Variable importance in linear regression versus random forest. Analysis over consumer loans prepayment rates.

Trabajo de investigación 019/020

Máster en Banca y Finanzas Cuantitativas

Autor:
Miguel Romero Olea

Directoras:
Raquel Bujalance Rodríguez
Eva Rodrigo Barral

Universidad Complutense de Madrid
Universidad del País Vasco
Universidad de Valencia
Universidad de Castilla-La Mancha

www.finanzascuantitativas.com

Índice general

Resumen

El objetivo de este trabajo es analizar el uso de modelos random forest para tratar de explicar los drivers del comportamiento de los clientes a partir de la determinación de las variables relevantes para el modelo (variable importance) y su comparación con los obtenidos mediante una regresión lineal clásica. Dicho objetivo se aborda desde un punto de vista teórico acompañado por la implementación de un caso práctico a partir del cual se desarrolla una aplicación web.

Como caso práctico se va analizar el comportamiento de prepago en créditos al consumo aplicando random forest y posteriormente utilizar una regresión lineal múltiple como modelo de contraste. Adicionalmente, se ha creado una aplicación web que permite a cualquier usuario realizar una comparativa del modelo random forest ante diferentes asunciones con el modelo lineal y obtener un análisis estadístico de sus resultados.

1 Introducción

En los últimos años la EBA y Basilea están impulsando el análisis del comportamiento del cliente, especialmente en el ámbito de los modelos predictivos y de estrés, por lo que cada vez cobra más importancia analizar los drivers del comportamiento, determinar las variables con un impacto mayor y analizar los cambios de los resultados ante distintos regímenes.

Este trabajo pretende abordar este reto desde dos perspectivas: la principal siguiendo la metodología de machine learning random forest y de manera secundaria desde la estadística clásica a través de la regresión multinomial. A partir de estos dos modelos se pretende analizar la idoneidad de cada método, así como su "precisión", realizar una clasificación de las variables según su importancia y su posterior selección, analizar desde un punto de vista estadístico la distribución de los errores de cada método y las diferencias entre ellos.

Este trabajo se ha realizado como un proyecto end-to-end. A partir de un prototipo (en marvelapp) de la herramienta con la que se quiere realizar el análisis, se crea un mapa de las funciones necesarias para llevar a cabo la calibración de los dos modelos y el análisis estadístico (análisis de la muestra, métodos calibración, importancia de las variables, test estadísticos,...), y se realiza una interfaz en formato app que facilite el análisis del modelo principal random forest y su documentación, para permitir a cualquier usuario experimentar directamente con el modelo.

Adicionalmente, se han implementado estos modelos en un caso práctico con el objetivo de predecir los prepagos de créditos al consumo que los clientes de una entidad bancaria pueden realizar.

Se ha utilizado el lenguaje R para el código y la librería *Shiny* para crear la aplicación web.

2

Modelos teóricos

Regresión lineal múltiple

Un modelo de regresión pretende modelar una variable Y que denominaremos dependiente o de respuesta, de carácter aleatorio, mediante un conjunto de variables $X_1, X_2, ..., X_p$ denominadas independientes, predictoras o regresores, también con carácter aleatorio. En nuestro caso, como tenemos más de un regresor se trata de un modelo de regresión múltiple.

La estructura básica del modelo es la siguiente,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon. \tag{1}$$

En forma matricial lo podemos expresar como,

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

donde $\boldsymbol{\beta}$ son los coeficientes de regresión desconocidos y fijos que deseamos estimar y $\boldsymbol{\epsilon}$ es una variable aleatoria conocida como perturbación aleatoria o residuo.

Los modelos de regresión sirven para estudiar las relaciones que hay entre las variables predictoras y la variable de respuesta, describir la estructura general de los datos, predecir observaciones futuras y valorar la importancia del efecto de las variables predictoras en la variable de respuesta.

En las regresiones lineales múltiples, los residuos (ϵ) tienen que comportarse de forma aleatoria y normal (en el sentido probabilístico). Para su comprobación se puede realizar un gráfico de residuos y valores ajustados, tratando de identificar patrones que contradigan dicha aleatoriedad, así como un Q-Q plot y un test Kolmogorov-Smirnov-Lilliefors[1] en el caso de la hipótesis de normalidad. Este modelo se puede aplicar si las perturbaciones tienen esperanza nula y varianza constante ($\sigma^2 > 0$). El test implementado en este trabajo para homocedasticidad es el de Breusch-Pagan.

Si el error no se distribuye normalmente y la varianza no es constante (como el caso de series temporales que la varianza crece con la media) o hay variables binarias, se podría considerar utilizar GLM (modelo lineal generalizado). Para ello se considera que la variable de respuesta sigue algún modelo probabilístico, que normalmente consideramos perteneciente a la familia exponencial.

Adicionalmente, se va a considerar el test de Dickey-Fuller de estacionariedad y se han valorado las observaciones influyentes utilizando la distancia de Cook y Leverages.

Respecto a la ecuación (1), la varianza de los regresores se denota como v_j, j = 1,...,p, la correlación entre los regresores como ρ_{jk} y se asume que la matriz de covarianza pxp entre represores es definida positiva, de modo que cualquier matriz de regresores con $n > p$ filas tiene rango de columna completo con probabilidad 1 (ausencia de multicolinealidad).

La estimación de los parámetros se basa en el método de mínimos cuadrados y, si las perturbaciones cumplen las hipótesis de aleatoriedad y normalidad, esta estimación coincide con la obtenida por el método de máxima verosimilitud teniendo además

[1] Las palabras destacadas en rojo están vinculadas al anexo.

buenas propiedades estadísticas (Teorema de Gauss-Markov). La expresión general del estimador obtenido es,

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{Y},$$

Los momentos condicionales $E(Y|X_1, X_2, ..., X_p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$ y $var(Y|X_1, X_2, ..., X_p) = var(\epsilon|X_1, X_2, ..., X_p) = \sigma^2$ y la varianza marginal del modelo

$$var(Y) = \sum_{j=1}^{p} \beta_j^2 \nu_j + 2 \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \beta_j \beta_k \sqrt{\nu_j \nu_k} \rho_{jk} + \sigma^2. \tag{2}$$

Los dos primeros sumandos de (2) constituyen la parte de la varianza que está explicada por los regresores y el R^2 de un modelo con n observaciones independientes es consistente con la proporción de los dos primeros sumandos de (2) en el total de var(Y). Su fórmula es

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}.$$

Árbol de regresión

Un árbol de regresión se construye mediante la partición recursiva de la muestra en grupos cada vez más homogéneos, denominados nodos, hasta llegar a los "nodos terminales". La muestra se divide en función de los valores de una variable, la cual se selecciona de acuerdo con un criterio de división. Una vez que se ha construido un árbol, la respuesta para cualquier observación se puede predecir siguiendo la ruta desde el nodo raíz hasta el nodo terminal apropiado del árbol, en función de los valores observados para las variables de división. El valor de la respuesta a predecir es simplemente la respuesta promedio en ese nodo terminal.

Los árboles de regresión son una técnica no paramétrica que trabajan de forma eficiente con relaciones no lineales, datos faltantes y cuando el número de variables es mayor que el número de observaciones. Además son robustos frente a datos anómalos y se pueden utilizar para la detección de los mismos.

Por otro lado, presentan cierta inestabilidad (pequeños cambios producen árboles muy diferentes) y pueden realizar predicciones inferiores a la óptima (divide las secciones en rectángulos, lo cual puede no ser óptimo). Para combatirlo se combinan una gran cantidad de árboles en un modelo llamado random forest.

Los dos tipos de árboles que estudiaremos son CART y CI (conditional inference).

Árboles CART

El algoritmo CART propuesto por Breiman et al. (1984) [1] elige la división para cada nodo de manera que se logre la reducción máxima de la impureza global en cada nodo (criterio de Gini). La impureza de Gini se puede calcular sumando la probabilidad de que un caso del nodo t sea de la clase i, p(i|t), multiplicada por la probabilidad de un error en la categorización de ese elemento (mide la “diversidad de clases” en un nodo):

$$i(t) = \sum_{i \neq j} p(i|t)p(j|t) = \sum_{i=1}^{m} p(i|t)(1 - p(i|t)) = \sum_{i=1}^{m} p(i|t) - p(i|t)^2 =$$

$$= \sum_{i=1}^{m} p(i|t) - \sum_{i=1}^{m} p(i|t)^2 = 1 - \sum_{i=1}^{m} p(i|t))^2.$$

Veámoslo con más detenimiento cómo se divide un nodo en un ejemplo:

Supongamos que queremos discernir si una persona "ha evadido impuestos" en función de los ingresos que percibe. Usando nuestra muestra de entrenamiento, se calcula el índice de gini para distintas divisiones de cada variable y se selecciona la que tiene menor índice, ya que indica si el nodo es impuro (sería totalmente puro un valor de 0). En este ejemplo tenemos 10 personas, las cuales se colocan en función de la regla de división que se ha aplicado a sus ingresos:

Income	10K		20K		30K		40K		50K		60K		70K		80K		90K		100K		110K	
Split	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
GINI Index split	0.42		0.4		0.375		0.343		0.417		0.4		**0.3**		0.343		0.375		0.4		0.42	

$$Gini\ index_{split} = \frac{6}{10}(1 - (3/6)^2 - (3/6)^2) + \frac{4}{10}(1 - (0/4)^2 - (4/4)^2)$$
$$Gini\ index_{split} = 0.6 * 0.5 + 0.4 * 0$$
$$Gini\ index_{split} = 0.3$$

Figura 2.1: Ejemplo Gini.

De las 10 personas, 4 perciben más de 70 mil euros (4/10). De esas 4 la probabilidad de que hayan evadido impuestos es 0/4 y la probabilidad de que no hayan evadido impuestos es 4/4. De esta manera se crea el segundo sumando que indica la diversidad de clases que habrá en el nodo "hijo", en este caso sólo habrá una lo cual es ideal. El primer sumando se realiza de manera idéntica y se suman para hallar el total de homogeneidad que crea esta división. En este caso la variable objetivo es categórica, la única diferencia con nuestro caso real es que en los nodos terminales se selecciona la categoría con mayor frecuencia de casos en vez de realizar el promedio como se hace con las numéricas.

CART primero hace crecer un árbol muy grande y luego lo "poda", es decir, corta ramas que no contribuyen a la predicción con un criterio de poda que puede diferir del criterio de división. La razón para podar un árbol grande en lugar de construir un árbol pequeño es para mejorar la predicción del árbol: detenerse demasiado pronto podría pasar por alto mejoras posteriores y no podar un árbol grande podría ocasionar problemas de sobre-ajuste. Si los árboles CART se usan en random forest, normalmente los dejan crecer bastante grandes y no se realiza ninguna poda.

Árboles de Inferencia Condicional

Una vez solucionado el problema de sobre-ajuste mediante una poda, los árboles CART presentan otro problema: la interpretación de los árboles está sesgada por la selección de variables en las divisiones de los nodos. Este sesgo se induce al maximizar un

criterio de división sobre todas las posibles divisiones simultáneamente y muchos investigadores lo identificaron como un problema (por ejemplo, Kass 1980; Segal 1988; Breiman et al. 1984, p. 42). La naturaleza del problema de selección de variables en diferentes circunstancias se ha estudiado intensamente (White y Liu 1994; Jensen y Cohen 2000; Shih 2004), y Kim y Loh (2001) argumentaron que los métodos de búsqueda exhaustiva están sesgados también hacia variables con muchos valores faltantes. En conclusión, este enfoque de división en los árboles CART es injusto en presencia de variables regresivas de diferentes tipos, variables categóricas con diferente número de categorías (tiene preferencia por variables con mayor número de categorías) o diferente número de valores faltantes. Para evitar este sesgo en la selección de variables, Hothorn, Hornik y Zeileis (2006b)[2] propusieron usar pruebas de multiplicidad ajustada condicional en lugar de la reducción máxima de impureza como criterio de división. Para que cualquier nodo se divida, el procedimiento realiza un test de permutación global con hipótesis nula de que no hay asociación entre ninguno de los regresores y la respuesta dentro del nodo. De lo contrario, se comprueban las hipótesis nulas individuales de no asociación a la respuesta para cada variable y la variable con el p-valor más pequeño se selecciona para la división. Posteriormente, la regla de división se determina en función de la variable seleccionada. Con este método la poda no es necesaria, ya que cuando no hay una división estadísticamente significativa los árboles no crecen más.

Este algoritmo está inspirado en test χ^2 de independencia y se realiza una partición recursiva insesgada basada en un test de hipótesis condicional.

Sea $D(Y|\mathbf{X})$ la distribución condicional de la respuesta sabiendo las variables predictoras X y L_n una muestra aleatoria de n observaciones independientes e idénticamente distribuidas,

$$L_n = \{(Y_i, X_{1i}, ..., X_{pi}; i = 1, ..., n)\}.$$

Un algoritmo genérico para la partición recursiva de la muestra L_n puede ser formulado usando pesos enteros no negativos $\mathbf{w} = \{w_1, ..., w_n\}$. Cada nodo del árbol es representado por un vector de pesos con valor distinto a cero cuando las correspondientes observaciones forman parte del nodo y cero en caso contrario. El siguiente algoritmo implementa la partición recursiva:

1. Para los pesos w se realiza un test con hipótesis nula global de independencia entre cualquiera de las p predictoras y la respuesta formulada en términos de las p hipótesis parciales $H_0^j : D(Y|X_j) = D(Y)$ con hipótesis nula global $H_0 = \cap_{j=1}^m H_0^j$. El criterio de parada se produce cuando esta hipótesis no puede ser rechazada con un nivel de confianza α. Si la hipótesis global puede ser rechazada, se mide la asociación entre Y y cada una de las X_j realizando tests estadísticos o con el p-valor y se selecciona la j^*-ésima variable X_{j^*} con asociación mayor a Y. Medimos la asociación entre Y y $X_j, j = 1, ..., p$ usando el estadístico lineal de la forma:

$$\mathbf{T}_j(L_n, \mathbf{w}) = vec(\sum_{i=1}^{n} w_i g_j(X_{ij}) h(Y_i, (Y_1, ..., Y_n))^T) \in \mathbb{R}^{pq}$$

donde g_j es una transformación no aleatoria de las variables X_j y h es la función de influencia que depende de las respuestas. En nuestro caso como sólo tenemos una respuesta $Y \in \mathbb{R}$, la función de influencia natural es la identidad

$h(Y_i, (Y_1, ..., Y_n)) = Y_i$. En las variables predictoras numéricas se pueden aplicar la identidad igualmente $g_{ij}(x) = x$ (otras transformaciones incluso no lineales también pueden ser usadas). Las variables categóricas en los niveles 1,...,K son representadas por $g_{ij}(k) = e_K(k)$, el vector unitario de longitud K con el k-ésimo elemento igual a uno. La distribución de $\mathbf{T}_j(L_n, \mathbf{w})$ bajo H_0^j depende de la distribución conjunta de Y y X_j, la cual es desconocida en la mayoría de los casos, pero se puede hallar la media condicionada μ_j y la covarianza Σ_j bajo H_0, pudiendo así estandarizar el estadístico

$$c_{max}(\mathbf{t}, \mu, \Sigma) = \max_{k=1,...,pq} |\frac{(\mathbf{T} - \mu)_k}{\sqrt{(\Sigma)_{kk}}}|.$$

Otra alternativa es usar su forma cuadrática $c_{quad}(\mathbf{t}, \mu, \Sigma) = (\mathbf{T} - \mu)\Sigma^+(\mathbf{t} - \mu)^T$, aunque es más costoso computacionalmente. De esta manera, seleccionamos la variable predictora con menor P-valor, esto es, X_{j^*} con $j^* = argmin_{j=1,...,p} P_j$ donde

$$P_j = \mathbb{P}_{H_0^j}(c(\mathbf{T}_j(L_n, \mathbf{w}), \mu_j, \Sigma_j) \geq c(\mathbf{j}_j, \mu_j, \Sigma_j)),$$

denota el P valor del test condicional para H_0^j. Para el test global se pueden aplicar aproximaciones para pruebas múltiples como P valor de Bonferroni-ajustado.

2. Criterio de división: Se elige el conjunto $A^* = argmax_A c(\mathbf{t}_{j*}^A, \mu_{j*}^A, \Sigma_{j*}^A)$ del espacio muestral y se determinan dos subgrupos disjuntos de pesos: $w_{izq,i} = w_i I(X_{j^*i} \in A^*)$ y $w_{der,i} = w_i I(X_{j^*i} \notin A^*)$ para todo $i = 1, ..., n$ ($I(.)$ denota función indicatriz). La bondad de la división es evaluada un estadístico que es un caso particular del estudiado anteriormente

$$\mathbf{T}_{j^*}^A(L_n, \mathbf{w}) = vec(\sum_{i=1}^{n} w_i I(X_{ij^*} \in A) h(Y_i, (Y_1, ..., Y_n))^T) \in \mathbb{R}^q.$$

3. Recursivamente se repiten los pasos 1 y 2 con modificaciones de los pesos respectivamente.

Random forest

Un árbol de decisión sufre problemas de sesgo (la diferencia en promedio de los valores predecidos con los valores reales) y varianza (cuán diferentes serán las predicciones de un modelo en un mismo punto si muestras diferentes se tomaran de la misma población). Al construir un árbol pequeño se obtendrá un modelo con baja varianza y alto sesgo. Normalmente, al incrementar la complejidad del modelo se verá una reducción en el error de predicción debido a un sesgo más bajo en el modelo hasta llegar a un punto en el que éste será muy complejo y se producirá un sobre-ajuste del modelo, el cual empezará a sufrir de varianza alta. El modelo óptimo debería mantener un balance entre estos dos tipos de errores, lo que se conoce como "trade-off" (equilibrio) entre errores de sesgo y varianza. El uso de ensambladores (un grupo de modelos predictivos) como random forest es una forma de aplicar este "trade-off" ya que se

reduce la varianza de las predicciones a través de la combinación de los resultados de varios árboles, cada uno de ellos modelados con diferentes subconjuntos tomados de la misma población y al no ser podados tienen además bajo sesgo. Un random forest consiste en un gran número de árboles (*ntree*) que permiten alcanzar una mejor precisión y estabilidad del modelo.

Random forest es aleatorio por dos razones: (i) cada árbol se basa en un subconjunto aleatorio de observaciones, y (ii) cada división dentro de cada árbol se crea en base a un subconjunto aleatorio de variables candidatas *mtry* (esta selección aleatoria de características dentro de cada nodo disminuye la correlación entre los árboles en el bosque, disminuyendo así la tasa de error del bosque). Los árboles son bastante inestables, por lo que esta aleatoriedad crea diferencias en las predicciones de los árboles individuales. El algoritmo[3] es el siguiente:

Sea Ω la muestra original. Para cada $b = 1,...,B$, se toma una muestra de n observaciones de Ω. Esta es la llamada muestra bootstrapeada Ω_b. Usamos Ω_b para construir cada árbol del random forest:

1. En cada nodo t, se seleccionan aleatoriamente m de las p variables independientes.

2. Para cada $k = 1,...,m$ variables, se busca la mejor división s_k de todas las posibles divisiones para cada una de las k variables.

3. Se elige la mejor división s^* de entre las mejores divisiones s_k para dividir el nodo t. Esta variable j en su punto de corte c_{s^*} es utilizada para dividir el nodo t.

4. Separamos los datos enviando las $i = 1,...,n$ observaciones con $x_{ij} < c_{s^*}$ al nodo izquierdo descendiente y las observaciones con $x_{ij} \geq c_{s^*}$ al nodo derecho.

5. Se repiten los pasos 1-4 con todos los nodos descendientes creando un árbol con tamaño máximo, T_b.

En cada árbol del random forest se obtiene la clase predicha para cada observación. La predicción general del random forest es el promedio de las predicciones de los árboles individuales. Se considera que un gran número de árboles es particularmente importante cuando queremos discernir la importancia de las variables ya que los resultados teóricos sobre random forest son asintóticos en el número de árboles.

RF-CART y RF-CI

RF-CART y RF-CI son random forests basados en árboles CART y CI (condicional inference) respectivamente. RF-CART y RF-CI utilizan diferentes métodos de muestreo por defecto: RF-CART usa una muestra con reemplazamiento de tamaño n y RF-CI una muestra sin reemplazamiento de tamaño 0.632 * n (en el muesteo con reemplazamiento haciendo bootstrap[2] alrededor del 63.2% de los datos acaban perteneciendo a la muestra bootstrapeada). Los valores predeterminados de RF-CART y RF-CI también dan como resultado árboles de tamaño muy diferente: RF-CART utiliza árboles grandes sin podar que crecen con un límite inferior para el tamaño de los nodos a considerar para la división (nodos de tamaño 5 o menor no se dividen). Los árboles RF-CI

[2]Método de remuestreo en el que se toma para cada árbol una submuestra de tamaño n de la muestra original, reemplazando los valores una vez tomados.

se construyen por defecto con la versión de Strobl et al. (2007)[4]: el criterio de parada está basado en un p-valor univariado usando el estadístico en su forma cuadrática, pero no se realizan pruebas de significación (las pruebas sin ajuste de multiplicidad se realizan a un nivel de significación predeterminado del 100 %) para que surjan árboles grandes, limitando el tamaño del árbol al restringir el tamaño mínimo de división de un nodo a 20 y el tamaño mínimo del nodo terminal a 7. Este criterio es más estricto ya que, junto con el enfoque de muestreo, implica que los árboles en RF-CI son por defecto sustancialmente más pequeños que los de RF-CART. Los ajustes tanto para RF-CART como para RF-CI podrían realizarse de manera que los árboles se vuelvan más grandes o más pequeños ajustando los criterios de división de nodos, o, en el caso de RF-CI, introduciendo un nivel de significación menor a 1. Segal, Barbour y Grant (2004)[5] propusieron aumentar el tamaño mínimo de nodo para RF-CART para mejorar la precisión de la predicción.

Error Cuadrático Medio

De acuerdo con el muestreo aleatorio de observaciones, independientemente de si existe reemplazamiento o no, (un promedio de) el 36.8 % de las observaciones no se usan para ningún árbol individual, es decir, están "fuera de la bolsa" = OOB para ese árbol. La precisión de la predicción de un random forest se puede estimar a partir de estos datos OOB como

$$OOB - MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{\hat{y}}_{iOOB})^2,$$

donde $\bar{\hat{y}}_{iOOB}$ denota la predicción promedio para la observación i de todos los árboles para los cuales esta observación ha sido OOB y n se corresponde al hiperparámetro *ntree*. De manera análoga a la regresión lineal, con la suma total de cuadrados $SST = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2$, $OOB - R^2$ puede ser obtenido como 1-OOB-MSE/SST.

Importancia de las variables

Regresión lineal múltiple

En regresión multilineal, los métodos de importancia de las variables que descomponen R^2 se basan en descomponer los dos primeros sumandos de la varianza. Es bien sabido que esta varianza (o equivalentemente R^2) puede ser descompuesta de forma única si los regresores están incorrelados, pero para regresores correlados se obtienen diferentes resultados con métodos diferentes. En particular, el aumento en R^2 asignado a un cierto regresor X_k depende de qué regresores ya están presentes en el modelo antes de agregar X_k. Para cualquier orden particular de regresores se puede determinar una asignación única, asignando a cada regresor el aumento en R^2 (o varianza explicada) cuando se agrega al modelo. Para una asignación única y justa, Lindeman, Merenda y Gold (1980)[6] propusieron promediar esas asignaciones orden-dependientes sobre todos los órdenes. Esta aproximación se denomina LGM. Feldman (2005) propuso una versión modificada con pesos dependientes de los datos para favorecer mejores predicciones. Este método se denomina PMVD: Proportional Marginal Variance Decomposition.

Random Forest

En los modelos random forest y árboles CART, los método más utilizados para analizar la importancia de las variables son la "importancia de Gini" (utilizado en modelos de clasificación) y su análogo para random forest, "reducción de impureza promedio" (media ponderada de cada árbol individual de la mejora en las divisiones según la importancia de Gini).

Sin embargo, debido al sesgo de impureza para seleccionar variables divididas, las métricas de importancia de variables resultantes también están sesgadas. Breiman (2002) sugirió una reducción en la MSE al permutar una variable, denominada "reducción de la MSE" como el método de elección. Aunque descartó esta métrica por una variabilidad excesiva en situaciones con muchos regresores en una versión posterior del manual de Random Forests (Breiman 2003)[7], varios autores adoptaron la reducción de MSE basada en la permutación como el enfoque más avanzado. La idea se basa en que al permutar aleatoriamente una variable predictora (dejando el resto sin permutar) se rompre la asociación que tenía con la variable respuesta y la precisión de la predicción decrece sustancialmente. Se determina de la siguiente manera: Para el árbol t, el error cuadrático medio OOB se calcula como el promedio de las desviaciones cuadradas de las respuestas OOB de sus predicciones respectivas:

$$OOBMSE_t = \frac{1}{n_{OOB,t}} \sum_{\substack{i=1: \\ i \in OOB_t}}^{n} (y_i - \hat{y}_{i,t})^2,$$

donde el $\wedge$ indica predicciones, OOB_t = {i: observación i es el OOB para el árbol t}, es decir, la suma es solo sobre observaciones OOB, y $n_{ooB,t}$ es el número de observaciones OOB en el árbol t. Si el regresor X_j no tiene un valor predictivo para la respuesta, no debería hacer una diferencia si los valores de X_j se permutan aleatoriamente antes de que se generen las predicciones en los datos OOB . Así,

$$OOBMSE_t(X_j permutado) = \frac{1}{n_{OOB,t}} \sum_{\substack{i=1: \\ i \in OOB_t}}^{n} (y_i - \hat{y}_{i,t}(X_j permutado))^2,$$

no debe ser sustancialmente más grande (o podría incluso ser más pequeño) que $OOBMSE_t$. Para cada variable X_j en cada árbol t, la diferencia $OOBMSE_t$(X_j permutada) - $OOBMSE_t$ se calcula en base a una permutación de los datos de fuera de bolsa de la variable para el árbol (esta diferencia es, por supuesto, 0 para una variable que no está involucrada en ninguna división del árbol t). La reducción de la MSE según el regresor X_j para el bosque completo se obtiene como el promedio de todos los *ntree* árboles de estas diferencias. Este método cubre el impacto de cada variable predictora individualmente y las interacciones multivariadas con otras variables predictoras.

Como dijimos anteriormente, la importancia de las variables asignadas usando el método Gini no es justa cuando hay muchas categorías y mediante permutaciones es mucho más seguro. Sin embargo debemos observar que este último método sobreestima la importancia de las variables si están altamente correlacionadas como muestra Strobl et al. (2007)[4].

Strobl et al. (2008)[8] proponen una variante utilizando variables insesgadas en un marco de inferencia condicional ya que consideran que la selección de variables con la

medida de importancia de las variables del método de random forest original conlleva el riesgo de que se prefieran artificialmente variables para predicción subóptimas. Los dos mecanismos que subyacen a esta deficiencia son la selección de la variable sesgada en los árboles individuales utilizados para construir el random forest y los efectos inducidos por el muestreo bootstrap con reemplazamiento.

Archer et al. (2008)[3] propone un método para tratar la inestabilidad de la muestra cuando tenemos muchas variables en comparación con el número de observaciones. Dicha estabilidad puede ser obtenida realizando bootstrap de la muestra o *bagging*.

Mientras que LMG y PMVD naturalmente descomponen R^2, tal descomposición natural no ocurre para la reducción de MSE de los random forest. Por lo tanto, a efectos de comparación, todas las métricas de importancia de variables se han normalizado para sumar el 100 %.

Adicionalmente se considera un método alternativo para no sólo medir la importancia de las variables sino también la fuerza de los efectos de las interacciones (dependencia entre las variables) y que permite usar distintos algoritmos de aprendizaje automático combinados en un conjunto llamado *super learner* mediante un proceso denominado *model stacking*. Este método se denomina Partial Dependence Plots (PDP).

PDP son representaciones gráficas de baja dimensionalidad de la función de predicción $\hat{f}(x)$ que permite al analista entender con mayor facilidad la relación entre la respuesta y predictores de interés. Estos gráficos son especialmente útiles para interpretar las salidas en modelos denominados "cajas negras".

Sea $\mathbf{x} = x_1, ..., x_p$ predictores cuya función de predicción es $\hat{f}(x)$. Si particionamos $\mathbf{x}$ en un conjunto de interés, z_s, y su complementario, $z_c = \mathbf{x} \backslash z_s$ entonces la "dependencia parcial" de la respuesta en z_s se define como:

$$f_s(z_s) = E_{z_c}[\hat{f}(z_s, z_c)] = \int \hat{f}(z_s, z_c)\rho(z_c)dz_c,$$

donde $\rho(z_c)$ es la densidad de la probabilidad marginal de z_c. Esta ecuación puede ser estimada a partir de datos de entrenamiento como :

$$\bar{f}_s(z_s) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(z_s, z_{i,c}),$$

donde $z_{i,c}$ $(i = 1, ..., n)$ son los valores de z_c que suceden en la muestra de entrenamiento, es decir, es el promedio de todas las otras variables predictoras del modelo.

La PDP para x se obtiene graficando los pares $\{x_{ij}, \bar{f}_i(x_{ij})\}$ para $j = 1, ..., k_i$. Las PDP planas significan que no tienen mucha influencia en los valores predichos, es decir, los valores de dependencia parcial $\bar{f}_i(x_{ij})$ tienen poca variabilidad.

La medida de importancia de las variables para el predictor x_s es:

$$\mathrm{i}(\mathrm{x}_s) = \begin{cases} \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}[\bar{f}_s(x_{si}) - \frac{1}{k}\sum_{i=1}^{k}\bar{f}_s(x_{si})]^2} & x_s \text{ continua} \\ [\max_i(\bar{f}_s(x_{si})) - min_i(\bar{f}_s(x_{si}))]/4 & x_s \text{ categórica} \end{cases}$$

Selección de variables

La selección de variables es un tema crucial en muchos problemas aplicados de regresión, especialmente cuando trabajamos con modelos de regresión lineal. Es interesante tanto para el análisis estadístico como para la modelización o la predicción eliminar variables irrelevantes, seleccionar todas las importantes o determinar un subconjunto suficiente para la predicción. Estos objetivos, diferentes desde una perspectiva de aprendizaje estadístico, implican la selección de variables para simplificar los problemas estadísticos, para ayudar en el diagnóstico e interpretación y para acelerar el procesamiento de datos.

En caso de los random forest la eliminación de variables irrelevantes no es un paso crucial en la determinación del modelo, ya que solo suponen incrementar el tiempo de computación para mejorar la precisión, pero la selección de variables dentro del árbol puede ayudar para determinar las variables relevantes y sus iteraciones a considerar en un modelo de regresión lineal (si es necesario su uso).

Regresión lineal múltiple

La idea se basa en introducir o eliminar una variable predictora considerando una medida de adecuación del modelo a nuestro objetivo. El mejor modelo será el que optimice este criterio. Si disponemos de p variables predictoras, hay 2^p modelos que chequear, por lo que se requiere un alto coste computacional.

El criterio de selección de variables que nos disponemos a implementar es el criterio de información de Akaike (AIC) que, basado en la verosimiltud, busca un equilibrio entre el ajuste y el número de parámetros utilizados. Además, se utiliza un prodedimiento forward (hacia delante) en el que se parte del modelo "vacío" y usando la función *stepAIC* del paquete *MASS* se van añadiendo variables creando un modelo reducido del inicial que minimice el valor Akaike del modelo.

Random Forest

Es habitual distinguir tres tipos de métodos de selección de variables: "filter" para el cual la puntuación de importancia variable no depende de un modelo dado; "wrapper" que incluye las predicciones en el cálculo de puntuación; y, finalmente,"embedded", que combina de manera más complicada la selección de variables y la estimación del modelo. Dentro de la familia de modelos "wrapper" destacamos el propuesto por Díaz-Uriarte et al. (2006)[9] al ser muy conocido y cercano al procedimiento que vamos a usar. Se trata de una estrategia basada en la eliminación recursiva de variables. Primero computan la importancia de las variables del random forest y luego en cada paso eliminan el 20% (parámetro arbitrario, no depende de los datos) de las variables con menor importancia y construyen un nuevo random forest con las variables que quedan. Finalmente seleccionan un conjunto de variables con menor tasa de error OOB.

En este trabajo usaremos un método propuesto por Genuer el al.(2010)[10] aplicado al algoritmo *randomForest* implementado anteriormente. Este método utiliza un procedimiento distinto en función de dos objetivos:

1. Encontrar variables importantes altamente relacionadas con la respuesta para conseguir una mejor interpretación.

2. Encontrar un número pequeño de variables suficientes para una buena predicción de la variable respuesta.

El método que vamos a utilizar es un procedimiento de dos pasos, el primero es común mientras que el segundo depende del objetivo:

1. Computar la importancia de las variables del random forest y ordenarlas de forma decreciente eliminando aquellas con menor importancia hasta quedarnos con m variables restantes.

   De forma más precisa, a partir de este orden, considerar la secuencia ordenada de las desviaciones estándar correspondientes de la importancia de las variables (IV) y utilizarla para estimar un valor de umbral para IV. Como la variabilidad de IV es mayor para las variables verdaderas en comparación con las irrelevantes, el valor de umbral viene dado por una estimación de la desviación estándar de IV de las variables irrelevantes. Este umbral se establece en el valor de predicción mínimo dado por un modelo CART en el que Y es la de IV y X son los rangos. Entonces solo se retienen las variables con una IV promediado que excede este umbral.

2. Selección de variables:

   - Para *interpretación*: construir la colección anidada de random forests que involucren las k primeras variables, para $k = 1, ..., m$, empezando con un modelo sólo con la variable de mayor importancia y terminando con otro que involucre a todas las variables retenidas previamente y seleccione las variables involucradas en el modelo con error OOB más pequeño. De hecho, para evitar problemas de inestabilidad, se selecciona el modelo más pequeño con un error OOB más pequeño que el que tiene el error OOB mínimo aumentado por su desviación típica.
   - Para *predicción*: a partir de las variables ordenadas retenidas para la interpretación, construir una secuencia ascendente de modelos random forest, tomando y probando las variables paso a paso: se agrega una variable sólo si la ganancia del error supera un umbral. La idea es que la reducción del error debe ser significativamente mayor que la media de la variación obtenida añadiendo variables que provocan ruido. El umbral se establece como la media de los valores absolutos de la diferencia de los errores OOB del primer modelo que seleccionamos para interpretación, p_{interp}, y el que contiene todas las variables, p_{elim}:

     $$\frac{1}{p_{elim} - p_{interp}} \sum_{j=p_{interp}}^{p_{elim}-1} |OOB(j+1) - OOB(j)|.$$

     Se seleccionan las variables del último modelo.

Se debe tener en cuenta que si se desea estimar el error de predicción, ya que el ranking y la selección se realizan en el mismo conjunto de observaciones, se recomienda una evaluación de error en un conjunto de prueba o usar un esquema de validación cruzada.

Aplicación Web

Como complemento al trabajo se ha creado un "toy model" en formato de app reactiva en la que se compara la predicción de un random forest con la regresión multilineal. La aplicación web está creada un el paquete de R llamado *Shiny*. En ella se permite al usuario de introducir cualquier base de datos y poder analizar con gran rapidez ambos modelos, seleccionando distintas parametrizaciones del random forest. La guía de uso de dicha aplicación se puede encontrar dentro de la propia aplicación a través del enlace: https://miro.shinyapps.io/AppWeb/.

3

Caso Práctico

Análisis previo y simulación de datos

La base de datos proporcionada para este estudio está formada por datos de clientes de financiación de productos de consumo (datos simulados en base al comportamiento habitual de dichos segmentos). La base está determinada por 15 variables para las cuales se han obtenido datos a nivel cliente mensuales desde 2012 hasta 2018. Como el presente trabajo se va a enfocar en el análisis de sección cruzada se ha seleccionado Diciembre 2017 (la base cuenta en dicho mes con 12418 clientes) como periodo de implementación.

Las variables consideradas son las siguientes[1]:

Variables	Descripción
feoperac	Fecha operación
COUNTERPARTY	Contrapartida legal o natural
AM_METH	Francesa o alemana
Maturity	Vencimiento mayor o menor 3 meses
contra1	ID contrato
fechaper_2fm	Fecha apertura
VCTO_RES_V1	Vencimiento residual en días
fecve2_2_fm	Fecha de vencimiento
imp1limi	Importe concedido
importh_abs	Importe absoluto
TIP_INT_1	Tipo interés del contrato
tenor	Indicador meses entre apertura y vencimiento
DEFAULT	Incumplimiento en algún momento
cpr_v1	Prepago parcial
prep_tot	prepago total

Cuadro 3.1: Variables

Versión 1 del Modelo

Limpieza y transformación de la base de datos

El siguiente paso consistió en estudiar las variables que teníamos en la base de datos y la información que aportaban para la implementación de nuestros modelos.

(a) Variables con datos faltantes

- *Default* y *cpr_v1*: Para estas dos variables los datos faltantes se han sustituido por cero, ya que dichas variables indican si algún contrato ha realizado incumplimiento o ha realizado un prepago. Al tener información sólo sobre los que sí han sucedido dichas acciones hemos interpretado que los datos faltantes corresponden a los casos en los que no se han producido.

[1] *tenor:* "3" aquellos contratos cuya diferencia entre fechas de apertura y vencimiento es menor o igual a 3 meses. "12" aquellos contratos cuya diferencia entre fechas de apertura y vencimiento es mayor a 3 meses y menor o igual a 12 meses. "36" aquellos contratos cuya diferencia entre fechas de apertura y vencimiento es mayor a 12 meses y menor o igual a 36 meses. "60" aquellos contratos cuya diferencia entre fechas de apertura y vencimiento es mayor a 36 meses y menor o igual a 60 meses."61": aquellos cuya diferencia entre fechas es mayor a 60 meses.

- *importh_abs*: Se ha sustituido por la media, ya que es un procedimiento sencillo y este trabajo no pretende abordar de manera más profunda el tratamiento de datos faltantes. El principal inconveniente es que tiende a subestimar la variabilidad real de la muestra al sustituir los faltantes por valores centrales de la distribución.

(b) Eliminación de variables

- *Maturity*: Se elimina porque es una variable que indica si faltan más o menos de tres meses hasta el vencimiento y ya existe una variable numérica que especifica exactamente los días hasta vencimiento.
- *contra*1, $fechaper\_2fm$, *feoperac* y $fecve2\_2\_fm$: Se eliminan porque el número de identificación de cada contrato y las fechas son variables categóricas con un nivel para cada caso, por lo que no nos aportan información relevante y hacen que nuestro modelo aumente considerablemente de dificultad.
- *imp*1*limi*: Se elimina porque tiene una alta correlación con *importh_abs* y menos correlación con la variable objetivo. El siguiente gráfico muestra la matriz de correlaciones entre las distintas variables:

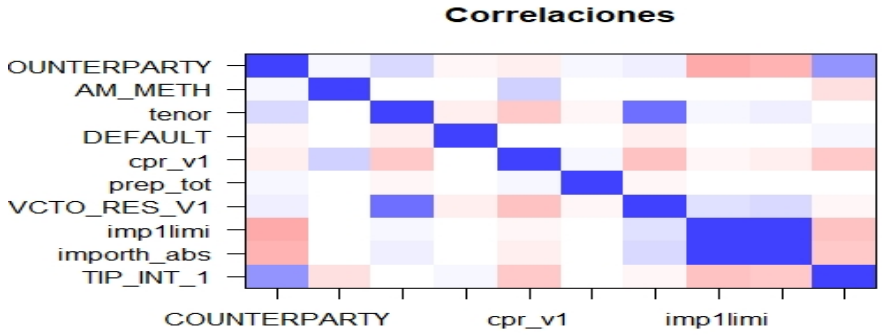

Figura 3.1: Matriz correlaciones.

Analizando la matriz de correlaciones observamos que además de la colinealdad que existe entre *imp*1*limi* e *importh_abs*, la variable cpr_v1 tiene una ligera correlación positiva con la variable AM_METH y una ligera correlación negativa con *tenor*, $VCTO\_RES\_V1$ y TIP_INT_1.

(c) Variable explicativa. Muestra desbalanceada.

Posteriormente se analizó la variable que queríamos predecir en nuestros modelos, cpr_v1, y se concluyó que la muestra estaba desbalanceada ya que el 90.8 % de los prepagos eran cero. Una alternativa para abordar dicho problema era realizar un "upsampling" aumentando el número de casos distintos de cero en la muestra de entrenamiento para que los modelos supiesen reconocer los casos en los que se realizan prepagos (valores extremos), que son los interesantes para nuestro estudio. Este proceso tiene cierta controversia ya que disminuye la precisión del modelo en general e introduce sesgo al modelo. La alternativa con la que finalmente se trabajó fue abordar el problema considerando sólo aquellas observaciones que

hubiesen realizado prepago, metodología que se puede combinar en análisis futuros con una clasificación previa para discernir si se ha realizado prepago o no.

Las siguientes tablas describen la frecuencia de los datos en las variables categóricas sobre la muestra finalmente considerada y algunos estadísticos básicos como máximos y mínimos, primer y tercer cuartil, la media y la mediana, que nos sirven para tener una primera intuición sobre algunos de los valores que pueden ser atípicos:

COUNTERPARTY	AM_METH	tenor	DEFAULT	cpr_v1
LEG ENT: 91	French:954	3 : 10	0:953	Min. :0.000615
NAT PER:868	German: 5	12: 8	1: 6	1st Qu.:0.008460
		36: 80		Median :0.012322
		60:376		Mean :0.080347
		61:485		3rd Qu.:0.013364
				Max. :1.000000

Cuadro 3.2: Datos

prep_tot	VCTO_RES_V1	imp1limi	importh_abs	TIP_INT_1
0:953	Min. :-1.542661	Min. :-0.083454	Min. :-0.073595	Min. :-2.21508
1: 6	1st Qu.: 0.007621	1st Qu.:-0.061513	1st Qu.:-0.050522	1st Qu.:-0.05732
	Median : 0.525232	Median :-0.041402	Median :-0.032166	Median : 0.31077
	Mean : 0.617928	Mean : 0.017204	Mean : 0.024818	Mean : 0.19616
	3rd Qu.: 1.387067	3rd Qu.:-0.007809	3rd Qu.:-0.001402	3rd Qu.: 0.83117
	Max. : 7.709313	Max. : 5.968104	Max. : 6.282421	Max. : 2.10045

Cuadro 3.3: Datos

Análisis de datos atípicos

Para finalizar con el tratamiento de datos, se procedió a realizar un análisis de los datos atípicos eliminando aquellas observaciones consideradas valores extremos reduciendo así el número de observaciones a 957. Este tratamiento se encuentra en el anexo para agilizar la lectura del documento y porque afecta también a la siguiente versión del modelo.

Análisis de relaciones bivariantes

La siguiente figura proporciona una visión general de las relaciones bivariadas para esta base de datos a la que denotaremos como **base datos original**:

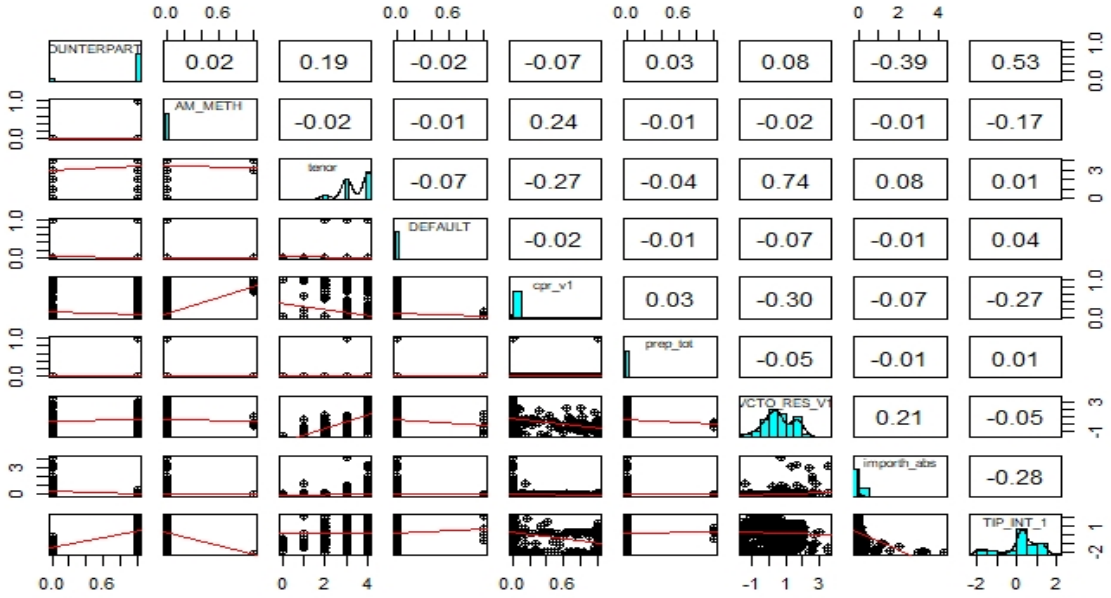

Figura 3.2: Scatterplot datos.

En la diagonal está dibujado un histograma y la densidad de cada variable. La matriz triangular superior muestra la correlación de pearson con dos decimales. La matriz triangular inferior dibuja un diagrama de puntos en color negro con una regresión de color rojo para cada par de variables.

Versión 2 del Modelo

Posteriormente, pensamos que añadir una dimensión temporal al problema podría aportar resultados interesantes por lo que se ha realizado una simulación de variables sintéticas que permitan recoger la evolución temporal en el modelo de sección cruzada.

Para la creación de dichas variables, se seleccionó la variable *importh_abs*, que informa sobre la deuda que le queda al cliente por abonar, ya que puede ser un indicador bastante fiable de los prepagos debido a que su variación entre meses indica el dinero que ha abonado en ese tiempo, tanto lo que le corresponde pagar como si ha realizado un prepago. Por lo tanto, hallamos los rendimientos absolutos ($a_t - a_{t-i}$, siendo i = 1, 2, 3, 6, 9 y 12 meses) normalizados[2]. Estas variables fueron llamadas como: *v.ia.1m*, *v.ia.2m*, *v.ia.3m*, *v.ia.6m*, *v.ia.9m* y *v.ia.12m*.

Una vez realizada dicha simulación, nos dimos cuenta de que los créditos eran muy jóvenes ya que conforme aumentábamos el periodo (la máxima diferencia de tiempo era 1 año) resultaba que cada vez más contratos no existían elevando el porcentaje de datos faltantes en los rendimientos: 3 % (1m), 7.12 % (2m), 16.19 % (3m), 25.1 % (6m), 34.98 % (9m) y 43.69 % (12m). Este tratamiento de datos faltantes lo abordamos desde dos puntos de vista:

[2]Cada variable fue normalizada de manera independiente para todos los contratos de Diciembre, Noviembre, Octubre, Septiembre, Junio, Marzo de 2017 y Diciembre de 2016.

1. Sustituyendo los datos faltantes de cada columna de rendimientos por la media de dicha columna sin los faltantes. Por ejemplo, los datos faltantes de *v.ia.1m* se sustituyeron por la media de lo que pagaron todos los clientes en el periodo de tiempo de un mes. La base de datos formada por las variables originales y las simuladas de esta manera se le denominó **base de datos simulada 1**. A estos datos se les realizó un análisis de atípicos igual que el de la base de datos anterior reduciendo así el número de observaciones a 952 y se graficó un mapa de calor de la matriz de correlaciones al igual que la gráfica de relaciones bivariantes.

   Analizando la matriz de correlaciones observamos que la variables simuladas tienen mucha correlación entre sí por lo que decidimos quedarnos sólo con *v.ia.3m* y *v.ia.6m*.
2. Sustituyendo los datos faltantes en las variables simuladas para cada contrato por la media de las variables simuladas anteriores. Por ejemplo, los datos faltantes para un determinado cliente en las variables simuladas *v.ia.9m* y *v.ia.12m* se sustituyeron por la media de lo que pagó esa persona cuando pasaron 1, 2, 3 y 6 meses. A esta base de datos formada por las variables originales y las simuladas de esta manera se le denominó **base de datos simulada 2**. A estos datos se les realizó un análisis de atípicos igual que el de la base de datos anterior reduciendo así el número de observaciones a 952 y se graficó un mapa de calor de la matriz de correlaciones al igual que la gráfica de relaciones bivariantes.

Metodología de calibración

Una vez creada la base de datos se divide en una muestra de entrenamiento (2/3 de las observaciones recogidas de manera aleatoria), "trainingSet", y una muestra de prueba (1/3 restante), "testSet". El proceso que se aplicará a los datos viene resumido en la siguiente figura:

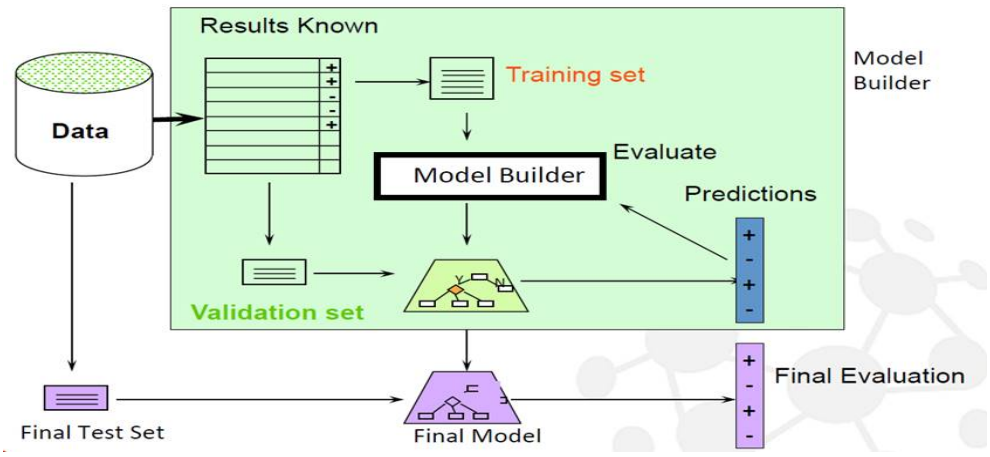

Figura 3.3: Esquema proceso.

En ella observamos cómo se calibra el modelo a partir de los datos de entrenamiento a los que les aplicamos validación cruzada y finalmente evaluar cómo funciona con una parte de la muestra reservada (testSet) para implementarla en el modelo final.

Random Forest

RF-CART

Tras el análisis previo de los datos y los distintos tratamientos, en esta sección procedemos a implementar el RF-CART para regresión con el objetivo de comprobar la relación que tienen los prepagos (cpr_v1) con el resto de variables.

La implementación de este modelo depende de sus dos hiperparámetros, *ntree* y *mtry*. Según la bibliografía consultada los valores estándares serían la raíz cuadrada del número de variables para *mtry* y se considera aceptable un valor de *ntree* de 1000. Sin embargo, hemos considerado optimizar dichos hiperparámetros mediante una 5-Cross-Validation para cada base de datos y comprobar aquella con la que se obtengan mejores resultados:

1. Para la base de datos original, teóricamente el valor óptimo de *mtry* sería 3, lo cual coincide con el mejor tras la optimización:

mtry	RMSE	Rsquared	MAE
2	0.1674525	0.4861822	0.0757015
3	0.1649312	0.4867868	0.0704074
4	0.1649216	0.4829681	0.0691621
5	0.1654567	0.4797856	0.0692413

Cuadro 3.4: Mtry RF-CART

   Una vez establecido el modelo estimamos las predicciones de los datos test para nuestra variable objetivo y observamos que el coeficiente R^2 es 42.40 %.

2. Para la base de datos simulada 1 teóricamente el valor óptimo de *mtry* sería 4 pero tras la optimización consideramos que con un valor de 5 se comete un menor error:

mtry	RMSE	Rsquared	MAE
2	0.1480833	0.6297556	0.0639226
3	0.1420939	0.6516247	0.0577781
4	0.1408170	0.6559279	0.0560706
5	0.1406545	0.6565184	0.0550772

Cuadro 3.5: Mtry RF-CART

   Una vez establecido el modelo estimamos las predicciones de los datos test para nuestra variable objetivo y observamos que el coeficiente R^2 es 55.09 %.

3. Para la base de datos simulada 2 teóricamente el valor óptimo de *mtry* sería 4,lo cual coincide con el mejor tras la optimización:

mtry	RMSE	Rsquared	MAE
2	0.1467566	0.6108495	0.0611025
3	0.1441018	0.6180036	0.0574337
4	0.1433811	0.6190456	0.0563196
5	0.1443659	0.6121812	0.0558954

Cuadro 3.6: Mtry RF-CART

   Una vez establecido el modelo estimamos las predicciones de los datos test para nuestra variable objetivo y observamos que el coeficiente R^2 es 64.11 %.

Comparando estos resultados parece evidente que utilizar las variables sintéticas que introducen la evolución temporal mejora considerablemente nuestro modelo y, para poder continuar con los análisis, seleccionamos la base de datos simulada 2 ya que obtenemos una ligera mejoría respecto a la otra base de datos simulada[3].

Una vez establecido el modelo estimamos las predicciones de los datos test para nuestra variable objetivo. Al tratarse de un elevado número de predicciones representamos una pequeña muestra en la siguiente tabla, formada por los datos reales y los estimados, de todas las que representamos en el gráfico:

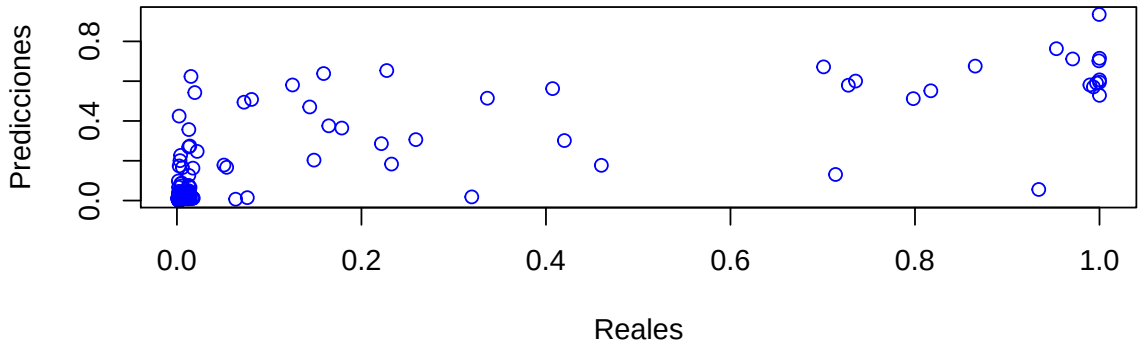

Figura 3.4: Predicciones RF-CART.

	Reales	Predicciones
1	0.00	0.01
2	0.08	0.51
3	0.14	0.47
4	0.01	0.01
5	0.41	0.56
6	0.01	0.04

Cuadro 3.7: Predicciones RF-CART

A continuación, implementamos la medida de importancia de las variables enunciadas en los modelos teóricos para poder compararlas con aquellas que considere importantes en regresión lineal múltiple:

[3]Estas estimaciones se realizaron fijando la semilla con un valor de 200 por lo que para evitar que fuesen casos anómalos ejecutamos estos modelos para los valores 50, 100, 150, ... ,500 obteniendo resultados muy parecidos.

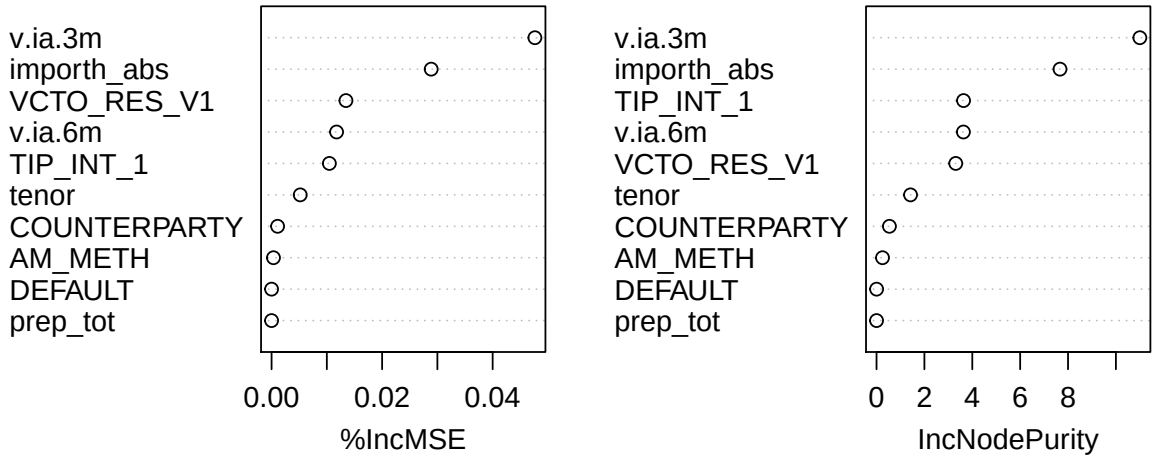

Figura 3.5: Importancia variables RF-CART.

	%IncMSE	IncNodePurity
COUNTERPARTY	8.48	0.54
AM_METH	9.15	0.25
tenor	20.04	1.42
DEFAULT	1.32	0.00
prep_tot	1.10	0.00
VCTO_RES_V1	26.12	3.31
importh_abs	42.14	7.66
TIP_INT_1	33.73	3.63
v.ia.3m	71.93	11.00
v.ia.6m	24.67	3.63

Cuadro 3.8: IV RF-CART

En esta gráfica nos centramos en %IncMSE ya que es la medida más robusta e informativa. Se trata de la medida de aumento del MSE explicada en los modelos teóricos escalada: ($OOBMSE_t(X_j$ permutada) - $OOBMSE_t) * 100/OOBMSE_t$.

Cuanto mayor sea esta la cantidad más importante será la variable, por lo que la variable *v.ia.3m* es la que tiene mayor relevancia seguida de *importh_abs*. También observamos que la otra variable simulada está ordenada en el puesto cuarto.

La medida IncNodePurity se relaciona con la función de pérdida (MSE para regresión) que se elige por mejores divisiones. Las variables más útiles logran mayores incrementos en la pureza de los nodos, es decir, la diferencia entre MSE antes y después de la división que se suma a todas las divisiones para esa variable en todos los árboles. Sin embargo, IncNodePurity es una medida que está sesgada y solo debe usarse si el tiempo de cálculo adicional del cálculo de %IncMSE es inaceptable.

Metodología complementaria Importancia Variables

Adicionalmente, se han implementado una metodología complementaria para medir la importancia de las variables llamada *Partial Dependence Plots*:

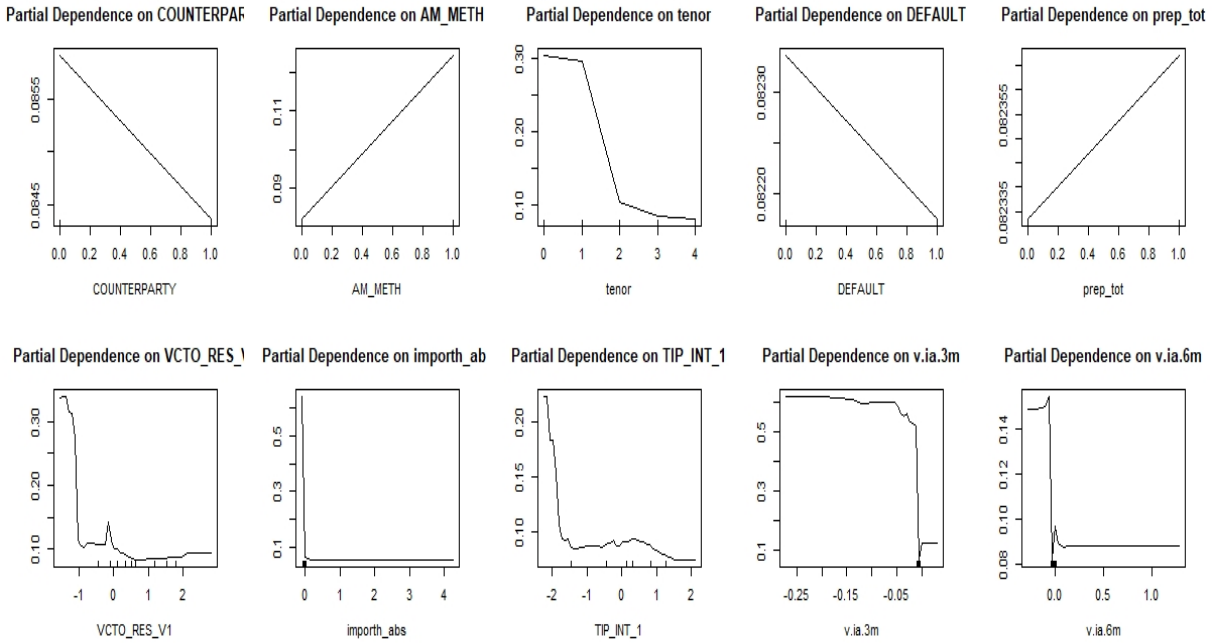

Figura 3.6: PDP

Los gráficos PDP muestran cómo varían en promedio las predicciones a medida que se modifica el valor de un predictor, manteniendo constantes el resto.

Una PDP plana indica poca importancia de las variables. El gráfico PDP de la variable simulada 3 meses muestra que la relación entre la prepagos y variaciones de la deuda cada tres meses es decreciente hasta que la deuda no exista, es decir, conforme vamos pagando nuestra deuda las predicciones de los prepagos disminuyen acercándose a cero. Cuando aumentamos nuestra deuda la media de las predicciones de los prepagos se mantiene constante y ligeramente superior a 0.1.

Sensibilidad al número de árboles

En relación al hiperparámetro *ntree* en la siguiente gráfica observamos cómo el error cuadrático medio (MSE) cometido converge asintóticamente como se mencionaba en la bibliografía cuando aumentamos el número de árboles:

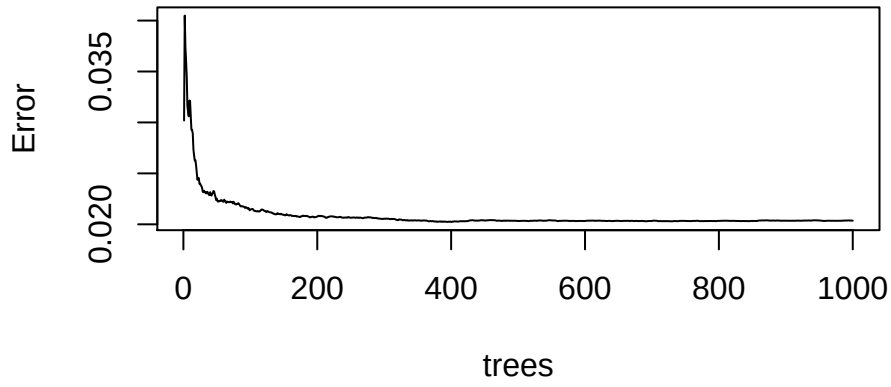

Figura 3.7: Error-ntree RF-CART.

Sin embargo, quisimos comprobar si esta tendencia se mantiene aumentando considerablemente dicho parámetro (*ntree* = 1000, 1500, 2000 y 2500) obteniendo los siguientes resultados:

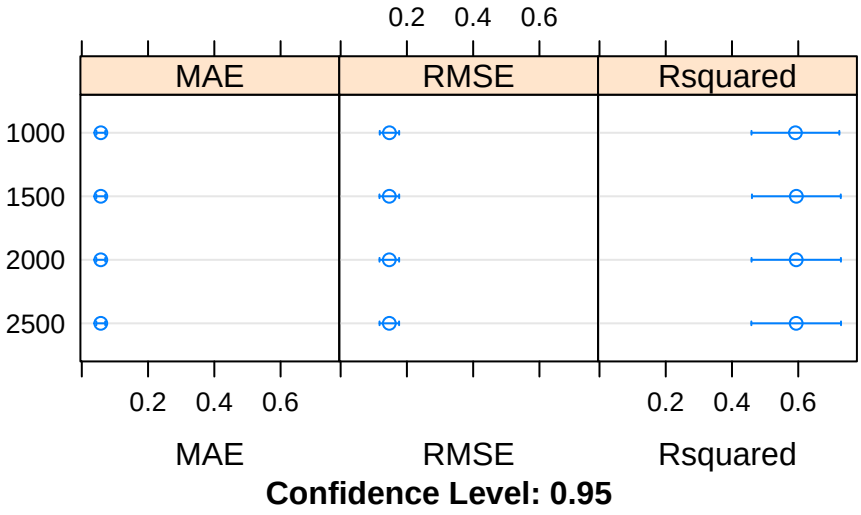

Figura 3.8: ntree RF-CART.

Observamos que tanto el R^2 como los errores se mantienen prácticamente constantes por lo que para agilizar los procesos de computación y para seguir la línea de la documentación se decidió aceptar el hiperparámetro *ntree* con valor de 1000.

Selección de Variables

Para finalizar, decidimos realizar una selección de variables:

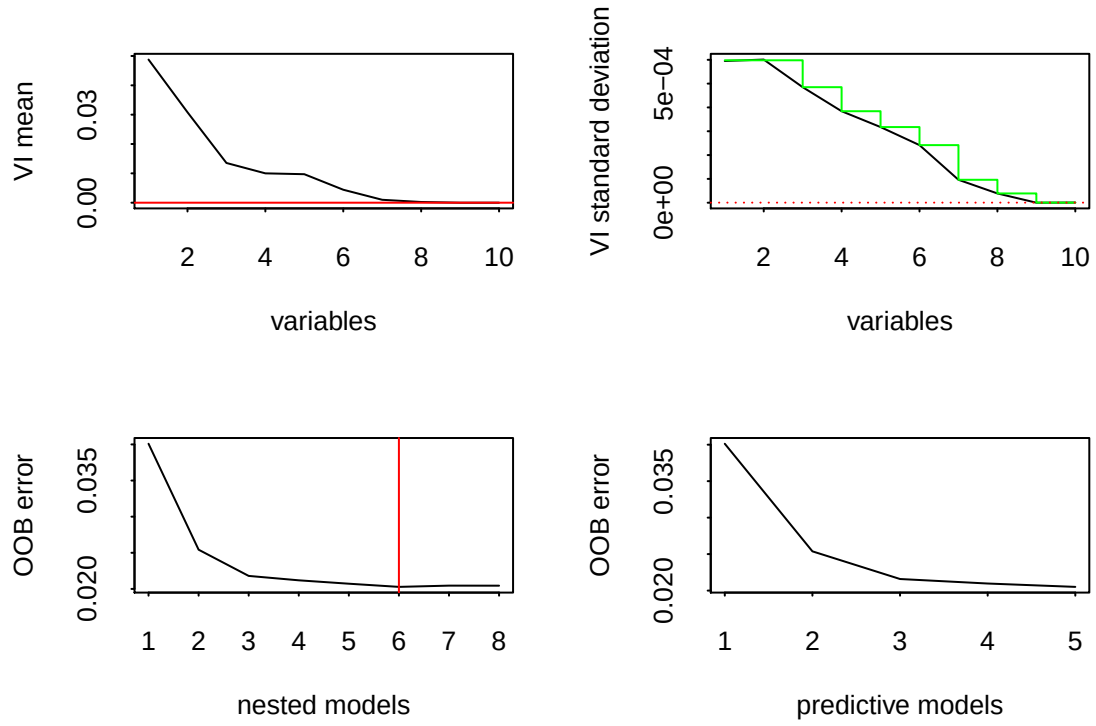

Figura 3.9: Selección variables RF-CART.

Los dos primeros gráficos indican la media y desviación típica de la importancia de las varibles del modelo ordenadas de forma decreciente. Con el segundo gráfico se establece el umbral (3.417595e-07) para seleccionar las variables (8 en nuestro caso) que lo superen en media como teóricamente decía el primer paso del algoritmo.

En los dos últimos gráficos observamos que obtenemos el menor OOB-error para interpretación con seis variables pero para predicción selecciona solamente cinco variables con un OOB-error de 0.02050717. Dichas variables son TIP_INT_1, $VCTO\_RES\_V1$, *prep_tot*, *v.ia.3m* e *importh_abs*.

RF-CI

En esta sección procedemos a implementar el RF-CI que al igual que RF-CART depende de dos hiperparámetros, *ntree* y *mtry*, de los caules consideramos aceptable un valor de ntree de 1000 y optimizamos el hiperparámetro *mtry* mediante una 5-Cross-Validation obteniendo el mejor resultado para el valor 4:

mtry	RMSE	Rsquared	MAE
2	0.1720358	0.4796429	0.0833136
3	0.1676384	0.4785617	0.0754045
4	0.1673003	0.4751445	0.0733164
5	0.1674326	0.4710268	0.0723851

Cuadro 3.9: Mtry RF-CI

Observamos que el coeficiente R^2 es 47.51 %, menor que el obtenido el el RF-CART por lo que decidimos seleccionar el modelo anterior para compararlo con la regresión

lineal múltiple. Al igual que en el caso anterior, comprobamos que la variable *v.ia.3m* resulta ser la de mayor importancia:

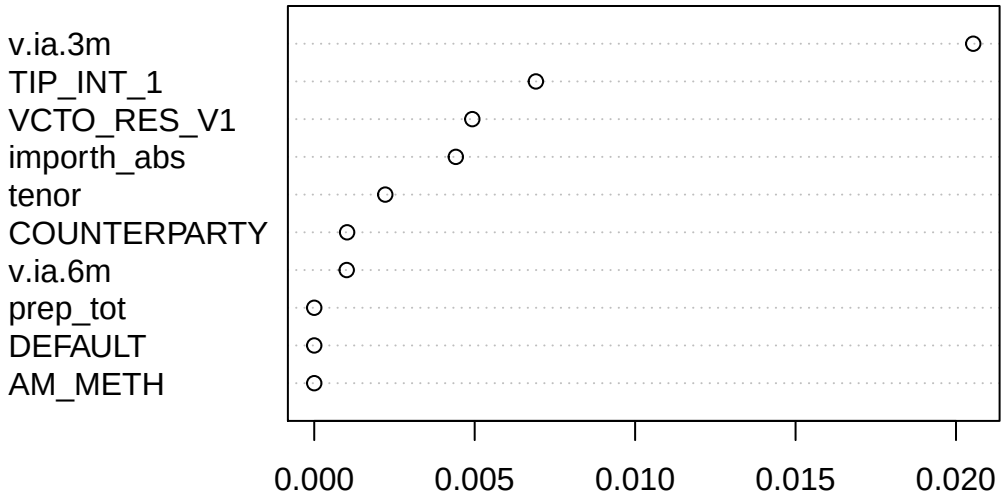

Figura 3.10: Importancia variables RF-CI

	IV
COUNTERPARTY	0.0009908
AM_METH	0.0000000
tenor	0.0022183
DEFAULT	0.0000000
prep_tot	0.0000000
VCTO_RES_V1	0.0049725
importh_abs	0.0044466
TIP_INT_1	0.0070204
v.ia.3m	0.0208668
v.ia.6m	0.0009810

Cuadro 3.10: IV RF-CI

Regresión lineal múltiple

Tras el análisis realizado para random forest, en esta sección vamos a comprobar la relación lineal que tienen los prepagos (*cpr_v*1) con el resto de variables para posteriormente poder comparar los resultados con el modelo anterior.

La tabla aporta la siguiente información:

- **Estimate:** Estimación coeficientes en la regresión.
- **Std. Error:** Desviación típica.

- **t value:** Estadístico que verifica si los coeficientes de la regresión significativos para el modelo.
- **Pr(>|t|):** Probabilidad de que la variable no sea relevante.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.1597	0.0569	2.81	0.0052
COUNTERPARTY	0.0903	0.0360	2.51	0.0123
AM_METH	0.4622	0.1187	3.89	0.0001
tenor	-0.0382	0.0161	-2.37	0.0181
VCTO_RES_V1	-0.0463	0.0141	-3.29	0.0011
importh_abs	-0.0695	0.0345	-2.01	0.0447
TIP_INT_1	-0.0805	0.0097	-8.34	0.0000
v.ia.3m	-1.2865	0.5381	-2.39	0.0171

Cuadro 3.11: Regresión lineal múltiple

Como se ha mencionado anteriormente, se ha utilizado una selección "forward" basada en el criterio de información de Akaike, donde el modelo reducido para el prepago es una combinación lineal de las variables seleccionadas con los coeficientes estimados incluyendo la estimación del término independiente. De esta manera, hemos reducido el modelo y aumentado el R-cuadrado (R^2 ajustado 23.44 %). A continuación, comprobaremos la bondad del ajuste, ¿es el modelo lineal adecuado para nuestros datos?

El coeficiente R^2 ajustado para el modelo reducido es 23.49 % (analizamos el ajustado para evitar que al añadir variables predictoras tengamos mejor modelo) por lo que la calidad del ajuste es bastante mala.

Procedemos a realizar un contraste general sobre el modelo:

La hipótesis nula es que el modelo está generado sólo por el término constante, es decir,

$$\begin{cases} H_0 : \beta_1 = \cdots = \beta_p = 0 \\ H_1 : \beta_i \neq 0 \text{ para algún i} \end{cases}$$

Se realiza un ANOVA donde el estadístico aparece en los resultados de la regresión como F-statistic (28.8) junto con su correspondiente p-valor (<2.2e-16). Como el p-valor es bajo se rechaza la hipótesis nula. Este contraste también se puede realizar con todas las variables individualmente donde si todos los p-valores asociados a los coeficientes son pequeños rechazamos la hipótesis de que sean nulos.

Además, observaremos algunos gráficos que pueden ayudarnos en la verificación de las hipótesis que debemos comprobar para la aplicabilidad de nuestro modelo:

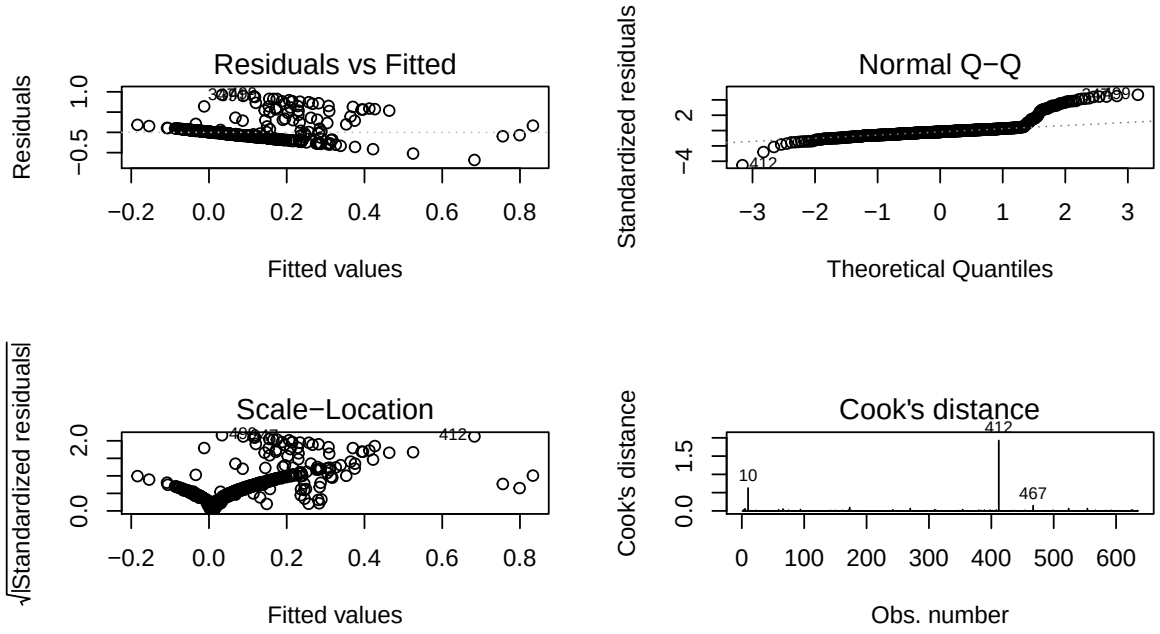

Figura 3.11: Residuos regresión multilineal.

- Para contrastar la normalidad de los residuos realizamos el test de Kolmogorov-Smirnov-Lilliefors obteniendo un p-valor pácticamente 0 (menor que 0,05) por lo que los residuos no se distribuyen según una distribución normal. Esto se puede comprobar también observando la gráfica Normal Q-Q, que es un diagrama de cuantiles, donde los residuos serían normales si discrepan poco de una línea recta. En este caso parece que ambas colas son más pesadas que las de una normal.
- En la gráfica de valores ajustados frente a residuos no aparece un patrón claro aunque ciertos residuos parece que conforman una línea recta, por lo que podría haber anomalías de aleatoriedad de los residuos. En esta gráfica también podemos discernir que la varianza no es constante.
- La última grafica representa la distancia de Cook, que es una medida invariante entre los coeficientes de regresión cuando la i-ésima observación está presente y ausente, midiendo así los datos influyentes. Otra manera de discernir dichos valores es mediante "Leverages (hat)", veámoslo gráficamente:

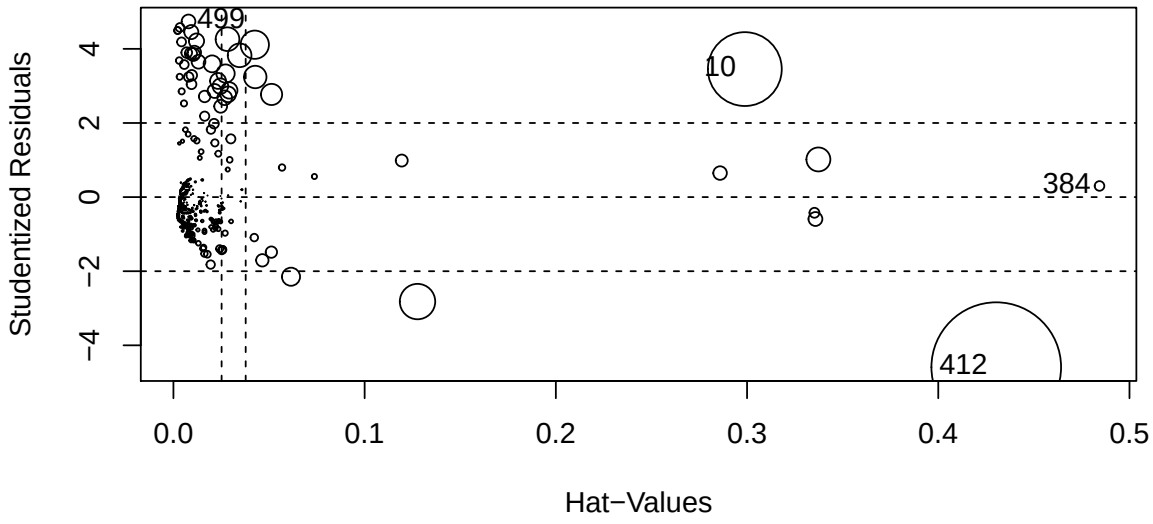

Figura 3.12: Influyentes regresión multilineal.

	StudRes	Hat	CookD
10	3.45	0.30	0.62
384	0.30	0.48	0.01
412	-4.59	0.43	1.93
499	4.74	0.01	0.02

Cuadro 3.12: Influyentes

Para buscar posibles variables influyentes analizamos:

- *Leverages (Hat)*: Se consideran observaciones influyentes aquellas cuyos valores hat superen 2.5((p+1)/n), siendo p el número de predictores y n el número de observaciones.
- *Distancia Cook (CookD)*: Se consideran influyentes valores superiores a 1, por lo que podríamos considerar muy influyente la observación 412.

- La media de los residuos es prácticamente 0.
- Para contrastar la homocedasticidad de los residuos realizamos el test de Breusch-Pagan. Obtenemos un p-valor cercano a 0 por lo que hay evidencias de falta de homocedasticidad.
- Para comprobar estacionariedad realizamos el test Dickey-Fuller obteniendo un p-valor = 0.01 y, por tanto, rechazando la hipótesis nula de no estacionariedad.
- Para el estudio de la multicolinealidad una de las medidas más utilizadas es el factor de inflación de varianza, VIF. Valores entre 5 y 10 indican posibles problemas y valores mayores o iguales a 10 se consideran muy problemáticos. En nuestro caso ninguna de las variables tienen una multicolinealidad preocupante, como esperábamos tras el análisis de correlaciones ya que las variables fuertemente correlacionadas han sido previamente eliminadas del modelo. La multicolinealidad

se presenta cuando las variables predictoras están altamente correlacionadas provocando que las estimaciones de los parámetros en los modelos lineales sean muy inestables.

	VIF
COUNTERPARTY	1.63
AM_METH	1.05
tenor	2.44
VCTO_RES_V1	2.41
importh_abs	1.23
TIP_INT_1	1.45
v.ia.3m	1.21

Cuadro 3.13: Multicolinealidad

Una vez analizadas todas las hipótesis se observa que el modelo presenta heterocedasticidad. Este problema podría tratarse viendo si es necesario una transformación logarítmica de los datos o utilizando un estimador robusto a heterocedasticidad, pero como la regresión lineal no es el objetivo del trabajo, se va a seguir con el modelo resultante como comparación de la importancia de las variables y procedemos a aplicar el modelo para realizar predicciones para los datos test para nuestra variable objetivo. Al tratarse de un elevado número de predicciones representamos una pequeña muestra en la siguiente tabla formada por los datos reales y los estimados de todas las que representamos en el gráfico:

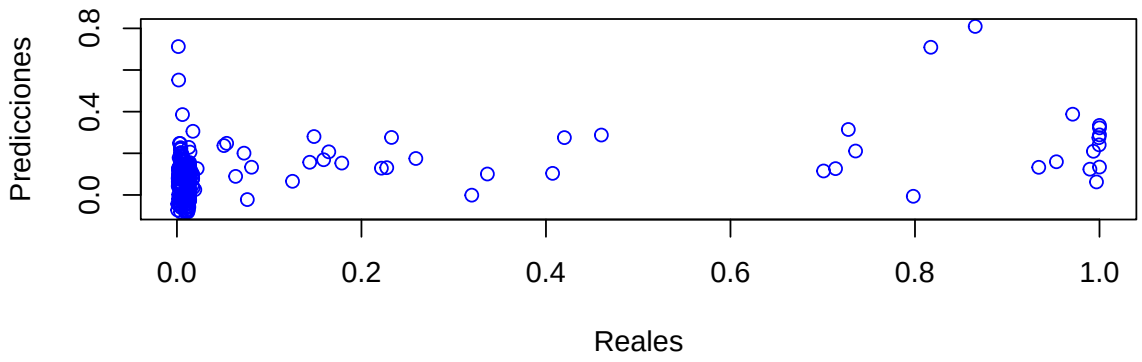

Figura 3.13: Predicciones regresión multilineal.

	Reales	Predicciones
1	0.00	-0.06
2	0.08	0.13
3	0.14	0.16
4	0.01	-0.03
5	0.41	0.10
6	0.01	0.03

Cuadro 3.14: Predicciones RL

Observamos que el coeficiente R^2 para las predicciones es 19.58 %, confirmando una vez más que este modelo es inadecuado para los datos que estamos considerando. Aún así, implementamos la medida de importancia de las variables enunciadas en los modelos teóricos para poder compararlas con aquellas que considere importantes el random forest:

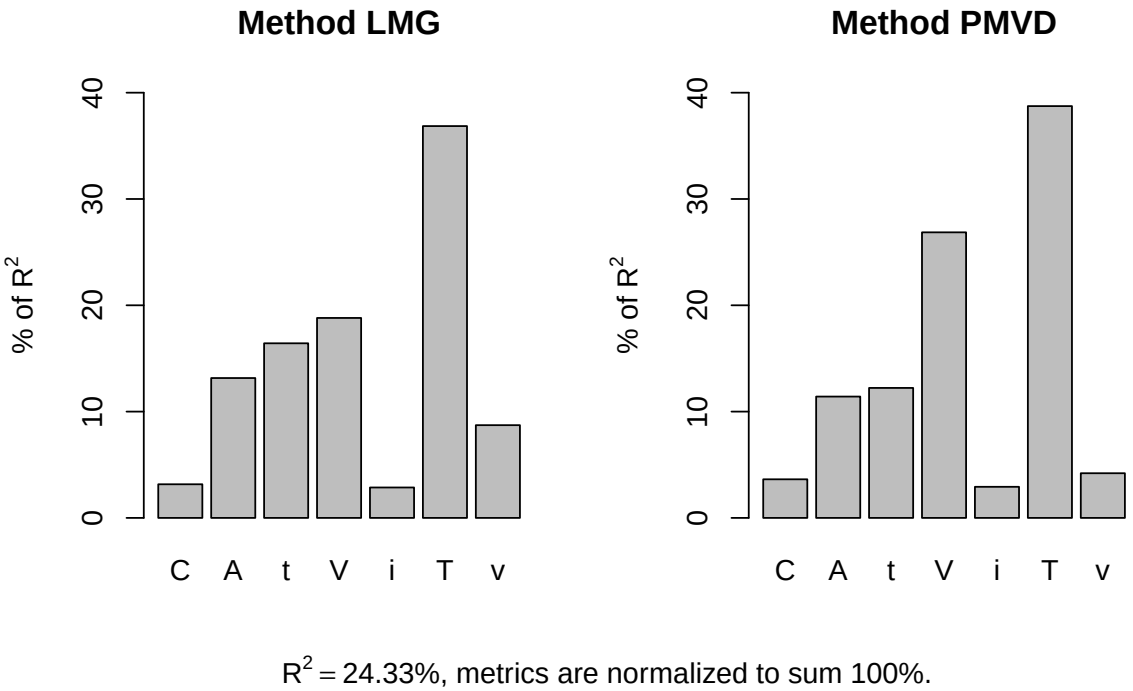

Figura 3.14: Importancia variables regresión multilineal.

Observamos que con ambas metodologías obtenemos el mismo orden decreciente de las variables según su importancia: TIP_INT_1, $VCTO\_RES\_V1$, *tenor*, AM_METH, *v.ia.3m*, $COUNTERPARTY$ e *importh_abs*. Resulta curioso que las dos con más importancia en random forest sean de las que menos tengan regresión lineal.

Comparación de errores

En esta sección realizaremos una comparativa entre los residuos generados por la regresión lineal múltiple y el RF-CART mediante el test no paramétrico Kolmogorov-Smirnov para dos muestras.

La hipótesis nula es que los residuos tienen una distribución idéntica. Como el p-valor es 0, no existen evidencias significativas para aceptar la hipótesis nula por lo que los residuos tienen distribuciones diferentes.

Contraste de modelo seleccionado en random forest

En esta sección vamos a comprobar cómo funcionaría la regresión lineal con las variables seleccionadas para random forest:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.1278	0.0119	10.74	0.0000
prep_tot	-0.0879	0.1025	-0.86	0.3914
VCTO_RES_V1	-0.0679	0.0099	-6.87	0.0000
importh_abs	-0.0978	0.0336	-2.91	0.0038
TIP_INT_1	-0.0753	0.0086	-8.74	0.0000
v.ia.3m	-1.3602	0.5321	-2.56	0.0108

Cuadro 3.15: Regresión lineal múltiple

Para las predicciones se obtuvo un R^2 de 13.59 %, empeorando de manera significativa los resultados de random forest. Esto es un indicador de que las variables seleccionadas en random forest recogen en gran medida la no linealidad de los datos y para incorporarlas a la regresión lineal haría falta incorporar interacciones entre las distintas variables ($X_1 * X_2$, $X_1 * log(X_2)$, etc), lo que queda para una investigación futura.

4 Conclusiones

En este trabajo se han implementado modelos de random forest para el análisis de comportamiento de clientes. Adicionalmente se ha analizado cómo obtener la importancia de las variables por distintos métodos y el uso de estos resultados para la selección de variables.

Para el análisis se ha utilizado como caso práctico una base de datos de clientes con datos de productos de crédito al consumo (datos simulados), con el objetivo de estudiar el prepago que estos pueden realizar sobre el esquema de amortización pactado. El análisis se ha centrado en determinar la cantidad de prepago una vez que se tienen identificados los clientes que prepagan.

El trabajo realizado muestra los siguientes resultados:

- El modelo random forest parece adecuado para analizar este tipo de comportamientos a pesar que las variables disponibles en la muestra son muy limitadas y, a priori, esperaríamos resultados mejores si se pudiesen incorporar variables tipo cliente (informan de las características del cliente: edad, lugar geográfico, ingresos...). De los dos algoritmos analizados el RF-CART presenta unos mejores resultados que el algoritmo RF-CI, a pesar de las limitaciones de RF-CART para trabajar con variables categóricas.
- El algoritmo random forest mejora visiblemente cuando se incorporaran variables sintéticas que recojan la evolución temporal. Al tratarse de un modelo de sección cruzada a priori no tiene en cuenta la dimensión temporal. En el modelo esta dimensión se ha incorporado creando variables sintéticas que recojan la evolución de los últimos meses del saldo pendiente.
- Los distintos modelos utilizados para analizar la importancia de las variables presentan resultados parecidos, siendo las variable más importante para determinar el prepago *v.ia.3m*. En dicha variable, se producen variaciones al realizarse prepagos por lo que tiene sentido que sea importante a la hora de intentar explicarlos.
- Se ha utilizado un modelo de regresión lineal múltiple como modelo de contraste presentando, como ya avanzábamos al analizar las hipótesis que debe cumplir el modelo, resultados muy inferiores a los algoritmos de random forest.
  La siguiente tabla muestra los resultados obtenidos por los distintos modelos considerados en términos de R^2 para la muestra de train y test.

	RL	RF-CART
R2train	23.49	61.21
R2test	19.58	64.11

Cuadro 4.1: Resultados

- Los modelos de random forest y de regresión lineal parecen estar considerando distintas iteraciones entre las variables. Este resultado viene confirmado por el análisis de residuos entre los dos modelos (en los que se aprecia que provienen distribuciones diferentes) así como por el análisis de la importancia de las variables ya que los dos modelos obtienen distintos subgrupos de variables relevantes.
- Como investigación futura queda mejorar el modelo de regresión lineal, considerando otro tipo de modelos (familia GLM) pero sobre todo analizando cómo tratar

de incorporar las variables relevantes encontradas en el random forest a través de la selección realizada, ya que la iteración entre éstas debe ser no lineal, y habría que estudiar qué combinación de dichas variables podría ser relevante ($X_1 * X_2$, $X_1 * log(X_2)$, etc). Adicionalmente, descubrir dichas iteraciones para conseguir que el modelo de regresión lineal se comporte de forma equivalente al RF permitiría explicar mediante "betas" el comportamiento de RF y mejorar así su interpretabilidad.

5

Anexo

Tests

Test Kolmogorov-Smirnov-Lilliefors

$$\begin{cases} H_0 : \text{La muestra procede de una distribución normal} \\ H_1 : \text{La muestra no procede de una distribución normal} \end{cases}$$

El estadístico de contraste habitual es:

$$D_n = sup_{-\text{ínf}<x<\text{ínf}} |S_n(x) - \phi(x - \bar{x})/S_x|,$$

donde ϕ es la distribución normal estándar.

Test Breusch-Pagan

$$\begin{cases} H_0 : \text{Existe homocedasticidad} \\ H_1 : \text{Existe heterocedasticidad} \end{cases}$$

El estadístico de contraste es:

$$\text{BP=SCRS/2} \sim \chi^2_p,$$

donde SCRS es la suma de los residuos cuadrados de la regresión. La regla de decisión será rechazar H_0 si el estadístico BP es mayor que el valor de las tablas: $BP > \chi^2_{p,\alpha}$.

Test Dickey-Fuller Aumentado

La hipótesis nula de este test indica la existencia de una raíz unitaria, es decir, que no haya estacionariedad.

$$\begin{cases} H_0 : \rho = 0 \\ H_1 : \rho < 0 \end{cases}$$

Este test incorpora tres tipos de modelos de regresión lineal. Representamos el que tiene deriva y tendencia lineal (los otros dos se corresponden al mismo sin tendencia lineal y el último sin tendencia lineal ni deriva):

$$dy_t = \alpha + \beta t + \rho y_{t-1} + \sum_j \phi_j dy_{t-j} + \epsilon_t$$

El estadístico se define como:

$$ADF = \frac{\hat{p}}{\sigma(\hat{p})}$$

donde $\hat{p}$ es la estimación del coeficiente.

Validación cruzada

Los algoritmos de machine learning como random forest están parametrizados para que se puedan adaptar de la mejor manera a un problema. La dificultad reside en que esos mejores parámetros para un problema en particular son desconocidos a priori.

El proceso de búsqueda de los parámetros óptimos es llamado "grid search". Esta técnica está basada en investigar de manera empírica con experimentos controlados. El proceso es el siguiente:

1. El usuario especifica los parámetros a probar.
2. El algoritmo es entrenado para todas las combinaciones de parámetros.
3. Se calculan medidas de rendimiento del modelo en cada prueba.
4. Los mejores parámetros son aquellos que proporcionan el mejor modelo.

El rendimiento del modelo mencionado anteriormente se calcula en una muestra que no ha sido usada para entrenar al modelo (para evitar sobreestimación). Para hacer esto, usaremos una técnica llamada validación cruzada mediante la que se reserva una muestra de los datos para "training" sobre la cual se valida el modelo.

Existen varias formas de implementar esta técnica pero nosotros usaremos "repeated k-fold cross validation":

Los datos de entrenamiento se dividen en k submuestras (cada submuestra es llamada fold) y el método es repetido k veces. Para cada vez, una de las k submuestras son usadas como datos de prueba y las otras k-1 submuestras se juntan para formar el "training set". Este proceso se podría repetir t veces y se computa el promedio del error para todas las k x t pruebas. La ventaja de este método es que es poco relevante cómo se dividen los datos. Cada dato llega a estar en un conjunto de prueba exactamente t veces (como mínimo una vez) y en un conjunto de entrenamiento (k-1) x t veces. La varianza de la estimación resultante se reduce a medida que k y t aumentan. La desventaja de este método es que el algoritmo de entrenamiento debe volver a ejecutarse desde cero k x t veces, lo que significa que se necesita k x t veces más tiempo para realizar una evaluación. Una variante de este método es dividir aleatoriamente los datos en una prueba y conjunto de entrenamiento k veces diferentes. La ventaja de hacer esto es que puede elegir de forma independiente el tamaño de cada conjunto de pruebas y con cuántas pruebas promedia más. En nuestro caso, se ha aplicado una 5-fold cross validation a los datos "training", por lo que k = 5 y t = 1. Veámoslo gráficamente:

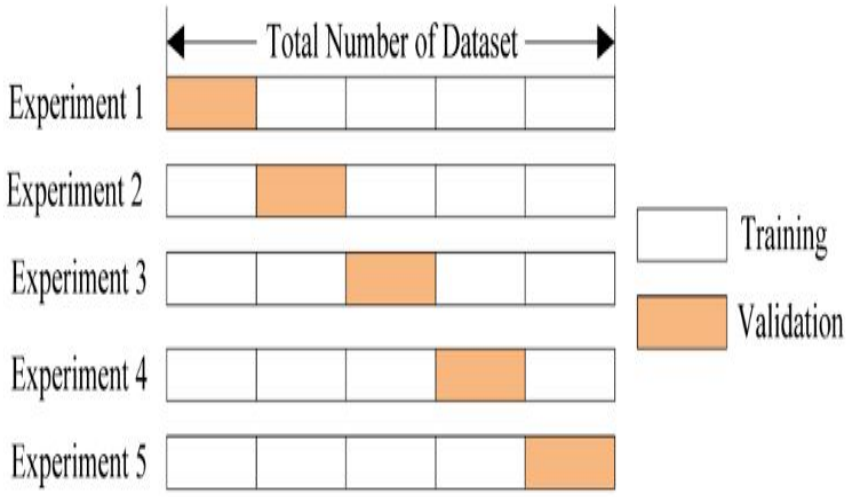

Figura 5.1: 5-fold cross validation

Valores atípicos

Para discernir los posibles valores atípicos graficamos mediante diagramas de cajas todas las variables numéricas y analizamos con más detenimiento aquellas que tengan valores muy alejados del primer y tercer cuartil:

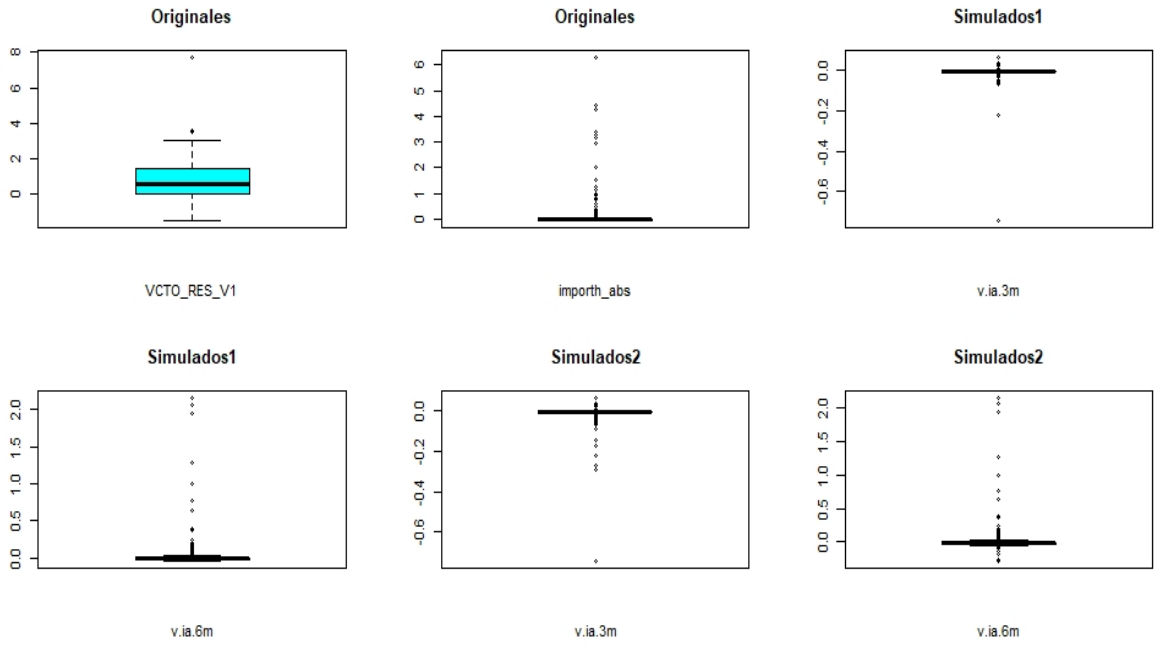

Figura 5.2: Datos originales

Observando estas seis variables considero que hay dos observaciones atípicas para la base de datos original, cinco en las variables simuladas 1 y cuatro en las variables simuladas 2:

- El mayor dato de $VCTO\_RES\_V1$ tiene un valor de 7.709 y además es el segundo mayor en la variable *importh_abs* con un valor de 4.41, ambos muy alejados de su media.
- El mayor valor de la variable *importh_abs*, 6.28.
- Para las variables simuladas 1 eliminamos los dos mayores datos de *v.ia.3m* (-0.74 y -0.219) y los tres mayores de *v.ia.6m* (2.15, 2.06 y 1.94).
- Para las variables simuladas 2 eliminamos el mayor valor de *v.ia.3m* (-0.74) y los tres mayores de *v.ia.6m* (2.15, 2.06 y 1.94).

Figuras

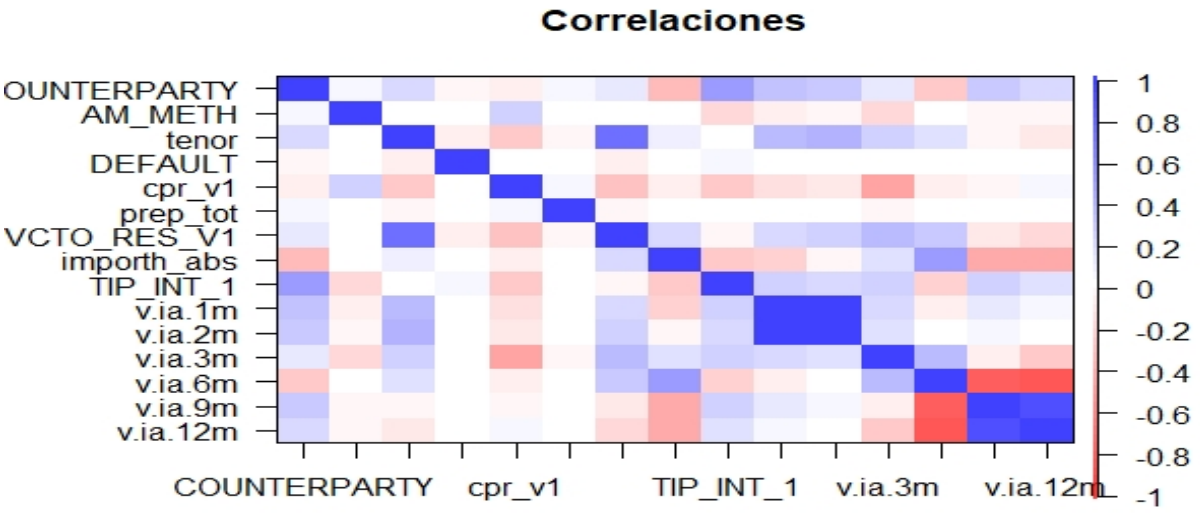

Figura 5.3: Correlaciones simuladas 1

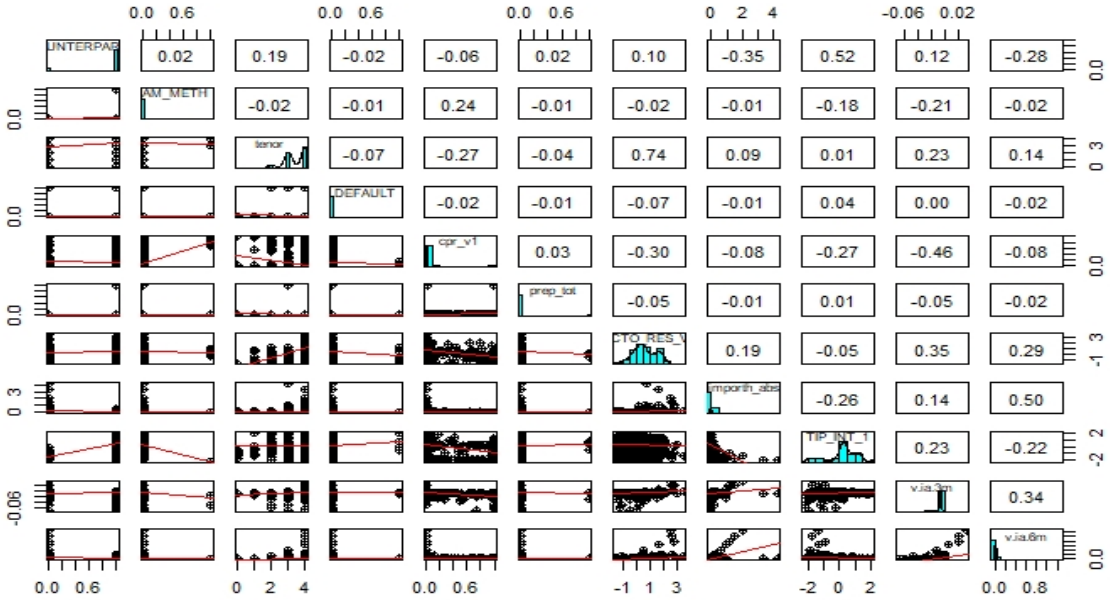

Figura 5.4: Scatterplot simuladas 1

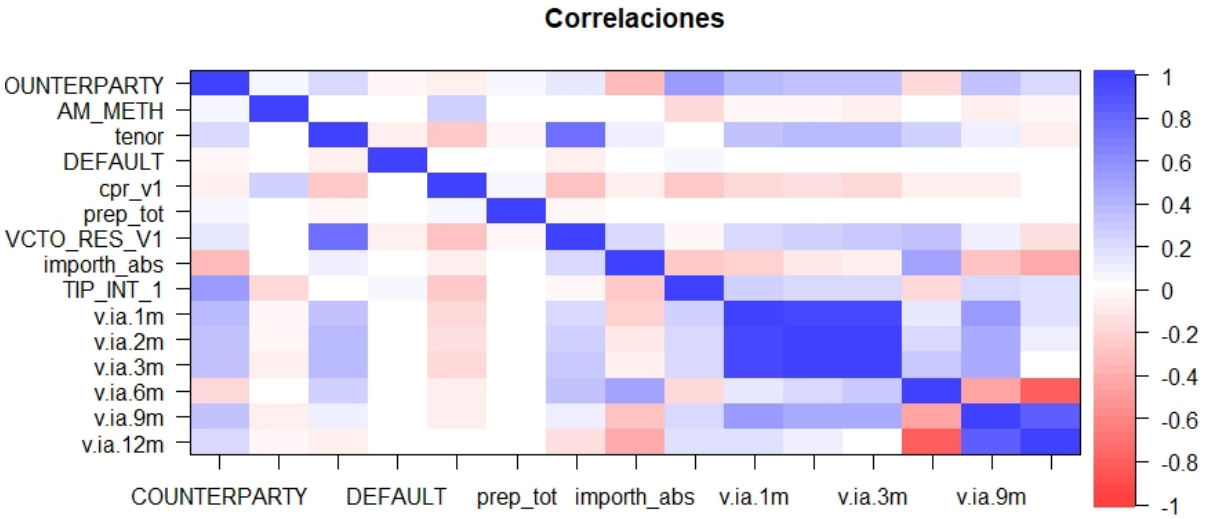

Figura 5.5: Correlaciones simuladas 2

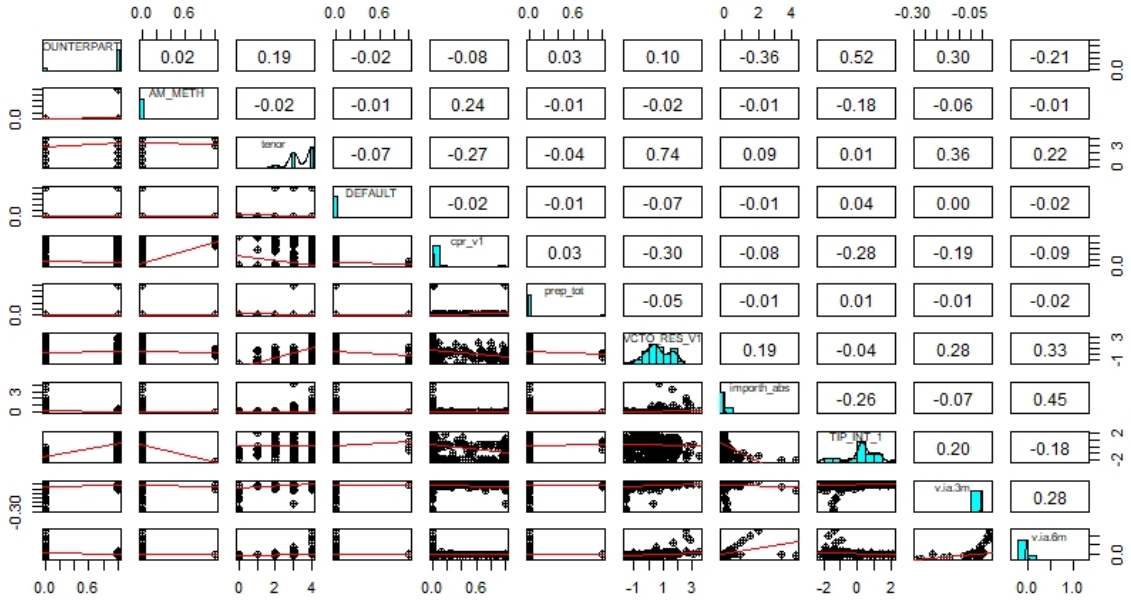

Figura 5.6: Scatterplot simuladas 2

Bibliografía

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees. belmont, ca: Wadsworth," *International Group*, p. 432, 1984.

[2] T. Hothorn, K. Hornik, and A. Zeileis, "Unbiased recursive partitioning: A conditional inference framework," *Journal of Computational and Graphical statistics*, vol. 15, no. 3, pp. 651–674, 2006.

[3] K. J. Archer and R. V. Kimes, "Empirical characterization of random forest variable importance measures," *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 2249–2260, 2008.

[4] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC bioinformatics*, vol. 8, no. 1, p. 25, 2007.

[5] M. R. Segal, J. D. Barbour, and R. M. Grant, "Relating hiv-1 sequence variation to replication capacity via trees and forests," *Statistical applications in genetics and molecular biology*, vol. 3, no. 1, pp. 1–18, 2004.

[6] R. H. Lindeman, P. Merenda, and R. Z. Gold, "Introduction to bivariate and multivariate analysis, glenview, il," *Scott: Foresman and company*, 1980.

[7] L. Breiman and A. Cutler, "Setting up, using, and understanding random forests v4. 0," *University of California, Department of Statistics*, 2003.

[8] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC bioinformatics*, vol. 9, no. 1, p. 307, 2008.

[9] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC bioinformatics*, vol. 7, no. 1, p. 3, 2006.

[10] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.

[11] U. Grömping, "Estimators of relative importance in linear regression based on variance decomposition," *The American Statistician*, vol. 61, no. 2, pp. 139–147, 2007.

[12] U. Grömping, "Variable importance assessment in regression: linear regression versus random forest," *The American Statistician*, vol. 63, no. 4, pp. 308–319, 2009.

[13] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Vsurf: an r package for variable selection using random forests," *The R Journal*, vol. 7, no. 2, pp. 19–33, 2015.

[14] B. M. Greenwell, B. C. Boehmke, and A. J. McCarthy, "A simple and effective model-based variable importance measure," *arXiv preprint arXiv:1805.04755*, 2018.

[15] "Random forest." https://cran.r-project.org/web/packages/randomForest/randomForest.pdf.

[16] "Party." https://cran.r-project.org/web/packages/party/party.pdf.

[17] "Vsurf." https://cran.r-project.org/web/packages/VSURF/VSURF.pdf.

[18] "Shiny." https://shiny.rstudio.com/.