Predicción de préstamos fallidos

Pablo Badenes Palomar

Facultad de ciencias sociales y jurídicas

Estadística Empresarial

4º Curso

Tutor: Mercedes Landete Ruíz

ÍNDICE

INTRODUCCIÓN

Empecé a trabajar en **InterMoney Titulización S.G.F.T., S.A.,** (IM) en marzo de este año, es una sociedad gestora de Fondos de Titulizacion. La titulización consiste en agrupación en una misma cartera de un conjunto de derechos de crédito similares que son cedidas por una Entidad (por ejemplo un banco) que financia el precio de compra colocando los títulos entre los inversores. (En 1997, el cantante y compositor David Bowie recibió 55 millones de dólares al vender bonos garantizados con ingresos futuros de su catálogo de álbumes; es decir titulizó sus canciones por 55 millones. Fue la primera titulización de los royalties de un artista)

A día de hoy, en IM se procesa y analiza más de un millón y medio de activos financieros de diferente tipología con un saldo superior a los 27 mil millones de Euros cada mes. Estos activos son la base de más de 40 Fondos.

IM como entidad experta en el mercado de titulización, ha sido asesora del Banco de España para la revisión de operaciones del programa de compra de bonos de titulización, del Banco Central Europeo (ABSPP) y de SAREB (Sociedad de Gestión de Activos procedentes de la Reestructuración Bancaria) para el estudio de alternativas de titulización de su cartera de activos.

En este trabajo, para mejorar la selección de los préstamos de una cartera, hemos desarrollado un método para predecir cuándo un préstamo va a entrar en fallido, es decir estar una cantidad de meses sin pagar su cuota (La cantidad de meses está definida por el propio fondo).

Hemos trabajado con préstamos al consumo, son el tipo de préstamo que te pueden dar para comprarte una nevera o un televisor; tienen una vida más corta que dificulta el análisis pero existe una muestra de datos más grande. Se dan más préstamos de este tipo que hipotecarios y para pymes juntos.

El modelo de predicción para saber si va a entrar en fallido un préstamo lo hemos desarrollado mediante la ejecución de un algoritmo de aprendizaje con nuestros datos.

El algoritmo de aprendizaje que hemos utilizado ha sido ***RandomForest***. Conocí el concepto del ***RandomForest*** estando en 3º curso, el nombre de la asignatura era

"Técnicas estadísticas en análisis de mercados", cuyo profesor era Xavi Barber. La asignatura la trabajamos completamente en ***Rstudio***, y se basaba en la modelización de problemas y sus respectivos análisis junto con la creación de informes con el objetivo de realizar la toma de decisiones más adecuada en cada caso.

La primera vez que se nombró el concepto de *Randomforest*, veníamos de estudiar regresión lineal y logística, y por tanto, fue una sorpresa ya que en lugar de estudiar las variables una a una y realizar comparaciones entre los modelos, generalmente usando ANOVA (es una de las técnicas más utilizadas en los análisis de los datos de los diseños experimentales y se utiliza cuando queremos contrastar más de dos medias), teníamos la posibilidad de introducir nuestro dataset y lanzar el modelo. Ahorrábamos más de 2 horas previas de análisis de los datos y no necesitábamos saber cuáles eran las variables que entraban en el modelo ya que este nuevo algoritmo lo había por nosotros.

Debido al poco tiempo que duran las asignaturas en el plan Bolonia, no nos dio tiempo a seguir profundizando en este nuevo método, y desde entonces, siempre me ha suscitado interés el seguir investigando.

Por otra parte, sabíamos que era un método novedoso dentro de los modelos predictivos y que alcanza una tasa de precisión mucho mayor que el resto, empezando a utilizarse dentro del campo de la investigación médica y las finanzas entre otros.

Respecto a la investigación que hemos realizado, cabe destacar dos líneas de investigación:

- **Cálculo de los parámetros de la función**: las funciones que vienen implementadas en Rstudio, generalmente, los parámetros vienen asignados por defecto. En la mayoría de las ocasiones esto nos reduce el tiempo que debemos invertir en realizar la programación adecuada, pero otras veces, como es en este caso, los parámetros óptimos deben de calcularse en función de los datos que partimos. Esta nueva línea que yo desconocía, ha sido una gran innovación, está toda documentada en el libro *An Introduction to Statistical Learning (*Vease bibliografía)
- **Explicación gráfica del Randomforest**: Los métodos basados únicamente en un árbol, son susceptibles de una representación gráfica sencilla. Este método

agrupa un conglomerado de árboles, por tanto, los gráficos necesarios para una explicación visual, son más complejo, además, utilizan otro tipo de medidas que no suelen ser habituales, y requieren una extensa explicación.

Esta otra línea de investigación, en mi opinión, la más interesante y novedosa procede de una tesis doctoral de Paluszyńska A (vease en bibliografia). Esta persona ha creado una nueva librería para Rstudio con diferentes herramientas para la creación ed gráficos exclusivos para los RandomForest, dichos gráficos son capaces de mostrar de una manera simplificada (a pesar de que son gráficos para gente con conocimientos estadísticos o matemáticos) los resultados obtenidos del modelo de predicción, y muestran como se comportan las diferentes variables y medidas utilizadas en función de las mismas o del número de bosques utilizados.

Una de las características más importantes del trabajo, es la novedad. Actualmente el RandomForest está a la cabeza de los algoritmos de aprendizaje automático, y por tanto, es un campo en continua investigación que está aportando unos niveles de precisión superiores a cualquier otro método y reduciendo la necesidad de un preprocesado largo de los datos.

Cabe añadir que estos métodos de predicción sirven para generar carteras de activos más sanas, de cara a que los Fondos contengan un menor riesgo. Además se pueden implantar en otras fases de gestión del fondo, como la corrección y chequeo de los activos.

MODELOS DE PREDICCIÓN

¿Qué es un modelo de predicción?

Cuando hablamos de modelo predictivo o modelo de predicción nos referimos a la relación del término fallido con el resto de variables del préstamo.

Este análisis predictivo está formado por un conglomerado de gestión, tecnologías de la información y modelado de datos, orientándolo para trabajar con grandes volúmenes y para contribuir al cálculo de riesgo de los préstamos en las entidades financieras.

Existen diversos tipos de métodos predictivos:

- **Modelos propiamente predictivos**: tienen como objetivo principal detectar nuevas oportunidades de negocio y factores críticos de riesgo, mediante el análisis de datos históricos y, sobre todo, actuales, a tiempo real, mientras se está realizando una determinada operación.

- **Modelos de descripción**: tienen como objetivo detectar la relación que puede existir entre servicios y/o productos, y el cliente (actual o potencial), estableciendo un vínculo de función entre ambos según los datos y las informaciones obtenidas sobre las preferencias del consumidor.

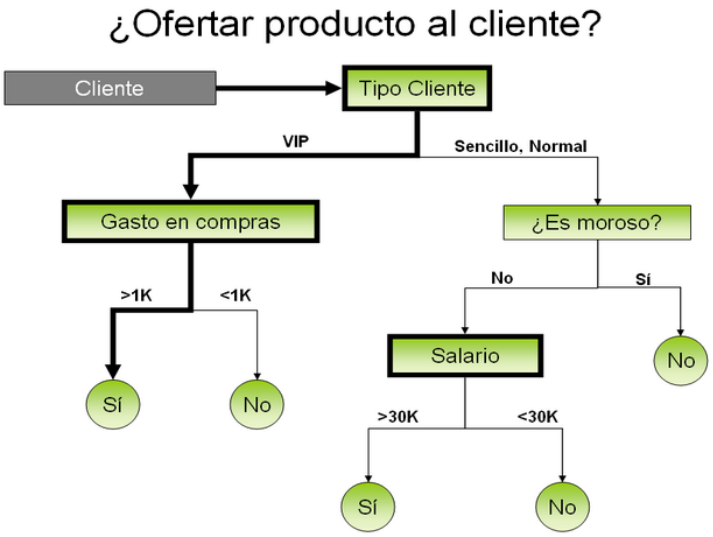
- **Modelos orientados a la toma de decisiones**: si los anteriores modelos nombrados están basados en una naturaleza técnica. El matiz de este tipo de modelo de análisis predictivo es eminentemente ejecutivo, es decir, toma las decisiones pertinentes en base a los resultados obtenidos tras llevar a cabo operaciones de analítica predictiva poniéndolos al servicio de los responsables en la toma de decisiones corporativas.

¿Qué son los métodos predictivos basados en árboles?

Los métodos estadísticos basados en árboles engloban a un conjunto de técnicas supervisadas no paramétricas que consiguen segmentar el espacio de los predictores en regiones simples, dentro de las cuales es más sencillo manejar las interacciones.

Los métodos basados en árboles los podemos dividir en dos tipos:

- **Árboles de regresión**: son un subtipo de árboles de predicción que utilizan como respuesta variables continuas. La interpretación del árbol se hace en sentido descendente, y por tanto la variables más importante es la que se sitúa en la parte de arriba del árbol. Queda patente que la principal ventaja de los árboles de decisión frente a otros métodos de regresión es su fácil interpretación y la gran utilidad de su representación gráfica.



- **Árboles de clasificación**: son otro subtipo de árboles, muy parecidos a los de regresión con la diferencia que la variable respuesta es cualitativa.

CONSTRUCCIÓN DE LOS ÁRBOLES

El proceso de construcción de los árboles predictivos se divide en dos etapas:

- **División sucesiva del espacio de los predictores** generando regiones no solapantes (nodos terminales) $R1$, $R2$, $R3$, …, Rj. Aunque, desde el punto de vista teórico las regiones podrían tener cualquier forma, si se limitan a regiones rectangulares (de múltiples dimensiones), se simplifica en gran medida el proceso de construcción y se facilita la interpretación.
- **Predicción de la variable respuesta en cada región**.

La metodología que se encarga de decidir donde se introducen las divisiones que permita crear las regiones $R1$, $R2$, $R3$, …, Rj es donde se diferencian los algoritmos de árboles de regresión y clasificación.

VENTAJAS Y DESVENTAJAS DE LOS ÁRBOLES

Ventajas

1. Tienen una estructura fácil de comprender e interpretar, aun cuando la relación entre las variables predictoras es compleja.

2. Los modelos de regresión basados en árboles de decisión son susceptibles de representación gráfica.
3. Los árboles pueden contener y analizar tanto predictores cuantitativos como cualitativos sin tener que recurrir a las variables *dummy* (son variables que indican la presencia o ausencia de una cualidad o atributo, estas pueden tomar valor 1 ó 0).
4. Los datos no deben seguir ningún tipo de distribución, ya que los árboles están basados en métodos no paramétricos.
5. El procesado de los datos antes del análisis requiere menos trabajo comparado con otros métodos de aprendizaje estadístico.
6. Los valores extremos de la muestra (*outliers*) no afectan a su estructura ya que la media no es un parámetro que se use para su creación.
7. Si el valor de un predictor no está disponible para alguna observación, a pesar de no poder llegar a ningún nodo terminal, se puede conseguir una predicción empleando todas las observaciones que pertenecen al último nodo alcanzado. En este caso la precisión de la predicción se verá reducida pero al menos podrá obtenerse.
8. son realmente eficientes a la hora de analizar una gran cantidad de variables, ya que permiten identificar de forma rápida y eficiente las variables más importantes sin un pre procesado de los datos.
9. Son capaces de seleccionar predictores de forma automática.

Desventajas

1. Los modelos de regresión y clasificación basados en un único árbol tienen una menor capacidad predictiva que otros modelos que usan múltiples árboles debido a su tendencia al *overfitting* (que muestra la facilidad con la que los árboles se ramifican adquiriendo estructuras complejas para explicar la muestra actual y por ello la predicción es inferior cuando introducimos nuevos datos). Por ello se han creado nuevas técnicas de combinación de múltiples árboles para reducir este fenómeno, en este caso, vamos a trabajar con *RandomForest*.

2. Las variables continuas son categorizadas para poder realizar la división de los nodos y por tanto, pierden parte de su información. Por ello los resultados obtenidos en clasificación son superiores a los obtenidos en regresión.
3. La creación de la estructura interna (ramificación) está basada en el algoritmo de *recursive binary splitting*. Dicho algoritmo evalúa las posibles divisiones de los predictores usando como referencia una determinada medida (*Residual Sum of Squares (RSS)*, Gini, entropía…). Los predictores continuos o predictores cualitativos con muchos niveles tienen mayor probabilidad de contener, solo por azar, algún punto de corte óptimo, por lo que suelen verse favorecidos en la creación de los árboles.

ALGORITMOS DE APRENDIZAJE

¿Qué es un algoritmo de aprendizaje?

Denominamos algoritmo a un conjunto ordenado y finito de operaciones simples con el objetivo de hallar una posible solución a un problema.

"Esta palabra, como tal, proviene del latín tardío alborarismus, y este a su vez es una abreviación del árabe clásico ḥisābu lgubār, que significa 'cálculo mediante cifras arábigas".

Los algoritmos permiten ejecutar una acción o resolver un problema mediante una serie de pasos o instrucciones definidas, ordenadas y finitas. Así, Partiendo de un estado inicial y siguiendo los sucesivos pasos indicados, se llega al estado final y se obtiene una solución.

Algoritmo de árboles de regresión

Existen dos criterios frecuentes en los árboles de regresión para identificar y calcular las posibles divisiones:

- ***Residual Sum of Squares (RSS)***. El objetivo de este método es encontrar las *J* regiones (*R*1,…, *Rj*) que minimizan el *Residual Sum of Squares* (RSS) total:

$$RSS = \sum_{j=1}^{J} \sum_{I \epsilon R_j} (y_i - \hat{y}_{Rj})^2$$

Donde $\hat{y}_{Rj}$ es la media de la variable respuesta en la región Rj. Este método busca la máxima reducción posible del sumatorio de las desviaciones al cuadrado dividido entre las observaciones y la media de la clasificación a la cual pertenecen. El mayor problema de este método es que es imposible hacer todas las combinaciones de partición de los predictores.

- ***Recursive binary splitting***: este método soluciona este último problema ya que no evalúa todas las regiones posibles. El objetivo de este algoritmo es encontrar predictor Xj y el punto de corte (umbral) s en cada iteración para reducir al máximo posible el RSS. La secuencia de pasos que sigue es la siguiente:

1. El proceso se inicia en lo más alto del árbol, donde todas las observaciones pertenecen a la misma región.

2. Se determinan los posibles puntos de corte s para cada uno de los predictores ($X1$, $X2$,…, Xp). Distinguimos dos puntos de corte diferente según la naturaleza de la variable:
   - Predictores cualitativos: los posibles puntos de corte son cada uno de sus niveles.
   - Predictores continuos, se usa como punto de corte el valor intermedio entre cada par de valores y se ordenan ascendentemente

3. Se calcula el RSS total, donde el primer término viene determinado por el RSS de la primera región y el segundo corresponde al RSS de la segunda región, que se consigue con cada posible división identificada en el paso anterior.

$$\sum_{i:x_i \varepsilon R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \varepsilon R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

4. Se selecciona el predictor Xj y el punto de corte S que resulta en el menor RSS total, ya que este resultado es el que produce divisiones más homogéneas. Si existen dos o más divisiones que consiguen la misma mejora, la elección se realiza de manera aleatoria.

5. Se repiten de forma iterativa los pasos 1 al 4 para cada una de las regiones que se han creado en la iteración anterior hasta que se alcanza alguna norma de parada. Entre ellas destacan:
    - Que ninguna región contenga un mínimo de n observaciones
    - Que el árbol tenga un máximo de nodos terminales
    - Que la incorporación del nodo reduzca el error en al menos un % mínimo.

Predicción en arboles de regresión

Los nodos contienen las observaciones de entrenamiento agrupadas, una vez creado el árbol. Para predecir una nueva observación, se recorre el árbol en función de los valores que tienen sus predictores hasta llegar a uno de los nodos terminales. En el caso de regresión, el valor predicho suele ser la media de la variable respuesta de las observaciones de entrenamiento que están en ese mismo nodo. Entre los valores más empleados para realizar las divisiones se encuentran la media, mediana, moda y cuantil.

Algoritmo de árboles de clasificación

El método que se usa para la construcción de los árboles de clasificación es el *recursive binary splitting*, al igual que en los árboles de regresión. No obstante, hay una clara diferencia, ya que la variable respuesta de este tipo de árbol es cualitativa y por tanto no se puede usar el RSS para realizar las divisiones óptimas de los predictores.

Existen diferentes métodos para poder obtener los nodos más homogéneos posibles, entre ellos cabe destacar:

- ***Classification error rate***: su fórmula viene dada por la proporción de observaciones que no pertenecen a la clase más habitual dentro del nodo.

$$E_m = 1 - max_k(\hat{p}_{mk})$$

- ***Gini index*:** calcula la varianza total en el conjunto total de las k clases del nodo m. Su objetivo es medir la pureza del nodo. Cuanto mayor sea la pureza del nodo, el valor del índice es menor

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- ***Information gain (cross entropy)*:** es una medida que se utiliza para cuantificar el desorden del sistema, en este caso, el desorden corresponde con la impureza de los nodos. Si un nodo contiene únicamente observaciones de una clase, es decir, es puro, su entropía es cero. Por otra parte, si la frecuencia de cada una de las clases es la misma, la entropía adquiere su valor máximo, 1

$$D = -\sum_{k=1}^{K} \hat{p}_{mk}\ log(\hat{p}_{mk})$$

- ***Chi-square*:** esta fórmula se usa para evaluar si existe una diferencia significativa entre los nodos hijos y el nodo parental. Cuanto mayor sea el estadístico x^2, mayor evidencia de una diferencia.

$$X^2 = \sum_{k} \frac{(observado_k - esperado_k)^2}{esperado_k}$$

Predicción en árboles de clasificación

El proceso de predicción en los árboles de clasificación es el mismo que se usa en los de regresión, pero con una pequeña variación. La moda es la variable que se usa como valor de predicción, es decir, la clase más frecuente del nodo. Además suele acompañarse con un porcentaje de cada clase en el nodo terminal aportando información sobre la confianza de la predicción.

Overfitting

En el proceso de construcción de los árboles, que hemos mostrado en apartados anteriores, se tiende a reducir muy rápidamente el error de entrenamiento, esto significa que el modelo creado se ajusta muy bien a las observaciones de entrenamiento. Este proceso crea un comportamiento en el cual los árboles se ramifican con cierta facilidad creando estructuras complejas (*overfitting*). De hecho, si no se impusieran límites en las

divisiones se crearía un nodo terminal por observación, creando un modelo que se ajustase perfectamente a las observaciones. Existen dos caminos para prevenir el efecto del *overfitting* de los árboles:

- **Controlar el tamaño del árbol:** Existen reglas de parada para que el tamaño de un árbol pueda controlarse deteniendo la división de los nodos en función de si cumplen o no determinadas condiciones. Entre ellas, las más habituales son:

  -Observaciones mínimas para cada división: marca el número mínimo de observaciones que debe albergar los nodos terminales.

  -Observaciones mínimas de nodo terminal: limita el número mínimo de observaciones que pueden contener los nodos terminales.

  -Profundidad máxima: define la profundidad máxima del árbol, es decir, e número de divisiones de la rama más extensa del árbol.

  -Número máximo de nodos terminales: hace referencia al número máximo de nodos terminales que puede tener el árbol, y una vez superado, las divisiones se detienen.

  -Reducción mínima de error: establece la reducción mínima de error que tiene que obtener una división para que se realice dicha división.

- **TREE PRUNING**

  El procedimiento de ramificación se basa en seleccionar la división idónea en cada momento hasta llegar a una condición de parada. Este procedimiento tiene un pequeño problema, no tiene en cuenta las divisiones posteriores, es decir, no se elige la opción que resulta en el mejor árbol final, a no ser que esa división sea la misma que se genera en ese momento. A este tipo de estrategia se las conoce como *greedy*.

  Una alternativa a esta técnica que consigue evitar el *overfitting* consiste en generar árboles grande, utilizando como condiciones de parada únicamente las

computacionales, para después podarlos (*pruning*) con el objetivo de mantener la estructura principal que consigue un *test de error bajo*

$$\sum_{j=1}^{|T|} \sum_{i \varepsilon R_j} (y_i - \hat{y}_{R_j})^2 + \alpha |T|$$

ÁRBOLES VS MODELOS LINEALES

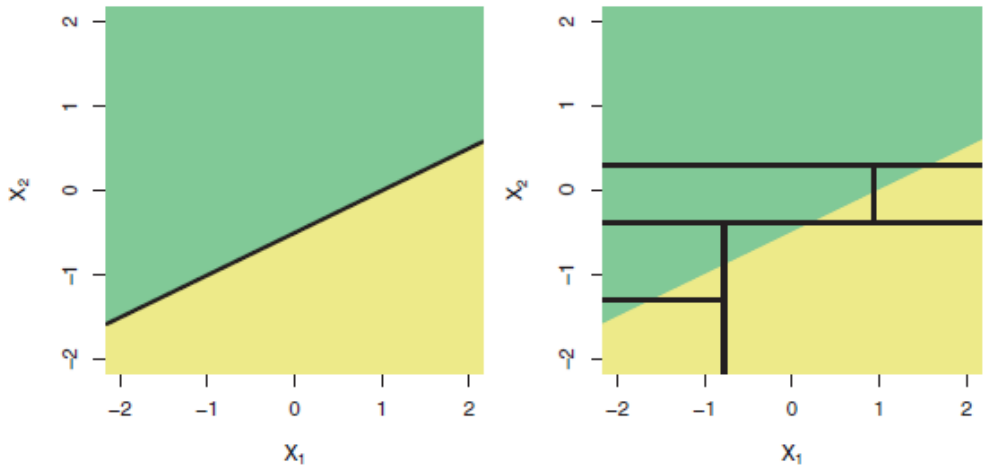
El método seleccionado para la creación de un modelo, depende del propio problema. Es decir, cuando la variable respuesta está relaciona de una manera lineal o aproximadamente lineal, los modelos de regresión lineal son superiores.

Por otra parte, si la relación entre la variable respuesta y los predictores es no lineal y compleja, los métodos predictivos basados en árboles suelen superar a las aproximaciones lineales.

Como se puede observar en los gráficos, mientras que un modelo lineal (gráfico izquierdo) la relación entre los predictores tiene una forma lineal, los modelos basados en árboles permiten captar la relación que hay entre ellos aunque no sean lineales.

En el gráfico situado a la izquierda, se observa como la relación entre las x es linealmente dependiente, ya que conforme aumenta el valor de la x en el eje horizontal, aumenta de la misa manera el valor de la x en el eje vertical.

Por contraposición, en el gráfico de la derecha, se observa que las dos variables no son linealmente dependientes ya que los valores de ambas variables no aumentan o disminuyen según lo haga la otra variable.



MÉTODOS DE ENSAMBLADO

Este tipo de métodos están basados en un conjunto de modelos que se usan juntos como un metamodelo (*es el análisis, la construcción y el desarrollo de los marcos, reglas, restricciones, modelos y teorías aplicables y útiles para el modelado de una clase predefinida de problema*), ya que usan conocimientos de distintas fuentes o modelos para tomar las decisiones.

Los árboles de predicción tienen los mismos problemas que cualquier modelo estadístico, uno de los más importantes es el equilibrio *bias-varianza*. La suma del *bias* (desviación) y la varianza nos muestran el error de clasificación obtenido.

El concepto de *bias* hace referencia al desvío calculado con respecto al error verdadero. Un alto valor en este parámetro indica que la distribución de los datos no se ajusta adecuadamente y no depende del número de muestras de entrenamiento.

Varianza es la variabilidad que presenta el clasificador para diferentes muestras. A mayor número de muestras de entrenamiento la varianza disminuye al contrario que un alto valor en la varianza nos muestra un sobre-ajuste.

Por tanto, los métodos de ensamble contienen un conjunto de métodos que combinan diversos modelos predictivos para lograr el mejor equilibro entre el bias y la varianza. Entre estos métodos destacan:

- ***Boosting***: es un proceso que consiste en ajustar secuencialmente modelos sencillos, que reciben el nombre de *weak learners* con el objetivo de que cada modelo aprenda de los errores de su predecesor.
  Como valor final se toma la media de todas las predicciones o la clase más frecuente. Dentro de las técnicas más conocidas de *boosting*, destacan *Stochastic Gradient Boosting, Adaboost y Gradient Boosting*.

- ***Bagging***: dicho término es el diminutivo de *bootstrap aggregation* basado en reducir la varianza mediante el muestreo repetido. Se obtienen múltiples muestras de la población, ajustando un modelo diferente con cada una de ellas para realizar la media (se usa la moda en el caso de las variables cualitativas) de las predicciones obtenidas.

Generalmente no se puede tener acceso a diferentes muestras de la población, por tanto, un recurso muy usado es generar pseudo-muestras, con los que ajustar diferentes modelos y después agregarlos. Este proceso es conocido como *bagging*. Este método ha demostrado incrementar de una manera notable la precisión de las predicciones. Los pasos que sigue son:

1. Generar β *pseudo_training* sets mediante *bootstrapping* a partir de la muestra original.

2. Realizar un entrenamiento con un árbol con cada una de las β obtenidas en el paso anterior. El objetivo es crear árboles con restricciones solamente computacionales y sin ser sometidos a *pruning*, con lo que obtenemos una varianza alta pero poco bias.

3. Obtener la predicción de todos los β árboles para cada nueva observación. El valor total de esta predicción se calcula usando la media de las β predicciones cuando se usan variables cuantitativas, mientras que para las variables cualitativas se usa la moda (la clase predicha más frecuente).

   En este proceso, la cantidad de árboles creados no supone un problema ya que por mucho que se incremente el número, el riesgo de *overfitting* no aumenta. Por tanto, cuando se alcanza un número determinado de árboles la reducción del *test error* se estabiliza. No obstante, hay que tener en cuenta la limitación computacional y por tanto, no conviene almacenar más arboles de los necesarios

Out-of-bag-error

El *out-of-Bag-Error* viene dado por el tercio de árboles que no se usa en los ajustes originales, ya que cada ajuste usa aproximadamente dos tercios de las observaciones originales. Para cada árbol ajustado usando el procedimiento de *bagging* se registran las observaciones empleadas, es posible predecir la respuesta de la observación i utilizando

aquellos árboles en los cuales dicha observación ha sido excluida y realizando el promedio.

Importancia de los predictores

El objetivo del *bagging* consiste en mejorar la capacidad predictiva del modelo en comparación a los modelos que están basados en un solo árbol. Pero este proceso, tiene un coste asociado, la interpretabilidad del modelo se ve reducida, ya que se trata de una combinación múltiple de árboles y por tanto su representación gráfica y la identificación de los predictores más importantes, es más complicada. Por otra parte, al contar con múltiples árboles, aparecen muevas estrategias para mejorar dichos problemas, entre ellas destaca:

- Incremento del MSE: Analiza la influencia de cada predictor sobre el error cuadrático medio del modelo (que se estima por el *out-of-bag-error).* El cálculo para cada predictor se realiza de la siguiente manera:

    1. Se crea el conjunto de árboles para formar el modelo de *bagging.*

    2. Se calcula el *out-of-bag* MSE del modelo.

    3. En cada predictor j:

        a. Intercambiar los valores del predictor en todos los árboles, dejando el resto constante.
        b. Recalcular el *out-of-bag* MSE tras el intercambio
        c. Obtener el incremento en el MSE causado por el intercambio del predictor j.

- Incremento de la pureza de los nodos: Para calcular dicho incremento se registra el descenso obtenido en la medida utilizada como criterio de división, que puede ser índice de Gini, MSE o entropía, en cada división de los árboles. Se calcula el decremento medio obtenido en el conjunto de árboles que forman el *ensemble,* y cuanto mayor sea dicho valor, mayor será la importancia de dicho predictor en el modelo.

RANDOMFOREST

Es una técnica de agregación desarrollada por Leo Breiman (1928-2005), que incrementa la precisión de la clasificación mediante la introducción de aleatoriedad en la construcción de cada clasificador individual.

Utiliza un algoritmo predictivo basado en la técnica de *Bagging*, la cual combina diferentes árboles, donde cada árbol es construido con observaciones y variables aleatorias. Es una variación de dicha técnica, con la diferencia que decorrelaciona los árboles que construye en el proceso, esta modificación consigue que al realizar el promedio de un conjunto de modelos, se consiga reducir la varianza. Ya que si la correlación entre los modelos es alta, la reducción de varianza que se puede conseguir es poco significativa.

El *Random forest* y el *bagging*, se basan en el mismo algoritmo con la diferencia que en el primero, se seleccionan aleatoriamente m predictores antes de realizar cada división. Por tanto, la diferencia en el resultado obtenido dependerá del valor del parámetro *m* escogido. Los resultados del bagging y random forest serán equivalentes si el valor m=p (siendo p el número de predictores totales). Sin embargo, el valor óptimo del parámetro *m* es aquel que minimice el *out-og-bag-MSE* para los diferentes valor de m, si los predictores están muy correlacionados, pequeños valores de m consiguen unos mejores resultados.

Al igual que con la técnica del *bagging*, en el *random forest* no se sufre problemas de *overfit* por aumenta el número de árboles generados en el proceso. Al alcanzar un número determinado, la reducción del error se estabiliza. Algunas de las características más importantes son:

- Es una técnica recomendada para grandes bases de datos
- Tiene la capacidad de gestionar miles de variables simultáneamente (siempre y cuando se tenga capacidad computacional)
- Tiene una técnica eficaz para la estimación de los registros que faltan y mantiene la precisión cuando una gran parte falta, a pesar de reducir la precisión.

- Permite estimar cuales son las variables más importantes en la clasificación.
- A medida que avanza la construcción del bosque, se calcula una estimación interna de la generalización del error.
- Es uno de los algoritmos de aprendizaje automático más precisos en la actualidad.

No obstante este algoritmo cuenta con una serie de desventajas:

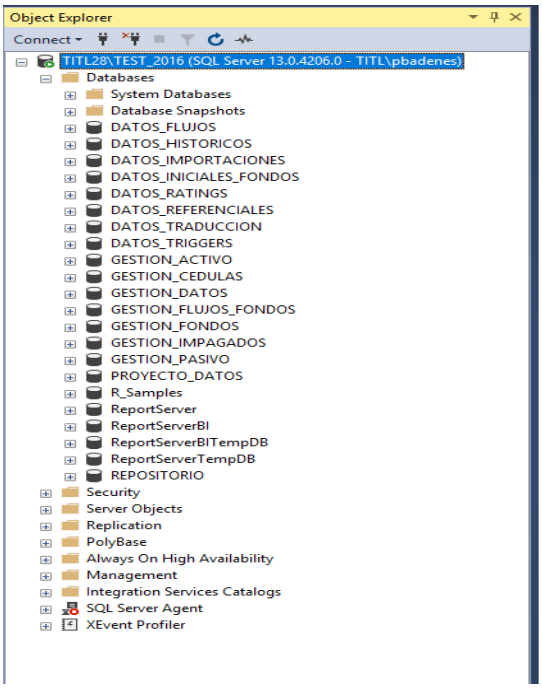
- La clasificación producida por *random forests* es difícil de interpretar y de mostrar gráficamente.
- Existen casos en los que puede sobre ajustar en ciertos sets de datos con tareas de clasificación/regresión ruidosas.

Si en los datos existen grupos de atributos que están correlacionados y relevancia similar para el rendimiento, entonces los grupos más pequeños estarán favorecidos sobre los grupos más grandes.

MI PROYECTO

¿De dónde parto?

Actualmente estoy trabajando en una empresa financiera dedicada a la gestión de fondos de titulización. La gran cantidad de datos y registros necesitan de una estructura basada en BBDD relacionales. La información de la empresa está almacenada en servidores de Microsoft SQL y se usará Rstudio como software de procesado y análisis de los datos mediante la creación de una conexión al servidor SQL (ODBC). ¿Qué información inicial tengo? Como se observa en las siguientes imágenes, la empresa cuenta con una varias BBDD muy extensas, algunas de ellas con más de 54 millones de registros. Para este trabajo, hemos seleccionado una muestra de 1.000 registros y 20 variables con el objetivo de probar la viabilidad de este método para más tarde proceder a la limpieza una a una de las bases y poder parametrizar el algoritmo en función a que base queramos aplicar este análisis.



Una parte de los datos que contienen las BBDD, han sido volcadas a mi máquina, con el objetivo de no obstaculizar las labores diarias que se realizan en los servidores de SQL y evitar cualquier problema de borrado o actualizado de los datos, ya que podría provocar daños muy severos en la empresa.

Análisis

Método

Después de analizar las posibles variantes de los árboles de clasificación, nos encontramos con el *Randomforest*, es uno de los métodos más eficaces para aplicarlos a grandes bases de datos con una extensa cantidad de variables.

Conexión a la BBDD

El primer paso en nuestro análisis será realizar la conexión a la BBDD mediante un *chunk* con código SQL. Una vez realizado este paso, tendremos acceso a toda la BBDD para poder extraer los datos necesarios para el análisis.

```
con <- dbConnect(odbc::odbc(), "LocalTilt28")
```

Una vez creada dicha conexión, se construirá una sentencia en lenguaje SQL que acceda a la BBDD y me proporcione los datos en formato *tibble (explicalo)* con los que poder trabajar de una manera sencilla en ***Rstudio***.

```
SELECT [checking_balance] as Saldo
      ,[months_loan_duration] as Duracion_prestamo
      ,[credit_history] as Historial_crediticio
      ,[purpose] as Finalidad
      ,[amount] as Cuantia_prestamo
      ,[savings_balance] as Ahorro
      ,[employment_length] as Duracion_empleo
      ,[installment_rate] as Tipo_interes
      ,[personal_status] as Estado_civil
      ,[other_debtors] as Otros_deudores
      ,[residence_history] as Historial_residencia
      ,[property] as Propiedad
      ,[age] as Edad
      ,[installment_plan] as Cuota
      ,[housing] as Vivienda
      ,[existing_credits] as Creditos_existentes
      ,[default] as Fallido
      ,[dependents] as Hijos

      ,[foreign_worker] as Extranjero
      ,[job] as Tipo_empleo
  FROM [dbo].[datos_prestamos]
```

Procesado de los datos

Una vez obtenida nuestra *tibble (*es la evolución moderna de un dataframe, es una tabla dividida en filas y columnas con unas determinadas características que sirven para facilitar el trabajo con los datos que contiene*)* almacenada en una variable, se procederá a la conversión de las variables en los formatos deseados. La mayoría de los registros que se encuentran en nuestras bases de datos están en formato carácter para poder facilitar la lectura y reducir el consumo del servidor a la hora de realizar diferentes procedimientos.

Por tanto, lo primero que se debe de hacer es la conversión de tipo carácter a factor. Uno de los paquetes preferidos por los científicos de datos que usan Rstudio como herramienta principal es la librería "*tidyverse*". Esta librería está creada por el científico de datos jefe de R y por tanto nos asegura una buena implementación.

```
datosc<-mutate(datosc,
          Saldo=parse_factor(Saldo, levels=NULL),
          Duracion_prestamo=parse_integer(Duracion_prestamo),
          Historial_crediticio=parse_factor(Historial_crediticio, levels= NULL),
          Finalidad=parse_factor(Finalidad, levels= NULL),
          Cuantia_prestamo=parse_double(Cuantia_prestamo),
          Ahorro=parse_factor(Ahorro, levels=NULL),
          Duracion_empleo=parse_factor(Duracion_empleo, levels=NULL),
          Tipo_interes=parse_factor(Tipo_interes, levels=NULL),
          Estado_civil=parse_factor(Estado_civil, levels=NULL),
          Otros_deudores=parse_factor(Otros_deudores, levels=NULL),
          Historial_residencia=parse_factor(Historial_residencia, levels=NULL),
          Propiedad=parse_factor(Propiedad, levels=NULL),
          Edad=parse_integer(Edad),
          Cuota=parse_factor(Cuota, levels=NULL),
          Vivienda=parse_factor(Vivienda, levels=NULL),
          Creditos_existentes=parse_factor(Creditos_existentes, levels=NULL),
          Fallido=parse_factor(Fallido, levels=NULL),
          Hijos=parse_factor(Hijos, levels=NULL),
          Extranjero=parse_factor(Extranjero, levels=NULL),
          Tipo_empleo=parse_factor(Tipo_empleo, levels=NULL))

glimpse(datosc)
```

Estimación del hiperparámetro mtry

La función *randomForest* de Rstudio, contenida en la librería "randomForest", viene dada con unos parámetros por defecto. En ocasiones puede resultar aceptado utilizar el valor por defecto, pero en el caos que nos ocupa, queremos obtener el valor del parámetro que nos proporcione el menor error posible.

Procedemos a la creación de una función que nos devuelva el cálculo del *out-of-bag clasification* error de un modelo *randomforest* en función del número de predictores que se evalúen.

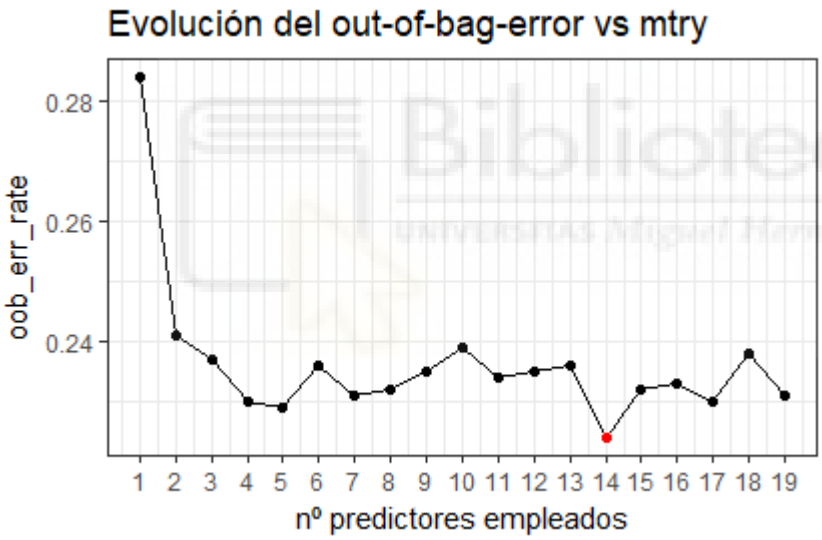
```
estimador_mtry <- function(df, y, ntree = 500){
  maximo_predictores <- ncol(df) - 1
  numero_predictores <- rep(NA, maximo_predictores)
  oob_err_rate <- rep(NA, maximo_predictores)
  for (i in 1:maximo_predictores) {
    set.seed(123)
    f <- formula(paste(y,"~ ."))
    modelo_rf <- randomForest(formula = f, data = df, mtry = i, ntree = ntree)
    numero_predictores[i] <- i
    oob_err_rate[i] <- tail(modelo_rf$err.rate[,1], n = 1)
  }
  results <- data_frame(numero_predictores, oob_err_rate)
  return(results)
}

hiperparametro_mtry <- estimador_mtry(df= datosc, y = "Fallido")
hiperparametro_mtry %>% arrange(oob_err_rate)
```

El resultado nos muestra una tabla de dos columnas, con el número de predictores evaluados y el error estimado por cada uno de ellos.

Número Predictores	Obb_error_rate	Número Predictores	Obb_error_rate
14	0.224	**9**	0.235
5	0.229	**12**	0.235
4	0.230	**6**	0.236
17	0.230	**13**	0.236
7	0.231	**3**	0.237
19	0.231	**18**	0.238
8	0.232	**10**	0.239
15	0.232	**2**	0.241
16	0.233	**1**	0.284
11	0.234		

Este resultado se puede observar gráficamente en la siguiente imagen. Como indica la tabla, el número de predictores empleado que minimice el error es 14 con un error estimado de 0.224.



Estimación del hiperparámetro nodesize

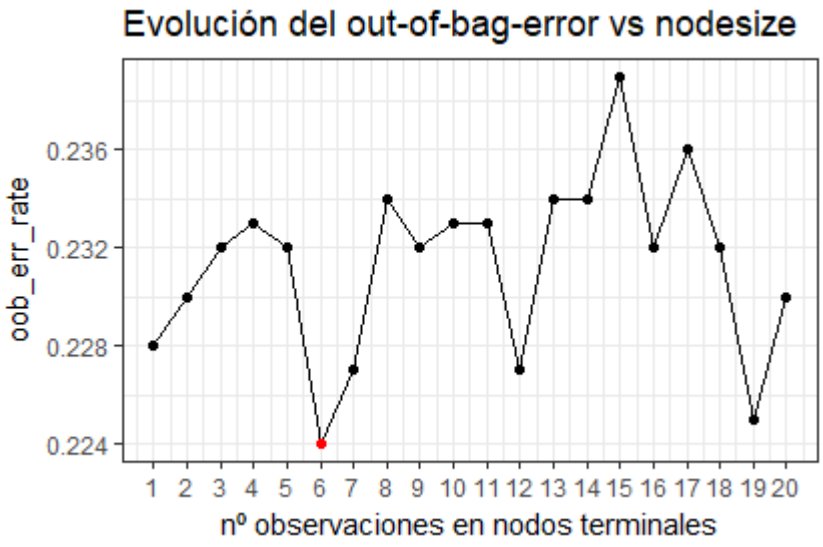
Otro hiperparámetro importante a la hora de realizar el modelo, es el tamaño de los nodos terminales. Esta función calcula, al igual que en el apartado anterior, el *out-of-bag clasification*, pero en este caso, en función del tamaño de los nodos terminales (*nodesize*).

```
tuning_rf_nodesize <- function(df, y, size = NULL, ntree = 500){
  if (is.null(size)){
    size <- seq(from = 1, to = nrow(df), by = 5) }
  oob_err_rate <- rep(NA, length(size))
  for (i in seq_along(size)) {
    set.seed(321)
    f <- formula(paste(y,"~ ."))
    modelo_rf <- randomForest(formula = f, data = df, mtry = 14, ntree = ntree,
                              nodesize = i)
    oob_err_rate[i] <- tail(modelo_rf$err.rate[, 1], n = 1)
  }
  results <- data_frame(size, oob_err_rate)
  return(results)
}
hiperparametro_nodesize <- tuning_rf_nodesize(df = datosc, y = "Fallido",
                                              size = c(1:20))
hiperparametro_nodesize %>% arrange(oob_err_rate)
```

Dicha función, nos devuelve una tabla dividida en dos columnas con el tamaño de los nodos y su error estimado.

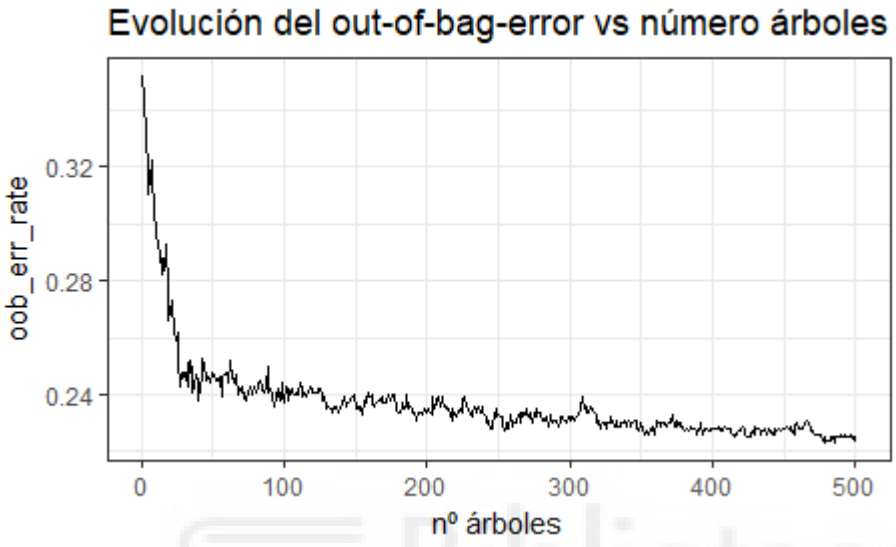
Tamaño del nodo	Obb_error_rate	Tamaño del nodo	Obb_error_rate
6	0.224	**16**	0.232
19	0.225	**18**	0.232
7	0.227	**4**	0.233
12	0.227	**10**	0.233
1	0.228	**11**	0.233
2	0.230	**8**	0.234
20	0.230	**13**	0.234
3	0.232	**14**	0.234
5	0.232	**17**	0.236
9	0.232	**15**	0.239

El resultado óptimo del tamaño del nodo es 6, como se puede observar gráficamente de una manera clara en la siguiente imagen:



Implantación del modelo y cálculo del error

Una vez se haya estimado los dos hiperparámetros anteriores, se lanzará el randomforest sustituyendo los dos parámetros calculados por los parámetros predefinidos que contiene está función. Una vez realizado dicho paso, obtendremos la evolución del *out-of-bag error* en función del número de árboles empleados.

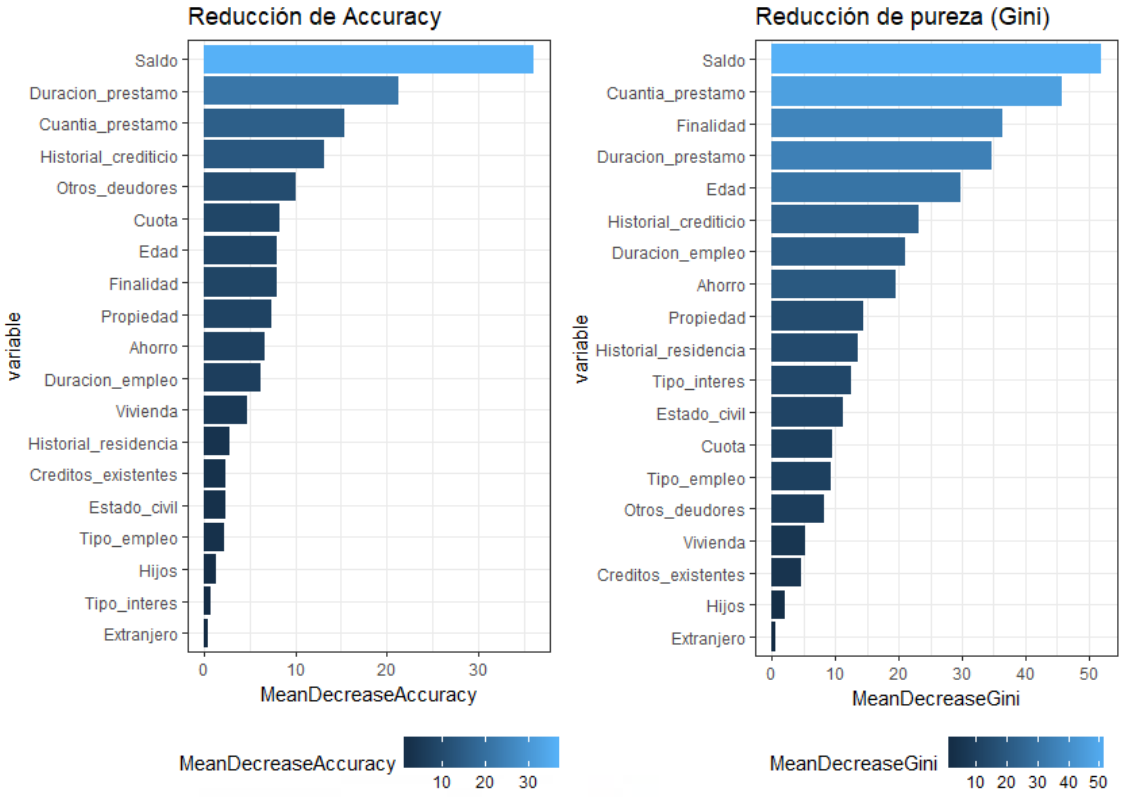


Como se puede observar en la imagen, el error comienza a estabilizarse alrededor de 400 árboles, por tanto, y aunque en este caso la limitación computacional no nos afecta, el valor óptimo son 400 ya que si estuviésemos trabajando sobre unos datos de un tamaño mucho mayor, podríamos tener problemas.

Una vez calculados los 3 hiperparámetros del randomforest, se procede a lanzar nuevamente el modelo para obtener la matriz de predicciones (véase en el apartado conclusiones).

Importancia de los predictores

Una de las ventajas de usar los randomforest frente a otro tipo de técnicas, es poder calcular la importancia de los predictores frente al modelo, ya sea con el porcentaje de variabilidad explicada o con la pureza de los nodos.

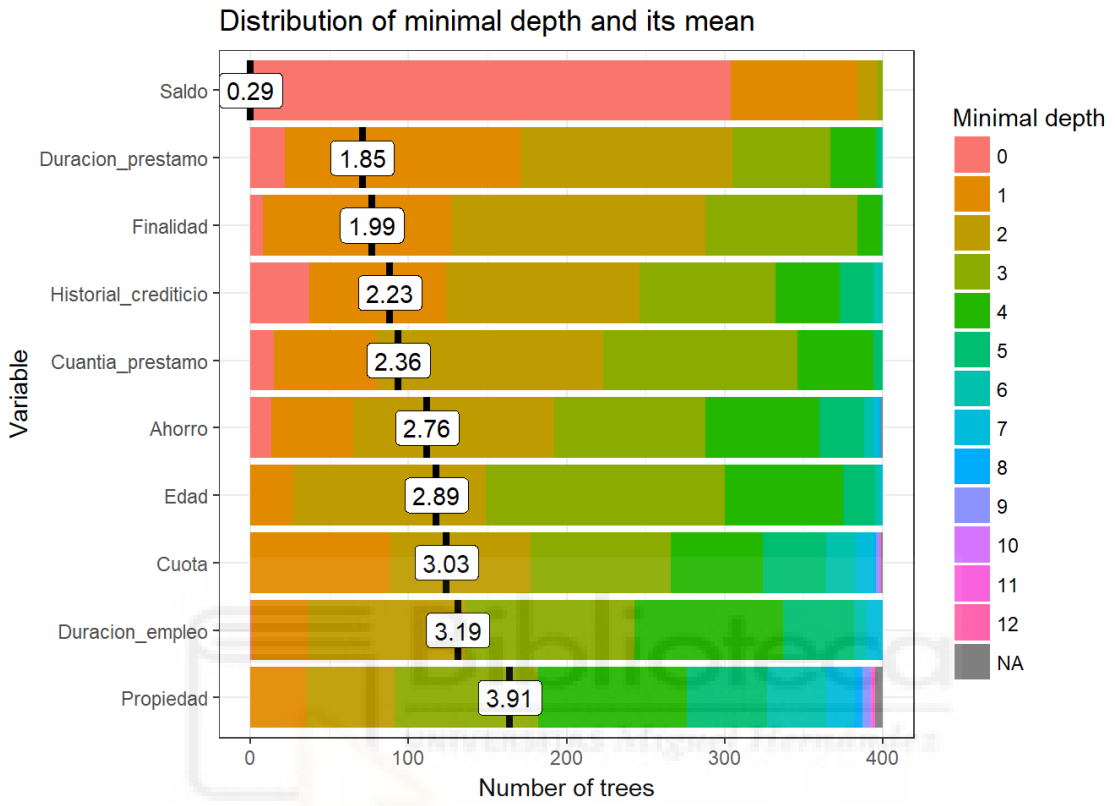


En esto gráficos se observa claramente que los predictores que tienen más importancia en el modelo son: Saldo, duración del préstamo, cuantía del préstamo, y el historial crediticio. También concuerdan con la reducción de la pureza del nodo, aunque en este caso, la variable finalidad entra a formar parte de las más importantes.

Por otra parte observamos que estas dos medidas, están estrechamente correlacionadas, ya que si nos fijamos en las variables en las variables que más precisión aportan a nuestro modelo son casi exactamente las mínimas que aumentan la pureza del nodos (observaciones de uno mismo tipo). Al igual que acabamos de comentar, sucede lo mismo con las variables que menos precisión aportan al modelo y reducen la pureza de los nodos.

Los predictores que aportan información sobre la extranjería, tipos de interés, si tienes hijos… se observa claramente que no son predictores de mucha importancia, pero en el caso de este tipo de algoritmos, como hemos comentado antes, no existe la necesidad de eliminar estas variables del modelo.

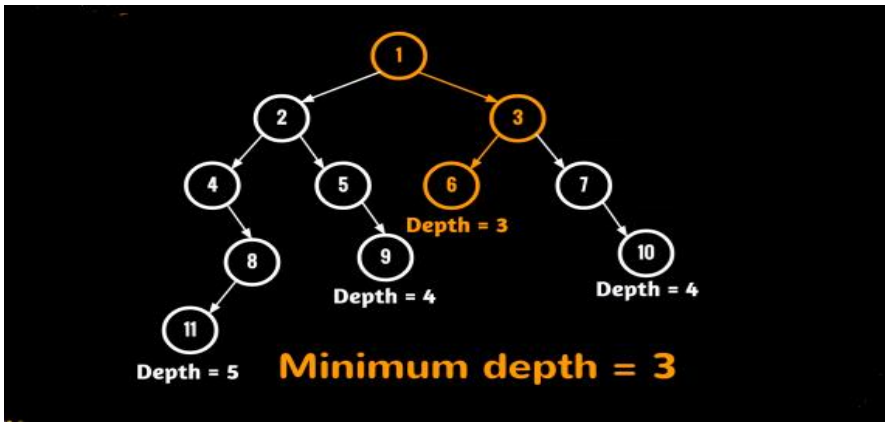
Explicación gráfica del árbol

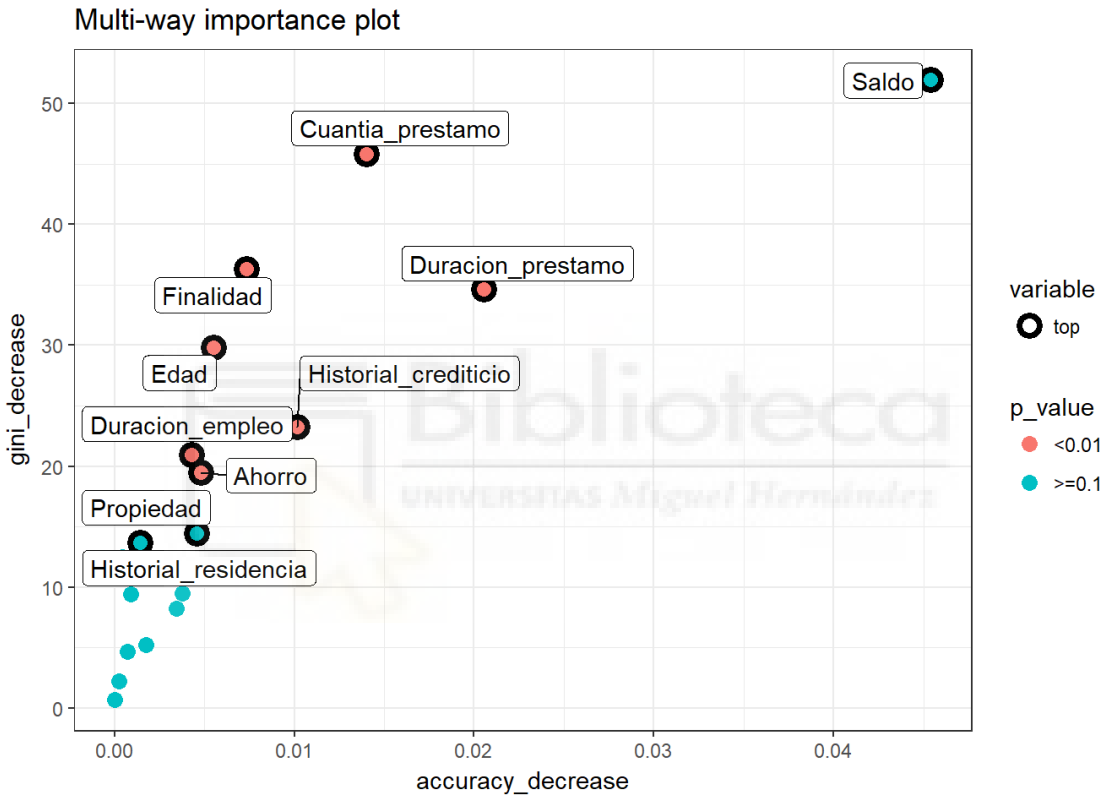


El concepto de *minimal depht*, hace referencia al camino más corto de entre todos los caminos desde el nodo parental hasta el nodo terminal en función del número de árboles utilizado. Observamos la variable más importante en nuestro modelo sigue siendo el saldo en cuenta, en este caso, muestra que es la variable que proporciona el camino más corto hasta el nodo terminal, lo que implica, que se reduce la complejidad de las ramificaciones y por tanto el error que se puede cometer en las sucesivas iteraciones.

Se vuelve a comprobar claramente que esta medida está relacionada con la mediada que indica la precisión del modelo además del índice que muestra la pureza de los nodos.

En el gráfico que viene a continuación, se muestra un ejemplo sencillo del significado de este concepto.



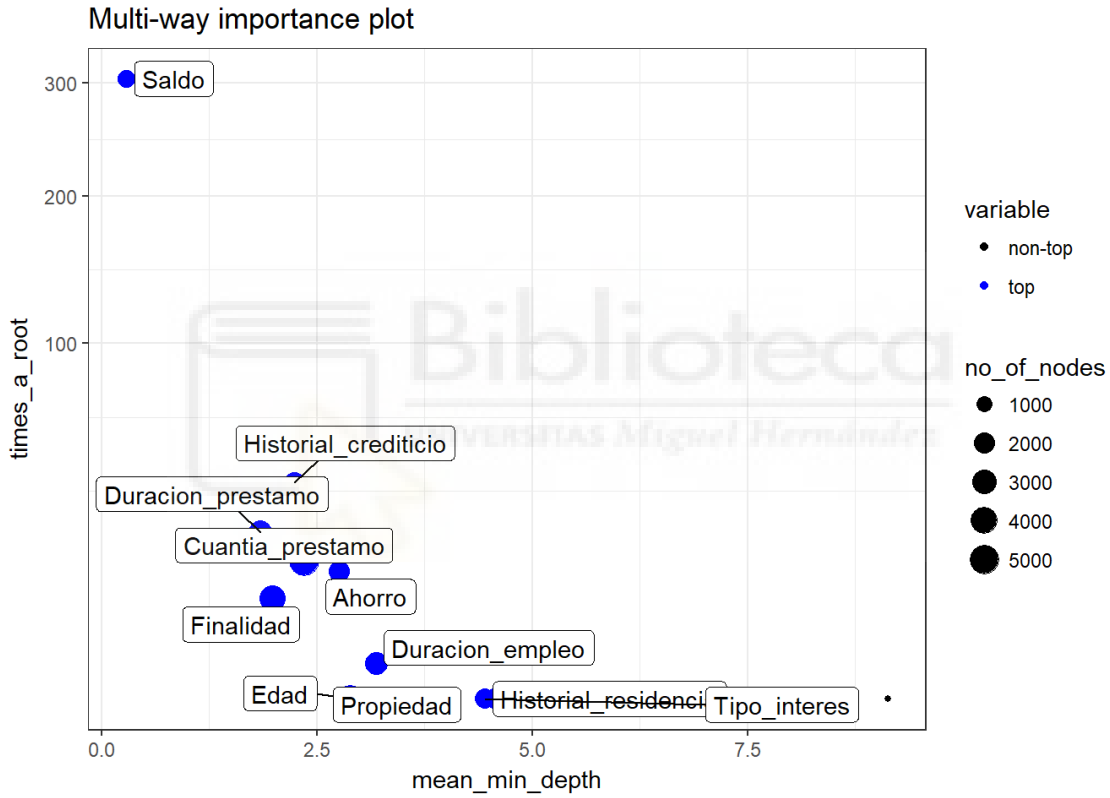


El segundo gráfico muestra la relación de los predictores con dos medidas de importancia para el análisis del modelo.

- Índice de Gini: explicado de una manera sencilla, cuanto más alto sea el valor, mayor es la pureza del nodo, es decir, el valor de las observaciones que contiene es el mismo.
- Precisión: hace referencia, a la influencia de las variables en la capacidad predictiva del modelo.

Se ha remarcado las 10 variables más importantes como se observa en la leyenda con la etiqueta “top”. Observamos claramente que los predictores más importantes dentro de nuestro modelo tanto por la precisión que aportan al modelo como por la influencia en la pureza de los nodos son: Saldo, cuantía del préstamo y su duración.

Por otra parte nos indica, analizando el p-valor, que las variables que son significativas se utilizan para realizar las divisiones de los subespacios con más frecuencia que si la selección de los predictores fuera al azar.



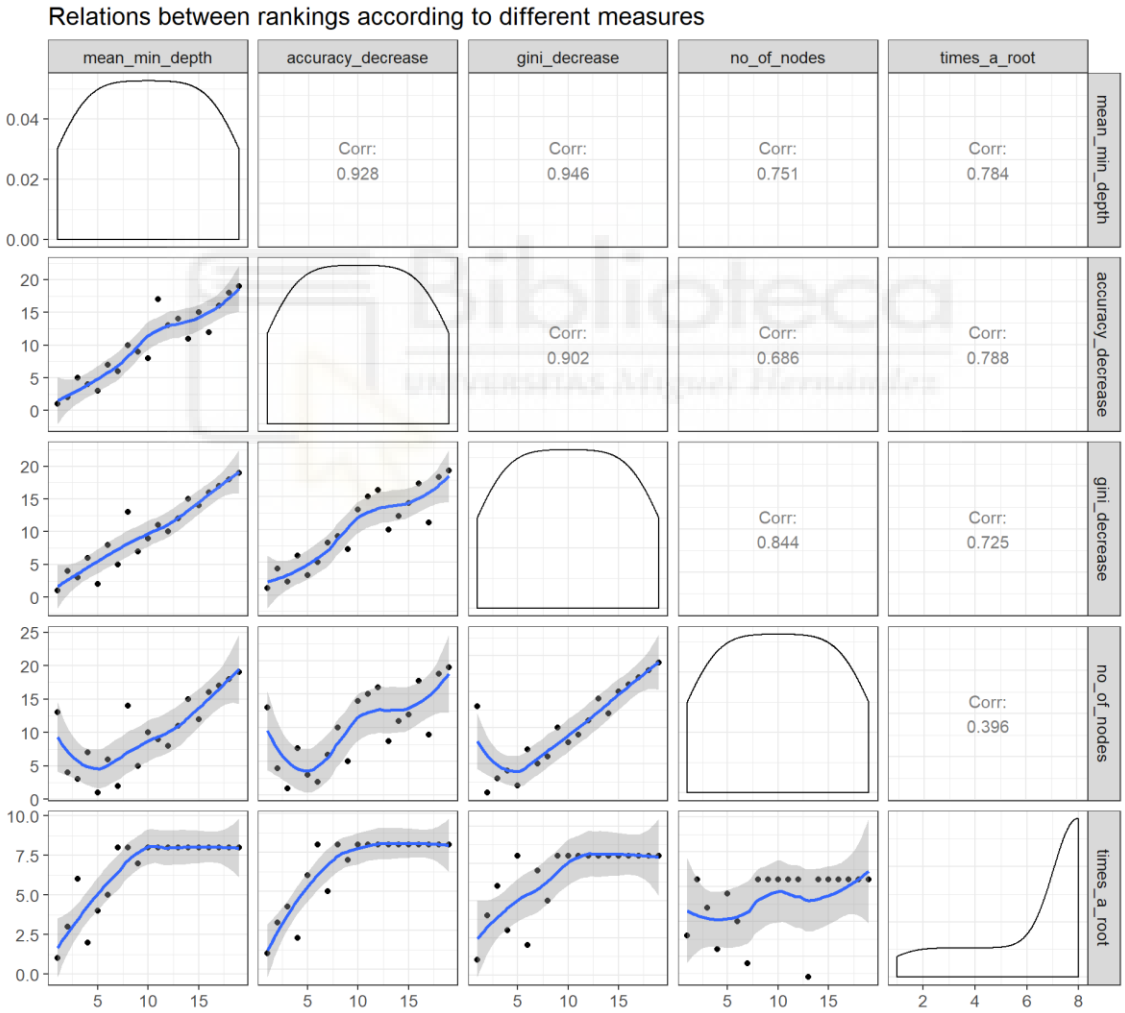
El gráfico de importancia multidireccional muestra la relación entre 11 de nuestros predictores y tres medidas de importancia para el modelo:

• Profundidad media de la primera división en la variable (mean-min-depth)

• Cantidad de árboles en los que la raíz se divide en la variable (times-a-root)

• La cantidad total de nodos en el bosque que se dividen en esa variable (no-of-nodes)

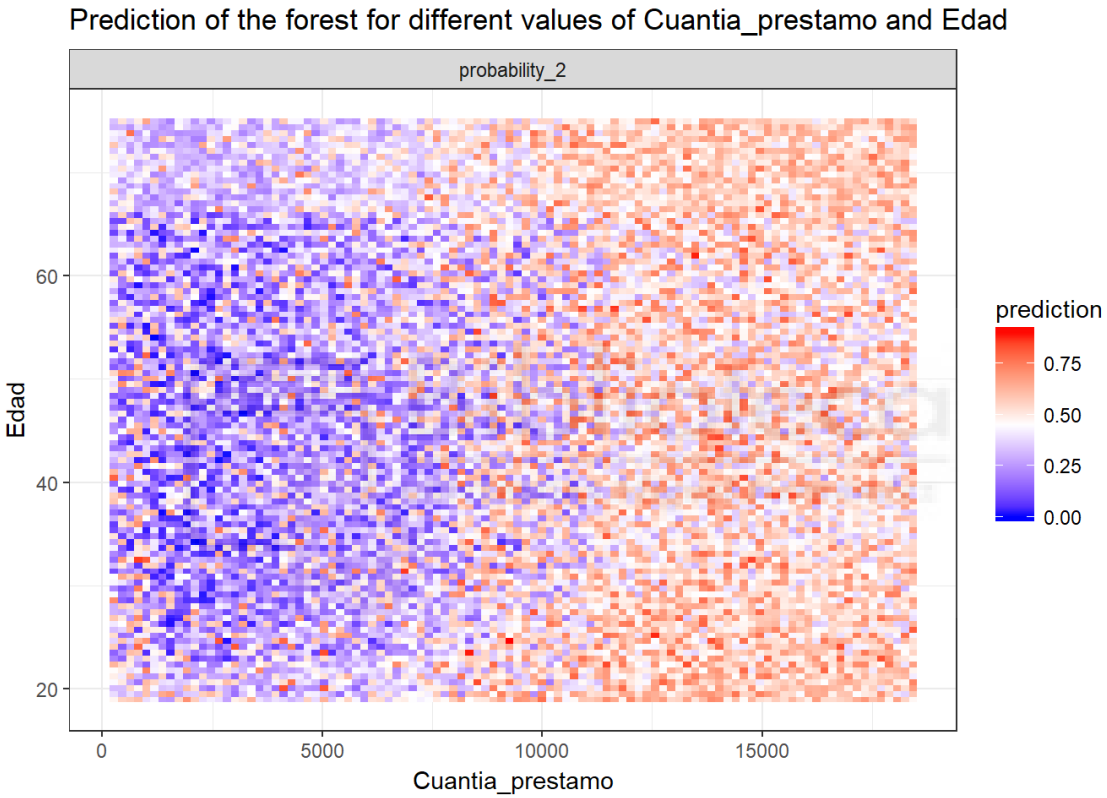
Lo primero que se observa a primera vista es que el número de nodos es muy similar para cualquiera de nuestras variables y que, nuevamente, la variable saldo es la que cobra más importancia a la hora de realizar la división de los subespacios conforme aumenta el número de ramificaciones del árbol.



El siguiente diagrama muestra las relaciones bilaterales que existen entre las clasificaciones de las variables según la medida de importancia que se usa dentro del

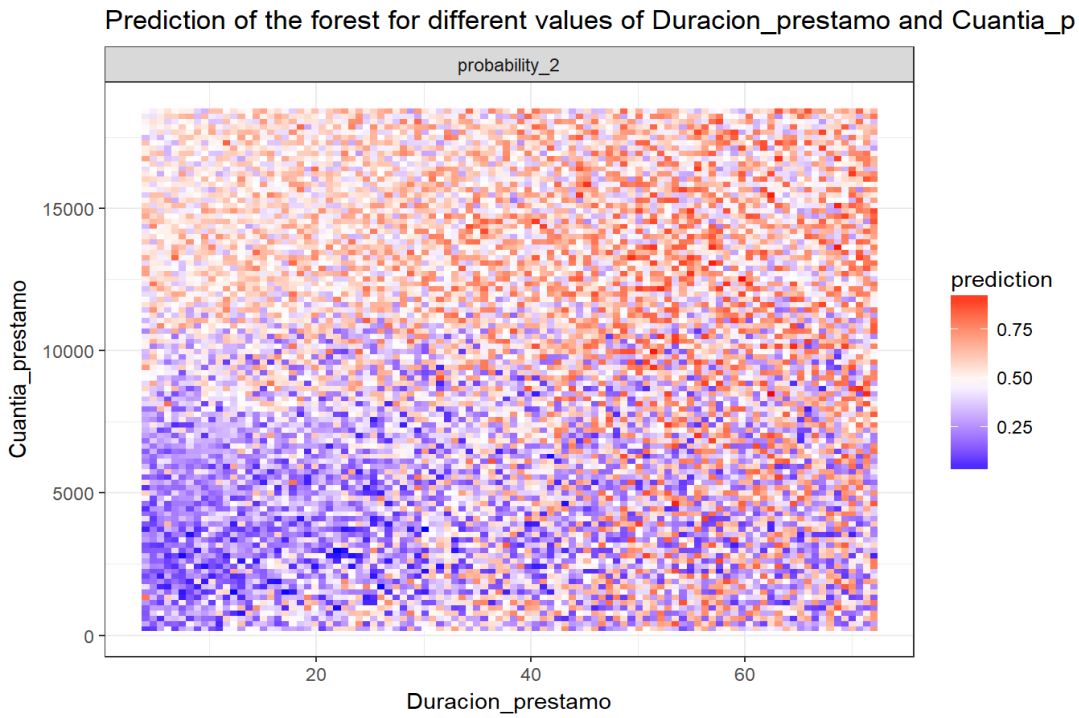
modelo. Como se observa claramente, la mayoría de estas medidas tienen una correlación muy alta.

Cabe destacar la relación existente entre las veces que se ramifica el árbol y el número de los nodos, ya que es la única relación que no alcanza una correlación superior al 50%.



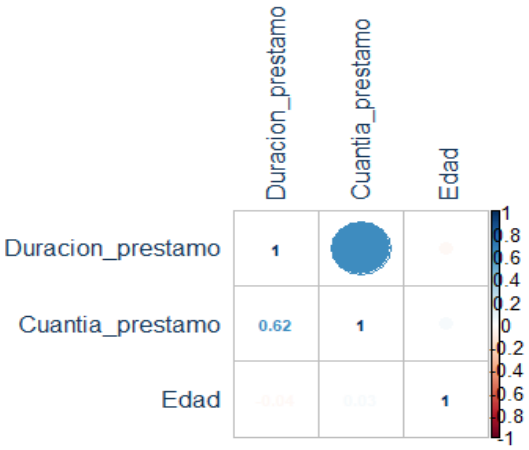
Este gráfico representado a través de una nube de puntos, muestra el porcentaje de predicción que obtenemos cuando cruzamos dos de nuestros predictores, la cuantía del préstamo y la edad.

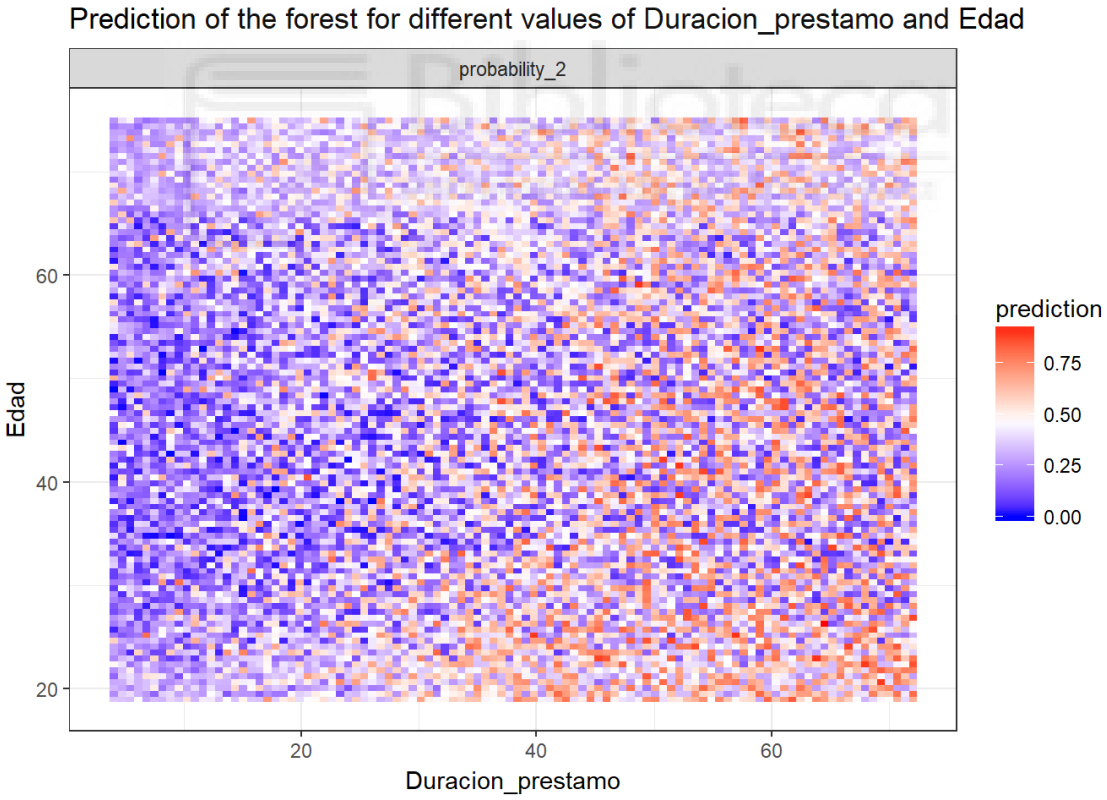
Lo primero que se observa a simple vista es que la edad del prestatario respecto a la cantidad del préstamo solicitada es indiferente, mientras que cuanto mayor es la cantidad del préstamos solicitado, mejor es la predicción de nuestro modelo, siendo superior al 75% los préstamos superiores a 10.000 euros.



Al igual que en el gráfico anterior, vamos a realizar la predicción para diferentes valores de la duración del préstamo frente a la cuantía del mismo. Cabe recordar que estas dos variables tienen una correlación superior al 60% como se muestra en la figura siguiente, siendo totalmente normal esta situación, ya que generalmente, a mayor cantidad del préstamo, la duración suele ser mayor.

Por otra parte, como se mostraba en el primer gráfico a partir de una cantidad de 10.000 euros de los préstamos obtenemos una predicción con un porcentaje bastante alto, pero la duración del préstamos no es una variable que debamos tener en cuenta a la hora de realizar las predicciones.





Por último, cruzamos la duración del préstamo con la edad del prestatario. Se observa claramente que esta interacción no aporta buenas predicciones para el modelo, ya que solamente a partir de 40 meses observamos una muestra pequeña de valores con un porcentaje medio de predicción.

Para finalizar el análisis gráfico del *Randomforest* que hemos construido, se ha creado una tabla a modo resumen, de las medidas anteriormente graficadas, ordenadas alfabéticamente.

variable	Mean.depth	Nº de nodos	Reducción precisión
Ahorro	2.759	2134	0.0061
Créditos existentes	6.314	748	0.0115
Cuantía préstamo	2.355	5435	0.0421
Cuota	3.026	1248	0.01066
Duración empleo	3.187	2580	0.0002
Duración préstamo	1.85	2883	0.0040
Edad	2.887	4397	0.0019
Estado civil	4.907	1562	0.00003
Extranjero	9.133	113	0.0012
Finalidad	1.985	3684	0.0014
Hijos	7.377	443	-0.00003
Historial crediticio	2.222	2145	0.0105
Historial residencia	4.367	2027	0.0013
Otros deudores	5.069	1069	0.0031
Propiedad	3.856	1709	0.0054
Saldo	0.257	1307	0.0476
Tipo empleo	5.012	1300	0.0016
Tipo interés	4.397	1817	0.0016
Vivienda	5.959	823	0.0005

Conclusiones

Una vez realizado el análisis completo del modelo, hemos obtenido, en términos generales un modelo que es capaz de predecir con un error ligeramente superior al 10% los préstamos que no han entrado en fallido.

Este porcentaje de acierto del 90% nos permitirá realizar las compras o recompras de préstamos, una vez el banco nos haya enviado los datos necesarios, pudiendo seleccionar los préstamos que sabemos con un porcentaje alto, que no van a entrar en fallido a futuro.

```
##
## Call:
##  randomForest(formula = Fallido ~ ., data = datosc, mtry = 14,      ntree = 400, importance = TRU
   nodesize = 6, norm.votes = TRUE)
##               Type of random forest: classification
##                     Number of trees: 400
## No. of variables tried at each split: 14
##
##         OOB estimate of  error rate: 22.9%
## Confusion matrix:
##     1   2 class.error
## 1 629  71   0.1014286
## 2 158 142   0.5266667
```

	Verdad acerca de la población	
Decisión basada en la muestra	H_0 es verdadera	H_0 es falsa
No rechazar H_0	Decisión correcta (probabilidad = 1 - α)	**Error tipo II** - no rechazar H_0 cuando es falsa (probabilidad = β)
Rechazar H_0	**Error tipo I** - rechazar H_0 cuando es verdadera (probabilidad = α)	Decisión correcta (probabilidad = 1 - β)

Nuestro algoritmo busca minimizar la probabilidad del Error tipo II, también denominado error β. En nuestro caso, dicho error, consiste en calcular que un préstamo no va a entrar en fallido, cuando realmente si entra.

Uno de los objetivos de esta empresa a la hora de realizar la constitución de un fondo, o las pertinentes recompras es seleccionar la mayor cantidad de préstamos (teniendo como límites los dados por el cedente), que no entren en fallido. Por tanto, respecto al error de tipo I, que consiste en calcular que un préstamo va a entrar en fallido, pero no entra, no es un error grave en este modelo.

En conclusión, obtenemos un algoritmo para poder introducir los datos procedentes del banco y poder elegir con una alta precisión cuales son los préstamos que vamos a seleccionar de la cartera y no van a entrar en fallido.

TRABAJO A FUTURO

No solo es práctico para el análisis de los prestamos fallidos, es una fácil modificar el trabajo para prever otros cambios en los préstamos, amortizaciones previas, impagos, primer mes de impago etc…

Todos estos análisis ganarían mucha calidad cuanto más información tengamos del deudor e información histórica del préstamo.

Luego habría que implementar el estudio de cara a otros tipos de activos: prestamos hipotecarios, para pymes, facturas, e incluso tarjetas de crédito y según las características del deudor, localización geográfica, edad, profesión. No tiene el mismo comportamiento un préstamo concedido para comprar un TV a alguien de Madrid que un préstamo hipotecario de alguien que vive del campo, el agricultor amortizara después de la cosecha y generara meses de impago durante la sequía, y el deudor de la Madrid no tiene por qué.

Entonces en un futuro se puede montar una batería de análisis de préstamos que permita saber la evolución completa de una cartera, dando al inversor una supuesta evolución de sus bonos dando el porcentaje de error de esta. Y previamente se pueden no haber seleccionado préstamos y deudores que puedan empeorar la calidad del Fondo. Una vez constituido el Fondo, todos los meses con la información recibida, se puede actualizar la evolución de la cartera.

ANÉCDOTA

Déjame explicarte que es un *RandomForest* con un ejemplo sencillos:

Supongamos que eres una persona muy indecisa, así que cuando quieres ver una película, primero, preguntas a tu amiga si ella piensa que te va a gustar. Para responder a esta cuestión, ella necesita averiguar qué tipo de películas te gustan, así que le comentas un montón de películas y le dices que te gusta y que no, de cada una de ellas. Entonces, cuando tú le preguntas si ella cree que te va a gustar la película X o no, ella te plantea 20 preguntas como en el juego de *¿Quién es Quién?* , por ejemplo "¿es X una película romántica?", "¿Aparece el actor *Johnny Depp* en la película? Y así, sucesivamente. Ella intenta obtener la máxima información posible de cada cuestión y te proporcionará un sí o un no al final.

De este modo, tu amiga se ha convertido en un árbol de decisión para las preferencias de tus películas.

Pero, tu amiga es solo una humana, así que ella no puede clasificar tus preferencias siempre. Entonces, la solución es preguntar a un gran grupo de tus amigos para obtener respuestas más precisas, y ver la película X, si la mayoría de ellos piensan que te va a gustar. Esa es la técnica, en lugar de preguntar solamente a tu amiga, le preguntas a ella y a todo su grupo de amigas y ellas votarán si te gusta o no te gusta la película.

Ahora, lo que quieres evitar es que cada una de tus amigas te de la misma respuesta, así que, primero les proporcionas datos ligeramente diferentes. Después de todo, ni siquiera tú tienes tus preferencias claras. Le dijiste a tu amiga que te encantaba el Titanic, pero puede ser que ese día estuvieras feliz porque era tu cumpleaños, así que tal vez algunos de tus amigos no deberían usar el hecho de que te gustaba Titanic al hacer sus recomendaciones. O tal vez le dijiste que te encantaba *Cenicienta*, pero en realidad te sigue gustando aun, por lo que algunos de tus amigos deberían darle más peso a Cenicienta. Entonces, en lugar de darles a tus amigas los mismos datos que le diste a la primera amiga, les das versiones ligeramente diferentes.

No cambies tus decisiones de amor y odio, solo dices que amas y odias algunas películas de una manera general (le das a cada una de tus amigas una versión inicial de tus datos de entrenamiento originales). Por poner un ejemplo, mientras que le dijiste a una que te gustaban Black Swan y Harry Potter y que no te gustaba Avatar, le dices a

otra que te gustó Black Swan tanto que la viste dos veces, no te gustó Avatar y no mencionas a Harry Potter en absoluto.

Al usar esta técnica, esperas que mientras cada uno de tus amigos dé recomendaciones bastante idiosincráticas (una piense que te gustan las películas de vampiros más que tú, otra cree que te gustan las películas de Pixar, y otra cree que odias todo), los errores se cancelan en la mayoría. Por lo tanto, tus amigas ahora forman un bosque compacto (bootstrap aggregated) de tus preferencias sobre películas.

Sin embargo, todavía hay un problema con los datos. Si bien te encantaron tanto Titanic como Inception, no fue porque te gusten las películas que protagoniza Leonardo DiCaprio, tal vez te gustaron ambas películas por otras razones diferentes. Por lo tanto, no quiere que sus amigos basen sus recomendaciones sobre si DiCaprio está en una película o no. Entonces, cuando cada amigo hace una pregunta, solo se permite un subconjunto aleatorio de las posibles preguntas (es decir, cuando se está construyendo un árbol de decisión, en cada nodo se usa aleatoriamente para seleccionar el atributo para dividir, digamos seleccionando al azar un atributo o seleccionando un atributo de un subconjunto aleatorio). Esto significa que tus amigas no pueden preguntar si Leonardo DiCaprio aparece en la película cuando ellas quieran. Entonces, mientras que previamente se introdujo aleatoriedad en el nivel de datos, al difuminar levemente las preferencias de la película, ahora se inyecta aleatoriedad en el modelo al hacer que sus amigas le hagan preguntas diferentes en momentos diferentes.

Y entonces tus amigas ahora forman un bosque aleatorio (Random Forest).

BIBLIOGRAFÍA

Barber, X. (s.f.). Obtenido de http://umh1480.edu.umh.es/

Bhalla, D. (Noviembre de 2014). *www.listendata.com*.

Fernandez, S. d. (2011).

Flores, W. G. (s.f.). *http://www.tamps.cinvestav.mx*. Obtenido de http://www.tamps.cinvestav.mx/~wgomez/diapositivas/RP/Clase14.pdf

https://blog.es.logicalis.com. (3 de 5 de 2015). Obtenido de https://blog.es.logicalis.com/analytics/modelos-predictivos-reforzando-el-valor-de-una-buena-decision

https://blog.es.logicalis.com. (3 de 25 de 2015). Obtenido de https://blog.es.logicalis.com/analytics/predictive-analytics-los-principales-modelos-del-analisis-predictivo

https://blog.es.logicalis.com. (25 de 3 de 2015). Obtenido de https://blog.es.logicalis.com/analytics/predictive-analytics-los-principales-modelos-del-analisis-predictivo

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York: Springer.

Loh, W.-Y. (s.f.). *www.stat.wisc.edu*. Obtenido de http://www.stat.wisc.edu/~loh/guide.html

Paluszyńska, A. (19 de Mayo de 2017). *https://cran.rstudio.com*. Obtenido de https://cran.rstudio.com/web/packages/randomForestExplainer/vignettes/randomForestExplainer.html

Paluszynska, A. (3 de Diciembre de 2017). *https://rawgit.com*. Obtenido de https://rawgit.com/geneticsMiNIng/BlackBoxOpener/master/randomForestExplainer_Master_thesis.pdf

randomforest2013.blogspot.com. (8 de Mayo de 2013). Obtenido de https://randomforest2013.blogspot.com.es/2013/05/randomforest-definicion-random-forests.html

Rodrigo, J. A. (Febrero de 2017). *www.Rpubs.com.* Obtenido de https://rpubs.com/Joaquin_AR/255596

Stephens, T. (19 de Enero de 2014). *trevorstephens.com*. Obtenido de http://trevorstephens.com/kaggle-titanic-tutorial/r-part-5-random-forests/

Walia, A. S. (24 de Julio de 2017). *www.r-bloggers.com*. Obtenido de https://www.r-bloggers.com/random-forests-in-r/