadro 4.4: Errores de clasificación y cantidad de aciertos de los cuatro modelos.

Podemos ver también a partir del Cuadro 4.5, que recoge las matrices de confusión de los modelos, que en los 4 casos la tendencia es a predecir que más equipos de los debidos entran a los *playoffs* (solo entran un total de 16), lo que se tratará de corregir en el apartado 4.3.3.

Random Forest				**LDA**				**GLM**				**GBM**			
Real\Pred	0	1	Total	Real\Pred	0	1	Total	Real\Pred	0	1	Total	Real\Pred	0	1	Total
0	6	8	14	0	7	7	14	0	6	8	14	0	5	9	14
1	2	14	16	1	4	12	16	1	4	12	16	1	4	12	16
Total	8	22	30	Total	11	19	30	Total	10	20	30	Total	9	21	30

Cuadro 4.5: Matrices de confusión de los 4 modelos.

Finalmente, podemos comparar la importancia de las variables para cada uno de los modelos. Para ello, se presentan en la Figura 4.7 las importancias calculadas a partir del método de permutación (Sección 3.4) para *Random Forest* y *GBM*, a partir de los valores de la combinación lineal mediante la que se crea la variable discriminante en *LDA* y a partir de los coeficientes del modelo en el *GLM*. En los dos últimos casos, se tomaron los valores absolutos. Estas importancias están escaladas de forma que para cada modelo sumen 1. Como podemos ver, existen discrepancias importantes entre los 4 modelos. Lo más destacable a primera vista es que para *LDA* se reparten más de la mitad de la importancia las variables $TS\,\%$ y $eFG\,\%a$, que como se puede ver en la Figura 4.1 no son las variables más correlacionadas con la variable respuesta.

Por otro lado, vemos que GBM reparte más o menos de forma equitativa la importancia salvo para la variable MOV, que tiene más del doble de importancia que el resto. El GLM coincide con el LDA en darle mucha importancia a $TS\%$ y $eFG\%a$, aunque también le da bastante importancia a $eFG\%$ y FT/FGA. Estos dos modelos le dan importancia prácticamente nula a la mitad de las variables. Podemos sacar en conclusión que por lo general las importancias que le da el modelo *Random Forest* a las variables son las que se ajustan mejor a las correlaciones con la variable respuesta.

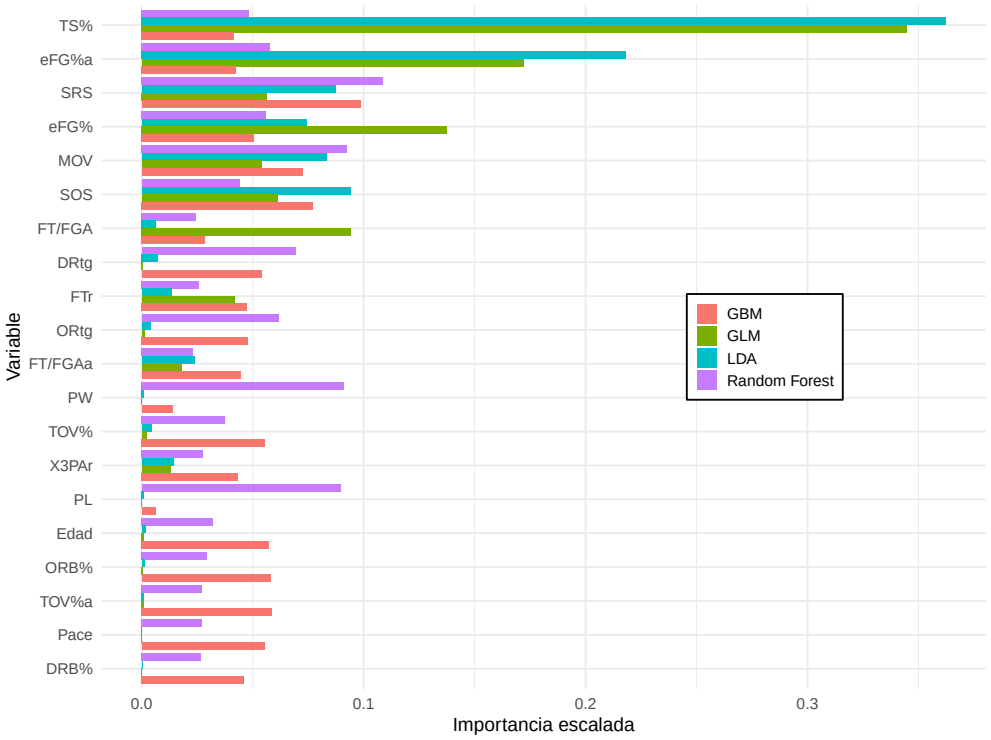


Figura 4.7: Comparación de importancia de variables con GBM y LDA.

4.3.3. Modelos con restricciones

Como ya se explicó antes, las respuestas para las observaciones que corresponden a una misma temporada no son independientes. Esto se debe a que cada año entran solo 16 equipos en *Playoffs*, 8 por cada conferencia, que producen las restricciones (4.1). Las predicciones que hemos realizado con los modelos de clasificación empleados no tienen en cuenta dichas restricciones. Sin embargo, podemos forzar de alguna manera a que las tengan en cuenta porque cada uno de los modelos asocia a cada observación una probabilidad de pertenecer a cada clase (en este caso 0 ó 1). Por lo tanto, como sabemos a qué conferencia pertenece cada equipo, se pueden escoger los 8 equipos de cada conferencia con mayor probabilidad de entrar en *playoffs*. En el Cuadro 4.6 se presentan los errores de clasificación y los aciertos de estos modelos modificados, mientras que

en el Cuadro 4.7 se muestran las matrices de confusión, donde efectivamente podemos ver que el número de equipos que se predice que entren en *playoffs* para los 4 modelos son 16.

	Random Forest	**LDA**	**GLM**	**GBM**
Error de clasificación	33.33 %	40.00 %	40.00 %	40.00 %
Aciertos	20	18	18	18

Cuadro 4.6: Errores de clasificación y aciertos de los modelos con restricciones.

Random Forest				**LDA**				**GLM**				**GBM**			
Real\Pred	0	1	Total	Real\Pred	0	1	Total	Real\Pred	0	1	Total	Real\Pred	0	1	Total
0	9	5	14	0	8	6	14	0	8	6	14	0	8	6	14
1	5	11	16	1	6	10	16	1	6	10	16	1	6	10	16
Total	14	16	30	Total	14	16	30	Total	14	16	30	Total	14	16	30

Cuadro 4.7: Matrices de confusión de los modelos con restricciones.

Vuelve a ser en este caso el *Random Forest* el modelo que más acierta con las predicciones. Además, los otros 3 modelos aciertan el mismo número de veces y cuentan con matrices de confusión idénticas. Es más, la predicción ha sido la misma para los 30 equipos, suceso cuanto menos curioso. Por lo general, parece que añadir las restricciones no consigue mejorar la precisión de los modelos; sin embargo, producen resultados más realistas.

4.4. Modelo de regresión

Con el modelo *Random Forest* de regresión nuestro objetivo es predecir el valor *Odds* de cada equipo la temporada siguiente a la que se recogieron los datos. Esta variable respuesta consiste en el número de victorias que consigue el equipo por cada derrota, dicho de otra forma, es el cociente entre victorias y derrotas. Como ya vimos antes, el modelo conseguido optimizando los hiperparámetros genera predicciones con un valor de RMSE de 0.362590 y un R^2 de 0.3107. Este segundo valor ya nos adelanta que el modelo tiene poca capacidad para realizar predicciones, ya que el R^2 indica la proporción de variabilidad de los datos que es capaz de explicar el modelo. A pesar de ello, se puede aprovechar el modelo para estudiar cómo afecta el variar los hiperparámetros a la precisión y para comparar los resultados con otros modelos de regresión utilizados habitualmente.

4.4.1. Dependencia de hiperparámetros

Se procederá de la misma forma que para el caso de clasificación, variando de cada vez el valor de un hiperparámetro y dejando fijos los valores óptimos del resto. Para empezar, podemos ver a través de la Figura 4.8 la dependencia de la correlación de los árboles con el número de variables a escoger en cada nodo, m. A diferencia del caso de clasificación, para valores bajos de m sí que se puede apreciar una dependencia clara de la correlación con m, siendo la mediana de la distribución directamente proporcional al número de variables. De hecho, se puede ver cómo ese crecimiento se frena para $m = 4$, que es el valor óptimo del hiperparámetro. De nuevo, parece que el modelo encuentra el equilibrio entre el aumento de la capacidad de predicción de cada árbol y el aumento de la correlación entre ellos una vez que la correlación deja de depender de m.

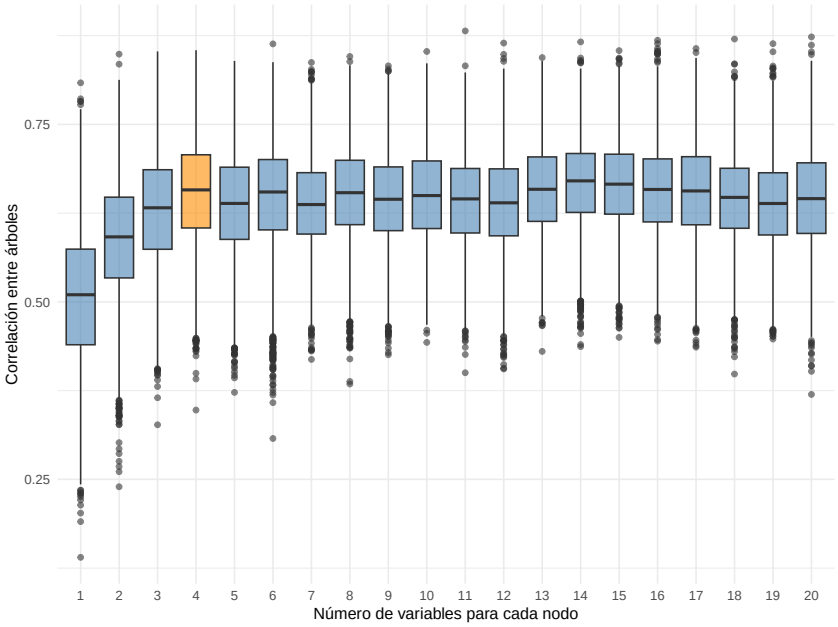
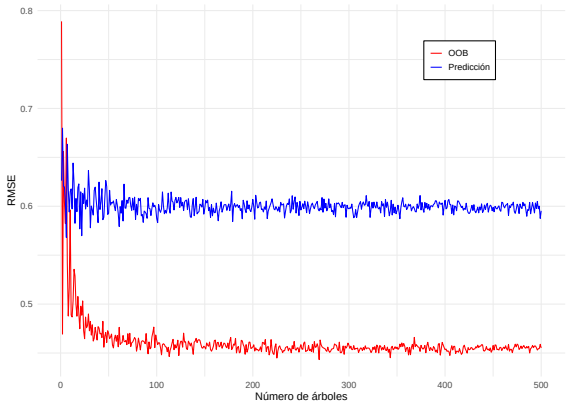


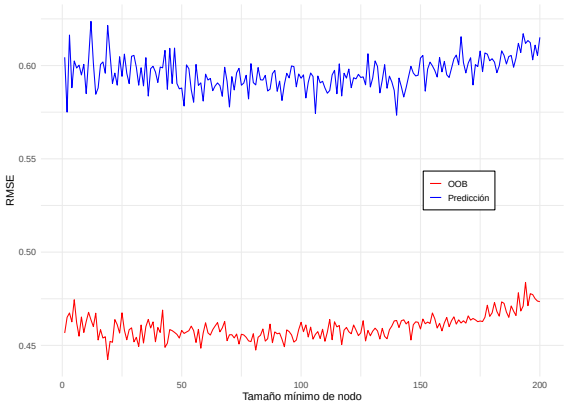
Figura 4.8: Correlación de los árboles frente a número de variables para cada nodo.

En la Figura 4.9a podemos ver la dependencia del RMSE de las predicciones en función del número de árboles que emplee el modelo. De nuevo, vemos que el error relativo a las observaciones *OOB* es inferior en todo caso al error cometido para las predicciones. Además, la dependencia con los árboles vuelve a ser decreciente para cantidades de árboles pequeñas, como es lógico, para luego llegar a un valor estable de RMSE sin depender del número de árboles para valores elevados. En este caso, el número óptimo calculado fueron 100, que es el valor más cercano a la zona en la que parece que se empieza a estabilizar el error.

Por otro lado, en la Figura 4.9b vemos la dependencia del RMSE con n_{min}, el número mínimo de observaciones en cada nodo. Parece que para valores bajos el error se mantiene estable, mientras que al acercarse a $n_{min} = 100$ se percibe una ligera tendencia creciente. Al igual que pasaba con el caso de clasificación, esto se debe a la excesiva simplicidad de los árboles para valores de n_{min} muy altos, restringiendo demasiado su ramificación.

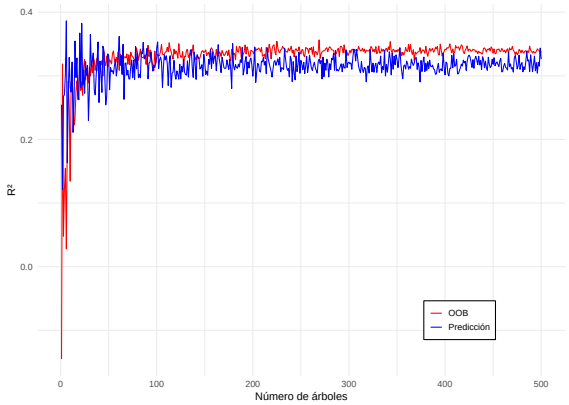


(a) RMSE - Número de árboles.

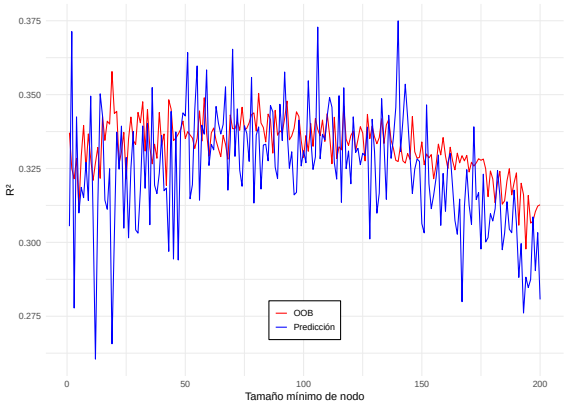


(b) RMSE - n_{min}.

Por completitud, se añade la dependencia de R^2 con los mismos hiperparámetros en las Figuras 4.10a y 4.10b. Vemos cómo se estabiliza para valores muy pequeños del número de árboles, prácticamente sin dependencia para valores altos. En cuanto a la dependencia con n_{min}, a pesar de la alta variabilidad, se puede percibir una ligera tendencia cóncava, es decir, directamente proporcional para valores bajos e inversamente proporcional para valores altos, alcanzando un máximo en un término medio.



(a) R^2 - Número de árboles.



(b) R^2 - n_{min}.

4.4.2. Comparativa con otros modelos

En esta sección se compararán los resultados del modelo *Random Forest* de regresión con los modelos GLM y GBM de regresión, que ya se introdujeron en el apartado 4.3.2. Como los modelos *Random Forest* y GBM consisten en ensamblados de árboles de decisión, se podrán comparar los errores que cometen en función del número total de árboles. Esto se representa en la Figura 4.11 con el MAE (expresión (4.4)) de las predicciones para los valores $mtry \in \{2, 4, 6\}$ en el caso de *Random Forest*[4] y los valores $interaction.depth \in \{2, 4, 6\}$ en el caso de GBM.

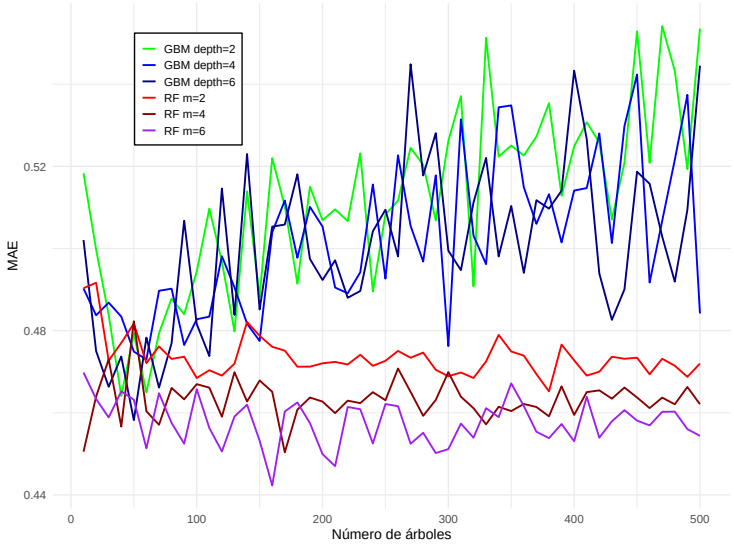
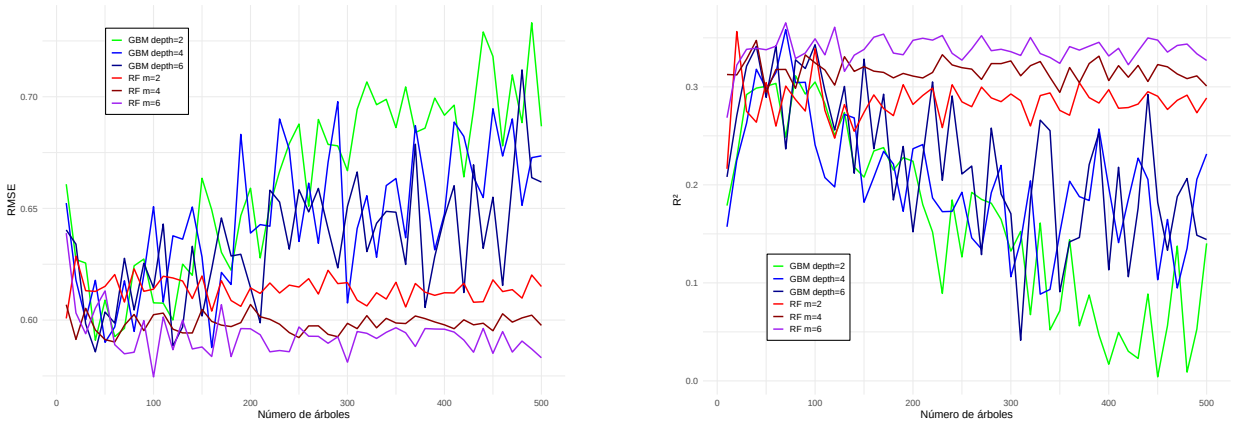


Figura 4.11: Comparación del MAE con los diferentes modelos de GBM.

Vemos que para pocos árboles los modelos registran errores similares, aunque parece que por lo general los GBM cuentan ligeramente con algo más de error. Sin embargo, a medida que el número de árboles aumenta, el error de los *Random Forest* se reduce y estabiliza mientras que el error de GBM aumenta en cualquiera de los 3 casos. De esta gráfica podemos sacar la conclusión de que los modelos se pueden comportar de forma similar para pocos árboles, aunque ligeramente mejor los *Random Forest*; pero para una gran cantidad de árboles los GBM funcionan mucho peor, obteniendo valores de MAE bastante superiores.

Añadimos también en las Figuras 4.12a y 4.12b la evolución de la comparativa del $RMSE$ y del R^2 para las predicciones en función del número de árboles. Estas gráficas llevan a las mismas conclusiones que la anterior, viéndose cómo los modelos se comportan similar para un número reducido de árboles, mientras que para grandes cantidades solo los *Random Forest* estabilizan el error.

[4]Para *maxnodes* y *nodesize* se escogieron los valores óptimos del Cuadro 4.2.



(a) RMSE. (b) R^2.

Figura 4.12: Comparación del RMSE y del R^2 con los diferentes modelos de *GBM*

Para completar el estudio de estos modelos, se realiza una comparativa de las importancias de las variables para los mejores modelos en cada caso, estando los valores óptimos de los hiperparámetros del *GBM* registrados en el Cuadro 4.3. Se añaden también las importancias para el modelo de regresión logarítmica para el caso de regresión.

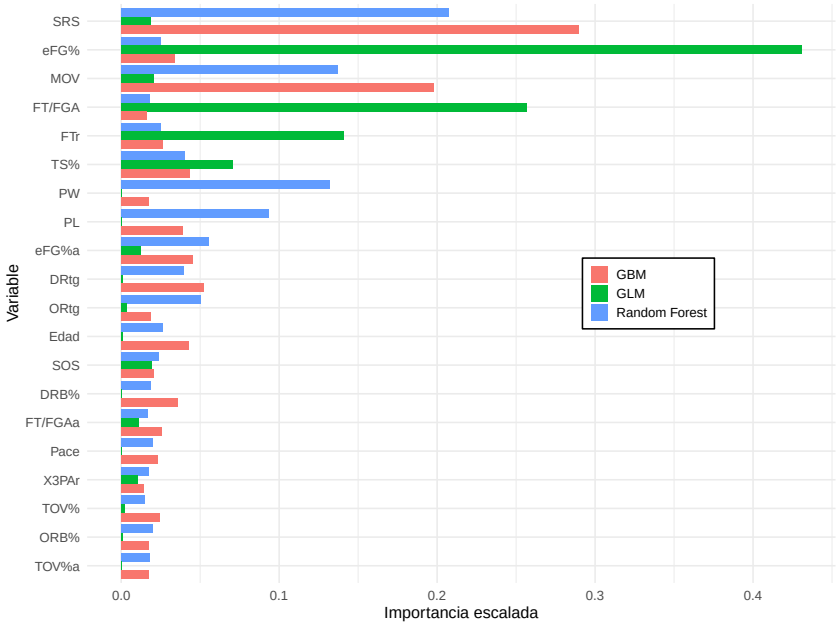


Figura 4.13: Comparación de importancia de variables con GBM.

En la Figura 4.13 se encuentran las importancias de las variables escaladas para poder realizar una comparación entre los modelos. Estas se calcularon con el método de permutación (ver Sección 3.4) en el caso del *Random Forest* y el *GBM*, y de forma análoga al caso de clasificación

para el GLM. De primeras, podemos ver que el modelo GBM acumula la mayor parte de la importancia total en las variables SRS y MOV, que son de las que más correlación tienen con la variable respuesta, tal y como se ve en la Figura 4.1. A grandes rasgos, parece que toda la importancia que reparte el *Random Forest* entre las variables SRS, MOV, PW y PL es la que acumula GBM en esas dos variables, ya que para el resto no hay mucha diferencia entre un modelo y otro. Cabe destacar también la poca importancia que le da GBM a las variables PW y PL ya que, como vimos, son de las que más correlación tienen con la respuesta. Lo mismo podemos decir del GLM, que reparte casi toda la importancia en $eFG\,\%$ y FT/FGA, mientras que ignora al resto de variables menos a FTr y $TS\,\%$. Por lo general, podemos afirmar que, al igual que pasaba con los modelos de clasificación, el *Random Forest* capta mejor la correlación de las variables explicativas con la respuesta que el GBM.

En el Cuadro 4.8 se presentan el $RMSE$, MAE y R^2 relativos a las observaciones de la temporada 2022/2023 para los modelos óptimos calculados. Los resultados son bastante similares en los 3 casos, habiendo diferencias de a lo sumo 0.01 tanto en el $RMSE$ como en el MAE. En cuanto al R^2, el modelo GLM consigue un valor ligeramente superior, por lo que se trata del modelo que es capaz de explicar la mayor proporción de variabilidad de los datos. Sin embargo, la diferencia es tan pequeña con los otros dos modelos que podemos considerar que prácticamente no hay diferencias en los resultados. De nuevo, sorprende que un modelo más sencillo en comparación con los otros como es el GLM consiga resultados competitivos, dando a entender que quizás no sea muy útil con estos datos emplear modelos más complejos para generar predicciones.

	RMSE	MAE	R^2
Random Forest	0.6022	0.4675	0.3107
GLM	0.5932	0.4777	0.3310
GBM	0.5987	0.4729	0.3185

Cuadro 4.8: Resultados de los modelos óptimos de regresión.

4.5. Conclusiones

A lo largo de esta sección se ha presentado la aplicación de modelos *Random Forest*, tanto en el caso de clasificación como en el de regresión, para una base de datos deportivos; en concreto, datos de los equipos de la NBA. Conseguir modelos capaces de predecir con un acierto elevado tanto la proporción de victorias como el acceso a la siguiente ronda de los equipos, aunque fuera el principal objetivo, no era el único. Otros objetivos importantes eran estudiar la dependencia de los modelos con la variación de sus hiperparámetros y comparar resultados con otros modelos

empleados habitualmente en el mismo ámbito.

En cuanto al primer objetivo, desde el primer análisis de la correlación de las variables se podía intuir que no se iba a poder conseguir como nos interesaría. La conclusión principal que podemos sacar de ello es que, antes de pasar a escoger entre los distintos modelos, es importante estudiar la base de datos con la que se trabaja, en particular, conocer las correlaciones entre las distintas variables. Así, si se conoce que las correlaciones entre variables explicativas y variables respuesta son bajas, en caso de ser posible, tratar de buscar otras variables que cuenten con una correlación mayor. En este caso no fue posible proceder de esta manera, puesto que no se encontraron otras variables con mayor correlación.

El análisis de la dependencia de los modelos con sus hiperparámetros fue muy útil para comprender la obtención de los hiperparámetros óptimos minimizando el error. El apoyo en el resultado que muestra el Teorema 3.12 fue fundamental para entender por qué en este caso los valores obtenidos del hiperparámetro *mtry* fueron tan bajos. Por otro lado, viendo la evolución gráfica de los errores en función del número de árboles e identificando a partir de qué valores estos se estabilizaban fue posible estimar sobre qué valores se encontraba el óptimo de número de árboles en cada caso.

Finalmente, a partir de la comparativa con modelos distintos a los *Random Forest* se pudo corroborar la utilidad de este modelo, ya que mejoró ligeramente a sus "competidores" en el caso de clasificación y no empeoró los resultados de los otros modelos de regresión. Sin embargo, a la vista de que modelos mucho menos costosos computacionalmente como LDA y GLM alcanzaban resultados similares, quizá resulte que para estos datos no merezca la pena emplear modelos complejos para generar predicciones.

Anexo I

Criterios de nodo terminal

- **Profundidad máxima del árbol**
  Consiste en limitar el número de niveles de un árbol, es decir, su profundidad. En este caso, la condición L dependerá de un parámetro, el número de niveles h. Este vendrá dado por h_{max}. Para representar la condición, se puede definir una función aplicada a las coordenadas del nodo (h,t) que devuelva 1 cuando se verifica y 0 cuando no.

  Definición I.1. Sean (h,t) las coordenadas de un nodo y sea $h_{max} \in \mathbb{N}$ el valor máximo de niveles fijado para el árbol. Definimos la **función de profundidad máxima del árbol** tal que

  $$L_{deep}(h,t) = \begin{cases} 1 & \text{si } h = h_{max}\,, \\ 0 & \text{si } h < h_{max}\,. \end{cases} \tag{I.1}$$

  No es necesario añadir el caso $h > h_{max}$ a la definición ya que es un caso imposible de alcanzar debido a la propia construcción del algoritmo. Podemos tomar de ejemplo el árbol de la Figura 2.1. Si se hubiera empleado el criterio de profundidad máxima con $h_{max} = 2$, los nodos con $h = 3$ no existirían y los de $h = 2$ serían terminales, como se ve en la Figura I.1.

- **Número mínimo de observaciones**
  Este criterio exige que haya un número mínimo de observaciones asociadas a un nodo para que este pueda ser dividido, es decir, para que no sea un nodo terminal. El número mínimo de observaciones lo denotaremos como n_{min}. Por ejemplo, si en un nodo $\left|O_{(h,t)}\right| = 5$ y $n_{min} = 10$, el nodo (h,t) será terminal. La función asociada al criterio se define de la siguiente forma:

  Definición I.2. Sean (h,t) las coordenadas de un nodo y $O_{(h,t)}$ las observaciones asociadas a dicho nodo. Sea $n_{min} \in \mathbb{N}$ el número mínimo de observaciones para que un nodo pueda

no ser terminal, fijado para el árbol. Se define la **función de mínimo de observaciones** de la siguiente forma:

$$L_{sample}\left(O_{(h,t)}\right) = \begin{cases} 1 & \text{si } \left|O_{(h,t)}\right| < n_{min}\,, \\ 0 & \text{si } \left|O_{(h,t)}\right| \geq n_{min}\,. \end{cases} \tag{I.2}$$

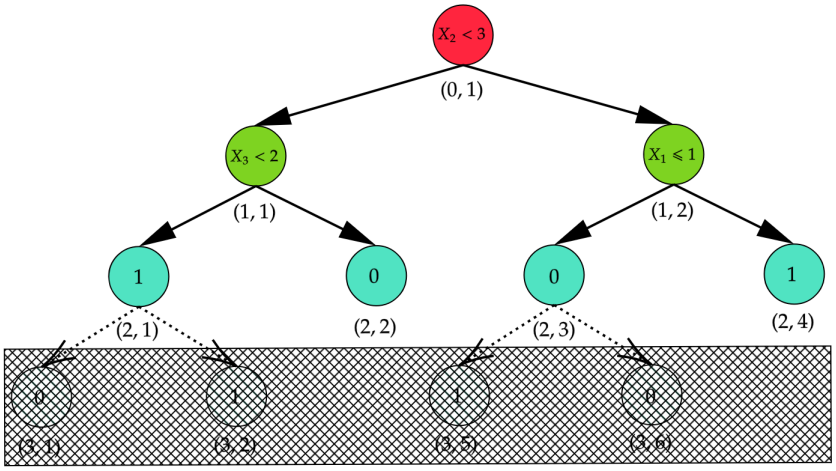


Figura I.1: Árbol de la Figura 2.8 si se hubiera fijado $n_{max} = 2$.

De esta forma, se controla el tamaño del árbol sin necesidad de fijar una profundidad, como se hizo en el caso anterior. Simplemente, se busca que la muestra sea lo suficientemente grande en cada nodo como para poder ser dividida en dos.

- **Criterio de pureza del nodo**

  Este criterio se centra en la diversidad de las respuestas de las observaciones. Si las observaciones tienen respuestas muy parecidas, se considerará que el nodo tendrá que ser terminal. Existe cierta diferencia entre el caso de clasificación y el caso de regresión.

  Para el caso de clasificación, supongamos que estamos trabajando con una variable Y dicotómica. Si en un nodo hay 100 observaciones y todas llevan al valor 0, el nodo se considerará completamente puro. Si de las 100 observaciones 50 llevaran al 0 y otras 50 al 1, la impureza del nodo se consideraría máxima. Esto resulta bastante intuitivo, pero para poder cuantificarlo se hace uso de métricas; las más empleadas son: índice de Gini, entropía y error de clasificación. Las dos primeras ya fueron presentadas en la Sección 2.3, mientras que el error de clasificación se define de la siguiente forma:

  Definición I.3. Sea $O = \{(x_i, y_i)\}_{i=1}^{n}$ una muestra de n observaciones, donde $y_i \in \{1, \ldots, N\}$, $\forall i$. Sea p_i la proporción de observaciones con respuesta i. Se define el **error**

de clasificación de la muestra tal que

$$EC = 1 - \max_{i \in \{1,\dots,N\}} (p_i) \ . \tag{I.3}$$

En este caso, un nodo se consideraría más impuro cuando mayor fuera el error de clasificación.

Para los problemas de regresión la idea es similar. La pureza de un nodo será mayor si sus observaciones son $(10.1, 10.4, 10.3)$ que si son $(10, 33, 4)$. La métrica más empleada es de nuevo el error cuadrático medio. Cuando mayor sea este, mayor será la impureza del nodo.

Para ambos casos, el criterio de pureza consistirá en seleccionar una métrica adecuada y fijar un valor límite. Si denotamos en general por $p_{(h,t)}$ al valor obtenido para el nodo (h,t) de la métrica seleccionada, y p_{min} el valor umbral de pureza que se quiere fijar para el árbol[1], se puede definir la función de pureza del nodo como sigue:

Definición I.4. Sean (h,t) las coordenadas de un nodo y sea $p_{(h,t)}$ el valor obtenido para una cierta métrica fijada a partir de las respuestas de $O_{(h,t)}$. Sea también $p_{max} \in \mathbb{R}$ un valor umbral fijado para la métrica. Se define la **función de pureza del nodo** tal que

$$L_{purity}\left(O_{(h,t)}\right) = \begin{cases} 1 & \text{si } p_{(h,t)} < p_{min} \ , \\ 0 & \text{si } p_{(h,t)} \geq p_{min} \ . \end{cases} \tag{I.4}$$

De esta forma, los nodos con muy poca impureza (valores de la métrica muy bajos) cuyas respuestas se consideran lo suficientemente homogéneas serán obligatoriamente nodos terminales. Los nodos con impurezas altas podrán seguir dividiéndose para aumentar la pureza en los siguientes nodos.

[1] Suponemos sin pérdida de generalidad que el valor de la métrica es mayor cuanto menor sea la pureza. En caso contrario, bastaría trabajar con $-p_{(h,t)}$ y p_{max} para que no hubiera problemas con la definición de la función.

Anexo II

Base de datos

- **Equipo**
  Variable discreta (de tipo factor). Etiqueta que indica a qué equipo de los 30 que pertenecen a la liga está asociada cada observación.

- **Conferencia (Conf)**
  Variable discreta (de tipo factor). Etiqueta que indica a qué conferencia, oeste (O) o este (E) pertenece cada equipo. Esta variable es relevante debido a que de los 16 equipos que se clasifican a *Playoffs*, 8 son de una conferencia y 8 son de otra. Por lo tanto, la conferencia a la que pertenezca un equipo influirá en sus resultados, ya que un número de victorias puede ser suficiente para acceder a *Playoffs* en una conferencia e insuficiente para la otra. En el Cuadro II.1 se presenta una lista de los equipos que conforman cada conferencia con la nomenclatura con la que se representan en las gráficas.

- **Temporada (Temp)**
  Variable discreta (de tipo factor). Etiqueta que refleja el año en el que acaba la temporada a la que está asociada la observación.

- **Edad**
  Variable real positiva. Indica la media aritmética de la edad de los jugadores que formaban parte del equipo la temporada a la que está asociada la observación.

  Definición II.1. Sea J el número de jugadores de un equipo. Se define la **edad de un equipo** como

  $$\text{Edad} = \frac{1}{J}\sum_{i=1}^{J}\text{Edad}_i\,, \tag{II.1}$$

  siendo Edad_i, $i \in \{1, \cdots, J\}$ la edad de un jugador.

OESTE (O)		ESTE (E)	
Dallas Mavericks (DAL)	New Orleans Pelicans (NOP)	Atlanta Hawks (ATL)	Miami Heat (MIA)
Denver Nuggets (DEN)	Oklahoma City Thunder (OKC)	Boston Celtics (BOS)	Milwaukee Bucks (MIL)
Golden State Warriors (GSW)	Phoenix Suns (PHO)	Brooklyn Nets (BKN)	New York Knicks (NYK)
Houston Rockets (HOU)	Portland Trail Blazers (POR)	Charlotte Hornets (CHA)	Orlando Magic (ORL)
Los Angeles Clippers (LAC)	Sacramento Kings (SAC)	Chicago Bulls (CHI)	Philadelphia 76ers (PHI)
Los Angeles Lakers (LAL)	San Antonio Spurs (SAS)	Cleveland Cavaliers (CLE)	Toronto Raptors (TOR)
Memphis Grizzlies (MEM)	Utah Jazz (UTA)	Detroit Pistons (DET)	Washington Wizards (WAS)
Minnesota Timberwolves (MIN)		Indiana Pacers (IND)	

Cuadro II.1: Equipos de la NBA agrupados por conferencias

- **Victorias/Derrotas Pitagóricas Esperadas (PW/PL)**
  Variables enteras. La idea de esta variable proviene del béisbol, donde se calculaba el cociente entre las carreras anotadas al cuadrado y la suma del cuadrado de las carreras anotadas con el de las carreras recibidas. De esta forma, se estimaba el número de victorias que iba a tener un equipo durante una temporada, resultando un estimador bastante bueno [19]. La idea se extrapola al baloncesto como sigue.

  Definición II.2. Sea G el número de partidos jugados por un equipo en una temporada. Sean PS el número de puntos anotados por el equipo y PA el número de puntos recibidos en una temporada. Se define el **número de victorias pitagóricas esperadas** como

  $$PW = G \cdot \frac{PS^{13.91}}{PS^{13.91} + PA^{13.91}}. \tag{II.2}$$

  Análogamente, se define el **número de derrotas pitagóricas esperadas** como

  $$PL = G \cdot \frac{PA^{13.91}}{PS^{13.91} + PA^{13.91}}. \tag{II.3}$$

  El uso del valor 13.91 en el exponente en vez de 2 tiene un origen empírico, ya que usando este exponente se comprobó que se corregía el desfase que se obtenía entre las victorias y las victorias pitagóricas esperadas. Esta corrección es debida al ejecutivo de la NBA Daryl Morey [25].

- **Margen de Victoria (MOV)**
  Variable real. Indica el promedio de puntos por el que gana/pierde un equipo en una temporada [6]. Para que sean valores enteros, los resultados de aplicar la fórmula son redondeados a la unidad.

  Definición II.3. Sea G el número de partidos jugados por un equipo en una temporada. Sean PS el número de puntos anotados por el equipo y PA el número de puntos recibidos en una temporada. Se define el **margen de victoria** como

  $$MOV = \frac{1}{G}(PS - PA) \tag{II.4}$$

- **Fuerza de Calendario (SOS)**

  Variable real. Es de utilidad para comparar el nivel de dificultad de los oponentes de cada equipo, ya que en la NBA no se juega el mismo número de partidos contra todos los equipos (depende de la proximidad entre estos). De hecho, la fuerza de calendario es usada a veces como criterio de desempate en algunas competiciones. Para calcular su valor, es necesario un proceso iterativo en el que interviene otra variable conocida como sistema de clasificación simple [11].

  Definición II.4. Sea MOV el margen de victoria del equipo en la temporada. Se define el valor **sistema de clasificación simple** de la siguiente manera:

  $$SRS = MOV + SOS\,, \tag{II.5}$$

  donde SOS es la fuerza de calendario.

  Definición II.5. Sea Ts el número de equipos existentes en una competición. Sea G_{ij} el número de partidos que juega el equipo i contra el equipo j ($i, j \in \{1, \cdots, Ts\}$). Se define la **fuerza de calendario** del equipo j de la siguiente forma:

  $$SOS_j = \frac{\sum_{i \neq j}(SRS_i \cdot G_{ij})}{\sum_{i=1}^{T} G_{ij}}\,, \tag{II.6}$$

  donde SRS_i es el valor de sistema de clasificación simple del equipo i. Se sobreentiende que $G_{ii} = 0$, $\forall i \in \{1, \cdots, Ts\}$.

  Como se puede apreciar fácilmente, las dos definiciones no son independientes entre sí. Es decir; para la primera definición necesitas la segunda, y viceversa. Es por ello que para obtener los valores de SOS y SRS de cada equipo es necesario un proceso iterativo que converja a los valores que se buscan. El algoritmo es el siguiente:

  1. Se calcula el margen de victoria, MOV, de cada equipo.
  2. Se asigna un valor inicial al SOS de cada equipo (por lo general, se escoge $SOS = 0$).
  3. Se calculan los SRS con la fórmula ya presentada: $SRS = MOV + SOS$.
  4. Se calculan los SOS a partir de los valores de SRS calculados en el paso anterior.
  5. Se repiten los dos pasos anteriores hasta que los valores de SOS y SRS converjan (cuando la diferencia de valores dada por la última iteración sea redundante).

- **Sistema de Clasificación Simple (SRS)**

  Variable real. Emplea la fuerza de calendario (la dificultad de los equipos rivales) y el margen de victoria obtenido por el equipo en la temporada para realizar una clasificación del nivel de los equipos. Puede tomar valores positivos y negativos, al igual que MOV y SOS. El valor 0 representa de alguna forma el promedio de los equipos, estando los equipos con SRS negativo por debajo de la media y viceversa.

- **Calificación Ofensiva (ORtg)**
  Variable real. Indica la cantidad de puntos anotados por el equipo cada 100 posesiones de ataque en la temporada. De esta forma, se valora la calidad de las posesiones ofensivas realizadas por el equipo, midiéndose el número de puntos por posesión conseguidos.

  Definición II.6. Sea PS la cantidad de puntos anotados por el equipo, y OP el número de posesiones ofensivas con las que contó. Se define la **calificación ofensiva** de tal forma que
  $$ORtg = \frac{PS}{OP} \cdot 100\,. \tag{II.7}$$

- **Calificación Defensiva (DRtg)**
  Variable real. Análoga a la calificación ofensiva, empleando esta vez los puntos recibidos y la cantidad de posesiones defensivas.

  Definición II.7. Sea PA la cantidad de puntos anotados contra el equipo, y DP el número de posesiones ofensivas con las que contó. Se define la **calificación defensiva** de tal forma que
  $$DRtg = \frac{PA}{DP} \cdot 100\,. \tag{II.8}$$

- **Ritmo (Pace)**
  Variable real. Mide el número de posesiones ofensivas con las que cuenta un equipo cada 48 minutos. Cuanto mayor sea el número de posesiones por unidad de tiempo, mayor será la velocidad con la que juega el equipo, de ahí el nombre de la variable. El uso de 48 minutos como intervalo temporal de referencia se debe a que es la duración de un partido de NBA sin prórrogas.

  Definición II.8. Sea OP el número de posesiones ofensivas que ha tenido el equipo en la temporada. Sea t el tiempo total (en minutos) que ha estado el equipo en cancha durante la temporada. Se define el **ritmo** del equipo como
  $$Pace = \frac{OP}{t} \cdot 48\,. \tag{II.9}$$

- **Tasa de Tiros Libres (FTr)**
  Variable real. Se trata del cociente entre los tiros libres intentados y los tiros de campo (todos los que no sean tiros libres) intentados. Mide el peso de los tiros libres en el total de tiros que intenta el equipo en la temporada.

  Definición II.9. Sean FTA y FGA el número de tiros libres y tiros de campo intentados durante la temporada, respectivamente. Se define la **tasa de tiros libres** como
  $$FTr = \frac{FTA}{FGA}\,. \tag{II.10}$$

- **Porcentaje de Triples Intentados (X3PAr)**
  Variable real. Mide el porcentaje de tiros de 3 puntos intentados por el equipo en la temporada con respecto del número total de tiros de campo intentados. Refleja la importancia que le da a los tiros de 3 puntos un equipo.

  Definición II.10. Sean $X3PA$ y FGA el número de tiros de tres y de tiros de campo, respectivamente, intentados por un equipo en una temporada. Se define el **porcentaje de triples intentados** como
  $$X3PAr = \frac{X3PA}{FGA}\,. \tag{II.11}$$
  Esta variable representa una proporción, a diferencia de la tasa de tiros libres, ya que los tiros de 3 entran dentro del global de tiros de campo, no ocurriendo esto con los tiros libres.

- **Porcentaje de Tiro Real (TS %)**
  Variable real. Se trata de una métrica que refleja la eficiencia en el tiro de un equipo. Para ello, pondera el valor de los tiros libres y los tiros de tres de manera que sean comparables a los tiros de dos puntos, y se calcula un valor que se acerca al número de puntos conseguidos entre número de puntos intentados. De esta forma, como se emplean para el cálculo tres tiros posibles y se ponderan en relación a su relevancia, esta métrica permite comparar la eficiencia en el tiro de equipos con formas de anotación muy distinta [29].

  Definición II.11. Sean FTM, $X2PM$ y $X3PM$ los tiros anotados de uno (tiro libre), dos y tres puntos, respectivamente, por un equipo en una temporada. Sean FTA y FGA los tiros libres y de campo, respectivamente, intentados por dicho equipo en la temporada. Se define el **porcentaje de tiro real** mediante la siguiente cantidad:
  $$TS\,\% = \frac{\frac{FTM}{2} + X2PM + \frac{3}{2}\cdot X3PM}{FGA + (0.44\cdot FTA)} = \frac{PS}{2\cdot[FGA + (0.44\cdot FTA)]}\,, \tag{II.12}$$
  donde PS son los puntos anotados por el equipo a lo largo de la temporada.

  El factor 0.44 se trata de un valor empírico que refleja, en promedio, cuánto "cuesta" en términos de posesiones cada intento de tiro libre. Esto se debe a que, aunque lo habitual es que haya dos tiros libre por posesión (en una falta personal común), también existen situaciones en las que solo se tira un tiro libre, o incluso tres. Por lo tanto, el factor 0.5 que se deduciría de la situación de 2 tiros libres se corregiría llegando al 0.44 resultante de la gran cantidad de datos de partidos con la que se cuenta. Nótese que el valor calculado puede alcanzar el 150 %.

- **Factor de tiro libre (FT/FGA)**
  Variable real. Trata de medir el peso que tienen los tiros libres anotados con respecto el conjunto de lanzamientos que realiza un equipo en un partido. En particular, se trata de una proporción entre los tiros libres anotados y los tiros de campo intentados.

Definición II.12. Sean FTM y FGA los tiros libres anotados y de campo intentados, respectivamente, por un equipo a lo largo de una temporada. Denotamos como **factor de tiro libre** al cociente entre ambas cantidades:

$$FT/FGA = \frac{FTM}{FGA}\,. \tag{II.13}$$

La variable $FT/FGAa$ se refiere a este mismo cálculo realizado con los tiros que efectuaron los rivales al jugar contra el equipo asociado a la observación en esa temporada. Lo mismo ocurre con $eFG\,\%a$ y $TOV\,\%a$.

- **Porcentaje Efectivo de Tiros de Campo (eFG %)**
  Variable real. Recoge la eficiencia de los tiros de campo (triples y lanzamientos de 2 puntos) de un equipo a lo largo de una temporada, dándole más valor (un 50 % más) a los tiros de 3 puntos que a los de dos.

  Definición II.13. Sean FGA los tiros de campo totales intentados por el equipo durante la temporada. Sean FGM los tiros de campo totales anotados y $X3PM$ los tiros de tres anotados por el equipo durante la temporada. Se define el **porcentaje efectivo de tiros de campo** de la siguiente forma:

$$eFG\,\% = \frac{FGM + \frac{X3PM}{2}}{FGA}\,. \tag{II.14}$$

  Cabe destacar que, de nuevo, se puede alcanzar hasta el 150 %. Nótese también que $FGM + \frac{X3PM}{2} = X2PM + \frac{3}{2}X3PM$, que es la ponderación que se busca entre los tiros de 2 y los de 3 puntos.

- **Pérdidas por Posesión (TOV %)**
  Variable real. Una pérdida sucede cuando un jugador del equipo en ataque pierde el balón ante la defensa. Esta variable recoge la cantidad de pérdidas de balón en ataque que tiene un equipo cada 100 posesiones de ataque a partir de los valores totales de la temporada. Es de gran ayuda a la hora de estudiar el aprovechamiento de las posesiones ofensivas.

  Definición II.14. Sea TOV el número total de pérdidas de un equipo en una temporada y OP el número de posesiones ofensivas con las que contó el equipo. Se define las **pérdidas por posesión** de la siguiente forma:

$$TOV\,\% = \frac{TOV}{OP} \cdot 100\,. \tag{II.15}$$

- **Porcentaje de Rebotes Ofensivos (ORB %)**
  Variable real. Un rebote ofensivo es aquel que coge un jugador que se encuentra atacando tras un tiro de su equipo a la canasta del rival. Esta variable recoge la cantidad de rebotes

que consiguió el equipo cada 100 posesiones durante la temporada. Ayuda a contabilizar la frecuencia con la que el equipo consigue nuevas posesiones ofensivas gracias a la captura del rebote.

Definición II.15. Sea ORB el número de rebotes ofensivos capturados por el equipo a lo largo de la temporada, y OP el número de posesiones ofensivas con las que contó el equipo. Entonces, la expresión para el **porcentaje de rebotes ofensivos** es

$$ORB\,\% = \frac{ORB}{OP} \cdot 100\,. \tag{II.16}$$

Las variables (FT/FGA, $eFG\,\%$, $TOV\,\%$ y $ORB\,\%$) conforman el grupo conocido como "4 factores ofensivos", a los que los analistas deportivos, en particular de la NBA, les dan especial relevancia. [7].

- **Porcentaje de Rebotes Defensivos (DRB %)**
  Variable real. Se define de forma análoga al porcentaje de rebotes ofensivos.

  Definición II.16. Sea DRB el número de rebotes ofensivos capturados por el equipo a lo largo de la temporada, y DP el número de posesiones ofensivas con las que contó el equipo. Entonces, la expresión para el **porcentaje de rebotes ofensivos** es

$$DRB\,\% = \frac{DRB}{DP} \cdot 100\,. \tag{II.17}$$

- **Acceso a *Playoffs***
  Variable discreta (tipo factor, dicotómica). Recoge si el equipo alcanzó los *playoffs* en la temporada. A esta fase de la competición, que sucede tras la temporada regular (a la que pertenecen los datos con los que contamos) acceden 16 de los 30 equipos, 8 por cada conferencia. El valor es 1 cuando el equipo alcanza los *playoffs* y es 0 cuando no lo hace.

- ***Odds***
  Variable real. Recoge el valor *odds* asociado al equipo en la temporada. La *odds* se define para una distribución de Bernoulli de la siguiente forma:

  Definición II.17. Sea X una variable aleatoria siguiendo una distribución de Bernoulli de parámetro $p \in [0, 1)$, $X \sim Bernoulli(p)$. De define el valor *odds* de la distribución como el cociente de la probabilidad de éxito, p, entre la probabilidad de fracaso, $1 - p$.

$$Odds = \frac{p}{1-p}\,. \tag{II.18}$$

  Nótese que en el caso $p = 1$ (suceso seguro), la función *odds* no estaría definida. Pero este caso es anecdótico, por lo que no influirá en los resultados posteriores. En este caso, la

estimación del valor *odds* será el porcentaje de victorias entre el porcentaje de derrotas, que es la variable que se recogerá en la base de datos:

$$\widehat{Odds} = \frac{\frac{W}{W+L}}{\frac{L}{W+L}} = \frac{W}{L}\,. \tag{II.19}$$

Tanto $Playoffs\_n$ como $Odds\_n$ se refieren a los valores definidos anteriormente referidos al mismo equipo la temporada siguiente a la que está asociada la observación. De esta forma, reflejan los resultados que consiguió el equipo la temporada siguiente a la de los datos con los que cuenta la observación.

Bibliografía

[1] J. Amat Rodrigo, "Random Forest con Python," *Ciencia de Datos*, 2020. Disponible en: https://cienciadedatos.net/documentos/py08_random_forest_python.html. [Consulta: 8 jun. 2025].

[2] A. A. Awan, *What is Bagging in Machine Learning? A Guide With Examples*, nov. 2023. [Consulta: 3 mar. 2025].

[3] S. Doroudi, "The Bias-Variance Tradeoff: How Data Science Can Inform Educational Debates," *AERA Open*, vol. 6, no. 4, pp. 1–18, 2020.

[4] B. Efron y R. J. Tibshirani, "An Introduction to the Bootstrap," *Chapman & Hall/CRC*, vol. 57, pp. 1–150, 1993.

[5] R. E. Schapire, "Explaining AdaBoost," en *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, Springer, cap. 5, pp. 37–52, 2013.

[6] BasketballReference, *Glossary NBA Stats*, 2025. Disponible en: https://www.basketball-reference.com/about/glossary.html. [Consulta: 12 mar. 2025].

[7] BasketballReference, *Four Factors*, 2025. Disponible en: https://www.basketball-reference.com/about/factors.html. [Consulta: 15 mar. 2025].

[8] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[9] BlueDreamV1B3, *NBA Teams Stat 2000-2023*, 2023. Disponible en: https://www.kaggle.com/datasets/bluedreamv1b3/nba-teams-stat-2000-2023. [Consulta: 7 abr. 2025].

[10] R. M. Crujeiras y M. Febrero Bande, *Modelos de Regresión y Análisis Multivariante: Tema 6. Regresión logística e introducción a los modelos lineales generalizados*, 2024.

[11] D. Drinen, *A very simple ranking system*, may. 2006. Disponible en: https://web.archive.org/web/20180531115621/https://www.pro-football-reference.com/blog/index4837.html?p=37. [Consulta: 28 feb. 2025].

[12] M. Borrajo García, W. González Manteiga y C. Sánchez Sellero, *Probabilidad y Estadística: Tema 1. Elementos básicos de un vector aleatorio*, 2022.

[13] M. Borrajo García, W. González Manteiga y C. Sánchez Sellero, *Probabilidad y Estadística: Tema 2. Vector de medias y matriz de covarianzas*, 2022.

[14] M. Borrajo García, W. González Manteiga y C. Sánchez Sellero, *Probabilidad y Estadística: Tema 8. Convergencia de sucesiones de variables aleatorias*, 2022.

[15] M. Borrajo García, W. González Manteiga y C. Sánchez Sellero, *Probabilidad y Estadística: Tema 9. Ley de los grandes números y teorema central del límite*, 2022.

[16] L. M. Varela Cabo, H. Montes Campos y T. Méndez Morales, *Mecánica estadística*, 1.ª ed., USC Editora, col. "24 USC Editora. Manuais", 2024. ISBN: 9788410142237.

[17] Wikipedia, *Gradient Boosting*, 2025. Disponible en: https://en.wikipedia.org/wiki/Gradient_boosting. [Consulta: 21 feb. 2025].

[18] Wikipedia, *Jensen's inequality*, oct. 2011. Disponible en: https://en.wikipedia.org/wiki/Jensen's_inequality. [Consulta: 9 abr. 2025].

[19] Wikipedia, *Pythagorean expectation*, ene. 2025. Disponible en: https://en.wikipedia.org/wiki/Pythagorean_expectation. [Consulta: 17 feb. 2025].

[20] A. Natekina y A. Knoll, "Gradient Boosting Machines, a tutorial," *Frontiers in Neurorobotics*, vol. 7, pp. 1–21, dic. 2013.

[21] A. F. Jadama, B. Jobarteh y M. M. Islam, "Ensemble Learning: Methods, Techniques, Application," *IEEE Access*, vol. 12, pp. 57489–57504, jun. 2024.

[22] T. Chen y C. Guestrin, "XGBoost: A Scalable Tree Boosting System," en *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[23] S. Lee, *A Practical Guide to Effective Leave-One-Out Cross-Validation Methods*, mar. 2025. [Consulta: 24 may. 2025].

[24] M. Febrero, *Regresión y Análisis Multivariante. Componentes Principales, Factorial Discriminante y Clustering*, 2019.

[25] D. Morey, *Modified Pythagorean Theorem*, 1994. Disponible en: https://morey.org/pythbook.gif. [Consulta: 3 jun. 2025].

[26] I. D. Mienye y N. Jere, "A Survey of Decision Trees: Concepts, Algorithms, and Applications," *IEEE Access*, vol. 12, pp. 86716–86727, 2024.

[27] GeeksforGeeks, *Feature Importance with Random Forest*, abr. 2024. Disponible en: https://www.geeksforgeeks.org/feature-importance-with-random-forests/. [Consulta: 1 feb. 2025].

[28] M. R. Olea, *Variable importance in linear regression versus random forest. Analysis over costumer loans prepayment rates*, tesis de máster, Universidad Complutense de Madrid, 2020.

[29] NBAStuffer, *True Shooting Percentage (TS)*, 2025. Disponible en: https://www.nbastuffer.com/analytics101/true-shooting-percentage/. [Consulta: 10 abr. 2025].

[30] T. Hastie, R. Tibshirani y J. Friedman, *The Elements of Statistical Learning*, 2.ª ed., Springer, 2009.