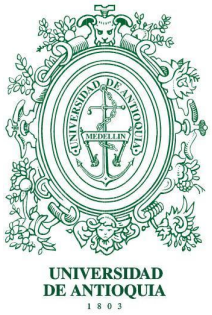



# Desempeño del bosque aleatorio en presencia de correlación entre variables explicativas

*Cristian Camilo Hidalgo García*

Universidad de Antioquia
Facultad de Ciencias Exactas y Naturales, Instituto de Matemáticas
Medellín, Colombia
2017

# Desempeño del bosque aleatorio en presencia de correlación entre variables explicativas

**Cristian Camilo Hidalgo García**

Trabajo de grado presentado como requisito parcial para optar al título de:
**Estadístico**

Director:
Juan David Ospina Arango

Universidad de Antioquia
Facultad de Ciencias Exactas y Naturales
Instituto de Matemáticas
Medellín, Colombia
2017

*Dedicado a*

*mi familia,*
*especialmente a mis padres y abuelos,*
*por creer que puedo.*

# Agradecimientos

Agradezco a la Universidad de Antioquia por abrirme las puertas al conocimiento que allí crece y se mantiene, a mis profesores, a quienes debo la disciplina y constancia en lo querido, la comprensión de las bases que sostienen la Estadística -*El arte de la conjetura*- y por hacer de mí un profesional idóneo. A mis compañeros, con los que he compartido un proceso de formación y experiencia de vida que ha transformado nuestros propios proyectos e ideas. A mi familia, por apoyarme en los momentos difíciles, su compañía y palabras me ayudaron a ser el *amo de mi destino, el capitán de mi alma* y a no doblegarme ante *las azarosas garras de la circunstancia*. Agradezco a mi tutor, por darme la oportunidad de adentrarme en esta exploración, esta corta sinfonía que suena en pañales.Agradezco a las personas que se unieron a mi en esta etapa de la vida de una manera tardía, pero que su compañía y apoyo hicieron de mí algo menos caótico -*M.*-. Agradezco finalmente, aunque suene patético, a mi perro tito -*que ya no vive*- por enseñarme el silencio y la compañía.

# Resumen

En este trabajo se estudió cómo la fuerte correlación entre variables explicativas afecta la predicción de los bosques aleatorios. Para ello, se tomó la media cuadrada de los errores de predicción en datos simulados de normales multivariadas y se comparó con la media cuadrada de los errores del mejor estimador bajo estos supuestos, que es el modelo lineal de regresión múltiple. Esto se realizó en varios escenarios de multicolinealidad. Se estudió la razón entre estos errores para evaluar el comportamiento del bosque aleatorio a medida que el número de variables relacionadas crece. Se analizó el caso donde las covariables/ variables predictoras presentan alta correlación, ya que en presencia de multicolinealidad, el bosque aleatorio disminuye la importancia de estas variables aunque sean importantes para explicar la variable dependiente. Se realizó una aplicación con datos reales sobre la calidad del aire en Medellín, entrenando el algoritmo de bosques aleatorios para hacer predicción del material particulado *pm* 2.5.

**Palabras clave:** Bosques aleatorios, regresión lineal multivariada, árboles de decisión, media cuadrada de errores.

# Abstract

1031/5000 In this work we studied how the strong correlation between explanatory variables affects the prediction of random forests. For this, the square mean of the prediction errors in simulated multivariate normal data was taken and compared with the square mean of the errors of the best estimator under these assumptions, which is the linear multiple regression model. This was done in several multicollinearity scenarios. The reason among these errors was studied to evaluate the behavior of the random forest as the number of related variables grows. We analyzed the case where the covariates / variables predictors present high correlation, since in the presence of multicollinearity, the random forest decreases the importance of these variables although they are important to explain the dependent variable. An application was made with real data on air quality in Medellin, training the algorithm of random forests to make prediction of particulate material $pm$ 2.5.
**Keywords :**Random forest, multivariate linear regression, decision trees, mean square error.

# Índice general

# Índice de figuras

# Índice de tablas

# Notación

La notación que utilizaremos a lo largo de los capítulos la definimos a continuación:
$MSE_{\text{RF}}$: Error cuadrático medio del bosque aleatorio.
$MSE_{LM}$: Error cuadrático medio del modelo lineal múltiple.
$MSE_{tree}$: Error cuadrático medio del árbol de decisión. $X$ Matriz con las variables predictoras.
$X_j$ Variable predictora $j$.
$\tau$: Correlación entre $X_j$ y $Y$
$c$ Correlación entre entre la variable $X_j$ y $X_i$.
$\Omega$: Universo o espacio muestal.
$\chi$: Conjunto de vectores descriptivos.
$RSS$:Suma de cuadrados de los errores.
$EC_\alpha(T)$: Función de costo de complejidad.
$err(T)$: Estimación de error de un árbol de decisión $T$.
$\alpha$ : Parámetro de complejidad.
$\widetilde{T}$: Cantidad de nodos terminales o hojas que contiene el árbol T.
$\Theta$: Árbol perteneciente al bosque aleatorio (*random forest*).
$errOOB$: Error fuera de la bolsa *error out of bag*.
$\beta$: Vector de parámetros del modelo lineal.
$R^2$: Coeficiente de determinación.
$\tau$: Correlación entre la variable $X_j$ y $y$.
$c$: Correlación entre $X_j$ y $X_j$.
$I(X_j)$: Importancia de la variable $X_j$ en el bosque aleatorio.
$C$: Matriz de varianzas covarianzas entre los $X_i$.
$\Sigma$: Matriz de varianzas covarianzas.

# Introducción

Desde que la inteligencia artificial comenzó su auge, ha sido prioritaria la ambición por comprender y descubrir relaciones complejas en los datos. En otras palabras, ha sido de interés encontrar modelos que no solo produzcan predicciones exactas, sino que además nos ayuden a extraer conocimiento de los datos de una manera eficaz.
El aprendizaje automático es la salida a estas ambiciones y ha resultados en una deriva de algoritmos que van en todas direcciones, rescatando de ellos, los métodos que tienen como base los árboles de decisión, siendo muy eficaces y útiles a la hora de producir resultados confiables y comprensibles en cualquier tipo de datos.
Históricamente, los bosques aleatorios (*Random Forest*) tuvieron su primer aparición en un artículo de Leo Breiman et al.[4] donde se hacía un tratado que hasta hoy sigue vigente (aunque ahora podemos encontrar variantes del método original). Algunos programas de software libre como *R project* utiliza en su paquete *randomForest* este método. Los bosques aleatorios ganan cada vez más cancha entre investigadores, esto lo podemos evidenciar dando un vistazo a *Google trends*.

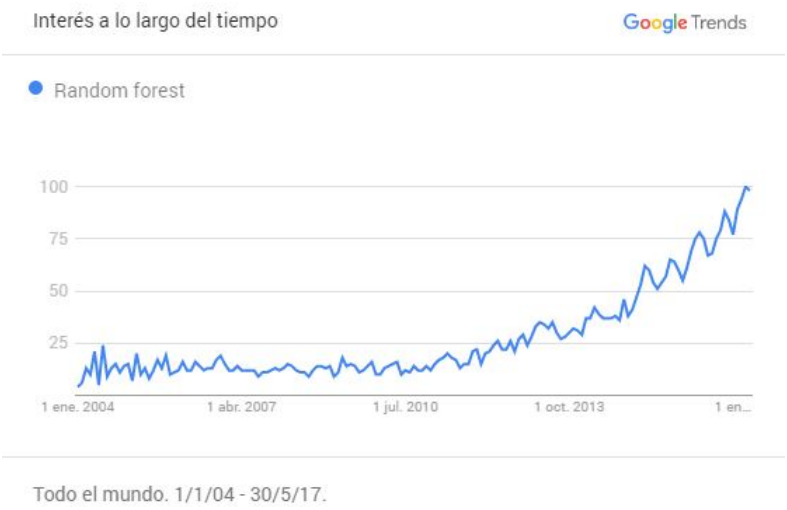



Figura 1: Tendencia en la búsqueda de los últimos trece años en google sobre *random forest*. *Google trends*.

Un tema de investigación sobre el algoritmo de aprendizaje de máquinas bosques aleato-

rios ha sido el de la afectación de la presencia de variables correlacionadas en la importancia de las variables, medida que adjudica el algoritmo a las variables predictoras dando mayor importancia a aquellas que registran cambios importantes en la variable respuesta al ser modificadas. Los estudios en cuanto a predicción no se han dado en el campo y el objetivo de este trabajo es ver si esta correlación presente en variables predictoras afecta el desempeño (predicción) del algoritmo.

En artículos recientes, Greggoruti et al. [9] trabajaron sobre la afectación de la presencia de correlación en variables explicativas en la importancia de la variable (*Variable importance*) del bosque aleatorio, proponiendo escenarios a partir de los cuales tratar el tema. Basado en estos escenarios (y otros más) he abordado el problema a tratar en este trabajo. Teniendo como base inicial un algoritmo de alto rendimiento como lo es la regresión lineal múltiple, comparé en cada escenario propuesto los errores cuadráticos medios de la estimación de los modelos, obteniendo mediante una razón de errores una medida del desempeño del bosque aleatorio.
Los experimentos han mostrado que bajo escenarios de correlación el desempeño del bosque es cercano al desempeño del modelo lineal generalizado (caso 2, 4, 5), en el caso en que agregamos variables incorrelacionadas entre sí pero con alta correlación con la variable dependiente, observamos una mejora sustancial del modelo lineal generalizado (caso 1, 3). Cuando las variables regresoras son numerosas (ej caso 6), el modelo de regresión lineal se comporta muy bien dado el elevado número de variables explicativas.
El siguiente trabajo presenta como capítulo inicial una breve introducción a los árboles de decisión y los bosques aleatorios. En el segundo capítulo se trata el modelo de regresión múltiple. El tercer capítulo muestra las simulaciones realizadas y los resultados obtenidos en cada escenario. En el cuarto capítulo se dan las conclusiones obtenidas dados los resultados y finalmente, se tiene un apéndice donde se deja el código utilizado para cada simulación en el programa *R project*.

# Capítulo 1

# Árboles de regresión y bosques aleatorios

## 1.1. Árboles de decisión

Los árboles de decisión es un método de aprendizaje de máquinas supervisado que utiliza datos históricos para encontrar una estructura en los mismos y crear, a partir de reglas simples, una clasificación de los datos.

La metodología CART (*Classification and Regression Trees*) fue creada en los años 80 [3]. Para la construcción de un árbol de decisión, utilizamos un conjunto de datos conocido como *datos de entrenamiento*[1] cuya respuesta es asignada (aprendizaje supervisado).

Los árboles de decisión han tenido una gran acogida a la hora de hacer interpretaciones de datos ya que entenderlo es relativamente fácil, esto sobre todo, porque sus criterios se basan en reglas simples que guían al intérprete hacia una 'decisión' mediante preguntas del tipo *¿Es hombre?*, *¿El mayor de 25?* (dependiendo del contexto).

El éxito de los árboles de decisión (y todos los métodos basados en árboles) se explica por varios factores que los hacen muy atractivos en la práctica:

- Es un método no paramétrico, puede modelar estructuras complejas en los datos sin ningún supuesto *a priori*.
- Maneja datos heterogéneos, es decir que variables numéricas y categóricas pueden ser parte del mismo análisis.

[1] Podemos particionar la base de datos en dos mediante un muestro aleatorio. Utilizaremos una parte de esta base de datos para entrenar el árbol (usualmente el 70-80 %) y la otra parte, para testear la eficacia del método, ya sea por medio de la suma cuadrada de errores para regresión o una tabla de confución para clasificación.

- Los árboles de decisión implementan intrínsecamente la selección de características, haciéndolos robustos a variables irrelevantes o ruidosas (Por lo menos hasta cierto punto)[11].
- Son robustos a datos atípicos o error en las etiquetas.
- Son fácilmente interpretables incluso por personas que no saben estadística.

Es muy importante comprender el funcionamiento de los árboles de decisión, ya que son la base para muchos algoritmos modernos, incluyendo *random forest*, *boosting*, donde son utilizadons como bloques de construcción para crear modelos más grandes.

### 1.1.1. Estructura de los árboles de decisión

Cuando el espacio de salida es un conjunto de valores finitos $y = \{a_1, a_2, \dots a_n\}$, así que podemos representar $y$ como una partición del universo $\Omega$, esto es [11]:

$$\Omega = \Omega_{a_1} \cup \Omega_{a_2} \cup \cdots \cup \Omega_{a_n}$$

Donde $\Omega_{a_i}$ es el conjunto de valores donde $y$ tiene valor $a_i$. Similarmente, un clasificador$\varphi$ también puede considerarse como una partición del universo $\Omega$, ya que define una aproximación de $y$ mediante $\hat{y}$. Sin embargo, esta partición se define en el espacio de entrada $\chi$ en vez de en $\Omega$, como sigue:

$$\chi = \chi^{\varphi}_{a_1} \cup \chi^{\varphi}_{a_2} \cup \cdots \cup \chi^{\varphi}_{a_n}$$

Donde $\chi^{\varphi}_{a_k}$ es el conjunto de vectores descriptivos $x \in X$ tal que $\varphi(x) = a_k$.

### 1.1.2. Árboles de regresión

Los árboles de regresión son un tipo de regresión no- paramétrico que les da robustez a la hora de tratar con datos atípicos y facilita su manejo cuando se tienen bases de datos con elevado número de variables. Una propiedad importante de los árboles de decisión es que el hecho de transformar los datos con funciones monótonas (exponencial, logaritmo,...) no cambia la estructura del árbol. Una vez construido, podremos usarlo para clasificar nuevos individuos mediante sus reglas de agrupamiento.

El criterio de partición en cada nodo se da por la suma cuadrada de los errores, separando la muestra donde esta suma sea mínima para dos grupos de la variable dependiente.

$$RSS = \sum_{left}(y_i - y_l^*)^2 + \sum_{left}(y_i - y_r^*)^2$$

donde $y_l^*$ es la media de los datos del nodo izquierdo y $y_r^*$ es la media de los datos en el nodo derecho, este proceso se repite hasta que se alcanza algún criterio de parada, por ejemplo, que el número de datos en cada nodo terminal sea $< 2$, se le llama a esto árbol máximo o crecido, y en ocasiones, especialmente en regresión, puede tornarse compleja su interpretación por el gran número de criterios a cumplir antes de clasificar un nuevo individuo.

### 1.1.3. Poda de árboles

Como vimos anteriormente, un árbol crecido torna complejo la interpretabilidad del mismo, por lo que existen maneras de reducir el número de reglas de decisión antes de usarlo. A esto le llamamos, poda del árbol.

Existen dos métodos de poda que son en los que se basa normalmente un analista para reducir el tamaño del árbol: Optimización del número de puntos en cada nodo y validación cruzada [14].

#### Optimización del número de puntos en cada nodo

El objetivo de este método es disminuir el número de particiones aumentando el número de observaciones por nodo, es decir que, en vez de parar cuando las observaciones en un nodo son $n < 2$, se puede dejar de crecer el árbol cuando $n < 10$. Esto puede generar buenos resultados pero se está introduciendo un parámetro demás, en este caso, $n_{min}$, que es el número mínimo de individuos en cada nodo requeridos para dejar de crecer el árbol. En la práctica, se suele fijar este parámetro en $0.1N$, donde $N$ es el total de observaciones de la muestra.

#### Validación cruzada

El procedimiento de validación cruzada, está basado en buscar un equilibrio óptimo entre el número de nodos y la suma cuadrada de errores. Ya que con el aumento del número de nodos, el error del árbol disminuye, por ejemplo, si dejáramos crecer un árbol sin poda, en el que cada nodo terminal tiene una observación, el árbol se ha adaptado o sobre-entrenado y el error es cero. Por el contrario, si nuestro criterio de parada es muy ligero, es decir que el número de observaciones mínimas en cada nodo es alto, el árbol tendrá un alto error clasificación. Así que encontrar el óptimo entre estos dos caminos se logra a través de la función de costo de complejidad:

$$EC_\alpha(T) = err(T) + \alpha\widetilde{T}$$

donde,
$err(T)$ es la estimación del error del árbol T

$\widetilde{T}$ es la cantidad de nodos terminales o hojas que contiene el árbol T
$\alpha$ es llamado parámetro de complejidad y define el coste de cada hoja

Aunque la validación cruzada no requiere de parámetros, el costo computacional es alto si se tienen bases de datos grandes, ya que construir la secuencia de árboles con un conjunto de entrenamiento diferente cada vez puede tardar el proceso, incluso en ocasiones la estructura de los árboles puede diferir de vez en cuando debido a esta estructura seleccionada azarosamente.

### 1.1.4. Ejemplo

Hablaremos en este caso sobre la calidad del vino, usaremos una base de datos abierta al público que se puede encontrar en `https://github.com/stedy`.

Los datos incluyen ejemplos de vinos blancos Vinho Verde de Portugal, uno de los principales países productores de vino del mundo. Hay 11 variables que representan propiedades químicas del vino blanco y una muestra de 4898 vinos. Para cada vino, se analizaron en el laboratorio características como la acidez, contenido de azúcar, cloruros, azufre, alcohol, pH y densidad. Las muestras se clasificaron a continuación en una cata a ciegas por paneles de no menos de tres jueces en una escala de calidad que variaba de cero (muy malo) a 10 (excelente). En el caso de que los jueces no estuvieran de acuerdo con la calificación, se usó la mediana de las calificaciones.
En el artículo donde se trabajó pro primera vez la base de datos [5], se hace una comparación de diferentes métodos (redes neuronales, máquinas de soporte vectorial y regresión lineal múltiple) para comparar qué modelo se ajusta mejor en la predicción de la calidad del vino obteniendo como resultado que el algoritmo de máquinas de soporte vectorial obtenía un mejor desempeño pero su interpretación se hacía más compleja en comparación con el modelo lineal.

En este caso, usaremos el 75 % de los datos para entrenar el modelo y el restante 25 % para validarlo. Para más información sobre las variables ver Cortez et al. [5].

El modelo entrenado puede verse en la figura 1.1, observe que la variable más importante es el porcentaje de alcohol que contiene el vino blanco, lo que nos dice que podremos adjudicar una calificación de calidad a un vino según algunos criterios químicos. Por ejemplo, si tuviésemos una nueva observación tendríamos en cuenta si el porcentaje de alcohol es menor a 10.9, como lo indica el árbol y seguir uno de los caminos hasta llegar a un nodo terminal. En cada nodo terminal se representa el promedio de las observaciones que cumplen con dicho criterio. Una manera de observar si se hace una buena predicción es con el error cuadrático medio del modelo, así, el error cuadrático medio en los datos de validación es (Ver apéndice-

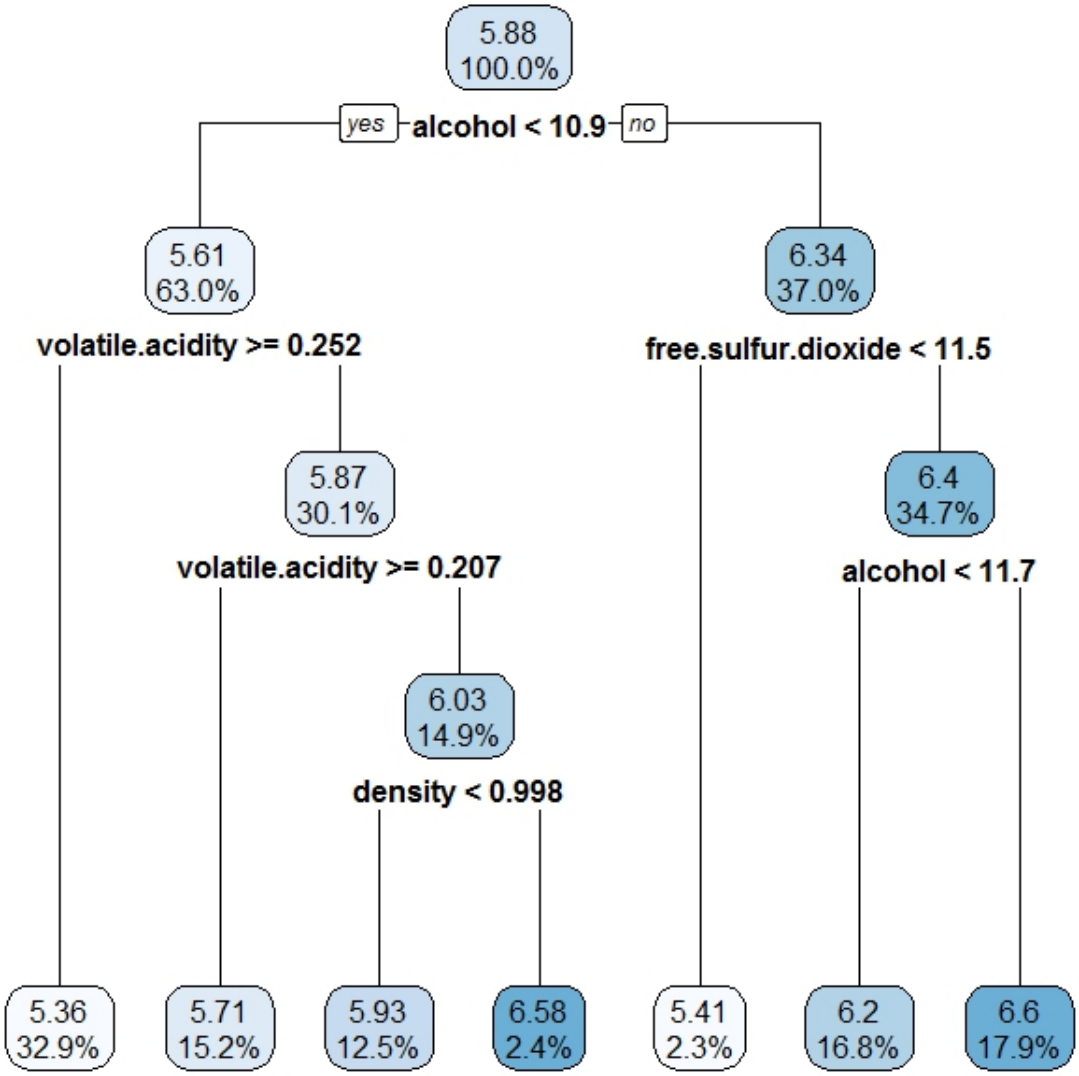



Figura 1.1: Árbol de regresión para la calidad del vino blanco

código capítulo 1):

$$MSE_{tree} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| = 0.534$$

Esto sugiere que la predicción del modelo está desviada en promedio 0.73, para ser en una escala de cero a 10, parece ser bueno. En el artículo original, la menor desviación se obtuvo con las máquinas de soporte vectorial obteniendo una desviación promedio de 0.67.

Entender el funcionamiento del árbol de regresión es fundamental para entender el bosque

aleatorio, introducción que daré en la próxima sección.

## 1.2. Bosques aleatorios

En estadística, sobretodo en las últimas décadas, se han venido popularizando algunos métodos analíticos debido a su eficiencia para analizar los datos, esto en parte, al crecimiento acelerado de la información y el desarrollo de la computación. Así, uno de los métodos que ha tenido una amplia acogida entre la comunidad de estadísticos para el análisis de la información, es el *Random Forest* o bosque aleatorio [4]. El bosque aleatorio, es un algoritmo de aprendizaje supervisado [2] que utiliza como base el *bagging* (*'boostrap aggregating'*) [2], metodología que consiste en la agregación de árboles de decisión [3] idénticamente distribuidos seleccionados por el tipo de muestreo *boostrap* [1], creando un predictor en cada árbol seleccionado para finalmente unificar todos los predictores y obtener uno muy eficaz. En este trabajo se hará un énfasis en los problemas de regresión, es decir, cuando la variable dependiente es cuantitativa.

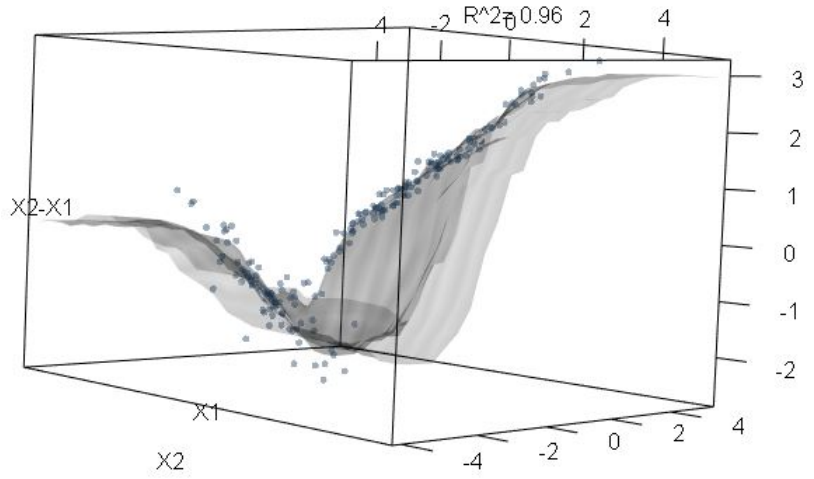



Figura 1.2: superficie de respuesta para una mezcla de normales del bosque aleatorio

En la figura anterior se observa la superficie de respuesta del algoritmo *random forest* ante una estructura de datos no lineal, mostrando su versatilidad ya que no necesita supuestos para generar predicciones y es capaz de encontrar una estructura en los datos y a partir de reglas simples, hacer una estimación puntual.

Más formalmente, si tenemos un conjunto de entrenamiento $L = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ de observaciones i.i.d de tamaño $n$ de un vector aleatorio $\{(X, Y)\}$, donde $X = (X_1, X_2, \ldots, X_p)$

[2] tendremos normalmente acceso a un conjunto de $p$ variables $X_1, X_2, \ldots, X_p$ medidas en $n$ observaciones y una respuesta $Y$ también medida en las $n$ observaciones, así que se usa el conjunto de $p$ variables para predecir $Y$ [10]

contiene $p$ variables explicativas o factores predictivos, $X \in \mathbb{R}^p$, y $Y \in \mathbb{R}$ el vector numérico de la variable dependiente o variable que nos gustaría predecir/entender dado por $f(X) = E[Y|X = x]$[10], entonces podremos estimar Y mediante $\hat{f}(X)$ como un problema clásico de regresión.

Así que los bosques aleatorios comienzan con una selección de $k$ muestras de un conjunto de entrenamiento $L$ de tamaño $n$ con reemplazamiento (*boostrap*), $\Theta = \{\Theta_1^n, \ldots, \Theta_k^n\}$, idénticamente distribuidas y tendremos aproximadamente

$$1 - \mathrm{P}\Big(1 - \Big(\frac{n-1}{n}\Big)^n\Big) \approx 0.632$$

63.2 % de observaciones únicas en cada muestra y con las cuales entrenaremos el conjunto de $k$ árboles de regresión binarios $\hat{f}(\Theta_1^n), \ldots, \hat{f}(\Theta_k^n)$ [3](Típicamente se escoge un $k$ entre 200 y 500 árboles de decisión). Cabe anotar que surgen dos diferencias respecto a los árboles de regresión clásicos en el bosque aleatorio, la primera, es que en cada nodo se elige un subconjunto de $m < p$ variables al azar y la partición se da respecto a este subconjunto (*ad hoc*, $m$ puede ser elegio como $\sqrt{p}$ o $[p/3]$ ); la segunda, es que no se realiza poda de los árboles entrenados.

Para medir el error del bosque aleatorio, Breiman [4] propuso tres medidas innovadoras en el bosque aleatorio que son la Importancia de la variable (*Variable importance or permutation variable*), el *z-score* y la importancia de GINI (*Gini importance*) [7]. Empezaremos mencionando el error *OOB*, por sus siglas (*out of bag*), que significa literalmente 'Fuera de la bolsa' y consiste en evaluar la predicción de los árboles entrenados $\hat{f}(\Theta_1^n), \ldots, \hat{f}(\Theta_k^n)$ con el conjunto de datos $\Psi$ que NO fueron tomados en el muestreo por *boostrap*, que son en promedio, el $36,7\,\%$ de las observaciones de $(Y, X)$. Lo calculamos como [7]:

$$errOOB = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{m}(\hat{f}(\Psi_{ij}) - Y_{\psi_{ij}})^2 \tag{1.2.1}$$

Donde $k$ es el número de árboles entrenados, $\hat{f}(\Psi_{ij})$ son las predicciones del i-esimo árbol en la observación j-ésima de los datos *out-of-bag* y $Y_{\psi_{ij}}$ son las observaciones del conjunto de datos evaluado correspondiente a cada $\hat{f}(\Psi_{ij})$.

### 1.2.1. Ejemplo

En Medellín - Colombia, se ha generado en los últimos meses controversia por la calidad del aire, dado que se ha notado un incremento en la cantidad de material particulado y ha habido una preocupación creciente en la ciudadanía ya que en el últimos dos años (2015 - 2016), se ha declarado emergencia naranja a causa de este fenómeno. En la ciudad existen

cinco estaciones de monitoreo del PM2.5 (*atmospheric particulate matter*) que se encuentran distribuidas en la *Casa de justicia Itaguí, Colegio Concejo Municipal de Itaguí, SOS Aburrá norte - Girardota, Tanques de la YE, Universidad Nacional de Colombia.*

El material particulado presente en la atmósfera de una ciudad (polvo, cenizas, hollín, partículas metálicas, cemento y polen, entre otras) se puede dividir, según su tamaño, en dos grupos principales. A las de diámetro aerodinámico igual o inferior a los 10 $\mu m$ o 10 micrómetros (1 $\mu m$ corresponde a la milésima parte de un milímetro) se las denomina PM10 y a la fracción respirable más pequeña, PM2.5. Estas últimas están constituidas por aquellas partículas de diámetro aerodinámico inferior o igual a los 2.5 micrómetros, es decir, son 100 veces más delgadas que un cabello humano. *disponible en* `https://blissair.com/what-is-pm-2-5.htm`.

La composición de estas partículas puede ser de diferentes fuentes. El material particulado PM2.5 es asociado a fuentes antropogénicas, es decir, creadas por el hombre, por lo que saber su densidad en el aire de una ciudad puede darnos una idea de la contaminación en dicha ciudad.

Las partículas PM2.5, tienen efectos negativos en la salud ya que se pueden acumular en el sistema respiratorio y están asociadas con el aumento de las enfermedades respiratorias y la disminución del funcionamiento pulmonar.

Los niveles según los cuales se declara uno u otro estado de la calidad del aire se presentan en la tabla 1.1

Se obtienen datos proporcionados por Siata, `https://siata.gov.co/siata_nuevo/` de registros de monitoreo para los años $2015 - 2016 - 2017(Febrero)$]. La matriz de datos consta de observaciones que incluyen fecha, estación de monitoreo y medición del material partículado en cada hora del día durante los dos años. La matriz consta de 64491 (Sin contar 2017) observaciones. Inicialmente se tomó el promedio de mediciones por hora en cada mismo día de cada mes, es decir que, por ejemplo, todos los lunes de enero fueron promediados en cada hora obteniendo así, un promedio de la medición por hora en cada mes, durante los siete días de la semana. La matriz de datos se reduce a 16115 (Sin contar el año 2017) observaciones. Se entrena el modelo de bosque aleatorio con los datos de los años $2015 - 2016$ y para realizar la validación se pretende usar los datos disponibles en el 2017 y compararlos con lo observado.

En la figura 1.1 se observa que en el modelo, las variables más importantes para explicar la cantidad de material particulado pm2.5 en Medellín son el Mes y la Hora, esto nos da un indicio a saber sobre los momentos en que la emisión de contaminantes es más alta. Pero ¿En qué momentos o meses se da esto?

En un análisis exploratorio de los datos se obtuvo que los momentos del día donde ma-

| ICA | COLOR | CLASIFICACIÓN | O3 8h ppm | O3 1h ppm | $PM_{10}$ 24h $\mu g/m^3$ | $PM_{2.5}$ 24h $\mu g/m^3$ | CO 8h ppm | $SO_2$ 24h ppm | $NO_2$ 1h ppm |
|---|---|---|---|---|---|---|---|---|---|
| 0 - 50 | Verde | Buena | 0.000<br>0.059 | - | 0<br>54 | 0<br>12 | 0<br>4.4 | 0<br>0.035 | 0<br>0.053 |
| 51 - 100 | Amarillo | Moderada | 0.060<br>0.075 | - | 55<br>154 | 12.1<br>35.4 | 4.5<br>9.4 | 0.036<br>0.075 | 0.054<br>0.100 |
| 101 - 150 | Naranja | Dañina a la salud para grupos sensibles | 0.076<br>0.095 | 0.125<br>0.164 | 155<br>254 | 35.5<br>55.4 | 9.5<br>12.4 | 0.076<br>0.185 | 0.101<br>0.360 |
| 151 - 200 | Rojo | Dañina a la salud | 0.096<br>0.115 | 0.165<br>0.204 | 255<br>354 | 55.5<br>150.4 | 12.5<br>15.4 | 0.186<br>0.304 | 0.361<br>0.649 |
| 201 - 300 | Púrpura | Muy Dañina a la salud | 0.116<br>0.374 | 0.205<br>0.404 | 355<br>424 | 150.5<br>250.4 | 15.5<br>30.4 | 0.305<br>0.604 | 0.650<br>1.249 |
| 301 - 400 | Marrón | Peligrosa | - | 0.405<br>0.504 | 425<br>504 | 250.5<br>350.4 | 30.5<br>40.4 | 0.605<br>0.804 | 1.250<br>1.649 |
| 401 - 500 | Marrón | Peligrosa | - | 0.505<br>0.604 | 505<br>604 | 350.5<br>500.4 | 40.5<br>50.4 | 0.805<br>1.004 | 1.650<br>2.049 |

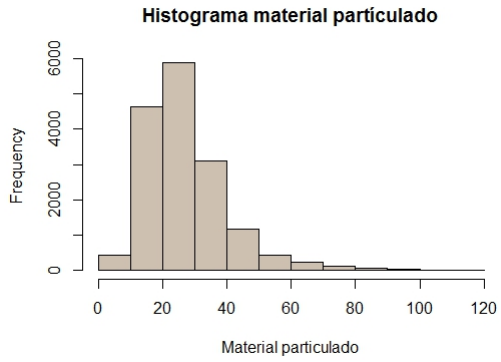
Tabla 1.1: Valores de ICA (Índice de la calidad del aire) para Medellín y el área Metropolitana `http://www.metropol.gov.co/CalidadAire/Paginas/ica.aspx`

yor es la emisión de material particulado pm2.5 son entre las 7 am y 12 m , mientras que los meses donde se reportaron mayores niveles son febrero, marzo y abril, siendo marzo el de producciones más elevadas a pesar de que la emergencia naranja fue declarada luego de obtenerse estos registros, en abril.

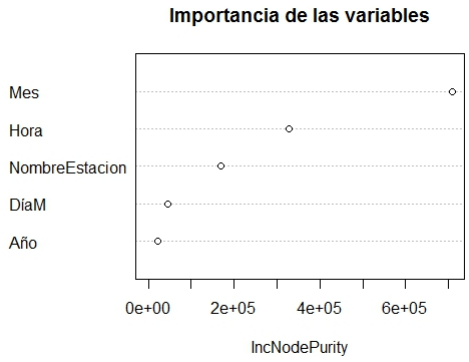
Ahora, ¿cómo se comportó el modelo a la hora de detectar estos cambios en el día y el mes? Esto es importante a la hora de hacer predicciones, pudimos observar en la figura 1.3*d* que los residuales se centran alrededor del cero aunque existen predicciones alejadas de la realidad, sobre todo en ciertos puntos donde hay una diferencia de hasta 40 puntos. Esto se debe a que hay datos atípicos debido a que las estaciones de monitoreo toman registro de la calidad del aire cada hora y puede darse el caso (muy frecuente) en que justo cuando pasaba registro había cerca un móvil con emisiones altas de material particulado (camiones, buses, tractomulas, entre otros). Así que el modelo encuentra una estructura en los datos y basa sus predicciones basado en esta estructura. La media del error en el modelo está dado por:

$$me = \frac{1}{n}\sum_{i=1}^{n}|e| = 5.67$$

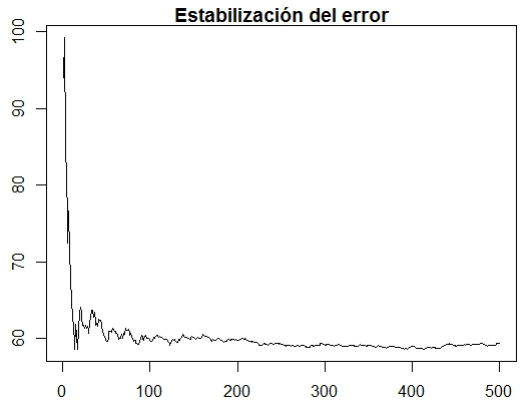
Ahora podremos realizar una validación con los datos de enero y febrero del 2017 y realizar predicciones para marzo del 2017. Así, obtenemos que En la figura 1.5 puede observarse que



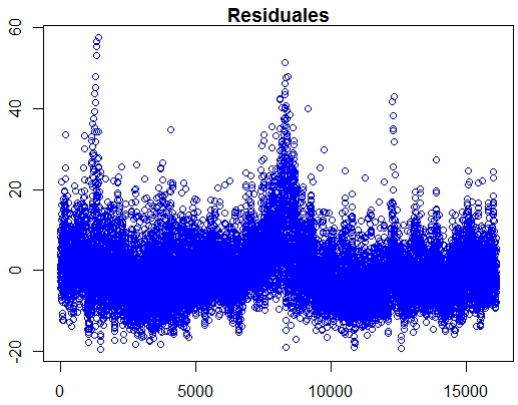

(a) Histograma pm 2.5




(b) Importancia de la variable del rf




(c) Estabilización del error



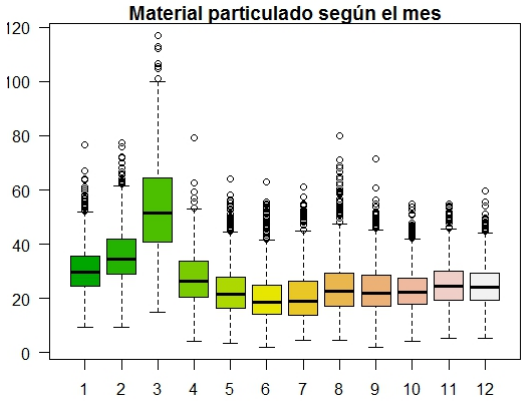

(d) Residuales del modelo

Figura 1.3: Hitograma para la variable dependiente y gráficos del modelo entrenado de *Random Forest*

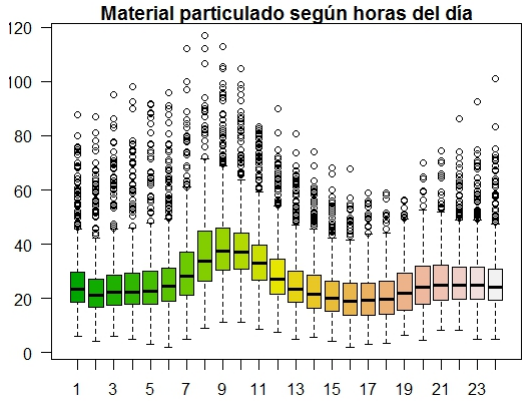
el modelo fue capaz de detectar la estructura observada en los datos, mostrando predicciones de niveles elevados en los meses de febrero, marzo (con mayores niveles) y en los momentos del día donde mayor emisión de pm2.5 hay es entre las 7 am y las 12 del medio día. A parte nos muestra la estructura en las estaciones y los días, mostrando que el domingo es el día donde menores niveles de pm2.5 se registran.

La media del valor absoluto de los residuales para los datos de enero y febrero del 2017 es de 5.21, los registros mínimos y máximos son de 2 y 17 ppm de pm2.5 por lo que una desviación media de 5.21 es buena.

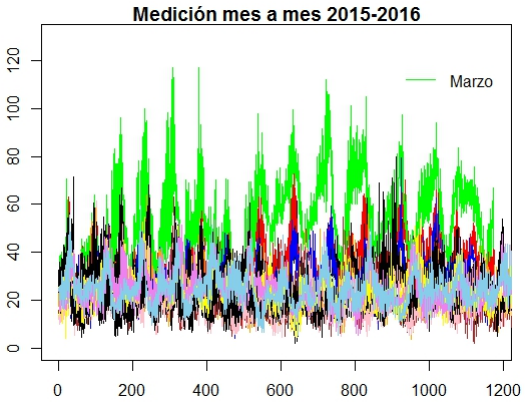
¿Cuál será el registro esperado de material particulado pm2.5 el día jueves 4 de marzo en los alrededores de la Universiad Nacional de Colombia sede Medellín si un estudiante tiene clases de 8 am a 12 del medio día? En la tabla 1.2 se observa una calidad del aire moderada para ese día.




(a) Diagramas de cajas y bigotes para la variable Mes




(b) Diagramas de cajas y bigotes para la hora




(c) Serie de tiempo mes a mes

Figura 1.4: Gráficos exploratorios de los datos, se observa que Marzo es el mes donde se encuentran mayores niveles de material particulado pm2.5 y entre las 7 am y 12 del medio día se registran mayor cantidad de pm2.5

| NombreEstacion | Hora | Día | Mes | Año | $\hat{y}_{rf}$ | Obs |
|---|---|---|---|---|---|---|
| Universidad Nacional de Colombia | 7 | 4 | 3 | 2017 | 51,065 | 43,8 |
| Universidad Nacional de Colombia | 8 | 4 | 3 | 2017 | 54,595 | 53,8 |
| Universidad Nacional de Colombia | 9 | 4 | 3 | 2017 | 55,36 | 57,4 |
| Universidad Nacional de Colombia | 10 | 4 | 3 | 2017 | 54,414 | 49,2 |
| Universidad Nacional de Colombia | 11 | 4 | 3 | 2017 | 51,11 | 30,3 |
| Universidad Nacional de Colombia | 12 | 4 | 3 | 2017 | 45,814 | 27,8 |

Tabla 1.2: Estimación de la calidad del aire de la estación Universidad Nacional de Medellín para los jueves de marzo de 2017

Cabe anotar que para marzo del 2017, se registraron valores más bajos de los niveles del material particulado 2.5, por lo que el comportamiento no es el usual al que se esperaría según los años anteriores. Entre más años se incluyan en el entrenamiento del modelo, se

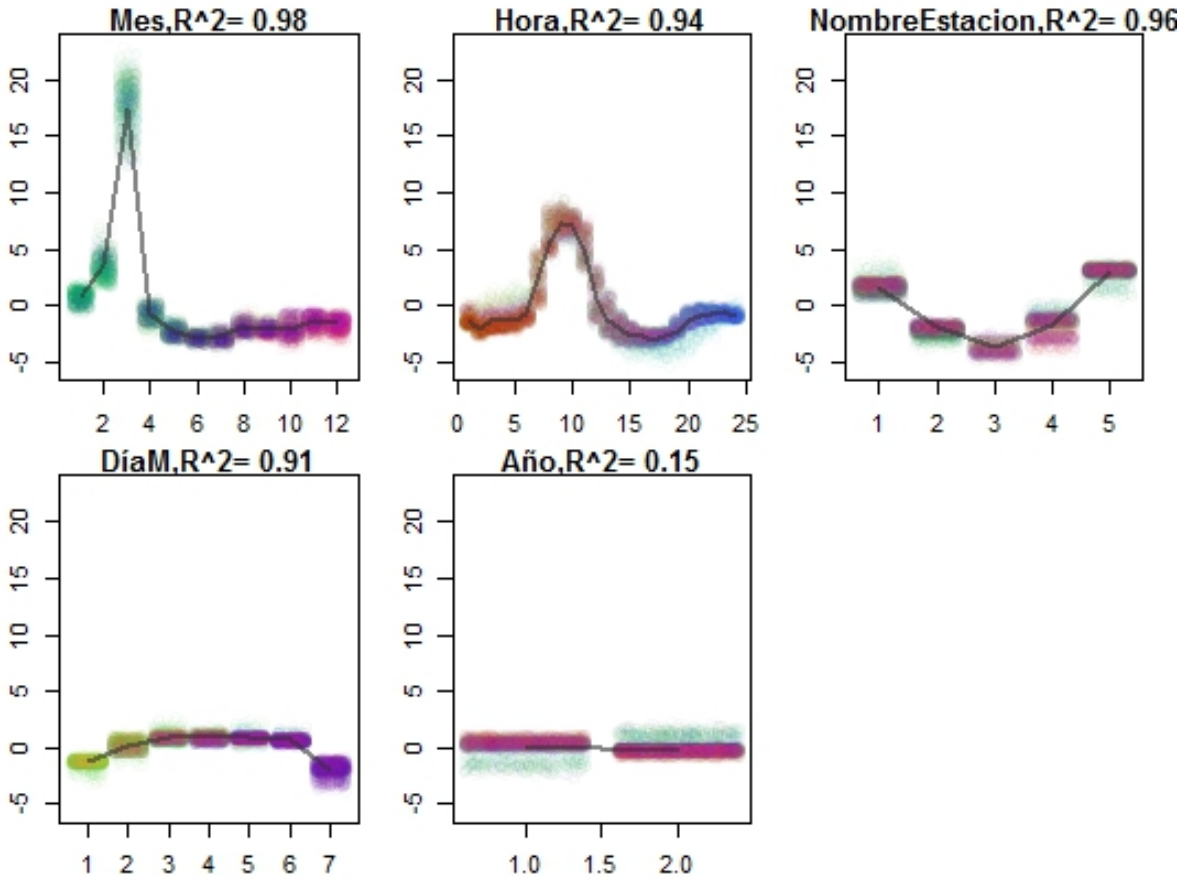
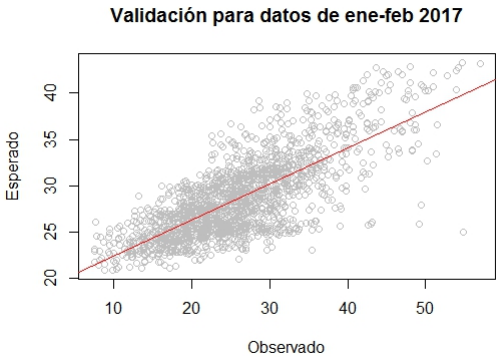
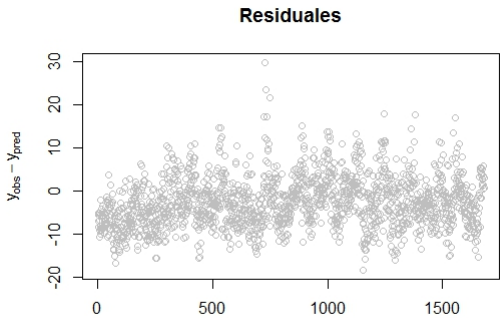



Figura 1.5: Predicción marginal del modelo con las variables en estudio con los datos fuera de la bolsa (*out in bag* - Validación).




(a) Gráfico de dispersión de los valores observados y predichos




(b) Gráfico de los residuales del modelo para los meses enero y febrero del 2017

Figura 1.6: Valores esperados y predichos del bosque aleatorio y sus residuales para los meses enero y febrero del 2017

esperaría que la predicción mejore. Esto puede dar pie a aplicaciones para la ciudadanía que estén interesados en seguir la calidad del aire en Medellín, preocupación creciente que se ha venido dando en los ciudadanos.

Ahora bien, el modelo de bosque aleatorio puede generar buenas predicciones y mostrar las variables más influyentes de un conjunto de datos siempre y cuando no exista alta correlación entre variables predictoras. No se estudiado el efecto de la predicción del bosque aleatorio cuando se da este caso. Entonces nos hacemos la pregunta. ¿Cómo la presencia de variables correlacionadas afectan el desempeño del bosque?

# Capítulo 2

# Regresión lineal múltiple

## 2.1. Método

El modelo de regresión lineal se utiliza para medir la relación entre una variable $Y$ (variable dependiente) y una o más variables predictoras/ explicativas, llamadas $X$. Los parámetros se calculan a partir de los datos y el interés se centra en la distribución de probabilidad condicional $E(Y|X)$ [13]. El término fue usado por primera vez por *Sir Francis Galton* en un estudio antropométrico [6] donde comparó la estatura de padres e hijos. Este modelo fue el primer tipo de modelos de regresión en ser estudiado rigurosamente, en esencia, por la facilidad de la estimación de sus parámetros y porque las propiedades estadísticas de los estimadores son fáciles de determinar.

Para introducir en el método, tendremos $y$ una variable aleatoria (v.a) que fluctúa alrededor de un parámetro desconocido $\eta$, esto es, $y = \eta + \epsilon$ donde $\epsilon$ es la fluctuación del error y $\eta = X\beta$ [13]. Es decir,

$$y_{nx1} = X_{nxp}\beta_{px1} + \epsilon_{nx1}$$

donde tenemos $n$ observaciones y $p$ variables predictoras

$$y = \beta_0 + X_1\beta_1 + X_2\beta_2 + \cdots + X_p\beta_p$$

El modelo de regresión lineal múltiple debe cumplir algunos supuestos.

### 2.1.1. Supuestos

- **Exogeneidad débil:** Esto significa que las variables predictoras $X$ no son variables aleatorias sino fijas, es decir, que están libres de errores. Aunque este supuesto no se da en muchos escenarios aplicados, no suponerlo conduce a modelos significativamente más difíciles.

- **Valor esperado de los error es cero:** Podemos expresarlo matemáticamente como $E(\epsilon) = 0$

- **Linealidad:** La media de la variable respuesta $y$ es una combinación lineal de los parámetros y las variables predictoras.

- **Varianza constante u homocedasticidad:** Con esto queremos decir que para diferentes valores de la respuesta $y$ el error será el mismo, en otras palabras, la varianza del error condicionado a las variable explicativas no varía en las diferentes observaciones.

- **Independencia de los errores:** Esto supone que los errores de la variable respuesta no están correlacionados entre sí.

- **No multicolinealidad:** No debe existir correlación significativa entre las variables predictoras, ya que podría traer problema en la estimación de los parámetros.

### 2.1.2. Interpretación

Utilizamos el modelo lineal para conocer la relación que existe entre uno de los predictores $X_j$ y $y$ cuando los demás predictores permanecen fijos, esto es, conocer la razón de cambio $\beta_j$ que ocurre en $y$ por cada unidad de cambio en $X_j$. Para ser más estrictos, $B_j$ es el cambio esperado en $y$ cuando $X_j$ aumenta una unidad; podríamos verlos como la esperanza de la derivada parcial de $Y$ respecto a $X_j$ ($E[\frac{\partial y}{\partial X_j}]$).

### 2.1.3. Estimación por mínimos cuadrados ordinarios

El estimador de mínimos cuadrados ordinarios es considerado el estimador óptimo del conjunto de parámetros $\beta$ mediante $\hat{\beta}$. La idea es minimizar la suma de cuadrados del error.

$$\begin{aligned} S(\beta) &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \epsilon'\epsilon \\ &= (y - X\beta)'(y - X\beta) \\ &= y'y - 2\beta' X'y + \beta' X' X\beta \end{aligned}$$

Derivamos respecto a $\beta$ evaluando en $\hat{\beta}$ e igualamos a cero para obtener finalmente la estimación de $\hat{\beta}$.

$$\begin{aligned} \frac{\partial S}{\partial \beta}\Big|_{\hat{\beta}} &= -2X'y + sX'X\hat{\beta} = 0 \\ X'X\hat{\beta} &= X'y \\ \hat{\beta} &= (X'X)^{-1}X'y \end{aligned}$$

Obteniendo finalmente la estimación de $\hat{\beta}$ por mínimos cuadrados ordinarios [12].

### 2.1.4. Validación del modelo

Luego que ajustamos un modelo de regresión a los datos, es necesario saber qué tanto los explica y cuán confiables pueden ser los pronósticos derivados del mismo. A este tipo de procedimientos se les llama validación del modelo.

#### Análisis de residuales

El tercer supuesto planteado nos dice que el valor esperado de los errores es cero, por lo tanto, es importante estudiar los residuos para examinar en qué medida la suposición tres puede ser violada. Esto permitirá reconocer patrones en los residuales que podrían aumentar la comprensión del modelo de regresión y eventualmente mejorarlo. Si tenemos el modelo

$$y = \beta_o + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

Entonces los residuales son de la forma:

$$\hat{\epsilon} = y - (\beta_o + \beta_1 X_1 + \cdots + \beta_p X_p)$$

Podemos realizar un análisis gráfico, ya sea un histograma para ver que los tengan la forma de una normal concentrada alrededor del cero, realizar pruebas de normalidad (Lilliefor's, K-S) o realizar el qqplot. Otra forma es graficar los residuales en $y$ y los valores predichos en $x$ y observar que no haya ningún patrón o tendencia.

#### $R^2$ y $R^2$ ajustado

El coeficiente de determinación o $R^2$ sirve como medida de bondad de ajuste del modelo, ya que indica el porcentaje de variabilidad de la variable dependiente $y$ explicado por la variabilidad conjunta de las variables predictoras. Está dado como:

$$R^2 = \frac{var(x\hat{\beta})}{var(y)}$$

La varianza de $y$ se puede descomponer en $var(y) = var(x\hat{\beta}) + var(\hat{\epsilon})$, así que también podríamos expresar el coeficiente de determinación como:

$$R^2 = 1 - \frac{var(\hat{\epsilon})}{var(y)}$$

El $R^2$ presenta un problema cuando las variables predictoras aumentan, ya que no puede decaer y por tanto será alto a pesar de que las variables no sean significativas [15], para remediar esto, se ha propuesto una alternativa llamado $R^2$ ajustado o también denotado $\bar{R}^2$, está dado por:

$$\bar{R}^2 = 1 - \frac{N-1}{N-k-1}(1-R^2)$$

Que lo que hace esencialmente es 'castigar' el modelo por el número de variables predictoras.

**Estadístico F**

Como vimos anteriormente, se puede escribir la variabilidad de y en términos de la variabilidad de $x\hat{\beta}$ mas la variabilidad del error. En otras palabras, $SST = SSR + SSE$, es decir, la variabilidad total (SST) es igual a la variabilidad de los regresores (SSR) mas la variabilidad del error (SSE) [12].

$$SST = y'y - \Big(\frac{1}{n}\Big)y'Jy = y'\Big[I - \Big(\frac{1}{n}\Big)\Big]y$$

$$SSE = \epsilon'\epsilon = (y - x\hat{\beta})'(y - x\hat{\beta}) = y'y - \hat{\beta}'x'y = y'(1-H)y$$

$$SSR = \hat{\beta}'x'y - \Big(\frac{1}{n}\Big)y'Jy = y'\Big[H - \Big(\frac{1}{n}\Big)J\Big]y$$

Donde $H_{nxn} = x(x'x)^{-1}x'$ y $J_{kxk} = 1_{kxk}$. Los grados de libertad asociados a $SST$ son $n-1$, a $SSR$ son $k-1$, siendo $k$ el número de parámetros y al error. $SSE$ son de $n-k$. Por último, para obtener la media cuadrada del error dividimos la suma cuadrada de errores entre sus grados de libertad.

El estadístico F se construye a partir de la media cuadrada de los errores como sigue:

$$F = \frac{\frac{SSR}{k-1}}{\frac{SSE}{n-k}} = \frac{MSR}{MSE}$$

Y lo contrastamos con una $F_{1-\alpha/2,k-1,n-k}$ y así, teniendo en cuenta esta comparación, tendríamos que si $P\text{-}valor < 0.05$ deducimos que al menos una de las variables es significativa.

**Estadístico t**

Con el estadístico F obtenemos la idea de si alguna de las variables tiene una relación significativa con la variable dependiente, por otra parte, el estadístico t nos indica cuáles variables son significativas; esto lo sabremos planteando una hipótesis nula que habla de la poca relación entre la variable explicativa y la dependiente. Así, realizamos el test sobre los coeficientes de la regresión, esto es, sobre los $\beta_j$, como sigue: $H_o : \beta_j = 0$, $Ha : \beta_j \neq 0$.

La hipótesis anterior la sometemos a prueba con el siguiente estadístico:

$$t = \frac{\beta_j}{SD(\beta_j)}$$

Donde $SD(\beta_j)$ es el error estándar de $\beta_j$. Si *p-valor*$<$ 0.05 concluimos que dada la evidencia, es poco probable que el coeficiente $\beta_j$ sea igual a cero.

# Capítulo 3

# Experimentos

En artículos recientes, Gregorutti et al. [9] [8] ha desarrollado un método que muestra cómo se afecta la importancia de las variables en el bosque aleatorio cuando existe correlación entre las variables predictoras, presentando mediante casos de correlación el comportamiento del error fuera de la bolsa. Se encontró que la importancia de la variable $X_j$ disminuye en razón de

$$I(X_j) = 2\Big(\frac{\tau_0}{1 - c + pc}\Big)^2$$

Donde $\tau_0$ es la correlación de la variable $X_j$ con $y$, $c$ es la correlación de $X_j$ con $X_i$ y $p$ el número de variables.

El objetivo de este trabajo es estudiar cómo se afecta el desempeño de la predicción del bosque aleatorio en presencia de correlación de variables, prescindiendo de las medidas remediarias propuestas en Gregorutti [9]. Para esto, mediante datos sintéticos, realizaremos un entrenamiento de dos modelos, bosque aleatorio y modelo lineal general, comparando sus errores cuadráticos medios.

La regresión lineal múltiple es el mejor estimador para un grupo de datos normales, así que será un referente para el estimador obtenido en el bosque aleatorio.

## 3.1. Simulación

Los experimentos consisten en realizar simulaciones de una población normal multivariada de la cual se extraerá una muestra que será dividida en un conjunto de entrenamiento (*training set:* 80 % del total de la muestra) y un conjunto de validación (*test set:* 20 %), el modelo de bosque aleatorio y regresión lineal múltiple serán ajustados a los datos de entrenamiento de la muestra y luego se evaluará la razón entre sus errores cuadrados medios.

**Caso 1**

En el caso uno queremos ver cómo es el desempeño del bosque cuando las variables explicativas son importantes en cuanto a su correlación con la variable $Y$, pero no están correlacionadas entre sí.

**Regresión lineal:** Este es el simple contexto donde $(X_1, X_2, Y) \sim N_3(0, \Sigma)$ con $\Sigma = \begin{pmatrix} C & \tau \\ \tau^t & 1 \end{pmatrix}$

y $C = \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix}$ La matriz de varianzas covarianzas de $X_1$ *y* $X_2$, y $\tau = (\tau_0, \tau_0)$ que sería:

$$\begin{pmatrix} 1 & c & \tau_0 \\ c & 1 & \tau_0 \\ \tau_0 & \tau_0 & 1 \end{pmatrix}$$

Obtenemos que

$$\Sigma = \begin{pmatrix} 1 & 0.1 & 0.95 \\ 0.1 & 1 & 0.9 \\ 0.95 & 0.9 & 1 \end{pmatrix}$$

El modelo lineal obtenido (ver apéndice A) será nuestro referente frente al desempeño del bosque, usaremos un gráfico de puntos donde compararemos los valores observados *vs* estimados. Esto nos dará una idea de qué tan bien se acerco el modelo a la estructura de datos. El segundo gráfico, cuando es posible (mdos menos variables explicativas), es el gráfico de la superficie de respuesta del modelo, el modelo de regresión lineal estima un plano mientras que el bosque aleatorio una superficie de respuesta. Finalmente, compararemos el error cuadrado medio de cada modelo y compararemos los dos mediante una razón de errores. Esto nos hablará del desempeño del bosque respecto al modelo lineal.

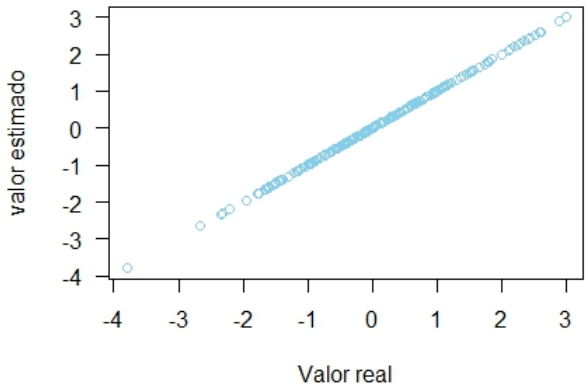
El método de comparación que usaremos en cada caso es la suma cuadrada de errores media, o $MSE$, expresada como

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{est} - y_{obs})^2$$

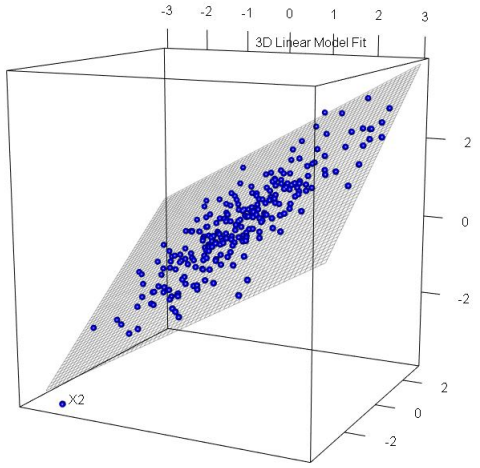
. El modelo lineal utiliza como base el estimador de mínimos cuadrados, siendo este, el mejor estimador insesgado de mínima varianza, por lo que la razón entre los errores $\frac{MSE_{rf}}{MSE_{lm}}$ sera un indicativo del desempeño del bosque.

El error cuadrático medio del modelo lineal para este caso es de: $MSE = 0.0000000411$

Para el algoritmo de aprendizaje *random forest* realizamos un entrenamiento del modelo con el *data ser* de entrenamiento, utilizaremos un $n = 800$ árboles y los demás parámetros vienen predeterminados ( Ver capítulo 2):

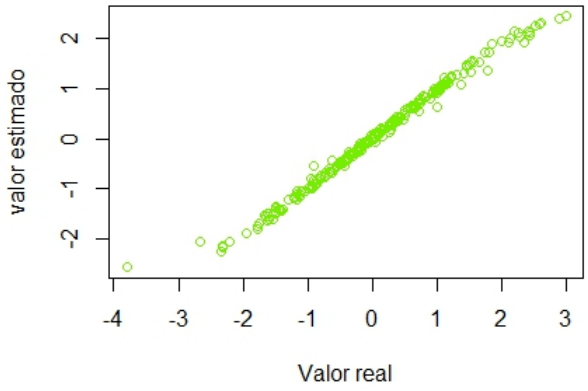



(a) Observado *vs* predicho

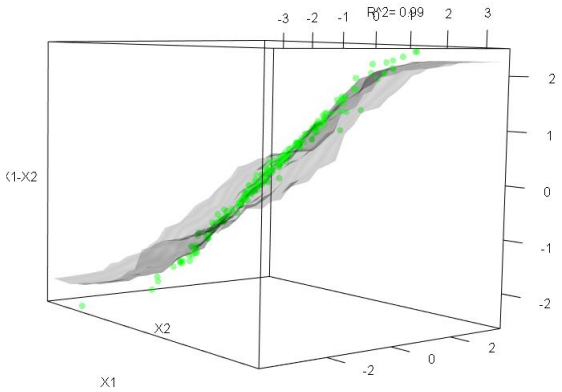



(b) Superficie de respuesta

Figura 3.1: Observado *vs* predicho y superficie de respuesta para el modelo de regresión lineal en el caso de dos variables explicativas incorrelacionadas.




(a) Observado *vs* predicho




(b) Superficie de respuesta

Figura 3.2: Observado *vs* predicho y superficie de respuesta para el modelo de *Random forest* en el caso de dos variables incorrelacionadas

Para este caso, los errores de ambos modelos son bajos, aun así, el modelo lineal muestra un ajuste perfecto. Esto se debe a la alta correlación de las variables predictoras respecto a la variable $Y$. El bosque presentó un $MSE = 0.0187$ y la razón entre ambos es de

$$\frac{MSE_{rf}}{MSE_{lm}} = \frac{0.0000000411}{0.0187} = 456050.3$$

**Caso 2**

**Regresión lineal:**En este caso estudiaremos la correlación de dos variables. El simple contexto donde $(X_1, X_2, Y) \sim N_3(0, \Sigma)$ con $\Sigma = \begin{pmatrix} C & \tau \\ \tau^t & 1 \end{pmatrix}$

y $C= \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix}$ La matriz de varianzas covarianzas de $X_1$ *y* $X_2$, y $\tau = (\tau_0, \tau_0)$ que sería:

$$\begin{pmatrix} 1 & c & \tau_0 \\ c & 1 & \tau_0 \\ \tau_0 & \tau_0 & 1 \end{pmatrix}$$

Obtenemos que

$$\Sigma = \begin{pmatrix} 1 & 0.9 & 0.95 \\ 0.9 & 1 & 0.9 \\ 0.95 & 0.95 & 1 \end{pmatrix}$$

en este caso encontramos dos variables altamente relacionadas con la variable dependiente (0.95) -ver Gregorutti [9], caso 1- y además, la correlación entre ellas es significativa (0.9).

El modelo lineal, expresado de la forma $y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2$ para el conjunto de entrenamiento (ver apéndice) es

$$y = -0.008654 + 0.497997 X_1 + 0.507555 X_2$$

. Ahora predecimos los valores de $y$ reemplazando cada valor de $X_1$ y $X_2$ en la ecuación y contrastamos contra los valores reales:

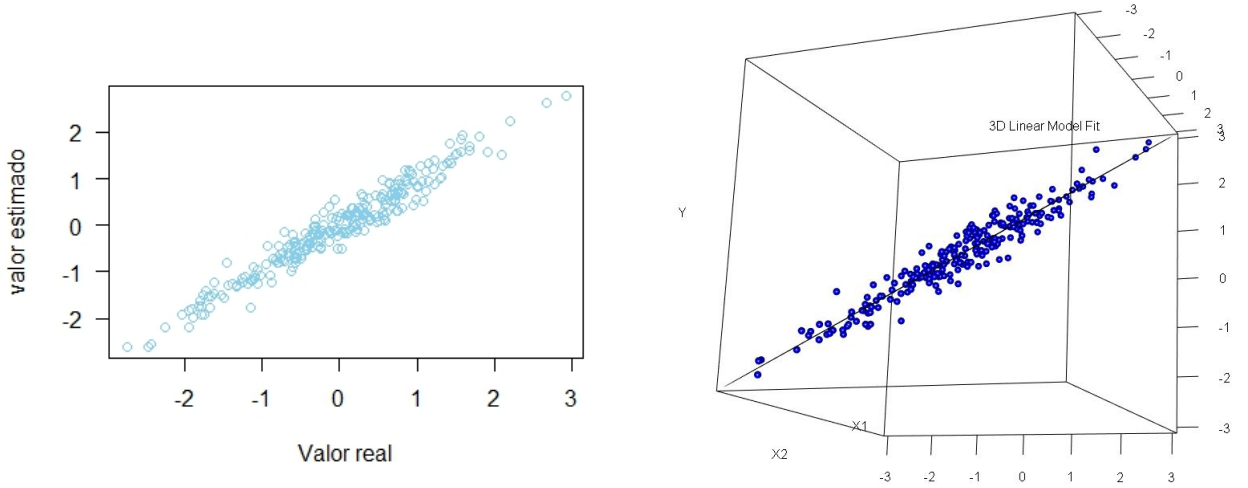
**Bosque aleatorio:** Entrenaremos el bosque con los datos obtenidos anteriormente (datos de entrenamiento). El número de árboles a entrenar será de 800, usaremos el algortimo predeterminado por *L Breiman*[4] (Ver apéndice A) implementado en el paquete *RandomForest* de R.

Así, para el caso dos: $\frac{MSE_{rf}}{MSE_{lm}} = 1.23565$

**Caso 3**

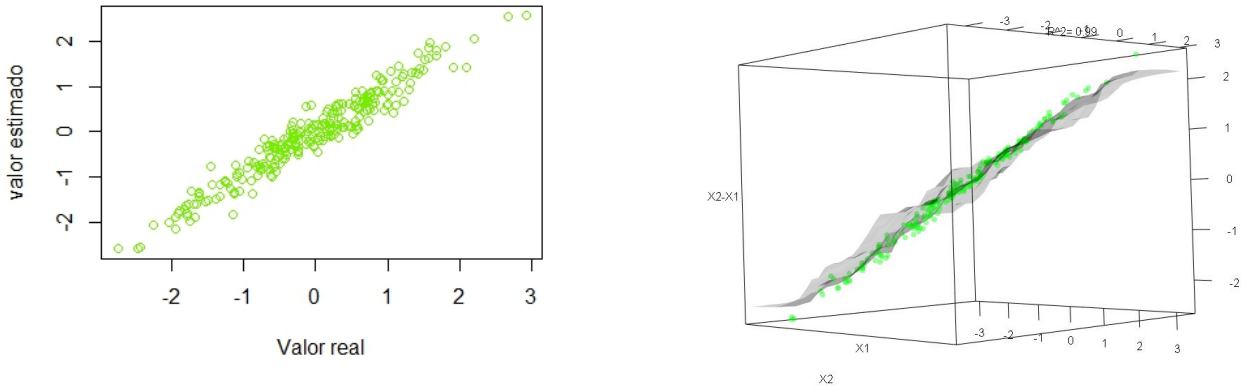
En este caso tenemos dos variables correlacionadas y una variable independiente. Adicionamos al caso anterior una variable adicional $X_3$ que será independiente de $X_1$ y $X_2$

$$C= \begin{pmatrix} 1 & c & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$




(a) Observado *vs* predicho (b) Superficie de respuesta

Figura 3.3: Errores y superficie de respuesta para el modelo de regresión lineal múltiple para el caso de dos variables altamente correlacionadas y a su vez correlacionadas con la variable dependiente



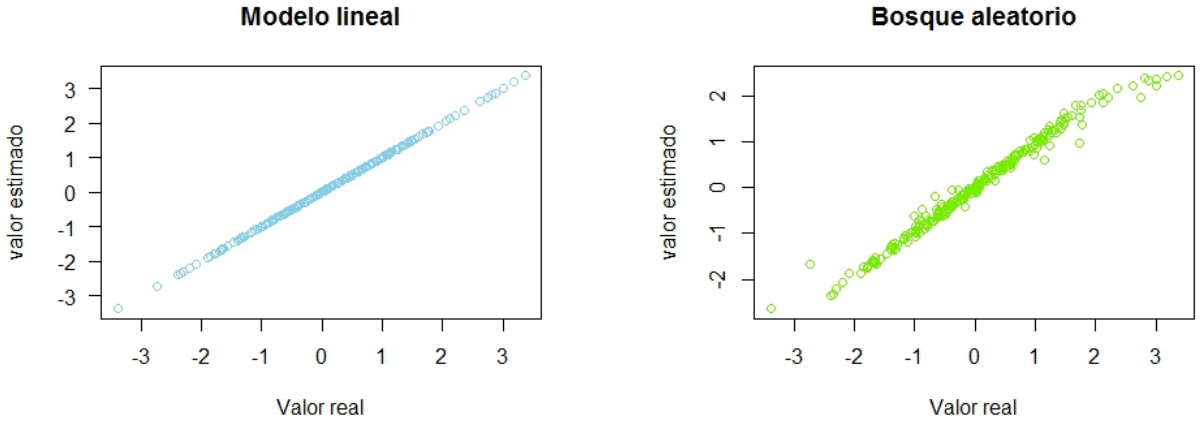

(a) Observado *vs* predicho (b) Superficie de respuesta

Figura 3.4: Errores y superficie de respuesta para el modelo de *Random forest* en el escenario de dos variables predictoras y una variable dependiente, altamente correlacionadas (caso 1-Gregorutti [9])

y $\tau^t = (\tau_0, \tau_0, \tau_3)$. Podemos verificar facilmente que $C^{-1}(\tau_0, \tau_0, \tau_3)^t = (\frac{\tau_0}{1+c}, \frac{\tau_0}{1+c}, \tau_3)^t$

Simulando el escenario anterior, suponiendo que $\tau_0 > \tau_3$, observamos:

El error cuadrado medio del modelo lineal es de $MSE_{lm} = 0.000000069488$, mientras que




(a) Observado *vs* predicho

(b) Superficie de respuesta

Figura 3.5: Observado *vs* predicho para ambos modelos en el caso de dos variables altamente correlacionadas y una tercera variable no correlacionada con $X_1$ y $X_2$ pero si con $Y$ (caso 2-Gregorutti [9])

el del bosque es de $MSE_{rf} = 0.06099569$ y la razón entre ambos es de

$$\frac{MSE_{rf}}{MSE_{lm}} = \frac{0.00000046891}{0.0339183} = 877777.8$$

**Caso 4**

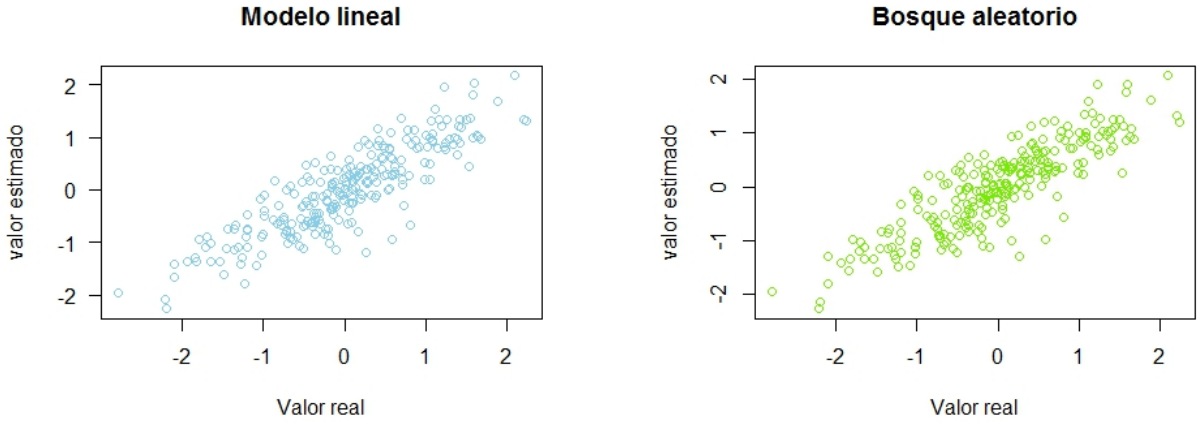
En este caso consideraremos $p$ variables correlacionadas, elegiremos $p = 20$, donde tendremos una correlación alta entre las 20 variables y a su vez estarán correlacionadas con la variable dependiente.

En este caso, el modelo lineal muestra un rendimiento menor al bosque aleatorio respecto a sus $MSE$, esto debido a la multicolinealidad entre las variables predictoras. El modelo lineal presenta un $MSE_{lm} = 0.2288982$ mientras que el erro cuadrático medio del bosque aleatorio es de $MSE_{rf} = 0.12584$ y su razón es de

$$\frac{MSE_{rf}}{MSE_{lm}} = \frac{0.12584}{0.2288982} = 0.5497824$$

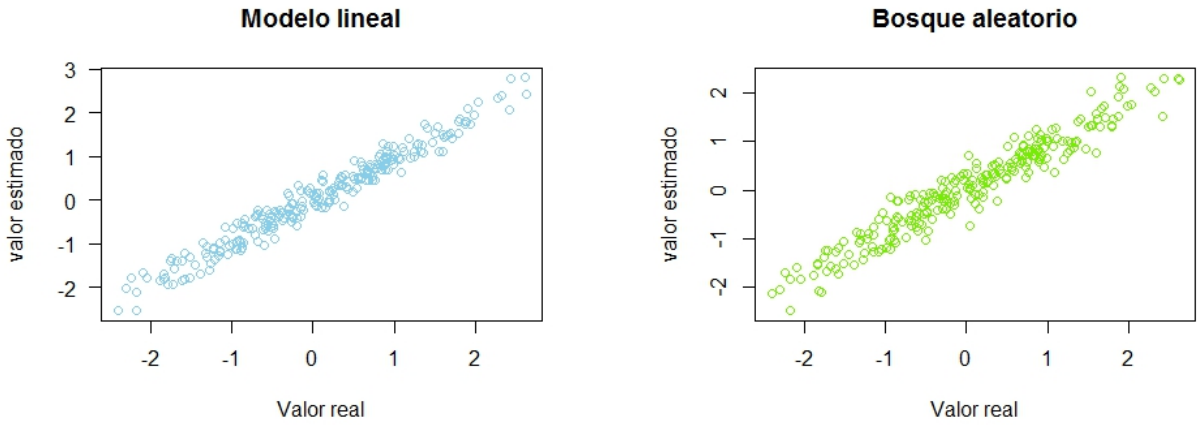
**Caso 5**

En este caso consideraremos un grupo $p$ de variables correlacionadas y un grupo $q$ de variables incorrelacionadas. En este caso $p = 15$ y $q = 5$ y evaluaremos el desempeño del bosque aleatorio (Ver apéndice A).




(a) Observado *vs* predicho (b) Superficie de respuesta

Figura 3.6: Observado *vs* predicho para ambos modelos en el caso de veinte variables altamente correlacionadas (caso 3- Gregorutti [9])




(a) Observado *vs* predicho (b) Superficie de respuesta

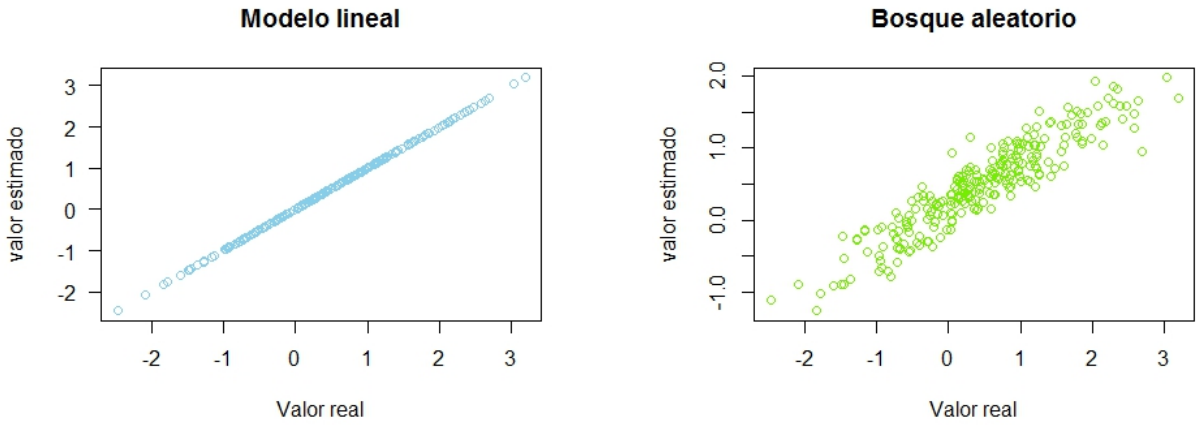
Figura 3.7: Observado *vs* predicho para ambos modelos en el caso de quince variables altamente correlacionadas y cinco variables sin correlación significativa con las demas pero sí con la variable dependiente (caso 4- Gregorutti [9])

El bosque aleatorio obtuvo una media cuadrática del error de $MSE_{rf} = 0.123464$ y el modelo de regresión lineal de $MSE_{lm} = 0.04554$ y su razón es de

$$\frac{MSE_{rf}}{MSE_{lm}} = \frac{0.123464}{0.04554} = 2.710874$$

### Caso 6

En el caso seis realizamos un análisis del rendimiento del bosque con 40 variables incluidas, seleccionadas al azar de un conjunto de distribuciones normales con media uniforme entre $-2$ y 2 y una varianza de cada distribución distribuida uniforme entre 1 y 10, la correlación de cada variable con $Y$ también es una variable aleatoria uniforme entre 0.2 y 0.99, esto con el objetivo de testear el rendimiento del bosque cuando aproximadamente el 62 % de las variables guardan una correlación con la variable dependiente $\rho \geq 0.5$




(a) Observado *vs* predicho (b) Superficie de respuesta

Figura 3.8: Observado *vs* predicho para ambos modelos en el caso de 40 variables donde aproximadamente el 60 % tiene una correlación $\rho \geq 0.5$

En este caso, el bosque aleatorio difiere considerablemente del modelo lineal (que presenta un ajuste casi perfecto a los datos), evaluando los errores obtenemos que los MSE para el bosque aleatorio es de $MSE_{rf} = 0.2390$ mientras que para el modelo lineal es $MSE_{lm} = 0.000000127$, así, su razón es

$$\frac{MSE_{rf}}{MSE_{lm}} = \frac{0.2390}{0.000000127} = 1881772$$

El buen rendimiento del modelo de regresión lineal múltiple se debe a que aumenta su eficacia cuando el número de variables explicativas crece y el tamaño de muestra es grande.

### Casos 7, 8, 9 y 10

La multicolinealidad afecta el desempeño del modelo lineal. Para evaluar cómo evoluciona el desempeño del bosque aleatorio en un conjunto de variables donde hay una alta correlación entre ellas, proponemos cuatro casos además de los tratados por Gregorutti. En el caso siete tenemos cuatro variables explicativas que presentan multicolinealidad entre ellas, en el caso

ocho mostramos el caso en que existen seis variables fuertemente correlacionadas entre sí. En el caso nueve y diez, se muestra el caso donde se tienen ocho y diez variables fuertemente correlacionadas entre sí, respectivamente. Se puede evidenciar que a partir del caso siete, los demás casos muestran un mejor desempeño del bosque aleatorio. Evidenciando que la multicolinealidad no afecta en gran medida su rendimiento en comparación con el modelo lineal. (Ver tabla 3.2)

## 3.2. Tabla resultados

Para resumir las simulaciones anteriores, resumiremos todos los casos presentados para tener una mejor idea global del problema, es decir, qué costo se paga a la hora de realizar un análisis que tiene multicolinealidad entre sus variables predictoras. Solo hubo un caso donde el bosque fue mejor, pero hubo alta multicolinealidad entre las variables por lo que la potencia del modelo de regresión disminuye.

| **Error/Casos** | **Caso 1** | **Caso 2** | **Caso 3** | **Caso 4** | **Caso 5** | **Caso 6** |
|---|---|---|---|---|---|---|
| $MSE_{rf}$ | 0.0187 | 0.06375425 | 0.0339183 | 0.12584 | 0.123464 | 0.2390 |
| $MSE_{lm}$ | 0.0000000411 | 0.051595 | 0.0000004689 | 0.228898 | 0.04554 | 0.0000000127 |
| $\frac{MSE_{rf}}{MSE_{lm}}$ | 456050.6 | 1.235649 | 877777.8 | 0.54978 | 2.7108 | 1881772 |

Tabla 3.1: Resumen de casos de simulación de los modelos regresión lineal múltiple y el algoritmo de aprendizaje *random forest* bajo algunos escenarios de datos normales

| **Error/Casos** | **Caso 7** | **Caso 8** | **Caso 9** | **Caso 10** |
|---|---|---|---|---|
| $MSE_{rf}$ | 0.1523 | 0.2387 | 0.1819 | 0.1712 |
| $MSE_{lm}$ | 0.1466 | 0.4828 | 0.3814 | 0.4141 |
| $\frac{MSE_{rf}}{MSE_{lm}}$ | 1.039071 | 0.4944 | 0.4770 | 0.4134 |

Tabla 3.2: Resumen de casos propuestos de simulación para estudiar el comportamiento de la predicción del bosque aleatorio *vs* la predicción del modelo lineal cuando la multicolinealidad afecta su desempeño

# Capítulo 4

# Conclusiones

Los experimentos han mostrado que bajo escenarios de correlación el desempeño del bosque es cercano al desempeño del modelo lineal(caso 2, 5, 7) o mejor que él (caso 4,8,9 y 10) debido a la inflación del error que sufre el modelo lineal en presencia de colinealidad entre dos o más variables explicativas. En el caso en que agregamos variables incorrelacionadas entre sí pero con alta correlación con la variable dependiente, observamos una mejora sustancial del modelo lineal generalizado (caso 1, 3, 6).
Comparando el bosque con el modelo lineal se ha podido observar que no muestra un mal desempeño a la hora de evaluar variables correlacionadas entre sí, atributo importante para tener en cuenta en futuros estudios y análisis. A pesar de que Gregorutti et al. [9] demostraron que la importancia de las variables decrece en la medida en que se correlacionan con otras, esto no afecta su poder predictivo y de hecho lo mantiene mientras que otros modelos clásicos como el lineal son afectados por la forma de calcular sus parámetros y también por la inflación de varianza que sufre por la multicolinealidad.

Es importante resaltar que el método de aprendizaje de máquinas *Random Forest* toma mayor relevancia cuando se comprende que en casos de la vida real, los datos no suelen presentar una estructura de normalidad, necesaria para ajustarlos al modelo lineal, es por tanto de interés comparar el desempeño del bosque aleatorio respecto a otros modelos estadísticos que se ajustan a otra estructura de datos. Así, un trabajo a futuro podría ser comparar el desempeño del bosque aleatorio *vs* modelos lineales generalizados ($GLM$) o modelos no lineales ($NLS$) y verificar si su desempeño es bueno y puede superarlos en algunos casos para la predicción.
Esto ayuda a entender el por qué algoritmos de aprendizaje de máquinas que son más cercanos a la ciencia de la computación han tenido buena acogida en la comunidad estadística. Cabe resaltar que estos algoritmos son usados en problemas de predicción donde no se tiene interés en entender un conjunto de datos. Actualmente, por la gran cantidad de datos que generan algunas empresas y por el corto tiempo de sus proyectos, se hace fundamental obtener predicciones precisas más que entender qué pasa con las variables (enfoque más es-

tadístico), es por esto último, que resulta importante conocer el comportamiento de estos algoritmos para comprender su alcance y hasta qué punto su utilización puede mejorar la toma de decisiones rápida y segura.

# Apéndice A

# Código

## A.1. Capítulo 1

```
######Ejemplo sobre árboles de regresión#########
####Base de datos calidad del vino####
#Datos tomados de https://github.com/stedy/Machine-Learning-with-R-datasets
vinos=read.csv("whitewines.csv",header=T)
set.seed(1)
##Entrenamiento
sampling=sample(1:nrow(vinos),round(0.75*nrow(vinos)))
#Conjunto entrenamiento
entren= data.frame(vinos[sampling,])
#Conjunto validación
valid=data.frame(vinos[-sampling,])
##Entrenamos el árbol de regresión##
library(rpart)
library(rpart.plot)
modelo= rpart(quality~.,data=vinos)
rpart.plot(modelo, digits = 3)
##Validación del modelo
Predichos=predict(modelo, valid)
#Erorres e= y - ypred
Errores= valid$quality- Predichos
##Media de errores cuadráticos
mean(Errores^2)

################################EJEMPLO MEZCLA DE NORMALES#########################
#https://artax.karlin.mff.cuni.cz/r-help/library/mixAK/html/MVNmixture.html
library(mixAK)
```

```
library(mvtnorm)
library(randomForest)
library(nortest)
library(forestFloor)
library(MASS)
library(scatterplot3d)
library(visreg)
library(car)
library(rgl)
### Choice
set.seed(1)
N=1000

mu <- matrix(c(-1, -1,-1, 1, 1, 1), nrow = 2, byrow = TRUE)

sigma=list()

sigman=matrix(c(1, -0.9,-0.95,-0.9,1,-0.95,-0.95,-0.95,1),ncol=3,nrow=3)
sigman=Matrix::nearPD(sigman)
for (i in 1:15){
sigman=Matrix::nearPD(sigman$mat)
diag(sigman$mat)=1
sigman=Matrix::nearPD(sigman$mat)
}

sigma[[1]] <- matrix(sigman$mat,ncol=3,nrow=3)
sigma[[2]] <- matrix(c(1, 0.9,0.95,0.9,1,0.95,0.95,0.95,1),ncol=3,nrow=3)

pesos=c(0.4,0.6)

Muestra=rMVNmixture2(n=N, weight=pesos, mean=mu, Sigma=sigma)

samplemix=data.frame(Muestra$x)
cor(samplemix)

##Entrenamiento
sampling=sample(1:nrow(samplemix),round(0.75*nrow(samplemix)))
#Conjunto entrenamiento
entren= data.frame(samplemix[sampling,])
#Conjunto validación
```

```
valid=data.frame(samplemix[-sampling,])
#################### Random Forest#######################

Bosque1=randomForest(X3~X1+X2, data=entren,ntree=800,importance=T,keep.inbag = TRUE)
plot(Bosque1)

X3_rf=predict(Bosque1,valid)
plot(valid[,3],X3_rf,xlab="Valor real", ylab="valor estimado",
main="Estimación RF")


#plot(samplemix[,1],samplemix[,2])
mean((valid[,3]-X3_rf)^2)
#
plot(residuos_rf, xlab="", ylab= "Residuos random fores", main="Modelo normal",
col="blue")

ff=forestFloor(Bosque1, X =valid[,1:2])
show3d(ff,plot.rgl=list(size=5))

################ANÁLISIS DATOS CALIDAD DEL AIRE################


############################EJEMPLO################################
###Datos de la calidad del aire Medellín y área metropolitana####
bd=read.csv2("datos.csv")
bd=bd[,c(-1)]
bd$Hora=as.factor(bd$Hora)
bd$Día=as.factor(bd$Día)
bd$Mes=as.factor(bd$Mes)
bd$Año=as.factor(bd$Año)
bd$DíaMejora=as.factor(bd$DíaMejora)
#Base inicial
bd0=data.frame(pm25=bd$pm25,NombreEstacion=bd$NombreEstacion,
Hora=bd$Hora,DíaM=bd$DíaMejora,Mes=bd$Mes,Año=bd$Año)
#Entrenamiento
bd1=bd0[which(bd0$Año != "2017"),]

bd2=bd0[which(bd0$Año == "2017" & (bd0$Mes== "1" | bd0$Mes =="2")),]
bd3=bd0[which(bd0$Año == "2017" & bd0$Mes== "3" ),]
```

```
#Extraemos promedio de días por mes en cada hora
DatosF=aggregate( pm25~., bd1, mean )
DatosF2=aggregate( pm25~., bd2, mean )

###Exploración de las variables
#boxplots
#Meses
boxplot(DatosF$pm25~DatosF$Mes, col=terrain.colors(12),las=1,
main="Material particulado según el mes",xlab="Mes",
ylab="Cantidad material part.")
#Hora
boxplot(DatosF$pm25~DatosF$Hora, col=terrain.colors(24), las=1,
main="Material particulado según horas del día",
xlab="Hora",ylab="Cantidad material part.")
##Gráfico serie de tiempo
ts.plot(DatosF$pm25[DatosF$Mes==1],ylim=c(0,130),ylab="Medición material part.",
main="Medición mes a mes 2015-2016")
lines(DatosF$pm25[DatosF$Mes==2],col="red")
lines(DatosF$pm25[DatosF$Mes==3],col="green")
lines(DatosF$pm25[DatosF$Mes==4],col="blue")
lines(DatosF$pm25[DatosF$Mes==5],col="orange")
lines(DatosF$pm25[DatosF$Mes==6],col="brown")
lines(DatosF$pm25[DatosF$Mes==7],col="pink")
lines(DatosF$pm25[DatosF$Mes==8],col="gray1")
lines(DatosF$pm25[DatosF$Mes==9],col="black")
lines(DatosF$pm25[DatosF$Mes==10],col="yellow")
lines(DatosF$pm25[DatosF$Mes==11],col="violet")
lines(DatosF$pm25[DatosF$Mes==12],col="skyblue")
legend(x=900,y=120, legend=c("Marzo"),col="green",lwd=1,bty="n")

set.seed(1)
#Histograma
hist(DatosF$pm25,main="Histograma material partículado",
col="antiquewhite3", xlab="Material particulado")

#Entrenamos el bosque aleatorio
library(randomForest)
library(forestFloor)
#Entrenamiento el bosque aleatorio
Modelrf=randomForest(pm25~.,data=DatosF,keep.inbag=TRUE)
```

```
#Importancia de la variable
varImpPlot(Modelrf,main="Importancia de las variables")

##Estabilización del error
par(mfrow=c(1,1))
plot(Modelrf,main="Estabilización del error")
#Medición del error
Predichos=predict(Modelrf,DatosF)
#Errores
Errores=DatosF$pm25-Predichos
#
plot(Errores, main="Residuales", col="blue")
##Media de errores
mean(abs(Errores))

##OUT INBAG ERROR
ff = forestFloor(Modelrf,DatosF)
#ggPlotForestFloor(ff,1:9)
plot(ff,col=fcol(ff))

##########VALIDACIÓN DATOS DEL 2017###############
dat2017=predict(Modelrf,DatosF2[,-6])

###
plot(DatosF2$pm25,dat2017, xlab="Observado", ylab="Esperado",
main="Validación para datos de ene-feb 2017",col="gray")
abline(lm(dat2017~DatosF2$pm25),col="red")
#Errores
Errdat2017=DatosF2$pm25-dat2017
##
plot(Errdat2017, ylab=expression(y[obs]- y[pred]),xlab="",main="Residuales", col="gray")

max(DatosF$pm25)
##Predicción marzo

pred=read.csv2("pred.csv")
pred$Hora=as.factor(pred$Hora)
pred$DíaM=as.factor(pred$DíaM)
pred$Mes=as.factor(pred$Mes)
```

```
pred$Año=as.factor(pred$Año)
predi=rbind(DatosF2[,-6],pred)
newpred=predict(Modelrf,predi)
newpred[1681:1686]
```

## A.2. Capítulo 3

```
####Simulaciónes########

library(knitr)
library(mvtnorm)
library(randomForest)
library(nortest)
library(forestFloor)
library(MASS)
library(scatterplot3d)
library(visreg)
library(car)
library(rgl)

#########################################################################
###################################CASO 1################################
#########################################################################

##Variables independientes independientes##
rm(list=ls())
set.seed(1)

#Simulacion de datos normales multivariados
# Simularemos varias distribuciones normales en tres dimensiones:
N=1000
mu=c(0,0,0)
sigma <- matrix(c(1,0.1,0.95,0.1,1,0.9,0.95,0.9,1),ncol=3,nrow=3)
sigma=Matrix::nearPD(sigma)
for (i in 1:15){
sigma[i]=Matrix::nearPD(sigma$mat)
diag(sigma$mat)=1
sigma=Matrix::nearPD(sigma$mat)
}
```

```
sigma$mat
## Simulacion de los datos incorrelacionados:
muestra1=MASS::mvrnorm(N, mu=mu, Sigma=sigma$mat)
colnames(muestra1)=c("X1","X2","Y")

##Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

####Modelo lineal múltiple
modelm= lm(Y~X1+X2,data=entren)#Entrenamos el modelo con el training set

#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciones
plot(valid[,3],valid[,4],xlab="Valor real", ylab="valor estimado"
, las=1, col="skyblue")#Gráfico para el caso uno
##Media de los errores
residuos_lm=valid[,3]-valid[,4]

lillie.test(residuos_lm)

plot(residuos_lm, xlab="", ylab= "Residuos", main="Modelo lineal multivariado",
col="blue")


with(valid,plot3d(X1,X2,Y, col="blue", size=0.7, type="s",
main="3D Linear Model Fit"))
#Creamos el plano
newdat <- expand.grid(X1=seq(-3,3,by=0.1),X2=seq(-3,3,by=0.1))
newdat$y_lm=predict(modelm,newdata=newdat)

with(newdat,surface3d(unique(X1),unique(X2),y_lm,
alpha=0.3,front="line", back="line"))

###Random forest
Bosque1=randomForest(Y~X1+X2, data=entren,ntree=800,importance=T,keep.inbag = TRUE)
plot(Bosque1)
```

```
y_rf=predict(Bosque1,valid)
plot(valid[,3],y_rf,xlab="Valor real", ylab="valor estimado", main="", col= "chartr
#
residuos_rf=valid[,3]-y_rf
#
lillie.test(residuos_rf)
##
plot(residuos_rf, xlab="", ylab= "Residuos random fores", main="",
col="chartreuse2")

ff=forestFloor(Bosque1, X =valid[,1:2])
show3d(ff,col="green",plot.rgl=list(size=8))

#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)


#########################################################################
###################################CASO 2################################
#########################################################################


#Ver gregorutti
# Simularemos varias distribuciones normales en tres dimensiones:
N=1000 ##Número de muestra extraída
mu=c(0,0,0)##" Media normal multivariada
sigma <- matrix(c(1, 0.9,0.95,0.9,1,0.95,0.95,0.95,1),ncol=3,nrow=3)#Matriz varianz

## Simulacion de los datos correlacionados:
muestra1=rmvnorm(N,mean=mu,sigma=sigma,method = "eigen")#Obtenemos muestra con dato
colnames(muestra1)=c("X1","X2","Y")#Nombres de las columnas

##Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])
```

```
#Modelos

####Modelo lineal múltiple
modelm= lm(Y~X1+X2,data=entren)#Entrenamos el modelo con el training set

#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciones
plot(valid[,3],valid[,4],xlab="Valor real", ylab="valor estimado"
, las=1, col="skyblue")#Gráfico para el caso uno
##Media de los errores
residuos_lm=valid[,3]-valid[,4]

lillie.test(residuos_lm)

plot(residuos_lm, xlab="", ylab= "Residuos", main="Modelo lineal multivariado",
col="blue")


with(valid,plot3d(X1,X2,Y, col="blue", size=0.7, type="s",
main="3D Linear Model Fit"))
#Creamos el plano
newdat <- expand.grid(X1=seq(-3,3,by=0.1),X2=seq(-3,3,by=0.1))
newdat$y_lm=predict(modelm,newdata=newdat)

with(newdat,surface3d(unique(X1),unique(X2),y_lm,
alpha=0.3,front="line", back="line"))

###Random forest
Bosque1=randomForest(Y~X1+X2, data=entren,ntree=800,importance=T,keep.inbag = TRUE)
plot(Bosque1)
y_rf=predict(Bosque1,valid)
plot(valid[,3],y_rf,xlab="Valor real", ylab="valor estimado", main="", col= "chartreuse2
mean((valid[,3]-y_rf)^2)
#
residuos_rf=valid[,3]-y_rf
#
lillie.test(residuos_rf)
##
plot(residuos_rf, xlab="", ylab= "Residuos random fores", main="",
col="chartreuse2")
```

```
ff=forestFloor(Bosque1, X =valid[,1:2])
show3d(ff,col="green",plot.rgl=list(size=8))

#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)
##############################################################################
###################################CASO 4#####################################
##############################################################################

rm(list=ls())
set.seed(1)
#Veinte variables
p=20
C=matrix(rep(0.88:0.88,400),ncol=p,nrow=p)
diag(C)=1
tau=c(rep(0.85:0.85,20))
vary=1
mu=c(rep(0:0,21))
sigma<-rbind(cbind(C,tau),c(tau,1))
sigma=Matrix::nearPD(sigma)
## Simulacion de los datos correlacionados:
muestra1=MASS::mvrnorm(n=1000, mu=mu, Sigma=sigma$mat)

colnames(muestra1)=c("X1","X2","X3","X4","X5",
"X6","X7","X8","X9","X10","X11","X12","X13","X14","X15",
"X16","X17","X18","X19","X20","Y")
##Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

#Modelos

####Modelo lineal múltiple
modelm= lm(Y~X1+X2+X3+X4+X5+X6+X7+X8+X9+
X10+X11+X12+X13+X14+X15+X16+X17+X18+X19+X20,
```

```
data=entren)#Entrenamos el modelo con el training set

#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciones
plot(valid[,21],valid[,22],xlab="Valor real", ylab="valor estimado"
, las=1, col="skyblue", main="Modelo lineal")#Gráfico para el caso uno
##Media de los errores
residuos_lm=valid[,21]-valid[,22]

lillie.test(residuos_lm)

plot(residuos_lm, xlab="", ylab= "Residuos", main="Modelo lineal multivariado",
col="blue")

###Random forest
Bosque1=randomForest(Y~X1+X2+X3+X4+X5+X6+X7+X8+X9+
X10+X11+X12+X13+X14+X15+X16+X17+X18+X19+X20,
data=entren,ntree=800,importance=T,keep.inbag = TRUE)
y_rf=predict(Bosque1,valid)
plot(valid[,21],y_rf,xlab="Valor real", ylab="valor estimado",
main="Bosque aleatorio", col= "chartreuse2")
#
residuos_rf=valid[,4]-y_rf
#
lillie.test(residuos_rf)
##
plot(residuos_rf, xlab="", ylab= "Residuos random fores", main="",
col="chartreuse2")
#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)
##############################################################################
####################################CASO 5####################################
##############################################################################

rm(list=ls())
set.seed(1)
#Veinte variables
```

```
N=1000
p=20
q=5
mu=c(rep(0:0,20))
C=matrix(rep(0.92:0.92,400),ncol=p,nrow=p)
C[16:20,]=0.2
C[,16:20]=0.2
diag(C)=1
tau=c(rep(0.87:0.87,15),rep(0.4:0.4,5))
vary=1
mu=c(rep(0:0,21))
sigma<-rbind(cbind(C,tau),c(tau,1))
sigma=Matrix::nearPD(sigma)
for (i in 1:15){
sigma=Matrix::nearPD(sigma$mat)
diag(sigma$mat)=1
sigma=Matrix::nearPD(sigma$mat)
}

## Simulacion de los datos correlacionados:
muestra1=MASS::mvrnorm(n=1000, mu=mu, Sigma=sigma$mat)
colnames(muestra1)=c("X1","X2","X3","X4","X5","X6","X7",
"X8","X9","X10","X11","X12","X13","X14",
"X15","X16","X17","X18","X19","X20","Y")


##Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

#Modelos

####Modelo lineal múltiple
modelm= lm(Y~X1+X2+X3+X4+X5+X6+X7+X8+X9+
X10+X11+X12+X13+X14+X15+X16+X17+X18+X19+X20,
data=entren)#Entrenamos el modelo con el training set
```

```
#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciones
plot(valid[,21],valid[,22],xlab="Valor real", ylab="valor estimado"
, las=1, col="skyblue", main="Modelo lineal")#Gráfico para el caso uno
##Media de los errores
residuos_lm=valid[,21]-valid[,22]

lillie.test(residuos_lm)

plot(residuos_lm, xlab="", ylab= "Residuos", main="Modelo lineal multivariado",
col="blue")

###Random forest
Bosque1=randomForest(Y~X1+X2+X3+X4+X5+X6+X7+X8+X9+
X10+X11+X12+X13+X14+X15+X16+X17+X18+X19+X20,
data=entren,ntree=800,importance=T,keep.inbag = TRUE)
y_rf=predict(Bosque1,valid)
plot(valid[,21],y_rf,xlab="Valor real", ylab="valor estimado",
main="Bosque aleatorio", col= "chartreuse2")
#
residuos_rf=valid[,21]-y_rf
#
lillie.test(residuos_rf)
##
plot(residuos_rf, xlab="", ylab= "Residuos random fores", main="",
col="chartreuse2")
#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)
#############################################################################
####################################CASO 6###################################
#############################################################################

rm(list=ls())
set.seed(1)
#cien variables
library(Matrix)
#Creamos una matriz aleatoria simétrica de correlación
C<-Matrix(rnorm(1600),40)#Creamos la matriz
C<-forceSymmetric(C)#La tornamos simétrica
```

```
diag(C)=1#Matriz de correlación
tau=runif(40, 0.2,0.99)#correlaciones con la variables Y
vary=runif(40,1,5)#Varianza de cad muestra extraída
mu=runif(41,-2,2)
sigma<-rbind(cbind(C,tau),c(tau,1))
sigma=Matrix::nearPD(sigma)
for (i in 1:15){
sigma=Matrix::nearPD(sigma$mat)
diag(sigma$mat)=1
sigma=Matrix::nearPD(sigma$mat)
}

## Simulacion de los datos aleatorios normales con media
#mu= runif(100, -3,3) y varianza runif(100,-10,10):
muestra1=MASS::mvrnorm(n=1000, mu=mu, Sigma=sigma$mat)


#Asignamos nombres a las variables creadas
nam=NULL
for(i in 1:40) {
nam <- rbind(nam,paste("X", i, sep = ""))
}
colnames(muestra1)=c(nam,"Y")

##Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

#Modelos

####Modelo lineal múltiple
fo=paste("Y~X1")
for (i in 2:40){
fo=paste(fo,"+",colnames(muestra1)[i])
}

#Modelo lineal
```

```
modelm= lm(formula(fo),
data=entren)#Entrenamos el modelo con el training set
#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciones
plot(valid[,41],valid[,42],xlab="Valor real", ylab="valor estimado"
, las=1, col="skyblue", main="Modelo lineal")#Gráfico para el caso uno
##Media de los errores
residuos_lm=valid[,41]-valid[,42]

lillie.test(residuos_lm)

plot(residuos_lm, xlab="", ylab= "Residuos", main="Modelo lineal multivariado",
col="blue")

###Random forest
Bosque1=randomForest(formula(fo), data=entren,ntree=800)

y_rf=predict(Bosque1,valid)
plot(valid[,41],y_rf,xlab="Valor real", ylab="valor estimado",
main="Bosque aleatorio", col= "chartreuse2")
#
residuos_rf=valid[,41]-y_rf
#
lillie.test(residuos_rf)
##
plot(residuos_rf, xlab="", ylab= "Residuos random fores", main="",
col="chartreuse2")
#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)
##############################################################################
############################CASOs 7, 8, 9, 10, 11 y 12#######################
##############################################################################


#######CASO 7 = cuatro variables correlacionadas
rm(list=ls())
set.seed(1)
#cien variables
library(Matrix)
```

```
p=4
caso7=matrix(rep(0.8:0.8,16),ncol=p,nrow=p)
diag(caso7)=1
tau=c(rep(0.85:0.85,4))
vary=1
mu=c(rep(0:0,5))
sigma<-rbind(cbind(caso7,tau),c(tau,1))
sigma=Matrix::nearPD(sigma)

## Simulacion de los datos correlacionados:
muestra1=MASS::mvrnorm(n=1000, mu=mu, Sigma=sigma$mat)
colnames(muestra1)=c("X1","X2","X3","X4","Y")

#Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

#Modelo lineal
modelm= lm(Y~.,
data=entren)#Entrenamos el modelo con el training set
#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciо
##Media de los errores
residuos_lm=valid[,5]-valid[,6]
mean(residuos_lm^2)
###Random forest
Bosque1=randomForest(Y~., data=entren,ntree=800)

y_rf=predict(Bosque1,valid)
#
residuos_rf=valid[,5]-y_rf
#
mean(residuos_rf^2)
##
#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)
```

```
#######CASO 8 = seis variables correlacionadas
rm(list=ls())
set.seed(1)
#cien variables
library(Matrix)
p=6
C=matrix(rep(0.8:0.8,36),ncol=p,nrow=p)
diag(C)=1
tau=c(rep(0.85:0.85,6))
vary=1
mu=c(rep(0:0,7))
sigma<-rbind(cbind(C,tau),c(tau,1))
sigma=Matrix::nearPD(sigma)

## Simulacion de los datos correlacionados:
muestra1=MASS::mvrnorm(n=1000, mu=mu, Sigma=sigma$mat)
colnames(muestra1)=c("X1","X2","X3","X4","X5","X6","Y")

#Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

#Modelo lineal
modelm= lm(Y~.,
data=entren)#Entrenamos el modelo con el training set
#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciones
##Media de los errores
residuos_lm=valid[,5]-valid[,6]
mean(residuos_lm^2)
###Random forest
Bosque1=randomForest(Y~., data=entren,ntree=800)

y_rf=predict(Bosque1,valid)
#
residuos_rf=valid[,5]-y_rf
```

```
#
mean(residuos_rf^2)
##
#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)

#######CASO 9 = ocho variables correlacionadas
rm(list=ls())
set.seed(1)
#cien variables
library(Matrix)
p=8
C=matrix(rep(0.8:0.8,64),ncol=p,nrow=p)
diag(C)=1
tau=c(rep(0.85:0.85,8))
vary=1
mu=c(rep(0:0,9))
sigma<-rbind(cbind(C,tau),c(tau,1))
sigma=Matrix::nearPD(sigma)

## Simulacion de los datos correlacionados:
muestra1=MASS::mvrnorm(n=1000, mu=mu, Sigma=sigma$mat)
colnames(muestra1)=c("X1","X2","X3","X4","X5","X6","X7","X8","Y")

#Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

#Modelo lineal
modelm= lm(Y~.,
data=entren)#Entrenamos el modelo con el training set
#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciо
##Media de los errores
residuos_lm=valid[,5]-valid[,6]
mean(residuos_lm^2)
```

```
###Random forest
Bosque1=randomForest(Y~., data=entren,ntree=800)

y_rf=predict(Bosque1,valid)
#
residuos_rf=valid[,5]-y_rf
#
mean(residuos_rf^2)
##
#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)

#######CASO 10 = diez variables correlacionadas
rm(list=ls())
set.seed(1)
#cien variables
library(Matrix)
p=10
C=matrix(rep(0.8:0.8,100),ncol=p,nrow=p)
diag(C)=1
tau=c(rep(0.85:0.85,10))
vary=1
mu=c(rep(0:0,11))
sigma<-rbind(cbind(C,tau),c(tau,1))
sigma=Matrix::nearPD(sigma)

## Simulacion de los datos correlacionados:
muestra1=MASS::mvrnorm(n=1000, mu=mu, Sigma=sigma$mat)
colnames(muestra1)=c("X1","X2","X3","X4","X5","X6","X7","X8","X9","X10","Y")

#Entrenamiento
sampling=sample(1:nrow(muestra1),round(0.75*nrow(muestra1)))
#Conjunto entrenamiento
entren= data.frame(muestra1[sampling,])
#Conjunto validación
valid=data.frame(muestra1[-sampling,])

#Modelo lineal
modelm= lm(Y~.,
```

```
data=entren)#Entrenamos el modelo con el training set
#Predicción
valid$y_lm=predict(modelm,valid)#Según el modelo estimado, realizamos las predicciо
##Media de los errores
residuos_lm=valid[,5]-valid[,6]
mean(residuos_lm^2)
###Random forest
Bosque1=randomForest(Y~., data=entren,ntree=800)

y_rf=predict(Bosque1,valid)
#
residuos_rf=valid[,5]-y_rf
#
mean(residuos_rf^2)
##
#Razón entre el error del random forest y el modelo lineal

mean(residuos_rf^2)/mean(residuos_lm^2)
```

# Bibliografía


[1] Bradley Efron, R. J. T. a. *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability 57. Springer US, 1993.

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

[3] Breiman, L., Ed. *Classification and regression trees*, repr ed. Chapman & Hall [u.a.], Boca Raton, 1998. OCLC: 247053926.

[4] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[5] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems 47*, 4 (Nov. 2009), 547–553.

[6] Galton, F. Presidential address, section h, anthropology. *British Association Reports 55* (1885), 1206–1214.

[7] Gregorutti, B. *Forêts aléatoires et sélection de variables: analyse des données des enregistreurs de vol pour la sécurité aérienne*. PhD thesis, Paris 6, 2015.

[8] Gregorutti, B., Michel, B., and Saint-Pierre, P. Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis 90* (2015), 15–35.

[9] Gregorutti, B., Michel, B., and Saint-Pierre, P. Correlation and variable importance in random forests. *Statistics and Computing* (Mar. 2016).

[10] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An Introduction to Statistical Learning*, vol. 103 of *Springer Texts in Statistics*. Springer New York, New York, NY, 2013. DOI: 10.1007/978-1-4614-7138-7.

[11] Louppe, G. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502* (2014).

[12] Montgomery, D. C. D. C., Peck, E. A., and Vining, G. G. *Introducción al análisis de regresión lineal*. No. 04; QA278. 2,. M6. 2001.

[13] Seber, G. A., and Lee, A. J. *Linear regression analysis*, vol. 936. John Wiley & Sons, 2012.

[14] Timofeev, R. *Classification and regression trees (CART) theory and applications*. PhD thesis, Humboldt University, Berlin, 2004.

[15] Verbeek, M. *A guide to modern econometrics*. John Wiley & Sons, 2008.