# FACULTAD DE ESTUDIOS ESTADÍSTICOS

## MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

**Curso 2018/2019**

## Trabajo de Fin de Máster

# *Aplicación y comparación de modelos de machine learning destinados a la puntuación del riesgo de crédito*


***Alumno*: Álvaro Alonso Pallarés**

***Tutor*: Lorenzo Escot Mangas**


Septiembre de 2019


UNIVERSIDAD
COMPLUTENSE
MADRID

*"Tanto si crees que puedes, como si crees que no puedes, tienes razón."*

Mi padre.

## AGRADECIMIENTOS

No creo que tenga la suficiente capacidad como para, a través de sólo unas palabras, lograr hacer justicia y conseguir transmitir verdaderamente lo agradecido que le estoy a todas estas personas, pero aquí va.

Quiero agradecer en primer lugar a mi familia, sobre todo a mis padres y hermanos los cuales, en estos momentos en los que nos encontramos, atravesando una de las mayores encrucijadas en las que una familia se puede encontrar, a diario me aguantaron, me apoyaron, me animaron a seguir avanzando, a no tirar la toalla y gracias a ellos logré terminar este proyecto. Porque siempre estuvisteis ahí, gracias.

En segundo lugar, quería dar las gracias a los diferentes grupos de amigos con los que cuento, en quienes siempre he encontrado tanto un lugar donde sentirme a gusto como un refugio donde desahogarme, especialmente en este último año. Quiero aprovechar esta ocasión y hacer hincapié en el grupo de amigos de la universidad, los cuales constantemente estuvieron dispuestos a ofrecerme ese recurso al que muy pocas personas están dispuestas a conceder, su tiempo, siempre preparados para echarte una mano y para escucharte. Además, y esta es una de las pocas cosas de mi vida que puedo afirmar con total seguridad, sin ellos no habría conseguido terminar tanto el grado como el máster a la velocidad a la que lo hice y logrando las notas con las que terminé. Por todo ello y por mucho más, gracias.

En tercer lugar, también quería dar las gracias al personal de la universidad y a todos los profesores con los que siempre he podido contar. A Javier Portela por la ayuda que me ha ofrecido últimamente resolviendo todas mis dudas, y especialmente a Lorenzo, por ser mi tutor y aguantarme tan pacientemente en tantas tutorías y dirigirme de la forma tan excelente en la que lo ha hecho.

Porque gracias a todos ellos, familia, amigos y profesores, me considero una persona excepcionalmente afortunada, a todos ellos, GRACIAS. Y este trabajo de fin de máster se lo dedico a la persona más especial, a la persona que más ha ocupado mi vida en estos últimos meses y con la que espero seguir contando siempre, va por ti Marga.

# RESUMEN

En este trabajo de investigación se estudian y comparan modelos que se emplean actualmente en el ámbito del scoring, como son la regresión logística, junto con otros modelos de machine learning que también se podrían implementar como son las redes neuronales, el Support Vector Machine (SVM) y técnicas basadas en árboles como son Bagging, Random Forest y Gradient Boosting.

Otros estudios ya han demostrado la superioridad en técnicas de clasificación de las redes neuronales frente a la regresión logística. En esta investigación queremos comprobar si logramos unos resultados acordes a estas otras investigaciones, sobre todo fijándonos en dos objetivos en particular. En primer lugar, encontrar el modelo que mejor resultados obtenga en función de su capacidad predictiva, y en segundo lugar, lograr encontrar, si existe, una diferencia significativa tanto en términos de predicción como en tiempos de ejecución a la hora de utilizar las habituales variables categóricas binarias frente a sus homólogas tipo WOE.

Los resultados demuestran que el modelo que mejores resultados obtiene es el SVM; y que las diferencias en calidad de predicción no son tan relevantes entre variables binarias y de tipo WOE, mientras que el tiempo de ahorro en ejecución por parte de este segundo tipo de variables es del todo significativo.

# Índice

# 1. INTRODUCCIÓN

Se entiende como crédito o contrato de crédito a aquella operación financiera en la que una persona (el acreedor) realiza un préstamo por una cantidad determinada de dinero a otra persona (el deudor) y en la que esta última se compromete a devolver la cantidad solicitada (además del pago de los intereses devengados, seguros y costos asociados si los hubiere) en el tiempo o plazo definido de acuerdo con las condiciones establecidas para dicho préstamo. Existen diferentes tipos de créditos, crédito tradicional, al consumo, comercial, hipotecario, personal, microcrédito, etc.

Las empresas encargadas de conceder estos créditos son casi siempre entidades financieras. Este tipo de entidades al poner en juego su capital y siendo su negocio en gran medida el conceder esta financiación económica a clientes a los que apenas conoce (una persona entra en una sucursal, pide un crédito, pero no se sabe si esta es de fiar o no), utilizan modelos de puntuación de riesgo para evitar impagos por parte de los deudores y de esta forma minimizar los riesgos y pérdidas para la empresa.

Las entidades financieras utilizan diferentes modelos de riesgo dependiendo de si están dirigidos a clientes nuevos o antiguos, modelos que no son de riego de concesión de crédito si no que son modelos de recobro, y todo un abanico de diferentes modelos orientados a estudiar, dentro de cada una de las ramas de mercado en las que se ve involucrada la entidad, cuáles son los factores más influyentes y cómo lograr un mayor beneficio junto con un menor riesgo.

De todos estos modelos, en este estudio nos centraremos en los modelos de riesgo de impago y es aquí donde entra en juego el concepto de "*Puntuación de riesgo para la concesión o rechazo de un crédito*". La puntuación de crédito es una de las medidas más importantes de solvencia. Actualmente uno de los sistemas más utilizados por las distintas entidades financieras en todo el mundo es la puntuación del informe crediticio de Fair Isaac Company **FICO®**. Este informe se basa en los criterios de medición desarrollados por Fair Isaac Corporation, el cual se centra en el análisis de datos generales del cliente y en los servicios de calificación crediticia en particular. La información sobre la que se basa este sistema es objetiva y consecuente. Por ley, la solvencia crediticia no se puede basar en la raza, sexo, religión o nacionalidad del solicitante y, por tanto, este es un sistema no discriminatorio.

El sistema de puntuación varía de 300 a 850 puntos y evalúa el riesgo de crédito al consumo, es decir, el nivel de riesgo que tiene el solicitante a la hora de devolver un préstamo. Cuanto mayor sea la puntuación, mayores posibilidades tiene el cliente de ser aceptado para un préstamo o tarjeta de crédito. Dentro de las tarjetas de puntuación se establece lo que se conoce como **Cutt-Off**, que es la puntuación mínima que debe alcanzar el cliente para que se le conceda el crédito, y dependerá de esta puntuación la calidad del crédito que se le ofrecerá (alcanzar el máximo capital a conceder, recibir mayores o menores intereses a pagar, etc).

La puntuación de crédito se basa en los siguientes cinco factores [1]:

- El **historial de pagos** representa el 35% de la puntuación. Esto muestra si la persona realiza los pagos puntualmente, con qué frecuencia omite pagos, cuántos días después de la fecha de vencimiento paga sus cuentas y cuándo fue la última vez que se omitieron pagos. Cuanto más elevada sea la proporción de pagos puntuales, mayor será su puntuación y cada vez que omita un pago se arriesgará a perder puntos.

- La **cantidad que adeuda una persona en préstamos y tarjetas de crédito** constituye el 30% de su puntuación. Esto se basa en la cuantía total que adeuda, la cantidad y los tipos de cuentas que tiene, y la proporción de dinero adeudado en comparación con la cantidad de crédito disponible. Los saldos altos y las tarjetas de crédito al límite disminuirán su puntuación de crédito, pero los saldos más bajos pueden aumentarla si paga puntualmente. Los nuevos préstamos con un historial de pagos breve pueden bajar su puntuación temporalmente, pero los préstamos que están más cerca de la liquidación pueden subirla porque muestran un historial de pagos exitoso.

- La **antigüedad de su historial de crédito** representa el 15% de su puntuación. Cuanto más antiguo sea el historial de pagos puntuales, más alta será su puntuación, ya que su puntuación se podría ver perjudicada si las entidades no tienen ningún historial de crédito para examinar.

- Los **tipos de cuentas que tiene** constituyen el 10% de su puntuación. Contar con una combinación de cuentas, incluidos préstamos a plazos, préstamos hipotecarios, tarjetas de crédito y préstamos de tiendas minoristas aumentará su puntuación.

- La **actividad de crédito reciente** constituye el último 10%. Si abre muchas cuentas últimamente o solicita que se abran cuentas, esto sugiere un posible problema financiero y puede bajar su puntuación. Sin embargo, si tiene los mismos préstamos o tarjetas de crédito durante un período prolongado y los paga puntualmente — incluso después de tener problemas de pago — su puntuación aumentará con el transcurso del tiempo.

Asociados a estos factores se encuentran también los datos generales del cliente como estado civil, rango salarial, estado civil, etc, con los que se termina de completar la tarjeta de puntuación.

A continuación se muestra un ejemplo de una tarjeta de puntuación y una tabla en la que se representan las distintas bandas de puntuación con sus calificaciones.

| Variable | Atributo | Puntuación |
|---|---|---|
| **Edad** | < 23 | 63 |
| **Edad** | 23 – 28 | 76 |
| **Edad** | 28 – 34 | 79 |
| **Edad** | 34 – 46 | 85 |
| **Edad** | 46 – 51 | 94 |
| **Edad** | > 51 | 105 |
| **Tipo Tarjeta** | AMEX, VISA, Sin TRJ | 80 |
| **Tipo Tarjeta** | MasterCard | 99 |
| **Salario** | < 600 | 85 |
| **Salario** | 600 – 1200 | 81 |
| **Salario** | 1200 – 2200 | 93 |
| **Salario** | > 2200 | 99 |
| **Estado Civil** | Casado | 85 |
| **Estado Civil** | Resto | 78 |

*Tabla 1*

| | |
|---|---|
| **300 – 549** | **Puntuación muy baja**. Prestatario con el mayor riesgo. |
| **550 – 619** | **Puntuación baja**. Prestatario de alto riesgo. |
| **620 – 649** | **Puntuación media**. Prestatario de medio riesgo. |
| **650 – 699** | **Puntuación alta**. Prestatario de bajo riesgo. |
| **700 – 749** | **Puntuación muy alta**. Prestatario de bajo riesgo. |
| **750 +** | **Puntuación excelente**. Prestatario de muy bajo riesgo. |

Este sistema de puntuación es un sistema ampliamente utilizado y que utiliza la regresión logística como método de predicción sobre las conductas de los clientes, ya que la regresión logística es el único método que es lineal en su transformación y que por tanto, permite construir una tabla de puntuación como la anterior de manera sencilla a partir de los parámetros estimados en la regresión logística. Pero este sistema de predicción como cualquier otro no es perfecto, y corre el riesgo de cometer los conocidos como errores de tipo I o tipo II. Es decir, conceder créditos a clientes que causarían impago; o rechazar créditos a clientes que pagarían puntualmente. Y es en esta búsqueda de un modelo óptimo, no necesariamente de regresión logística, donde entran en juego el resto de técnicas de machine learning debido a sus capacidades de predicción y creación de modelos de forma autónoma.

El machine learning, incluyendo a la propia regresión logística, es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de los datos, identificar patrones y tomar decisiones con una mínima intervención humana.

Este método nació del reconocimiento de patrones y de la teoría que indica que los ordenadores pueden aprender sin ser programados para realizar tareas específicas ya que investigadores interesados en la inteligencia artificial deseaban saber si los ordenadores podían aprender de datos.

El aspecto iterativo del machine learning es muy importante porque a medida que los modelos son expuestos a nuevos datos, éstos pueden adaptarse de forma independiente ya que aprenden de cálculos previos para producir decisiones y resultados confiables y repetibles.

Esta es una ciencia que no es nueva, pero que ha cobrado un nuevo impulso. El resurgimiento del interés en el machine learning se debe a los mismos factores que han hecho más populares que nunca a la minería de datos y el análisis Bayesiano. Cosas como disponer de volúmenes y variedades de datos crecientes, un procesamiento computacional económico y potente, y un almacenaje de datos asequible.

Todo esto significa que es posible producir modelos de manera rápida y automática que pueden analizar datos complejos y de gran tamaño y producir resultados rápidos y precisos, incluso a gran escala. Con unos modelos con estas características, toda organización que los use tendrá una mejor oportunidad de identificar oportunidades rentables o de evitar riesgos desconocidos. El machine learning se puede aplicar en distintos ámbitos como gobierno, salud, transporte, servicios financieros, marketing, ventas, etc.

En nuestro caso para el desarrollo de este trabajo compararemos un modelo de regresión logística que obtendremos inicialmente frente a otros modelos de machine learning de aprendizaje supervisado como son las redes neuronales, SVM, y algunas técnicas basadas en árboles como Bagging, Random Forest y Gradient Boosting.

# 2. OBJETIVOS Y JUSTIFICACIÓN DEL PROYECTO

Los objetivos de este trabajo de investigación son fundamentalmente dos, comparar los resultados de predicción de los modelos de regresión logística utilizados actualmente para el desarrollo de las tarjetas de puntuación, frente a otros modelos de machine learning de aprendizaje supervisado como son las redes neuronales, Support Vector Machine y junto con algunas técnicas de aprendizaje basadas en árboles como son Bagging, Random Forest y Gradient Boosting; y también analizar la predicción de estos modelos, tanto mediante variables explicativas binarias como mediante variables explicativas que han sufrido una transformación WOE (la cual explicaremos más adelante), para comprobar si, a costa de una reducida pérdida de precisión en la predicción, supone un significativo ahorro de tiempo de ejecución el utilizar este segundo tipo de variables frente a las variables binarias.

Con esto lo que se busca es comprobar si es posible una evolución en los modelos actuales utilizados por las entidades financieras a la hora de predecir posibles clientes impagadores, y comprobar si nuestros resultados se encuentran en consonancia con los resultados obtenidos en estudios anteriores.

Además, como ningún recurso es inagotable, también se adjuntará el tiempo de ejecución de cada método para proporcionar una percepción de los costes en tiempo que lleva aplicar cada uno de los métodos. De esta forma las conclusiones de los modelos no serán sólo en función de su calidad, sino también en función de su tiempo de ejecución.

La justificación del proyecto se basa en la realidad de que los modelos de riesgo tradicionales sólo utilizan regresión logística ya que es un modelo lineal y son sólo los modelos lineales los que, en principio, permiten crear tarjetas de puntuación con variables tramificadas. Este tipo de tarjetas son las utilizadas tradicionalmente por las entidades financieras porque son las que permiten una interpretación sencilla sin conocimientos técnicos, es decir, “sólo sumando los puntos”.

Desde hace algunos años han ido apareciendo nuevos modelos de predicción basados en machine learning, deep learning, etc. Y en ocasiones estos modelos han proporcionado mejores resultados que los modelos de regresión logística. [2]

Para este proyecto queremos lograr encontrar el mejor modelo de predicción, un modelo que no necesariamente tendrá por qué ser de regresión logística, y que por lo tanto el mejor modelo pueda ser no lineal.

# 3. REVISIÓN DE LA LITERATURA

Enfrentar modelos de regresión logística junto con modelos de redes neuronales, es un proyecto que actualmente se está llevando a cabo en investigaciones de distintos ámbitos debido a la eficacia de ambos métodos de predicción.

A través de diversos estudios [2] se ha descubierto que en tareas de predicción las redes neuronales y los modelos de regresión múltiple tienden a rendir por igual, pero en las tareas de clasificación en todo tipo de condiciones las redes neuronales rinden mejor que los modelos estadísticos de análisis discriminante y regresión logística.

Así mientras algunos trabajos empíricos no encuentran diferencias entre los resultados hallados por unos y otros modelos *(Croall y Mason, 1992; Michie et al, 1994; Ripley, 1993; Thrun, Mitchell y Cheng, 1991)*, otros resultados tienden a apoyar una ligera superioridad de las redes neuronales sobre las técnicas estadísticas tradicionales *(ver p.e. Garson, 1991; Huang y Lippman, 1987; White, 1994)*.

Los expertos en redes aducen que pese a que las redes neuronales a priori son capaces de asociar cualquier patrón de entrada con cualquier patrón de salida, su rendimiento depende del ajuste heurístico de numerosos parámetros (número de unidades de entrada, salida y ocultas; funciones de activación: lineal, sigmoidal, tangencial,...; regla de aprendizaje: Hebb, delta, retropropagación,...; coeficientes de aprendizaje y momentum, etc), ajustes que no siempre garantizan la solución deseada, dada además la estructura de "caja negra" *(Cherkassky, Friedman y Wechler, 1994)* de este tipo de modelos.

Las redes neuronales tienden a producir una proporción de clasificaciones correctas superior a la obtenida por los modelos estadísticos tradicionales (sin que existan diferencias entre regresión logística y análisis discriminante), e independientemente del patrón de correlaciones que mantengan las variables. Por otra parte, analizando el efecto de las distintas condiciones de relacionalidad, se observa la misma tendencia de resultados que la observada en la tarea de predicción.

Al igual que en el análisis anterior (modelos de clasificación binaria) estos resultados muestran que las redes neuronales consiguen una proporción de clasificaciones correctas significativamente superior a la obtenida por la técnica de análisis discriminante, indistintamente del patrón de correlaciones que mantengan las variables.

Únicamente en la tarea de predicción cuantitativa y bajo condiciones idóneas de aplicabilidad (condiciones que por otro lado pocas veces satisfacen los datos procedentes de investigaciones aplicadas), el procedimiento clásico de regresión obtuvo mejores resultados (del orden de un punto porcentual) que las redes neuronales. En el resto de condiciones, las redes neuronales y los modelos de regresión múltiple rinden por igual en este tipo de tarea. Estos resultados coinciden básicamente con los hallados en otros trabajos como el de Wilson y Hardgrave (1995) quienes no encontraron diferencias entre redes neuronales y modelos de regresión en una tarea de predicción. Esto sugeriría que un investigador, junto con su criterio, debería realizar un estudio previo del patrón de correlaciones que subcomprenden las variables predictoras antes de decidirse por utilizar un modelo de regresión o una red neuronal. Sólo en el caso de unas condiciones idóneas podría ser aconsejable utilizar modelos de regresión, pudiendo utilizar indistintamente redes neuronales y técnicas de regresión en todos los demás casos.

Sin embargo, en todo tipo de tareas de clasificación (binaria o no) las redes neuronales obtuvieron mejores resultados que los modelos estadísticos convencionales (bien análisis discriminante, bien regresión logística). La superioridad de las redes queda especialmente de manifiesto en la condición que satisface de forma idónea las condiciones de aplicabilidad de las técnicas estadísticas convencionales.

En este sentido coinciden los resultados hallados por otros autores como Fogelman (1994) quien obtuvo mejores resultados utilizando redes neuronales que modelos ARIMA en la predicción de series temporales. De igual modo, Navarro (1998) ha demostrado cómo la estimación de valores faltantes (missing values) a través de redes neuronales suele ser mejor que la conseguida por procedimientos estadísticos convencionales.

Este patrón de resultados favorece claramente a las redes neuronales sobre los modelos estadísticos clásicos como técnicas de clasificación, afirmación que se ve fortalecida por su mayor versatilidad de uso, al no depender su aplicabilidad del cumplimiento de los supuestos teóricos sobre los que se basan las técnicas estadísticas (normalidad, homocedasticidad, independencia ...). Otra ventaja adicional de las redes neuronales sobre los modelos estadísticos, importante en investigación aplicada, es que admiten como variables de entrada conjuntos mixtos de variables cuantitativas y cualitativas.

Un aspecto menor que llama la atención en relación a la tarea de clasificación binaria son los resultados similares obtenidos a través de las dos técnicas estadísticas empleadas (análisis discriminante y regresión logística), lo que contrasta con los resultados obtenidos por Press y Wilson (1978) para quienes la regresión logística es una mejor herramienta de categorización que el análisis discriminante, al no depender del cumplimiento de los supuestos paramétricos de la estimación mínimo-cuadrática.

De todo lo anterior no se concluye que se ampare el destierro de los métodos estadísticos convencionales a la hora de realizar tareas de clasificación, pues las redes neuronales pese a su mejor rendimiento, presentan una serie de inconvenientes que el investigador debe de sopesar antes de decidirse por su utilización. En primer lugar, el

entrenamiento de una red neuronal es un proceso demasiado creativo (Martín y Sanz, 1997) que generalmente se soluciona por un método heurístico de ensayo y error. Además, y esto es un aspecto especialmente delicado, la calidad de las soluciones dadas por la red elegida no puede ser siempre garantizada (Cherkassky et al, 1994) debido a su naturaleza de "caja negra" y otras causas: interferencia catastrófica, sobreaprendizaje, mínimos locales, etc. No hay que olvidar que una red neuronal no da información explícita sobre la importancia relativa de los distintos predictores, ni tampoco obviar el elevado costo computacional requerido en el entrenamiento de las redes neuronales, muy superior al de los modelos estadísticos. En última instancia deberá ser pues el investigador quien, sopesando tales limitaciones, decida si compensa decidirse por la utilización de una arquitectura de uno u otro tipo.

La consideración de todo lo dicho nos lleva a sugerir, con Fogelman (1994), Cherkassky et al (1994), Sarle (1994, 1998) y otros muchos autores, que técnicas estadísticas y redes neuronales deben comenzar a ser usadas conjuntamente, tal y como comienzan a utilizarse en determinadas aplicaciones técnicas (ver Martín y Sanz, 1997). De este modo, la estadística, centrada tradicionalmente en funciones lineales, y las redes neuronales más acostumbradas a tratar con problemas mal definidos o no lineales (Smith, 1993), se verán mutuamente enriquecidas.

# 4. TÉCNICAS ESTADÍSTICAS DE ANÁLISIS DE MACHINE LEARNING

Para el desarrollo de este proyecto construiremos diferentes modelos basados en el machine learning, como son la regresión logística, las redes neuronales, SVM y técnicas de aprendizaje basadas en árboles.

El machine learning es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de datos, identificar patrones y tomar decisiones con la mínima intervención humana.

Dos de los métodos de machine learning más ampliamente adoptados son el aprendizaje supervisado y el aprendizaje no supervisado, aunque también existen otros métodos de machine learning como el aprendizaje semi-supervisado y el aprendizaje con refuerzo [3]. Ésta es una descripción de los tipos más populares de aprendizaje.

- Aprendizaje supervisado

Los algoritmos de **aprendizaje supervisado** son entrenados utilizando ejemplos etiquetados, como una entrada donde se conoce el resultado deseado. Por ejemplo, un conjunto de datos podría tener puntos de datos etiquetados como “A” (acertados) o “F” (fallidos). El algoritmo de aprendizaje recibe un conjunto de datos de entrada junto con los resultados correctos correspondientes, aprende comparando su resultado real con los resultados correctos para encontrar errores y modifica el modelo en consecuencia. A través de métodos como la clasificación, regresión, predicción y aumento de

gradiente, el aprendizaje supervisado utiliza patrones para predecir los valores de la etiqueta en datos no etiquetados adicionales. El aprendizaje supervisado se utiliza comúnmente en aplicaciones donde datos históricos predicen eventos futuros probables. Por ejemplo, puede anticipar cuándo es probable que transacciones con tarjetas de crédito sean fraudulentas o qué cliente de una aseguradora tiene la probabilidad de iniciar un reclamo.

Existen fundamentalmente dos tipos de aprendizaje supervisado [4]:

1) **Regresión.** Tiene el objetivo de predecir valores continuos. Con el valor numérico de las etiquetas se utilizan diferentes variables para obtener los datos que interesan. Este tipo de aprendizaje sirve para una serie de finalidades concretas como predecir el precio de un producto, una propiedad, el valor del stock de una tienda, etc.

2) **Clasificación.** El algoritmo encuentra diferentes patrones y clasifica los elementos en diferentes grupos. Este modelo busca sacar conclusiones de los valores observados, ya que una o más entradas intentan predecir el valor de uno o más resultados. Un claro ejemplo es el filtro de correos electrónicos para ver si son spam o no: solo hay dos resultados posibles, ya que cuando analizamos los datos de la transacción podemos dividirlos en solo dos categorías, ya sea “Autorizado” o “Fraudulento”.

- Aprendizaje no supervisado

El **aprendizaje no supervisado** se utiliza contra datos que no tienen etiquetas históricas. No se da la "respuesta correcta" al sistema y el algoritmo debe descubrir lo que se muestra. El objetivo es explorar los datos y encontrar alguna estructura en su interior. El aprendizaje no supervisado funciona bien con datos de transacciones. Por ejemplo, puede identificar segmentos de clientes con atributos similares que después puedan ser tratados de manera semejante en campañas de marketing. O bien pueden encontrar los atributos principales que separan los segmentos de clientes. Algunas técnicas populares incluyen mapas con organización automática, mapping del vecino más cercano, k-means clustering y descomposición de valores singulares. Estos algoritmos se pueden utilizar también para segmentar temas de texto, recomendar elementos e identificar valores atípicos de datos.

En este caso también existen fundamentalmente dos tipos de aprendizaje no supervisado:

1) **Análisis clúster.** Extrae referencias de conjuntos de datos como parte de los datos de entrada, sin respuestas etiquetadas y sin conocer nada de cómo se clasifican. Se trata de clasificar un conjunto de datos en grupos lo más homogéneos entre sí y lo más heterogéneos entre ellos. Un ejemplo claro es la segmentación que llevan a cabo las empresas para sus campañas, dividiéndolas por tipo de cliente.

2) **Reducción de la dimensión.** Se trata de reducir el número de variables aleatorias mediante la obtención de un conjunto de variables principales, eliminando así variables irrelevantes y mejorando el rendimiento computacional. Se utiliza como método intermedio para reducir la complejidad del sistema. Donde más se utiliza es en el ámbito de la biología y la medicina, donde existen muchísimas variables y se requiere de la eliminación del máximo posible de ellas.

- Aprendizaje semi-supervisado

El **aprendizaje semi-supervisado** se utiliza para las mismas aplicaciones que el aprendizaje supervisado. Sin embargo, utiliza datos etiquetados y no etiquetados para entrenamiento, por lo general, una pequeña cantidad de datos etiquetados con una gran cantidad de datos no etiquetados (porque los datos no etiquetados son menos costosos y se requiere menos esfuerzo en su obtención). Este tipo de aprendizaje se puede utilizar con métodos como la clasificación, regresión y predicción. El aprendizaje semi-supervisado es de utilidad cuando el coste asociado con el etiquetado es demasiado alto para permitir un proceso de entrenamiento completamente etiquetado. Algunos ejemplos iniciales de este tipo de aprendizaje incluyen la identificación del rostro de una persona en una cámara Web.

- Aprendizaje con refuerzo

El **aprendizaje con refuerzo** se utiliza a menudo para robótica, juegos y navegación. Con el aprendizaje con refuerzo, el algoritmo descubre a través de ensayo y error qué acciones producen las mayores recompensas. Este tipo de aprendizaje tiene tres componentes principales: el agente (el que aprende o toma decisiones), el entorno (todo con lo que interactúa el agente) y las acciones (lo que el agente puede hacer). El objetivo es que el agente elija las acciones que maximicen la recompensa esperada en cierta cantidad de tiempo. El agente logrará la meta mucho más rápido si aplica una buena política, de modo que el objetivo en el aprendizaje con refuerzo es aprender la mejor política.

Las distintas técnicas de análisis supervisado que utilizaremos para el desarrollo de este proyecto son las siguientes:

- Regresión Logística.

- Redes Neuronales.

- Técnicas basadas en árboles como son:

    - Bagging.
    - Random Forest.
    - Gradient Boosting.

- Support Vector Machine.

## 4.1 REGRESIÓN LOGÍSTICA

La segmentación puede definirse como el proceso de dividir un todo (población, consumidores, etc) en grupos uniformes más pequeños con características similares denominados segmentos, según los valores de determinadas variables llamadas variables dependientes o variables respuesta. Debido a esta similitud de los elementos dentro de cada segmento, es probable que respondan de modo similar frente a determinadas estrategias como marketing, ventas, precios, distribución, tratamientos, etc.

Cuando el criterio es una variable cuantitativa se suele hablar de problemas de predicción o estimación, mientras que cuando es una variable cualitativa o categórica se habla entonces de problemas de clasificación. Tradicionalmente la solución a estos problemas se ha llevado a cabo desde la óptica de modelos estadísticos de regresión: regresión simple o múltiple para problemas de predicción (variable cuantitativa); análisis discriminante o modelos de regresión logística para problemas de clasificación (variable cualitativa).

Existen varias opciones para estimar un modelo de regresión de entre los que destacan por su facilidad de aplicación e interpretación, el modelo de regresión lineal y el modelo de regresión logística. Teniendo en cuenta el tipo de variable que deseemos estimar aplicaremos un modelo de regresión u otro, así cuando la variable dependiente es una variable continua el modelo de regresión más frecuentemente utilizado es la regresión lineal, mientras que cuando la variable de interés es dicotómica (únicamente toma dos valores) se utiliza la regresión logística.

Dado que en nuestro caso la variable respuesta es dicotómica, utilizaremos inicialmente la regresión logística la cual trata de encontrar el mejor ajuste para describir las relaciones entre la variable respuesta y un grupo de variables explicativas.

Dentro de los modelos de elección discreta en los que el conjunto de elección tiene sólo dos alternativas posibles mutuamente excluyentes, consideramos el modelo lineal de probabilidad, el modelo Logit y el modelo Probit.

Los modelos Logit y Probit tienen como modelos de respuesta binaria el siguiente modelo:

$$P(Y = 1|X_1, X_2, \dots, X_k) = G(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k)$$

En el que, para evitar los problemas del modelo lineal de probabilidad, se especifica como $Y = G(X\beta)$, donde G es una función que toma valores estrictamente entre 0 y 1 para todos los números reales. Según las diferentes definiciones de G tenemos los distintos modelos de elección binaria.

En nuestro caso, como las variables en estudio tienen todas forma binaria (las que inicialmente no la tenían les hemos realizado las transformaciones oportunas para que así sea). Utilizaremos los modelos Logit que tiene la forma $G(z) = \frac{e^z}{1+e^z}$ y cuya expresión es:

$$Y = P(Y = 1/X = x) = G(z) = G(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_k X_k}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_k X_k}}$$

Los modelos Logit al igual que los Probit son modelos no lineales, por lo que no se pueden estimar por MCO (mínimos cuadrados ordinarios) y se deben emplear métodos de máxima verosimilitud.

La transformación Logit tiene la siguiente forma:

$$g(x) = ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = ln\left[\frac{\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}}{1-\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}}\right] = ln\left[\frac{\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}}{\frac{1+e^{\beta_0+\beta_1 x}-e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}}\right] = ln\left[e^{\beta_0+\beta_1 x}\right] = \beta_0 + \beta_1 x$$

Para contrastar la hipótesis nula de que un conjunto de parámetros es igual a cero se pueden emplear los siguientes procedimientos:

- Estadístico de Wald.
- Contraste de razón de verosimilitudes (Likelihood Ratio test).

Las medidas de la bondad de ajuste que tenemos son:

- **Porcentaje de predicciones correctas.** Para cada i calculamos la probabilidad estimada de que $Y_i = 1$:

$$\hat{P}_i = \hat{P}(Y_i = 1|X_{1i}, \dots, X_{ki}) = G(\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \cdots + \hat{\beta}_k X_{ki})$$

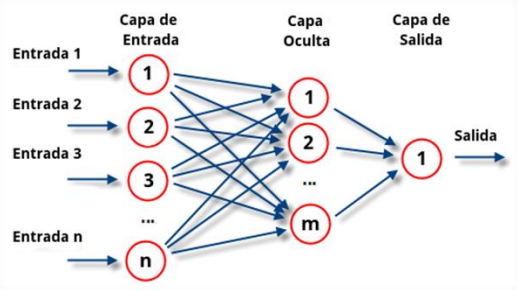
Si $\hat{P}_i > 0.2227$ nuestra predicción será que $Y_i = 1$ y si $\hat{P}_i \leq 0.2227$ nuestra predicción será que $Y_i = 0$ (se utiliza este número en vez del 0.5, ya que este es el porcentaje en tanto por uno de individuos que han causado DEFAULT en nuestra base de datos de 3.000 observaciones). El porcentaje de veces en que el valor de $Y_i$ observado coincida con la predicción, es el porcentaje de predicciones correctas.

- **Pseudo - $R^2$ (de McFadden).** Está entre 0 y 1.

- **Criterios de Información.** Medidas que tratan de buscar un equilibrio entre la bondad del ajuste, medida en base al valor del logaritmo de la función de verosimilitud y una especificación parsimoniosa del modelo. Algunos ejemplos son Akaike (AIC), Schwarz (SC) y Hannan-Quinn (HQ). Se escoge el modelo con menor valor del criterio de información.

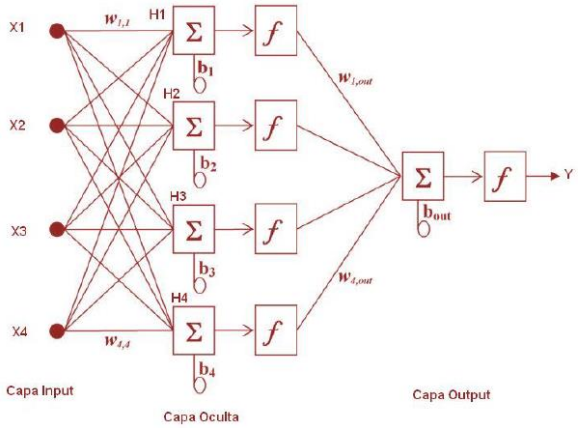
## 4.2 REDES NEURONALES

Las redes neuronales son una técnica de machine learning que simula las interconexiones de las neuronas cerebrales a la hora de procesar la información y de predecir la variable respuesta. Estas redes están compuestas fundamentalmente por tres elementos:




- **Nodos Input** o de entrada: Variables independientes del modelo.
- **Nodos Output** o de salida: Variables dependientes del modelo (puede haber más de una).
- **Capa Oculta**: Capa con nodos ocultos (variables artificiales que no existen como tal en los datos).

Una red neuronal es en realidad un modelo de la forma $y = f(x_1, x_2, x_3 \dots)$ donde la función $f$ es por lo general no lineal.




Esta es una red neuronal con 4 nodos input $x_1, x_2, x_3, x_4$, un nodo output $Y$, y una capa oculta con 4 nodos ocultos $H1, H2, H3, H4$.

La capa input se conecta a la capa oculta mediante la función de combinación, representada por Σ, donde los pesos $w_{ij}$ hacen el papel de parámetros a estimar.

Este tipo de técnica es la más adecuada en aquellos casos en los que se desconocen las leyes que relacionan las variables input con las output, en los que no hay linealidad o hay funciones desconocidas entre variables input y output, en los casos en que hay variables latentes, modelos a trozos según rangos de variables, en los que hay información redundante, información localizada importante en algunos rangos o variables, o incluso en los casos en los que hay datos missing. Y como esta técnica proporciona modelos tan flexibles que pueden abordar todas estas situaciones, es lo que los hace los más adecuados para estos casos.

Pero para utilizar una red neuronal con garantías se requieren muchas observaciones, ya que al no haber inferencia propiamente dicha se necesitan datos test para validar el modelo.

Y con todo esto, a pesar de que es una técnica de análisis eficaz, es importante saber que las redes neuronales son una caja negra y por lo tanto es difícil extraer información de ellas a partir de sus predicciones.

## 4.3 TÉCNICAS BASADAS EN ÁRBOLES

El aprendizaje basado en árboles de decisión utiliza un árbol de decisión como un modelo predictivo que mapea observaciones sobre un artículo hasta alcanzar conclusiones sobre el valor objetivo del artículo [5]. Es uno de los enfoques de modelado predictivo utilizado en estadística, minería de datos y aprendizaje automático (machine learning). Los modelos de árbol donde la variable objetivo puede tomar un conjunto finito de valores se denominan árboles de clasificación. En estas estructuras de árbol, las hojas representan etiquetas de clase y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase. Los árboles de decisión donde la variable objetivo puede tomar valores continuos (por lo general números reales) se llaman árboles de regresión.

En análisis de decisión, un árbol de decisión se puede utilizar para representar visualmente y de forma explícita decisiones y toma de decisiones. En minería de datos, un árbol de decisión describe datos, pero no las decisiones, más bien el árbol de clasificación resultante, puede ser un usado como entrada para la toma de decisiones.

Para la construcción de un árbol de regresión o clasificación se utilizan algoritmos iterativos que consisten en dividir los datos en regiones basadas en intervalos de las variables dependientes. Los árboles dan lugar a valores constantes según estas regiones.

Por lo general, habrá que encontrar las regiones que minimicen una función de error dada como puede ser el ASE (Alpha Standard Error) en regresión y la tasa de fallos en clasificación.

La combinatoria (regiones posibles) es astronómica, por lo cual se va dividiendo en regiones y subregiones de forma jerárquica mediante divisiones subóptimas. El orden sería el siguiente:

1) Se comienza por buscar un punto de corte en cada variable (obteniendo dos intervalos) y se observa el error cometido en la predicción al fijar los valores constantes. Se escoge la variable y el punto de corte óptimos, y si se trata de variables nominales, se establecen agrupaciones de categorías en lugar de puntos de corte.

2) Dentro de las divisiones obtenidas en el apartado anterior, se continúa subdividiendo el árbol hasta llegar a algún criterio de parada (número de hojas finales, por ejemplo).

En los árboles es necesario determinar a priori ciertos criterios para hacer o no divisiones o para parar el algoritmo. Aunque a priori el algoritmo parece simple, la casuística es muy compleja y no todos los programadores dan lugar al mismo resultado.

Las ventajas y desventajas de las técnicas basadas en árboles son las siguientes:

- Adaptabilidad a la forma funcional entre las variables objetivo y predictoras.
- Tratamiento automático de valores missing.
- Tratamiento automático de categorías poco representadas.
- Detección automática de regiones y puntos de corte (no tratada en otros algoritmos o técnicas).
- Resultados a menudo fáciles de interpretar.

- Poca capacidad predictiva y gran varianza.
- Sensibilidad a cambios en los datos, inestabilidad y poca robustez.
- Falta de suavidad (función escalonada) lo que a veces redunda en mayor error promedio de predicción en regresión.

Es importante saber que las desventajas de los árboles no se pueden solucionar mejorando las funciones de error o los algoritmos de construcción, pero sí combinando el resultado de muchos árboles.

### 4.3.1 BAGGING

Con el procedimiento Bagging la forma de actuar es la siguiente, dados los datos de tamaño *N*:

1) Repetir *m* veces i) y ii):

   i) Seleccionar *n* observaciones con reemplazamiento de los datos originales. Se admiten todo tipo de variaciones, tomar $n<N$ con o sin reemplazamiento, estratificación, etc.

   ii) Aplicar un árbol y obtener predicciones para todas las observaciones originales *N*. La complejidad del árbol a utilizar es un tema muy diverso. Probablemente lo mejor sean árboles débiles (pocas hojas finales) y muchas iteraciones *m*. Pero en algunas versiones se utilizan árboles desarrollados hasta el final sin prefijar el número de hojas o profundidad.

2) Promediar las *m* predicciones obtenidas en el apartado 1).

Si se trata de un problema de clasificación, pueden ser utilizadas dos estrategias:

a) Promediar las probabilidades estimadas y obtener una clasificación a partir de un punto de corte.

b) Clasificar en cada iteración y asignar a cada observación la clasificación mayoritaria entre todas las iteraciones (Majority voting).

Con cada submuestra se genera un modelo con el que se predicen los datos test. La predicción final será la media de las *m* diferentes predicciones. Al utilizarse diferentes submuestras, se reduce la dependencia de la estructura de los datos completos para construir el modelo y como consecuencia se reduce la varianza del modelo (y a menudo también el sesgo).

La idea del submuestreo es controlar el sobreajuste implícito en la selección rígida de variables y estimación fija de parámetros que caracteriza a los métodos directos.

En general, Bagging funciona bien en los siguientes casos:

- Cuando los modelos no están claros (muchas variables con relación débil pero estable con la variable dependiente, multiplicidad de modelos-opciones).
- Cuando existen relaciones no lineales (regresión) o separaciones no lineales (clasificación).
- Cuando existen interacciones ocultas, muchas variables categóricas, etc.

Los principales parámetros que controlar en Bagging son:

- El tamaño de las muestras *n,* y si se va a utilizar bootstrap (con reemplazo) o sin reemplazamiento.
- El número de iteraciones *m* a promediar.

Características de los árboles que son bastante influyentes:

- El número de hojas final o, en su defecto, la profundidad del árbol.
- El maxbranch (número de divisiones máxima en cada nodo).
- El p-valor para las divisiones en cada nodo. Más alto supone árboles menos complejos (es decir, más sesgo, menos varianza).
- El número de observaciones mínimo en una rama-nodo. Se puede ampliar para evitar sobreajuste (reducir varianza) o reducir para ajustar mejor (reducir sesgo).

Aunque se puede utilizar la técnica con algoritmos diferentes de los árboles, su influencia es mucho más grande con árboles al ser estos muy dependientes de los datos utilizados.

### 4.3.2 RANDOM FOREST

Random Forest es una modificación del Bagging que consiste en incorporar aleatoriedad en las variables utilizadas para segmentar cada nodo del árbol.

El procedimiento sería el siguiente, dados los datos de tamaño *N*:

1) Repetir *m* veces i), ii) y iii):

i) Seleccionar *n* observaciones con reemplazamiento de los datos originales.

ii) Aplicar un árbol de la siguiente manera: En cada nodo, seleccionar *p* variables de las *k* originales y de las *p* elegidas, escoger la mejor variable para la partición del nodo.

iii) Obtener predicciones para todas las observaciones originales *N*.

2) Promediar las *m* predicciones obtenidas en el apartado 1).

El algoritmo Random Forest da un paso más en soslayar el problema de selección de variables, evitando decidirse rígidamente por un set de variables y aprovechando a la vez las ventajas del Bagging. Se trata de incorporar dos fuentes de variabilidad (remuestreo de observaciones y de variables) para ganar en capacidad de generalización y reducir el sobreajuste conservando a la vez la facultad de ajustar bien relaciones particulares en los datos (interacciones, no linealidad, cortes, problemas de extrapolación, etc.)

Principales parámetros a controlar en Random Forest:

- El tamaño o porcentaje de las muestras *n* y si se va a utilizar bootstrap (con reemplazo) o sin reemplazamiento.
- El número de iteraciones *m* a promediar
- El número de variables *p* a muestrear en cada nodo (si es igual al número inicial de variables *k* el Random Forest es equivalente al Bagging)

### 4.3.3 GRADIENT BOOSTING

El algoritmo Gradient Boosting consiste en repetir la construcción de árboles de regresión o clasificación, modificando ligeramente las predicciones iniciales cada vez, intentando ir minimizando los residuos en la dirección de decrecimiento. Al plantear diferentes árboles constantemente, el proceso va ajustando cada vez más las predicciones a los datos y de esta forma unos árboles corrigen a otros con lo cual la flexibilidad y adaptación del método mejora respecto a la construcción de un único árbol.

Este proceso ha de ser monitorizado en principio mediante early stopping para determinar el número de iteraciones y por lo tanto necesitará datos de validación. Aunque a menudo el early stopping no es necesario pues la convergencia es lenta y van a la par los errores en training y validación (no hay sobreajuste).

Principales parámetros a controlar en Gradient Boosting:

- La constante de regularización *v* (shrink). Normalmente debe de estar entre (0.001 y 0.3) cuanto más alta, más rápido converge, pero demasiado alta es poco preciso y si se pone muy baja, hay que utilizar muchas iteraciones para que converja.

- El número de iteraciones *m* no es importante en general, pero hay que tener en cuenta la constante *v* de regularización.

Las ventajas y desventajas de Gradient Boosting son las siguientes:

- Es invariante frente a transformaciones monótonas, no es necesario realizar transformaciones logarítmicas, etc.
- Tiene un buen tratamiento de missings, variables categóricas, etc.
- Es muy fácil de implementar, tiene relativamente pocos parámetros a monitorizar (número de hojas o profundidad del árbol, tamaño final de hojas, parámetro de regularización…).
- Tiene una gran eficacia predictiva con un algoritmo muy competitivo.
- Es robusto respecto a variables irrelevantes, robusto respecto a colinealidad y detecta interacciones ocultas.
- A mayor complejidad de los datos (interacciones, missings, no linealidad, muchas variables categóricas, muchas variables en general), es más posible que el algoritmo Gradient Boosting supere a otros algoritmos.

Como todos los métodos basados en árboles, dependiendo de los datos puede ser superado por otras técnicas más sencillas. Con datos relativamente sencillos (pocas variables, no missings, no interacciones, linealidad (regresión) o separabilidad lineal (clasificación)), Gradient Boosting o Random Forest no tienen nada nuevo que aportar y pueden ser preferibles modelos más sencillos (regresión, regresión logística, análisis discriminante) o modelos Ad-Hoc que adapten aspectos concretos como la no linealidad (como las redes neuronales).

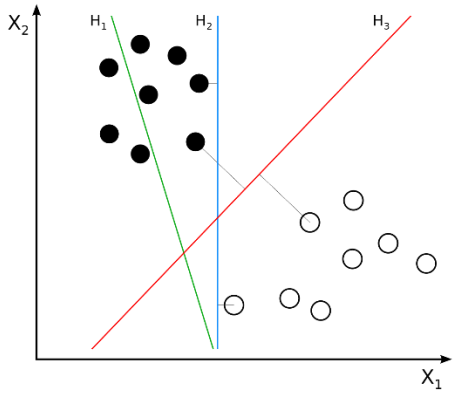
## 4.4 SVM

La técnica Support Vector Machine lo que hace es plantear el problema de la separación lineal de clases mediante métodos algebraicos, buscando el hiperplano de separación y se basa fundamentalmente en tres ideas [6].

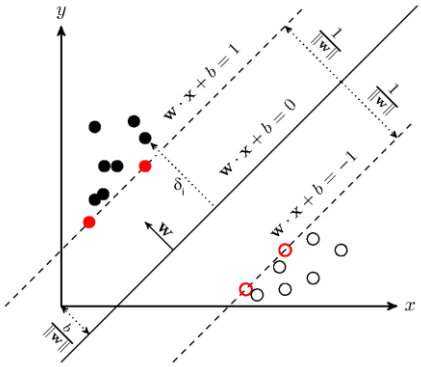
1. Maximal Margin

No se trata solamente de separar las clases por un hiperplano (función lineal), sino de incluir en la decisión de la construcción del separador el concepto de separador con máximo margen. Esto a menudo mejora tanto el sesgo como la varianza de los resultados.

Se trata de hallar el vector de parámetros *w* que maximice el margen. Las ecuaciones de los hiperplanos que delimitan el margen son $w \cdot x = 1$ y $w \cdot x = -1$. Denotando *y* como (-1,1) en un problema de clasificación, hay que maximizar la distancia entre los dos hiperplanos de separación $\left(\frac{2}{\|w\|}\right)$. Se suele hacer con métodos clásicos de optimización (Lagrange, Kuhn Tucker).

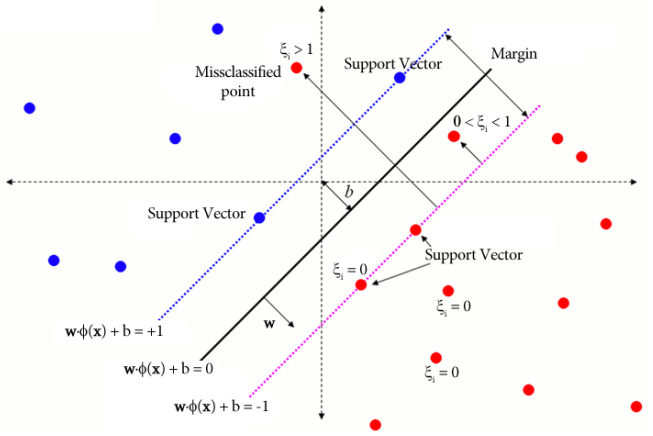



$H_1$ no separa las clases.
$H_2$ separa las clases pero con poco margen.
$H_3$ separa las clases con el máximo margen.




$$\arg\min = \frac{\|w\|^2}{2}$$
$$i = 1, \dots, n \rightarrow y_i(w \cdot x_i - b) \geq 1$$

2. Soft Margin

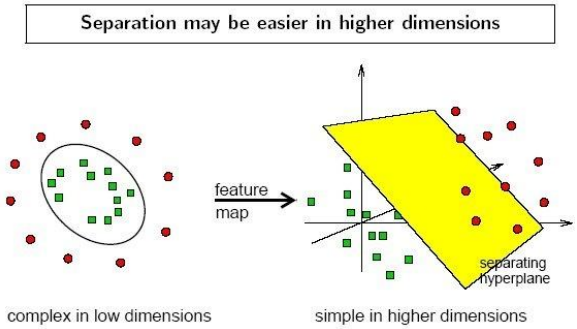



La separación perfecta no suele existir, y por lo tanto es necesario permitir observaciones mal clasificadas por los separadores para no incurrir en sobreajustes. Esto implica añadir una variable $\xi$ de "residuo" y una constante $C$ de regularización del margen, que están relacionadas inversamente con la anchura del margen y el "permiso para fallar" que vamos a permitir en la construcción del separador. A mayor $C$ (menores "residuos" $\xi_i$) y un menor margen; y a menor $C$ (permitimos mayores "residuos" $\xi_i$), y más permiso para fallar y más margen.

$$\arg\min = \left\{\frac{\|w\|^2}{2} + C\sum_{i=1}^{n}\xi_i\right\}$$
$$i = 1, \dots, n \rightarrow y_i(w \cdot x_i - b) \geq 1 - \xi_i \qquad \xi_i \geq 0$$

3. Kernel

A menudo ocurre que la separación entre clases en muchos problemas no es lineal, y una idea para aplicar a pesar de todo un algoritmo de separación lineal es trabajar en un espacio de dimensión superior donde sí tenga sentido la separación lineal.



Por ejemplo, supongamos que a la siguiente tabla le añadimos una variable nueva, $x_2^2$ calculada directamente:

| y | $x_1$ | $x_2$ |
|---|---|---|
| -1 | 3 | 4 |
| 1 | 4 | 6 |
| … | … | … |

→

| y | $x_1$ | $x_2$ | $x_2^2$ |
|---|---|---|---|
| -1 | 3 | 4 | 16 |
| 1 | 4 | 6 | 36 |
| … | … | … | … |

Entonces en ese espacio de 3 dimensiones sí se podría separar linealmente. Para ello, se pueden introducir nuevas funciones de las variables ($x^2, x^3, \mathrm{x}_1\mathrm{x}_2$, etc). Lo que aumenta la dimensión del vector de variables independientes, y en ese caso sería más fácil para el algoritmo encontrar separaciones lineales y un buen tamaño de margen.

El problema es que este aumento de dimensión hace a menudo impracticables los cálculos, pero aquí interviene el "truco Kernel" ("The Kernel trick"): cualquier algoritmo que dependa solo de los productos escalares (como es el caso del SVM) permite trabajar computacionalmente en una dimensión controlada a través de una función llamada Kernel, que tiene que cumplir: $K(x, y) = \langle \varphi\,(\mathrm{x}), \varphi\,(\mathrm{y}) \rangle$

La función ϕ(x) representa una función que extrapola de la dimensión original a una superior.

Las ventajas y desventajas de la técnica Support Vector Machine son las siguientes:

- Muy flexible, sobre todo por el truco Kernel. Hay versiones para regresión y para clasificación multinomial, y ambas tienen buenas propiedades.
- Puede competir en datos separables linealmente con la regresión logística.
- Buen rendimiento en la clasificación de imágenes.

- Lento a veces en la optimización.
- Dificultad en seleccionar la función Kernel y sus parámetros asociados.
- En los algoritmos clásicos son un problema los valores missing, categorías poco representadas, variables irrelevantes, etc.

# 5. BASE DE DATOS

Para poder realizar este estudio se ha utilizado un conjunto de datos que ofrece la plataforma KAGGLE. Este conjunto de datos está formado por 30.000 observaciones y contiene información sobre pagos predeterminados, factores demográficos, datos crediticios, historial de pagos y estados de cuenta de los clientes de tarjetas de crédito en Taiwán desde abril de 2005 hasta septiembre de 2005 [7].

Las variables con las que cuenta nuestro repositorio de datos son las siguientes.

- SOCIODEMOGRÁFICAS

**ID**: Identificador de cada cliente.

**SEX**: Género (*1=hombre, 2=mujer*)

**EDUCATION**: Nivel de educación (*0=otros, 1=graduado escolar, 2=universitario, 3=educación secundaria, 4=otros, 5=desconocido, 6=desconocido*)

**MARRIAGE**: Estado marital (*0=otro, 1=casado, 2=soltero, 3=divorciado*)

**AGE**: Edad en años.

- ECONÓMICAS

**LIMIT_BAL**: Cantidad de crédito otorgado en dólares taiwaneses (incluye crédito individual y familiar / suplementario)

**PAY_0**: Estado de reembolso en septiembre, 2005 (*-2=No consumido, -1=Pagado debidamente, 0=Crédito revolvente, 1=Pago atrasado por un mes, 2= Pago atrasado por dos meses, ... 8= Pago atrasado por ocho meses, 9= Pago atrasado por nueve meses o más*)
**PAY_2**: Estado de reembolso en Agosto, 2005 (escala igual que arriba)
**PAY_3**: Estado de reembolso en Julio, 2005 (escala igual que arriba)
**PAY_4**: Estado de reembolso en Junio, 2005 (escala igual que arriba)
**PAY_5**: Estado de reembolso en Mayo, 2005 (escala igual que arriba)
**PAY_6**: Estado de reembolso en Abril, 2005 (escala igual que arriba)

**BILL_AMT1**: Importe del estado de la cuenta en Septiembre, 2005 (dólar taiwanés)
**BILL_AMT2**: Importe del estado de la cuenta en Agosto, 2005 (dólar taiwanés)
**BILL_AMT3**: Importe del estado de la cuenta en Julio, 2005 (dólar taiwanés)
**BILL_AMT4**: Importe del estado de la cuenta en Junio, 2005 (dólar taiwanés)
**BILL_AMT5**: Importe del estado de la cuenta en Mayo, 2005 (dólar taiwanés)
**BILL_AMT6**: Importe del estado de la cuenta en Abril, 2005 (dólar taiwanés)

**PAY_AMT1**: Importe del pago anterior en Septiembre, 2005 (dólar taiwanés)
**PAY_AMT2**: Importe del pago anterior en Agosto, 2005 (dólar taiwanés)
**PAY_AMT3**: Importe del pago anterior en Julio, 2005 (dólar taiwanés)
**PAY_AMT4**: Importe del pago anterior en Junio, 2005 (dólar taiwanés)
**PAY_AMT5**: Importe del pago anterior en Mayo, 2005 (dólar taiwanés)
**PAY_AMT6**: Importe del pago anterior en Abril, 2005 (dólar taiwanés)

**VAR25 (default.payment.next.month)**: Falta de pago (*1=si, 0=no*)

# 6. ANÁLISIS DESCRIPTIVO, DEPURACIÓN Y TRANSFORMACIÓN INICIAL DE DATOS

Previamente dentro de cualquier tipo de análisis estadístico, primero se debe realizar una parte de depuración y análisis descriptivo de los datos. Esta parte la realizaremos a través del programa "SAS Enterprise Miner".

Pero antes de comenzar con la depuración y el análisis, hemos de realizar primero una pequeña reconstrucción de las variables.

## 6.1 RECONSTRUCCIÓN DE VARIABLES

Para que los resultados obtenidos de un estudio se puedan interpretar y aporten información relevante, es necesario que cada categoría de cada variable esté bien definida y no haya confusión con las demás, ya que si esto sucede la interpretación podría ser dudosa o estaría basada en matices entre distintas variables o categorías, que para la finalidad del estudio no son relevantes. Es por esto por lo que cuando una variable tiene un número excesivo de categorías, o cuando muchas variables son similares entre sí o aportan la misma información, se contempla la recategorización de categorías dentro de una misma variable, o la unión de distintas variables para de esta forma evitar confusión y simplificar tanto el estudio como la interpretación de los resultados.

Los cambios que realizamos en las variables son los siguientes:

- Lo primero que hacemos es que la variable objetivo que originalmente se llama VAR25 le cambiamos el nombre por "DEFAULT", para que de esta forma tenga un nombre acorde a su significado ya que lo que indica esta variable es qué individuo ha sufrido impago.

- Para la variable EDUCATION, la descripción de sus categorías indica que esta variable cuenta con siete niveles.

| EDUCATION | | | | |
|---|---|---|---|---|
| **EDUCATION** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **0** | 14 | 0.05 | 14 | 0.05 |
| **1** | 10585 | 35.28 | 10599 | 35.33 |
| **2** | 14030 | 46.77 | 24629 | 82.10 |
| **3** | 4917 | 16.39 | 29546 | 98.49 |
| **4** | 123 | 0.41 | 29669 | 98.90 |
| **5** | 280 | 0.93 | 29949 | 99.83 |
| **6** | 51 | 0.17 | 30000 | 100.00 |

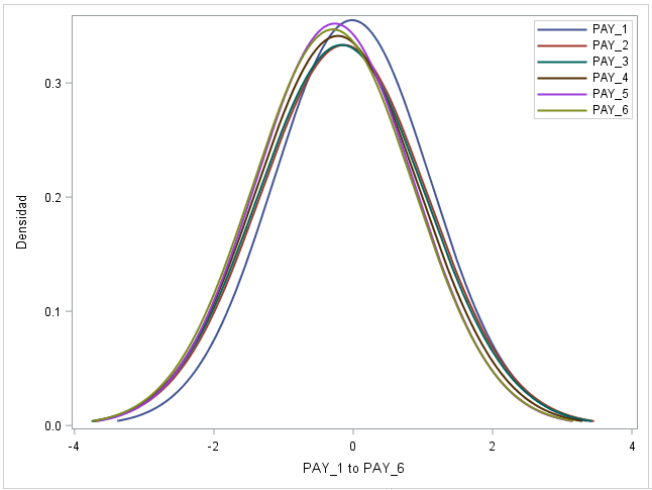
*Tabla 6.1.1*

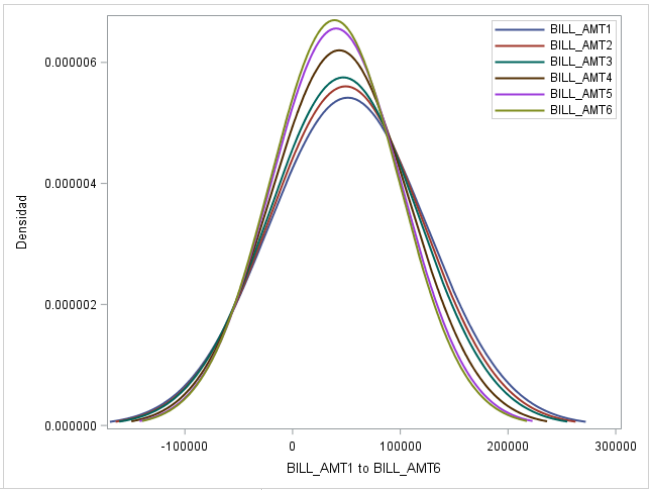
| EDUCATION | | | | |
|---|---|---|---|---|
| **EDUCATION** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **0** | 468 | 1.56 | 468 | 1.56 |
| **1** | 10585 | 35.28 | 11053 | 36.84 |
| **2** | 4917 | 16.39 | 15970 | 53.23 |
| **3** | 14030 | 46.77 | 30000 | 100.00 |

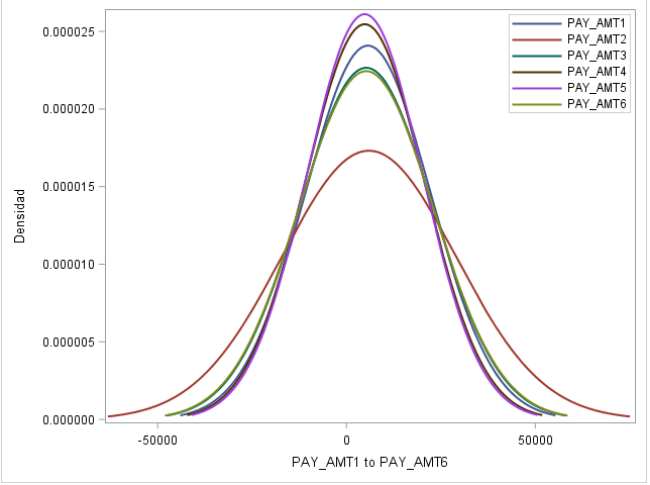
Pero tanto el nivel cero como el cuatro hacen referencia al nivel de educación “Otros”, así que unimos ambas categorías en una sola, la cero. Y viendo que tanto la categoría cinco como la seis hacen referencia a un nivel de educación desconocido, también recodificamos esas categorías uniéndolas a la categoría cero. De esta forma la variable pasa a tener sólo cuatro niveles, del cero al tres. Además, intercambiamos las categorías 2 y 3 ya que se sobreentiende que en un tipo de variable de este estilo a mayor categoría mayor nivel de estudios y como la categoría 2 hace referencia a nivel educacional universitario mientras que la categoría 3 hace referencia a nivel de estudios de educación secundaria, intercambiamos ambas categorías para que la interpretación de la variable sea coherente.

- Para el conjunto de variables PAY hacemos una recodificación de la variable PAY_0. Como todas las variables económicas están compuestas por seis variables que van del uno al seis (los seis meses que hay desde abril hasta septiembre), y esta variable pasa del cero al dos, simplemente nos creamos una nueva variable, PAY_1 a la que le asignamos los valores de PAY_0, para que de esta forma todas las variables económicas se encuentren en consonancia.

- Para las seis variables que hay dentro de PAY, BILL_AMT y PAY_AMT observamos los siguientes gráficos en los que se representa la distribución de los datos dentro de cada una de ellas.




*Gráficos 6.1.1*

Vemos que prácticamente las seis variables dentro de PAY, BILL_AMT y PAY_AMT aportan la misma información, así que decidimos conservar sólo la primera de cada una de las variables, es decir, PAY_1, BILL_AMT1 y PAY_AMT1.

- Para la variable PAY_1, como el tamaño mínimo de observaciones que debe recoger cada nivel en una variable discreta debe ser de al menos el 5% del conjunto de datos, y como el conjunto de datos cuenta con 30.000 observaciones, para que un nivel se tenga en cuenta en una variable de este tipo debe de contar al menos con 1500 observaciones. Luego en base a esto, reducimos los niveles de la variable PAY_1 unificando sus categorías de la 3 a la 8 en una sola. Y también como las categorías -2, -1 y 0 todas hacen referencia a no retrasarse en el pago, unimos las tres categorías en una sola. De esta forma, la variable PAY pasa a tener la siguiente distribución de valores en sus categorías.

| PAY_1 | | | | |
|---|---|---|---|---|
| **PAY_1** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **-2** | 2759 | 9.20 | 2759 | 9.20 |
| **-1** | 5686 | 18.95 | 8445 | 28.15 |
| **0** | 14737 | 49.12 | 23182 | 77.27 |
| **1** | 3688 | 12.29 | 26870 | 89.57 |
| **2** | 2667 | 8.89 | 29537 | 98.46 |
| **3** | 322 | 1.07 | 29859 | 99.53 |
| **4** | 76 | 0.25 | 29935 | 99.78 |
| **5** | 26 | 0.09 | 29961 | 99.87 |
| **6** | 11 | 0.04 | 29972 | 99.91 |
| **7** | 9 | 0.03 | 29981 | 99.94 |
| **8** | 19 | 0.06 | 30000 | 100.00 |

| PAY_1 | | | | |
|---|---|---|---|---|
| **PAY_1** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **0** | 23182 | 77.27 | 23182 | 77.27 |
| **1** | 3688 | 12.29 | 26870 | 89.57 |
| **2** | 2667 | 8.89 | 29537 | 98.46 |
| **3** | 463 | 1.54 | 30000 | 100.00 |

*Tabla 6.1.2*

- Por último, eliminaremos la variable ID ya que no la necesitamos para nuestro estudio.

## 6.2 ANÁLISIS INICIAL

Una vez terminada la reconstrucción inicial de variables lo primero que hacemos es establecer el rol de cada variable, donde indicaremos que la variable DEFAULT es la variable objetivo y las demás variables las dejaremos como input. La tipología de las variables es la siguiente.

- Categóricas (nominales): MARRIAGE
- Categóricas (binarias): SEX, DEFAULT
- Discretas (ordinales): EDUCATION, PAY
- Continuas (intervalo o razón): AGE, LIMIT_BAL, BILL_AMT, PAY_AMT

Sobre la variable objetivo observamos lo siguiente.

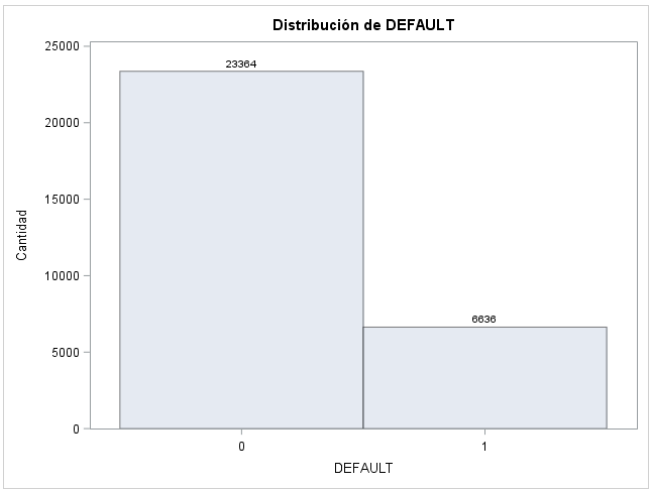



*Gráfico 6.2.1*

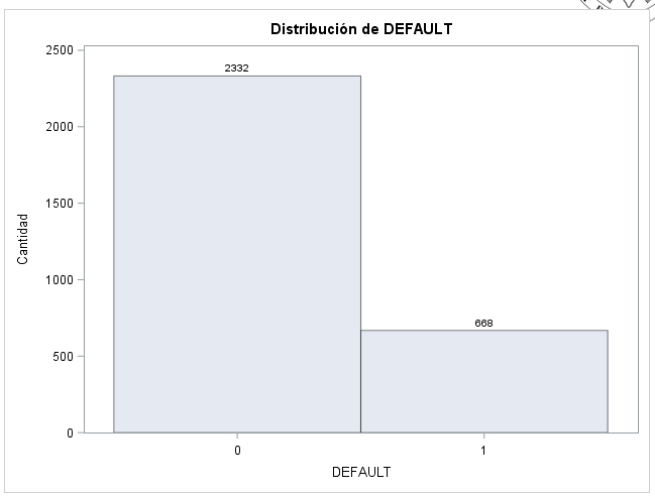



*Gráfico 6.2.2*

Inicialmente con las 30.000 observaciones, a través de este gráfico de frecuencias vemos que hay muchas más personas que pagan a tiempo que las que sufren impago, concretamente, un 77.88 frente a un 22.12%.

Una vez reducimos de 30.000 a 3.000 observaciones por problemas de hardware que más adelante explicaremos, seguimos viendo que la distribución de pagadores e impagadores apenas varía, concretamente, un 77.73 frente a un 22.27% (estos porcentajes indican que la selección de observaciones ha sido correcta ya que parece que hay la misma distribución de datos).

La nueva distribución de valores por categoría en cada variable explicativa discreta es la siguiente:

| SEX | | | | |
|---|---|---|---|---|
| **SEX** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **1** | 11888 | 39.63 | 11888 | 39.63 |
| **2** | 18112 | 60.37 | 30000 | 100.00 |

| EDUCATION | | | | |
|---|---|---|---|---|
| **EDUCATION** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **0** | 468 | 1.56 | 468 | 1.56 |
| **1** | 10585 | 35.28 | 11053 | 36.84 |
| **2** | 4917 | 16.39 | 15970 | 53.23 |
| **3** | 14030 | 46.77 | 30000 | 100.00 |

| MARRIAGE | | | | |
|---|---|---|---|---|
| **MARRIAGE** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **0** | 54 | 0.18 | 54 | 0.18 |
| **1** | 13659 | 45.53 | 13713 | 45.71 |
| **2** | 15964 | 53.21 | 29677 | 98.92 |
| **3** | 323 | 1.08 | 30000 | 100.00 |

| REP_OPT_AGE_NEW | | | | |
|---|---|---|---|---|
| **REP_OPT_AGE_NEW** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **1** | 3871 | 12.90 | 3871 | 12.90 |
| **2** | 11825 | 39.42 | 15696 | 52.32 |
| **3** | 14304 | 47.68 | 30000 | 100.00 |

| REP_OPT_BILL_AMT1_NEW | | | | |
|---|---|---|---|---|
| **REP_OPT_BILL_AMT1_NEW** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **1** | 17611 | 58.70 | 17611 | 58.70 |
| **2** | 12389 | 41.30 | 30000 | 100.00 |

| REP_OPT_PAY_AMT1_NEW | | | | |
|---|---|---|---|---|
| **REP_OPT_PAY_AMT1_NEW** | **Frecuencia** | **Porcentaje** | **Frecuencia acumulada** | **Porcentaje acumulado** |
| **1** | 5332 | 17.77 | 5332 | 17.77 |
| **2** | 14772 | 49.24 | 20104 | 67.01 |
| **3** | 9896 | 32.99 | 30000 | 100.00 |

*Tablas 6.2.1*

Vemos que en general todas las categorías de todas las variables tienen representación gracias a un considerable número de observaciones en cada una de ellas.

## 6.3 DATOS PERDIDOS O DATOS MISSING

A continuación, mostramos una serie de estadísticos descriptivos tanto para las variables continuas como para las variables discretas de nuestro modelo.

| Variable | Rol | Media | Desviación estándar | No ausente | Ausente | Mínimo | Mediana | Máximo | Asimetría | Curtosis |
|---|---|---|---|---|---|---|---|---|---|---|
| AGE | INPUT | 35.4855 | 9.217904 | 30000 | 0 | 21 | 34 | 79 | 0.732246 | 0.044303 |
| BILL_AMT1 | INPUT | 51223.33 | 73635.86 | 30000 | 0 | -165580 | 22381 | 964511 | 2.663861 | 9.806289 |
| LIMIT_BAL | INPUT | 167484.3 | 129747.7 | 30000 | 0 | 10000 | 140000 | 1000000 | 0.992867 | 0.536263 |
| PAY_AMT1 | INPUT | 5663.581 | 16563.28 | 30000 | 0 | 0 | 2100 | 873552 | 14.66836 | 415.2547 |

*Tabla 6.3.1*

| Rol de los datos | Nombre de la variable | Rol | Número de niveles | Ausente | Moda | Porcentaje moda | Moda2 | Porcentaje Moda2 |
|---|---|---|---|---|---|---|---|---|
| TRAIN | EDUCATION | INPUT | 4 | 0 | 3 | 46.77 | 1 | 35.28 |
| TRAIN | MARRIAGE | INPUT | 4 | 0 | 2 | 53.21 | 1 | 45.53 |
| TRAIN | PAY_1 | INPUT | 4 | 0 | 0 | 77.27 | 1 | 12.29 |
| TRAIN | SEX | INPUT | 2 | 0 | 2 | 60.37 | 1 | 39.63 |
| TRAIN | DEFAULT | TARGET | 2 | 0 | 0 | 77.88 | 1 | 22.12 |

*Tabla 6.3.2*

A través de los resultados de ambas tablas observamos que no hay variables con datos ausentes y que las variables BILL_AMT1 y PAY_AMT1 no se encuentran dentro de los límites de asimetría [-2, 2]. A parte de esto, no encontramos ningún error o dato irregular en ninguna variable con lo que no se necesita realizar recodificaciones ni tratamientos.

## 6.4 DATOS ATÍPICOS

Pasamos a establecer los límites superior e inferior de cada variable para poder determinar si existen atípicos. Este paso sólo se hará con las variables continuas ya que sobre las variables categóricas no existen datos atípicos. Para evitar errores a la hora de establecer atípicos seremos bastante estrictos y por eso utilizaremos dos métodos. Dependiendo de si las variables son simétricas o no, utilizaremos el método de la desviación típica (STDDEV) o el de la desviación absoluta media (MAD); y además utilizaremos el método del rango intercuartílico de tal forma que los límites serán aquellos en los que se solapen los límites de ambos métodos.

A la hora de establecer los límites mediante el método del rango intercuartílico, si no se tienen valores para alguno de los límites es que esa variable no tiene valores atípicos en ese límite. Y como se quiere ser lo más estricto posible, en FARLOW nos quedaremos con el límite superior y en FARHIGH nos quedaremos con el límite inferior.

| Variable | Técnica | Simetría | | RIQ | | | | Límite Inferior | Límite Superior |
|---|---|---|---|---|---|---|---|---|---|
| | | | | FARLOW | | FARHIGH | | | |
| | | Límite Inferior | Límite Superior | Límite Inferior | Límite Superior | Límite Inferior | Límite Superior | | |
| **AGE** | STDDEV | 7.831788 | 63.13921 | - | - | - | - | - | - |
| **BILL_AMT1** | MADS | -173823 | 218586 | - | - | 257884 | 964511 | - | 257884 |
| **LIMIT_BAL** | STDDEV | -221759 | 556727.3 | - | - | 1000000 | 1000000 | - | 1000000 |
| **PAY_AMT1** | MADS | -15288 | 19488 | - | - | 17029 | 873552 | - | 19488 |

*Tabla 6.4.1*

Una vez aplicados los límites obtenemos los siguientes resultados en relación a los atípicos.

| Variable | Rol | Media | Desviación estándar | No ausente | Ausente | Mínimo | Mediana | Máximo | Asimetría | Curtosis |
|---|---|---|---|---|---|---|---|---|---|---|
| AGE | INPUT | 35.4855 | 9.217904 | 30000 | 0 | 21 | 34 | 79 | 0.732246 | 0.044303 |
| REP_BILL_AMT1 | INPUT | 43254.92 | 54357.44 | 29215 | 785 | -165580 | 20652 | 257884 | 1.686592 | 2.435306 |
| REP_LIMIT_BAL | INPUT | 167484.3 | 129747.7 | 30000 | 0 | 10000 | 140000 | 1000000 | 0.992867 | 0.536263 |
| REP_PAY_AMT1 | INPUT | 3281.219 | 3619.735 | 28580 | 1420 | 0 | 2000 | 19469 | 1.735423 | 3.121694 |

*Tabla 6.4.2*

Nos encontramos con que las variables BILL_AMT1 y PAY_AMT1 sí que cuentan con valores atípicos, pero estos atípicos no los vamos a quitar porque si apareciese otra observación similar, el modelo no sería capaz de predecirla bien.

Lo siguiente que se hará es transformar las variables para intentar suavizar los valores y que este sea un modelo que permita predecir todo tipo de observaciones.

## 6.5 TRANSFORMACIONES DE VARIABLES

Una vez analizados los atípicos, pasamos a transformar variables. En ocasiones es necesario realizar alguna transformación en las variables para que el modelo de predicción funcione mejor o se pueda plasmar la verdadera relación con la variable objetivo. Como la variable objetivo es binaria, realizamos una transformación de variables a través del método "Mejor" (esta transformación de variables sólo modifica variables de intervalo).

A través del nodo "Transformar variables" de SAS Enterprise Miner se realizan las siguientes transformaciones:

| Input Name | Role | Input Level | Name | Level | Formula |
|---|---|---|---|---|---|
| AGE | INPUT | INTERVAL | OPT_AGE | NOMINAL | Optimal Binning(4) |
| BILL_AMT1 | INPUT | INTERVAL | OPT_BILL_AMT1 | NOMINAL | Optimal Binning(4) |
| LIMIT_BAL | INPUT | INTERVAL | LG10_LIMIT_BAL | INTERVAL | log10(LIMIT_BAL + 1) |
| PAY_AMT1 | INPUT | INTERVAL | OPT_PAY_AMT1 | NOMINAL | Optimal Binning(4) |

*Tabla 6.5.1*

Este software selecciona este tipo de transformaciones de entre otras muchas ya que con estas transformaciones es con las que obtiene un mayor $R^2$.

La transformación "Optimal Binning" lo que hace es tramificar variables continuas, y ha tramificado estas variables de la siguiente forma:

- **AGE**: La ha dividido en 3 tramos, del mínimo al 25.5 es 1, del 25.5 al 34.5 es 2, y del 34.5 al máximo es 3.
- **BILL_AMT**: La ha dividido en 2 tramos, del mínimo al 34381 es 1, y del 34381 al máximo es 2.
- **PAY_AMT**: La ha dividido en 3 tramos, del mínimo al 6.5 es 1, del 6.5 al 3989 es 2, y del 3989 al máximo es 3.

La transformación "log10" lo que hace es el logaritmo en base diez de (LIMIT_BAL+1). Si observamos los resultados de la tabla del $R^2$ según la mejor transformación observamos lo siguiente:

| Variable original | Variable calculada | Fórmula | RSquare |
|---|---|---|---|
| LIMIT_BAL | LG10_LIMIT_BAL | log10(LIMIT_BAL + 1) | 0.030126897980052702 |
| LIMIT_BAL | LOG_LIMIT_BAL | log(LIMIT_BAL + 1) | 0.03012689797984556 |
| LIMIT_BAL | OPT_LIMIT_BAL | Optimal Binning(4) | 0.029871913986550022 |
| LIMIT_BAL | SQRT_LIMIT_BAL | Sqrt(LIMIT_BAL + 1) | 0.02791538173357896 |
| LIMIT_BAL | CNTR_LIMIT_BAL | (LIMIT_BAL - 167484.32267) | 0.0235683524478843 |
| LIMIT_BAL | RANGE_LIMIT_BAL | (LIMIT_BAL - 10000) / (1000000-10000) | 0.02356835244788261 |
| LIMIT_BAL | STD_LIMIT_BAL | (LIMIT_BAL - 167484.32267) / 129747.66157 | 0.023568352447879522 |
| LIMIT_BAL | LIMIT_BAL | | 0.023568352447877805 |
| LIMIT_BAL | INV_LIMIT_BAL | 1 / (LIMIT_BAL + 1) | 0.022760590136619012 |
| LIMIT_BAL | SQR_LIMIT_BAL | (LIMIT_BAL + 1)**2 | 0.015005900013337231 |
| LIMIT_BAL | EXP_LIMIT_BAL | exp(LIMIT_BAL / 10000) | 9.467872611157256E-6 |

*Tabla 6.5.2*

Al ser esta transformación la que obtiene un mayor $R^2$ por eso selecciona esta, aunque también vemos que la transformación en función del logaritmo neperiano es prácticamente la misma que la transformación en función del logaritmo en base 10.

Pero de todas estas transformaciones, únicamente nos quedaremos con la que se hace sobre la variable LIMIT_BAL, ya que como posteriormente emplearemos el nodo "Agrupación Interactiva" [8] el cual también tramifica las variables continuas, no necesitaremos haberlas tramificado antes a través de la transformación "Optimal Binning".

## 6.6 TRAMIFICACIÓN Y AGRUPACIÓN DE CATEGORÍAS

Por último, procedemos con la agrupación de categorías a través del nodo "Agrupación Interactiva".

El nodo "Agrupación Interactiva" primero realiza la agrupación en la característica de intervalo. Puede elegir entre dos métodos de agrupación: cuantil y agrupación. El método cuantil genera grupos, y los grupos están formados por cantidades clasificadas con aproximadamente la misma frecuencia en cada grupo; mientras que el método agrupación genera grupos dividiendo los datos en intervalos espaciados uniformemente que se basan en la diferencia entre los valores máximo y mínimo. En nuestro caso hemos utilizado el método cuantil.

Después de que las variables de intervalo se hayan agrupado previamente, se ajusta un modelo de árbol de decisión para cada característica. Los procedimientos que se utilizan para producir los grupos son PROC ARBOR o PROC OPTBIN. Se puede elegir entre cuatro métodos de agrupación: criterio óptimo, cuantil, tasa de eventos monotónicos y óptima restringida. El método de criterio óptimo utiliza uno de dos criterios: reducción en la medida de entropía o el valor p del estadístico Chi-cuadrado de Pearson; el método cuantil genera grupos con aproximadamente la misma frecuencia en cada grupo; el método de tasa de eventos monotónicos genera grupos que resultan en una distribución monotónica de tasas de eventos en todos los atributos, la tasa de eventos es igual a P (evento | atributo), que esta es la probabilidad condicional de un evento dado que un solicitante exhibe un atributo particular; y el método óptimo restringido encuentra un conjunto óptimo de grupos y simultáneamente impone restricciones adicionales, como se especifica en la configuración del panel de propiedades del nodo. En nuestro caso hemos utilizado el método de agrupación criterio óptimo.

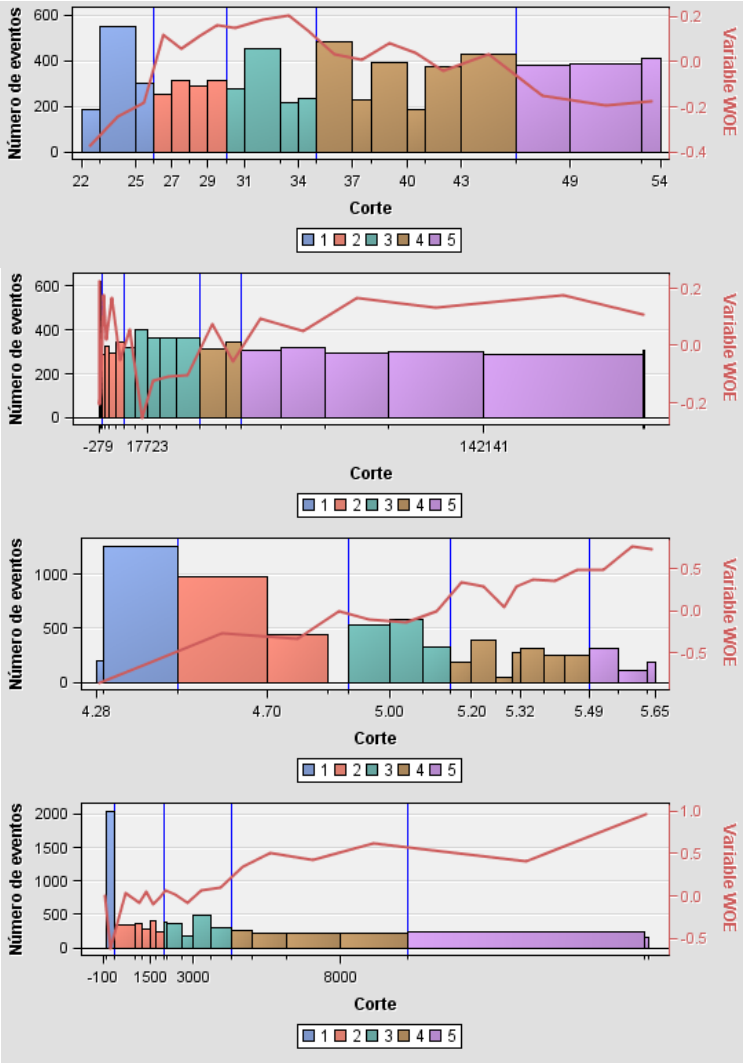
Las tramificaciones de variables, agrupamientos y asignación de valores WOE que se realiza en cada una de las variables son las siguientes.

| AGE | | |
|---|---|---|
| **Valor** | **Grupo** | **WOE** |
| Min ≤ AGE < 26 | 1 | -0.24673 |
| 26 ≤ AGE < 30 | 2 | 0.110727 |
| 30 ≤ AGE < 35 | 3 | 0.170947 |
| 35 ≤ AGE < 46 | 4 | 0.026868 |
| AGE ≥ 46 | 5 | -0.17232 |

| BILL_AMT1 | | |
|---|---|---|
| **Valor** | **Grupo** | **WOE** |
| Min ≤ BILL_AMT1 < 800.5 | 1 | -0.14102 |
| 800.5 ≤ BILL_AMT1 < 9157.5 | 2 | 0.074294 |
| 9157.5 ≤ BILL_AMT1 < 37046 | 3 | -0.10966 |
| 37046 ≤ BILL_AMT1 < 52205.5 | 4 | 0.008923 |
| BILL_AMT1 ≥ 52205.5 | 5 | 0.119986 |

| LG10_LIMIT_BAL | | |
|---|---|---|
| **Valor** | **Grupo** | **WOE** |
| Min ≤ LG10_LIMIT_BAL < 4.48 | 1 | -0.67677 |
| 4.48 ≤ LG10_LIMIT_BAL < 4.9 | 2 | -0.29267 |
| 4.9 ≤ LG10_LIMIT_BAL < 5.15 | 3 | -0.09246 |
| 5.15 ≤ LG10_LIMIT_BAL < 5.49 | 4 | 0.33815 |
| LG10_LIMIT_BAL ≥ 5.49 | 5 | 0.619579 |

| PAY_AMT1 | | |
|---|---|---|
| **Valor** | **Grupo** | **WOE** |
| Min ≤ PAY_AMT1 < 316 | 1 | -0.61215 |
| 316 ≤ PAY _AMT1 < 2000 | 2 | -0.03141 |
| 2000 ≤ PAY _AMT1 < 4309.5 | 3 | 0.045942 |
| 4309.5 ≤ PAY _AMT1 < 10300 | 4 | 0.474215 |
| PAY _AMT1 ≥ 10300 | 5 | 0.657854 |

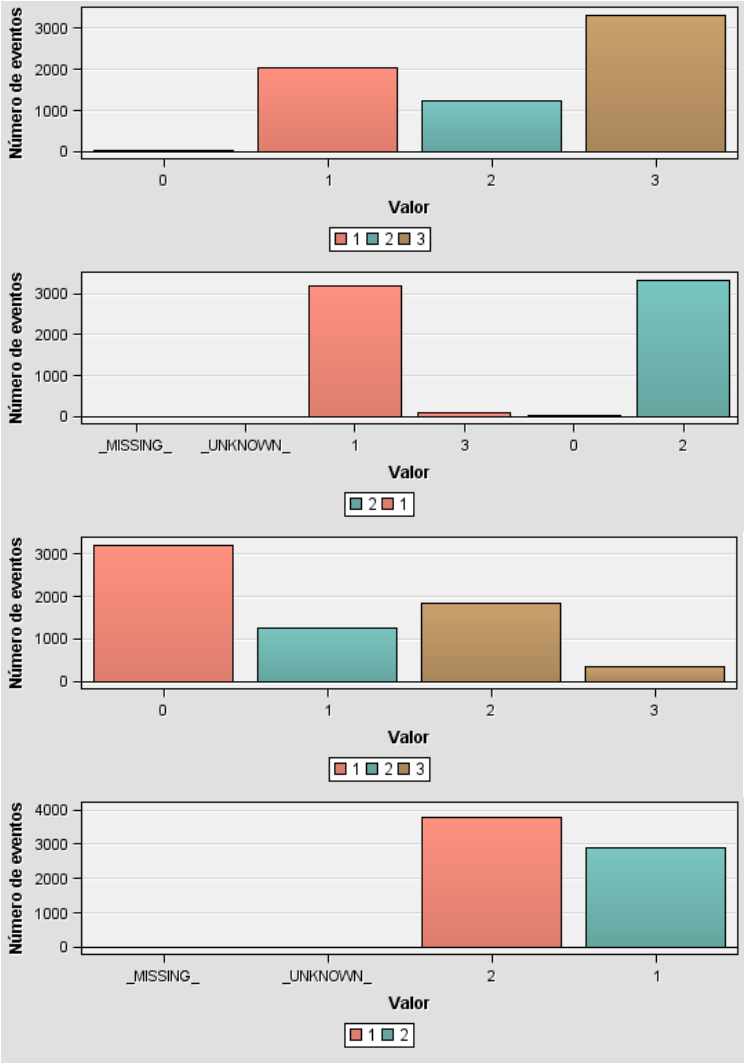


| EDUCATION | | |
|---|---|---|
| Valor | Grupo | WOE |
| Categorías 0 y 1 | 1 | 0.209693 |
| Categoría 2 | 2 | -0.16846 |
| Categoría 3 | 3 | -0.09142 |

| MARRIAGE | | |
|---|---|---|
| Valor | Grupo | WOE |
| Categorías 1 y 3 | 1 | -0.08008 |
| Categorías 0 y 2 | 2 | 0.072942 |

| PAY_1 | | |
|---|---|---|
| Valor | Grupo | WOE |
| Categoría 0 | 1 | 0.570459 |
| Categoría 1 | 2 | -0.59307 |
| Categorías 2 y 3 | 3 | -2.08478 |

| SEX | | |
|---|---|---|
| Valor | Grupo | WOE |
| Categoría 2 | 1 | 0.079777 |
| Categoría 1 | 2 | -0.11515 |




*Tablas y gráficos 6.6.1*

Una vez que se han agrupado las características, el nodo “Agrupación Interactiva” calcula el WOE para cada atributo de cada característica.

A continuación se explica en qué consisten las variables WOE y cuál es su utilidad.

# 7. SISTEMA WOE

El sistema WOE (Weight Of Evidence), consiste en utilizar el poder predictivo que tiene cada variable independiente en relación con la variable objetivo para seleccionar las variables más relevantes [9].

El peso de la evidencia (WOE) mide el riesgo relativo de un atributo o nivel de grupo. El valor depende del valor de la variable objetivo binaria, que es "no evento" (objetivo = 0) o "evento" (objetivo = 1). El WOE de un atributo se define de la siguiente manera:

$$WOE_{atributo} = ln\left(\frac{p^{no\ evento}_{atributo}}{p^{evento}_{atributo}}\right) = ln\left(\frac{N^{atributo}_{no\ evento}/N^{total}_{no\ evento}}{N^{atributo}_{evento}/N^{total}_{evento}}\right)$$

Las definiciones de las cantidades en la fórmula anterior son las siguientes:

- $N_{no\ evento}^{atributo}$: Número de registros sin eventos que exhiben el atributo.
- $N_{no\ evento}^{total}$: Número total de registros sin eventos.
- $N_{evento}^{atributo}$: Número de registros de eventos que exhiben el atributo.
- $N_{evento}^{total}$: Número total de registros de eventos.

Es decir, lo que hace es tramificar y agrupar aquellas categorías en función de si la aparición del suceso en estudio o no aparición de éste es acentuado dentro de esas categorías. De tal forma que logra obtener una serie de clases dentro de cada variable en las que, si un individuo se encuentra dentro de alguna de ellas, se le puede clasificar sin confusión sobre si logra o no el evento en estudio.

Pero la transformación WOE, no es lo mismo que la tramificación y agrupación de categorías. La transformación WOE lo que logra es que a esas categorías que se habían creado, y que para un análisis tradicional luego se tendrían que crear tantas variables dummys como categorías haya menos una, en realidad a esas categorías se les asocia un valor continuo. Es decir, las variables pasan a tener el nuevo número de categorías que se había establecido a través de la tramificación y agrupación, pero en vez de tener valores discretos, son valores continuos, lo que evita la necesidad de crear variables dummys a partir de ella. Luego la variable pasa a tener una serie de nuevos valores, pero que no son discretos, son continuos, y esto permite reducir la dimensión de la matriz de datos al tener únicamente estas variables en el modelo sin necesidad de variables dummy, lo que por consiguiente supone un considerable ahorro de tiempo de ejecución para cada modelo.

Además de las variables WOE en la puntuación de crédito, el "Information Value" (IV) es frecuentemente utilizado para comparar capacidades predictivas entre las variables. Cuando se desarrollan nuevas tarjetas de puntuación utilizando regresión logística, las variables son a menudo unidas y recodificadas utilizando el concepto de WOE.

Asumiendo que el lector tiene alguna experiencia básica en la puntuación de crédito, uno de los objetivos cuando se unen variables es maximizar el Information Value. El WOE para una única unión se define como:

$$WOE = ln\left(\frac{\%good_i}{\%bad_i}\right)$$

Y el IV por cada variable se define como:

$$IV = \sum_{i=1}^{n}\left(ln\left(\frac{\%good_i}{\%bad_i}\right) * (\%good_i - \%bad_i)\right)$$

Donde n es el número de variables.

Para ilustrar el concepto, aquí hay un ejemplo para la variable "Ingresos" de un dataset ficticio.

| Ingresos | No Eventos (Buenos) | Eventos (Malos) | % de No Eventos (% de Buenos) | % de Eventos (% de Malos) | WOE | IV |
|---|---|---|---|---|---|---|
| **0-50** | 197 | 20 | 0,05379574 | 0,0591716 | -0,09524737 | 0,00051204 |
| **51-100** | 450 | 34 | 0,12288367 | 0,10059172 | 0,20016823 | 0,00446214 |
| **101-150** | 492 | 39 | 0,134352813 | 0,11538462 | 0,15219824 | 0,00288693 |
| **151-200** | 597 | 51 | 0,163025669 | 0,15088757 | 0,07737265 | 0,00093916 |
| **201-250** | 609 | 54 | 0,166302567 | 0,15976331 | 0,04011539 | 0,00026232 |
| **251-300** | 582 | 55 | 0,158929547 | 0,16272189 | -0,02358157 | 8,9429E-05 |
| **301-350** | 386 | 41 | 0,105406881 | 0,12130178 | -0,14045353 | 0,00223249 |
| **351-400** | 165 | 23 | 0,045057346 | 0,06804734 | -0,41226757 | 0,00947803 |
| **>401** | 184 | 21 | 0,050245767 | 0,06213018 | -0,21230551 | 0,00252313 |
| **Total** | 3662 | 338 | | | | 0,02338566 |

*Tabla 7.1*

Para el atributo Ingresos 0-50 se lograrían los siguientes WOE e IV:

$$WOE = ln\left(\frac{197/3662}{20/338}\right) = ln\left(\frac{0{,}05379574}{0{,}0591716}\right) = -0{,}09524737$$

$$IV = ln\left(\frac{0{,}05379574}{0{,}0591716}\right) * (0{,}05379574 - 0{,}0591716) = 0{,}00051204$$

# 8. MÉTODOS DE SELECCIÓN DE VARIABLES

Una vez terminado todo el proceso de análisis descriptivo, depuración y transformación de los datos, pasamos a la selección de variables para la construcción de modelos.

Las redes neuronales tienen varias peculiaridades, y una de ellas es que en su construcción se utilizan todas las variables que se le indiquen en el modelo, es decir, no existen métodos de selección de variables para reducir la dimensionalidad de las variables input. Esto supone que a la hora de construir la mejor red, se utilizarán distintas técnicas de selección de variables para solventar este problema.

Existen distintos tipos de selección de variables dependiendo del propósito de la misma.

- Selección sesgada por el planteamiento lineal.
  - Métodos Stepwise, Backward y Forward en regresión.
  - Variable "importance" en regresión PLS.
  - Métodos que tengan en cuenta el $R^2$ directamente o penalizado (EM).

- Selección no sesgada por el planteamiento lineal.
  - Variable "importance" en árboles y en Gradient Boosting.

- Agrupaciones de categorías.
  - Mediante árboles y otros métodos de agrupación (EM).

Existen gran cantidad de métodos de selección de variables, cada uno con sus respectivos criterios, pero nosotros utilizaremos solo tres, con lo que para llevar a cabo la selección de variables lo haremos en función de aquel grupo de variables que haya sido resultado de la selección por cada uno de los métodos de selección de variables.

Los métodos de selección son los siguientes:

## 8.1 LÓGICA DE NEGOCIO

La primera selección de variables que hicimos fue al eliminar la variable ID junto con las variables que resultaban redundantes dentro de las variables PAY, BILL_AMT y PAY_AMT. Esta selección de variables no fue mediante ninguna técnica estadística, sino simplemente utilizando el sentido común, la distribución de los datos y aplicando una lógica de negocio ya que únicamente se quiere mantener en el estudio aquellas variables que son relevantes para el mismo.

## 8.2 ÍNDICE DE GINI E INFORMATION VALUE

El índice de Gini y el valor de información (Information Value) son dos criterios de poder predictivo que se encuentran dentro del nodo "Agrupación Interactiva". El coeficiente de Gini es una medida de desigualdad que se encuentra entre 0 y 1, siendo cero la máxima igualdad y 1 la máxima desigualdad, y el índice de Gini es el coeficiente de Gini multiplicado por 100, es decir, expresado en referencia a 100 como máximo.

Por defecto los valores de corte son de 20 y 0.1 respectivamente, y lo que permiten es seleccionar únicamente las variables más importantes para el modelo.

A continuación se muestra una tabla con los valores de ambos indicadores para cada una de las variables.

| Variable | Estadístico de Gini | Valor de información | Nivel para Interactivo | Nuevo rol | Rol calculado | Agrupamiento predefinido | Nivel | Etiqueta | Ordenación Gini |
|---|---|---|---|---|---|---|---|---|---|
| **PAY_1** | 39.818 | 0.861 | ORDINAL | Predeterminado | Input | | ORDINAL | | 1.0 |
| **LG10_LIMIT_BAL** | 23.075 | 0.175 | INTERVAL | Predeterminado | Input | | INTERVAL | Transformed: LIMIT_BAL | 2.0 |
| **PAY_AMT1** | 21.538 | 0.162 | INTERVAL | Predeterminado | Input | | INTERVAL | PAY_AMT1 | 3.0 |
| **AGE** | 7.974 | 0.021 | INTERVAL | Predeterminado | Rechazada | | INTERVAL | AGE | 4.0 |
| **EDUCATION** | 7.907 | 0.024 | ORDINAL | Predeterminado | Rechazada | | ORDINAL | EDUCATION | 5.0 |
| **BILL_AMT1** | 5.857 | 0.011 | INTERVAL | Predeterminado | Rechazada | | INTERVAL | BILL_AMT1 | 6.0 |
| **SEX** | 4.709 | 0.009 | BINARY | Predeterminado | Rechazada | | BINARY | SEX | 7.0 |
| **MARRIAGE** | 3.815 | 0.006 | NOMINAL | Predeterminado | Rechazada | | NOMINAL | MARRIAGE | 8.0 |

*Tabla 8.2.1*

Tanto mediante el índice de Gini como mediante el IV se seleccionan las variables PAY_1, LG10_LIMIT_BAL y PAY_AMT1.

## 8.3 HIGH PERFORMANCE FOREST

Otro de los métodos de selección de variables es mediante la construcción de un bosque de alto rendimiento a través del nodo "HP Forest" que se encuentra en SAS Enterprise Miner [10].

El nodo "HP Forest" crea un modelo predictivo llamado bosque. Un bosque consiste en varios árboles de decisión que difieren entre sí de dos maneras. Primero, los datos de entrenamiento para un árbol son una muestra sin reemplazo de todas las observaciones disponibles, y segundo, las variables de entrada que se consideran para dividir un nodo se seleccionan aleatoriamente de entre todas las variables de entrada disponibles. Por lo demás, los árboles en un bosque se entrenan como árboles de decisión estándar.

El nodo "HP Forest" acepta variables objetivo de intervalo y nominales. Para una variable objetivo de intervalo, el procedimiento promedia las predicciones de los árboles individuales para predecir una observación; y para una variable objetivo de tipo categórica, las probabilidades posteriores en el bosque son los promedios de las probabilidades posteriores de los árboles individuales. Además, el nodo realiza una segunda predicción por votación: el bosque predice la categoría objetivo que los árboles individuales predicen con mayor frecuencia.

Para la selección de variables, las variables se rechazan cuando su importancia de variable no es positiva. Para las variables objetivo de intervalo, la medida de importancia es el error absoluto promedio fuera de bolsa (OOB); y para objetivos categóricos, la medida de importancia es la reducción del margen fuera de bolsa.

A través del nodo HP Forest obtenemos la siguiente tabla de importancia de las variables.

Importancia de la variable de reducción de pérdida

| Variable | Número de reglas | Gini | Gini OOB | Margen | Margen OOB |
|---|---|---|---|---|---|
| GRP_PAY_1 | 2482 | 0.054902 | 0.05270 | 0.109805 | 0.107960 |
| GRP_PAY_AMT1 | 11146 | 0.008514 | 0.00327 | 0.017027 | 0.011895 |
| GRP_LG10_LIMIT_BAL | 13512 | 0.008247 | 0.00184 | 0.016494 | 0.010264 |
| GRP_BILL_AMT1 | 15340 | 0.006454 | -0.00067 | 0.012908 | 0.006115 |
| GRP_MARRIAGE | 7396 | 0.001951 | -0.00121 | 0.003903 | 0.000713 |
| GRP_SEX | 10379 | 0.002437 | -0.00186 | 0.004874 | 0.000591 |
| GRP_EDUCATION | 11899 | 0.003294 | -0.00257 | 0.006589 | 0.000818 |
| GRP_AGE | 19665 | 0.005220 | -0.00478 | 0.010440 | 0.001029 |

*Tabla 8.3.1*

Mediante esta tabla vemos que el valor del margen fuera de la bolsa es mayor que cero en todas las variables, luego mediante este método de selección se seleccionan todas las variables.

## 8.4 REGRESIÓN LOGÍSTICA

Uno de los métodos de selección de variables para la construcción de las redes neuronales será a través de la propia regresión logística. La regresión logística que llevaremos a cabo sigue un procedimiento stepwise, a través del cual cada variable de la base de datos se va introduciendo una a una en el modelo y se extrae si no es significativa para el mismo. De tal forma, que al final únicamente quedan en el modelo las variables que son significativas para este tipo de regresión.

Mediante el procedimiento ***logistic*** se seleccionan las siguientes variables:

GRP_BILL_AMT1, GRP_PAY_AMT1, GRP_SEX, GRP_MARRIAGE, GRP_PAY_1.

Y mediante la macro ***%randomselectlog*** la cual realiza un método stepwise repetidas veces con diferentes archivos train se seleccionan los siguientes grupos de variables ya que han sido los más elegidos (los modelos que salen elegidos más veces son los posibles candidatos para probar con validación cruzada):

GRP_BILL_AMT1, GRP_PAY_AMT1, GRP_SEX, GRP_MARRIAGE, GRP_PAY_1.

GRP_PAY_AMT1, GRP_SEX, GRP_MARRIAGE, GRP_PAY_1.

GRP_LG10_LIMIT_BAL, GRP_BILL_AMT1, GRP_PAY_AMT1, GRP_SEX, GRP_MARRIAGE, GRP_PAY_1.

Comparando todos los grupos de variables llegamos a la conclusión de que a través de la regresión logística se seleccionan las siguientes variables:

GRP_LG10_LIMIT_BAL, GRP_BILL_AMT1, GRP_PAY_AMT1, GRP_SEX, GRP_MARRIAGE y GRP_PAY_1.

## 8.5 TABLA RESUMEN

Como conclusión elaboramos una tabla en la que se encuentran las variables que se han seleccionado mediante cada uno de los métodos de selección.

| Variables \ Técnicas | Índice de Gini Information Value | HP Forest | Regresión Logística |
|---|---|---|---|
| **AGE** | X | ✓ | X |
| **BILL_AMT1** | X | ✓ | ✓ |
| **LIMIT_BAL** | ✓ | ✓ | ✓ |
| **PAY_AMT1** | ✓ | ✓ | ✓ |
| **EDUCATION** | X | ✓ | X |
| **MARRIAGE** | X | ✓ | ✓ |
| **PAY_1** | ✓ | ✓ | ✓ |
| **SEX** | X | ✓ | ✓ |

*Tabla 8.5.1*

Como al final por un método u otro se seleccionan todas las variables, para la construcción de los diferentes modelos utilizaremos todas las variables.

# 9. MODELOS DE PREDICCIÓN

La comparación de los modelos dentro de cada técnica se realizará mediante una representación gráfica en forma de caja y bigotes del ECM de cada modelo, y cada uno de los siguientes modelos de predicción de cada técnica los diseñaremos doblemente mediante dos semillas distintas, para de esta forma reducir aquellos resultados que podrían darse por la casuística de la selección de datos.

Utilizaremos el ECM como criterio de selección de los mejores modelos ya que el ECM mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima, lo cual nos indica cómo de acertado es cada modelo a la hora de predecir.

Conviene saber que si las predicciones son en forma de probabilidad, es necesario determinar cuál es el criterio o punto de corte (threshold) para asignar cada observación a cada clase. Por defecto, el punto de corte es p=0.5 (a partir de una probabilidad predicha de pertenecer a la clase A de 0.5, se asigna la observación a la clase A), pero nosotros cambiamos ese punto de corte.

Inicialmente con las 30.000 observaciones establecemos el punto de corte en 22.12 ya que es la frecuencia de una observación default en la muestra, pero como al desarrollar los distintos modelos nos encontramos con que los tiempos de espera para ejecutar cada modelo de redes neuronales eran excesivamente largos (más de 20 horas y no habían terminado de ejecutarse), tuvimos que ir reduciendo la cantidad de observaciones con las que trabajar hasta llegar a las 3.000. Con esta cantidad de observaciones el punto de corte se establece en 22.27, muy parecido al 22.12 inicial, lo que nos indica que la distribución de datos a través de la selección aleatoria ha sido correcta.

## 9.1 VARIABLES BINARIAS

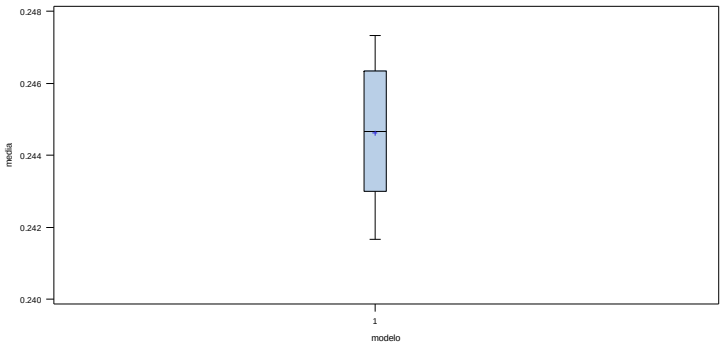
Primero realizaremos cada uno de los modelos con las variables iniciales transformadas en binarias tras la agrupación y tramificación inicial, y más tarde construiremos cada uno de los modelos con las variables WOE obtenidas.

### 9.1.1 REGRESIÓN LOGÍSTICA

Para la regresión logística utilizamos la macro ***%cruzadalogistica*** la cual hace validación cruzada logística para variables dependientes binarias. Los resultados con cada una de las semillas son los siguientes.

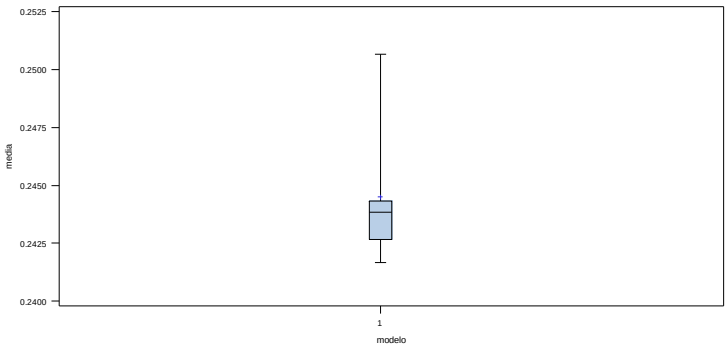
**Con la semilla original:** 12345 ~ 12350

**Media:** 0.2446111




**Tras cambiar la semilla:** 12340 ~ 12345

**Media:** 0.2445000




### 9.1.2 REDES NEURONALES

La determinación del mejor modelo de redes neuronales se llevará a cabo mediante un proceso de "ensayo y error" donde iremos añadiendo distintos aspectos de la red neuronal como número de nodos, función de activación, algoritmo, etc. E iremos variando cada uno de esos aspectos para comprobar cómo evolucionan los distintos modelos.

- Número de nodos

Para tener una idea inicial del número de nodos a utilizar en la red, daremos uso de la siguiente fórmula.

$$h(k+1)+h+1=\frac{nº\ observaciones}{nº\ de\ observaciones\ por\ parámetro}$$

*h*: nº nodos ocultos *k*: nº nodos input (variables)

En teoría debe haber entre 5 y 25 observaciones por parámetro, pero como la base de datos cuenta con un número considerable de observaciones (inicialmente 30.000, pero decidimos reducir a 3.000 por los tiempos de ejecución), se entiende que utilizar sólo 5 observaciones por parámetro es un número muy reducido que se suele dar uso cuando la base de datos que se usa tiene pocas observaciones. Como este no es el caso, decidimos utilizar entre 15 y 25 observaciones por parámetro y de esa forma obtenemos la siguiente tabla con las posibles observaciones por parámetro y los consiguientes números de nodos.

| Observaciones por parámetro | Ecuación | h |
|---|---|---|
| 15 | $h(8+1)+h+1=\frac{3000}{15}\rightarrow 10h+1=\frac{3000}{15}$ | $h=19$ |
| 20 | $h(8+1)+h+1=\frac{3000}{20}\rightarrow 10h+1=\frac{3000}{20}$ | $h=14$ |
| 25 | $h(8+1)+h+1=\frac{3000}{25}\rightarrow 10h+1=\frac{3000}{25}$ | $h=11$ |

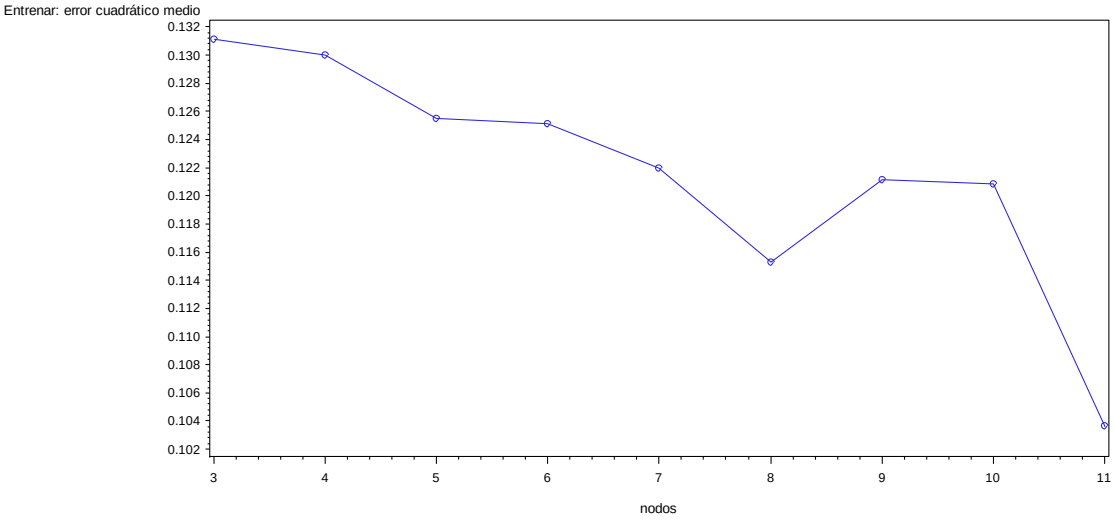
*Tabla 9.1.2.1*

Utilizamos la macro ***%repito*** en la que comprobamos con cada número de nodos obtenido en la tabla anterior y a través del error cuadrático medio, qué número de nodos muestran mejores resultados.

Los resultados son los siguientes.

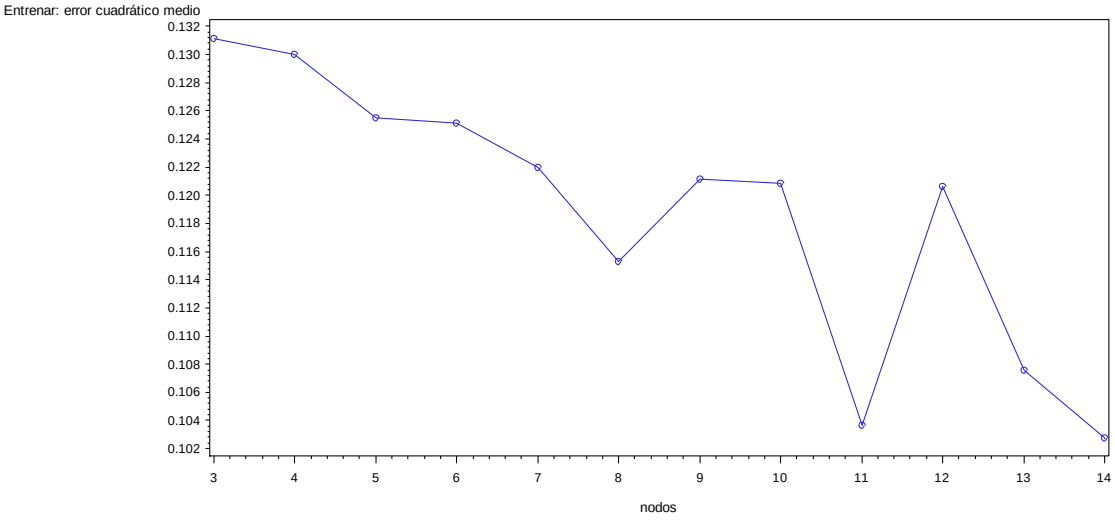
**Entre 3 y 11 nodos.**

El mejor es el 11.



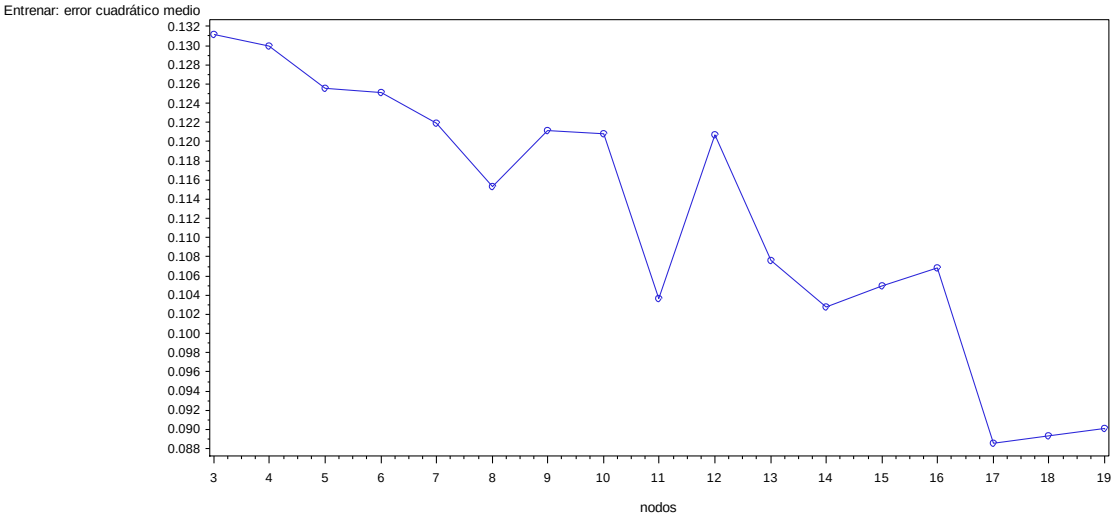

**Entre 3 y 14 nodos.**

Los mejores son los nodos 11 y 14.



**Entre 3 y 19 nodos.**

El mejor es el nodo 17.




Los resultados indican que a mayor número de nodos, menor error cuadrático medio. Esto tiene sentido ya que a un mayor número de nodos mejor se podrá ajustar la red a los datos, pero hay que tener cuidado de no utilizar un número excesivo de nodos para evitar el sobreajuste.

Una vez que hemos visto que los posibles mejores números de nodos para construir las redes neuronales son con 11, 14 o 17, para comprobar cuál es el mejor de todos ellos utilizaremos las macros ***%nodosvalcruza***, ***%numeronodos*** y ***%variar***. La macro ***%nodosvalcurza*** utiliza unos parámetros fundamentales de la red neuronal mediante validación cruzada repetida, siendo este un sistema de gran potencia que permite evaluar la variabilidad del modelo.

Probamos seleccionando los nodos de uno en uno desde el 10 al 20 para que haya margen y obtenemos el siguiente gráfico.

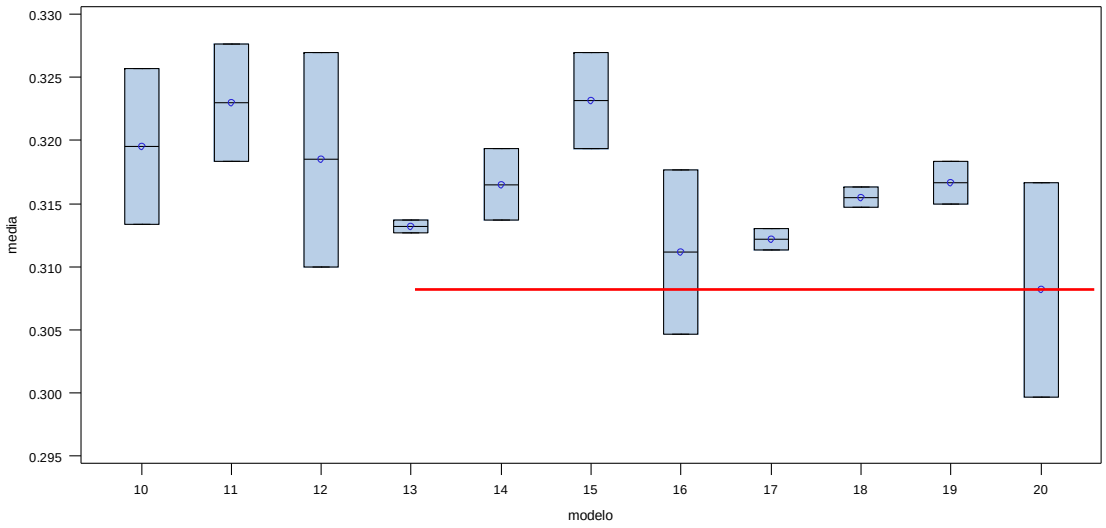


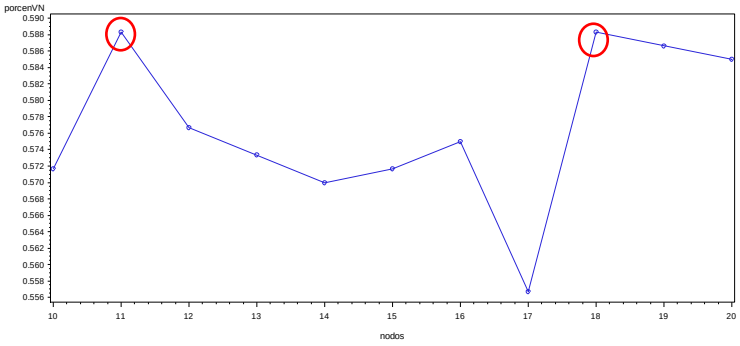

*Gráfico 9.1.2.2*

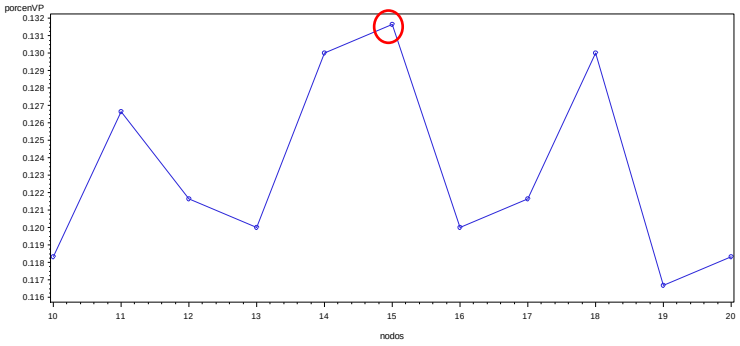
El mejor modelo se consigue con el vigésimo número de nodos, con 20 nodos.

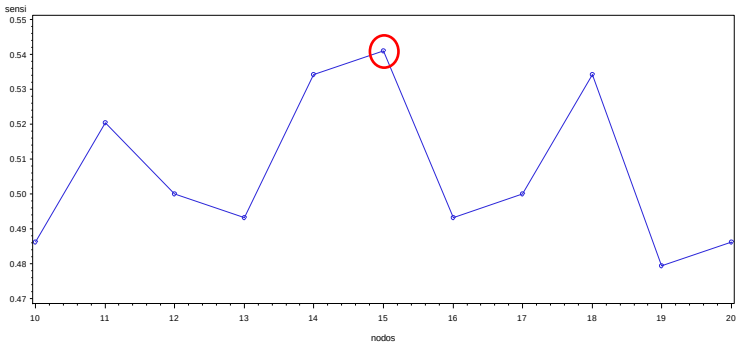
Ahora probamos con la macro ***%numeronodos***. Esta macro nos ofrece los siguientes valores con los que poder evaluar qué número de nodos obtiene los mejores resultados, porcentaje de verdaderos negativos, porcentaje de falsos negativos, porcentaje de verdaderos positivos, porcentaje de falsos positivos, sensibilidad, especificidad, tasa de fallos, tasa de aciertos, precisión y F_M $\left(F_M = \frac{2*sensibilidad*precisión}{sensibilidad+precisión}\right)$.

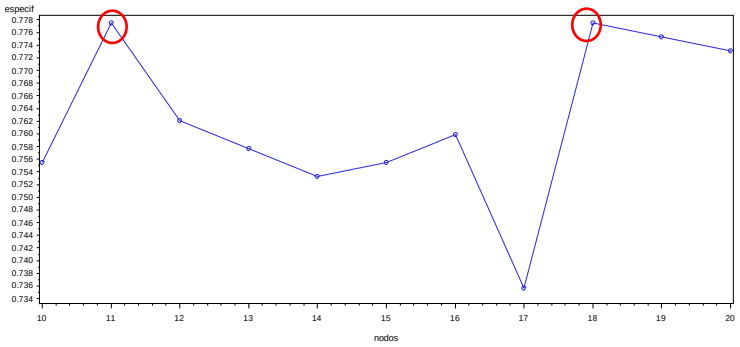
Como comparar estas diez tablas a la vez puede ser bastante complicado, nos vamos a centrar en los verdaderos negativos y positivos, sensibilidad, especificidad, tasa de aciertos y tasa de fallos.

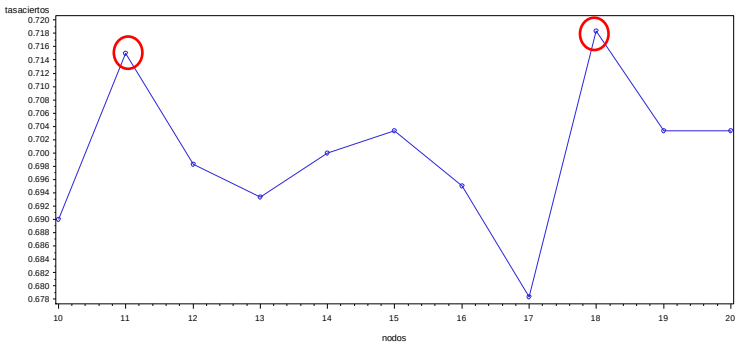
Los resultados son los siguientes.

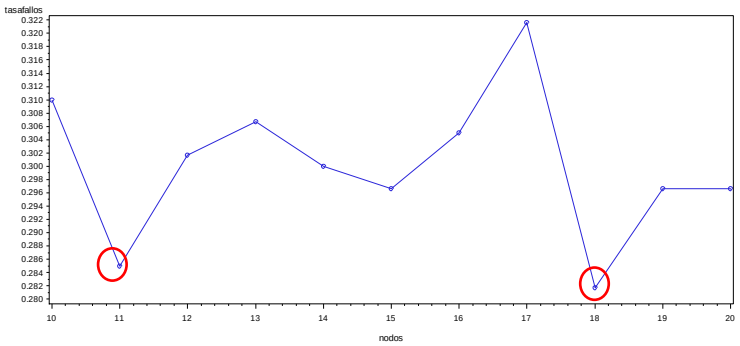


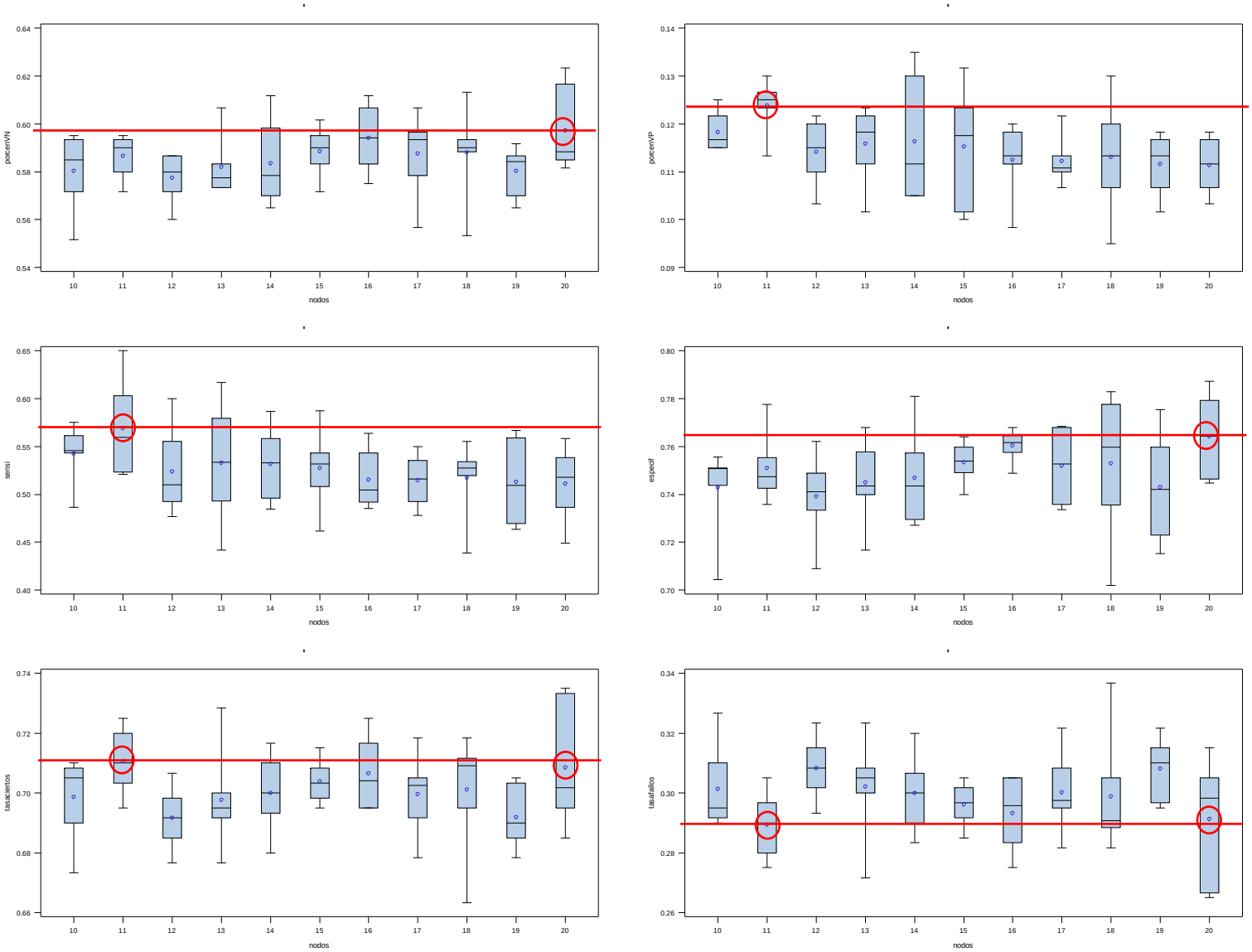




Vemos que los nodos que ofrecen mejores resultados son 11, 15 y 18.

Para terminar con la búsqueda de aquellos nodos que ofrecen mejores resultados, utilizamos por último la macro ***%variar***. Esta macro nos ofrece los mismos valores que la macro anterior, así que decidimos centrarnos únicamente en los verdaderos negativos y positivos, sensibilidad, especificidad, tasa de aciertos y tasa de fallos.

Los resultados son los siguientes.




Vemos que los nodos que ofrecen mejores resultados son 11 y 20.

Así que como conclusión obtenemos que, por dos de los tres métodos de selección de número de nodos, los números de nodos que ofrece mejores resultados son 11 y 20 nodos.

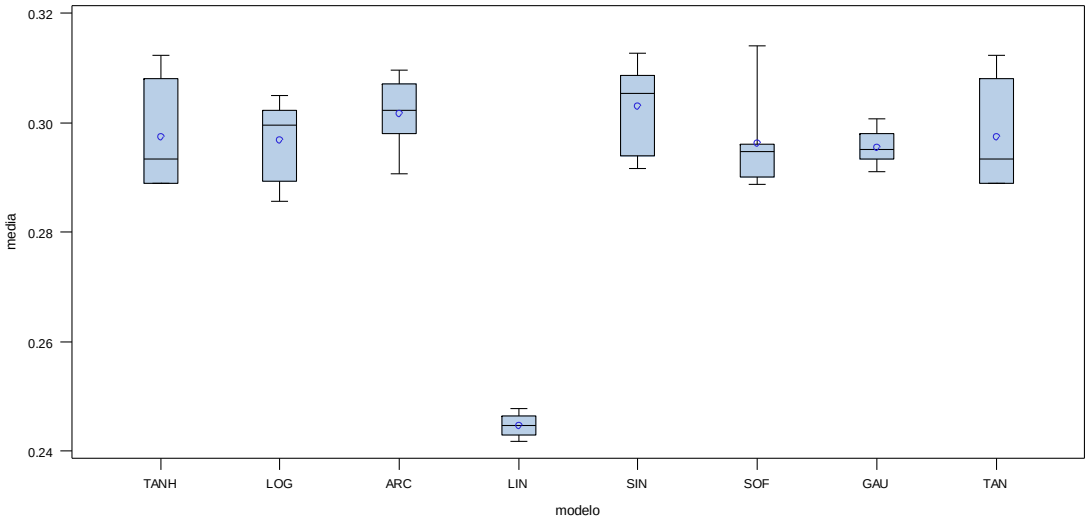
- Función de activación

El siguiente paso es comprobar qué función de activación ofrece mejores resultados con los mejores números de nodos establecidos anteriormente. Las funciones de activación que comprobaremos son TANH, LOG, ARC, LIN, SIN, SOF, GAU y TAN. Para ello utilizaremos la macro ***%activalcruza*** que compara cada función a través de una red neuronal mediante validación cruzada repetida.

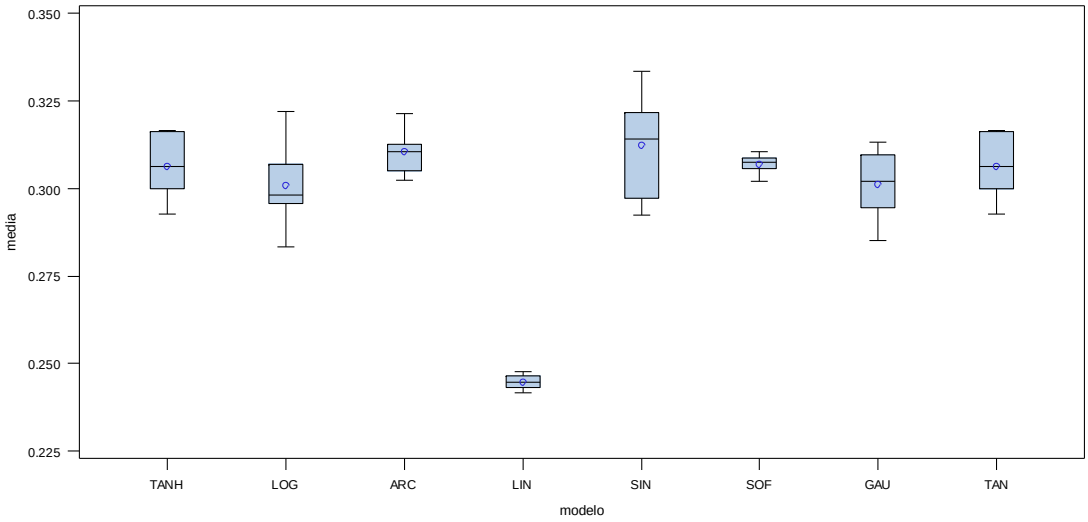
Para comprobar cómo responde cada función de activación con cada número de nodos, creamos una tabla donde recogemos de forma gráfica las distintas funciones de activación con los distintos números de nodos para de esa forma poder determinar cuál ofrece mejores resultados.

Los resultados son los siguientes.

**Con 11 nodos ocultos.**




**Con 20 nodos ocultos.**

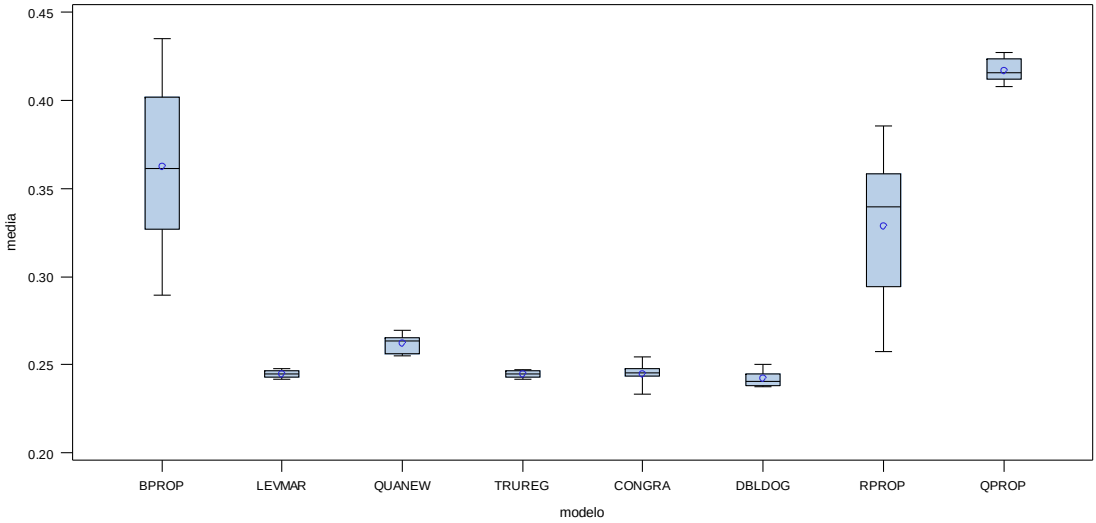


Observando los gráficos vemos que la función de activación que obtiene un menor ECM en media con cada uno de los números de nodos es la función LIN. El funcionamiento de la función de activación lineal es con diferencia la que mejor respuesta obtiene.

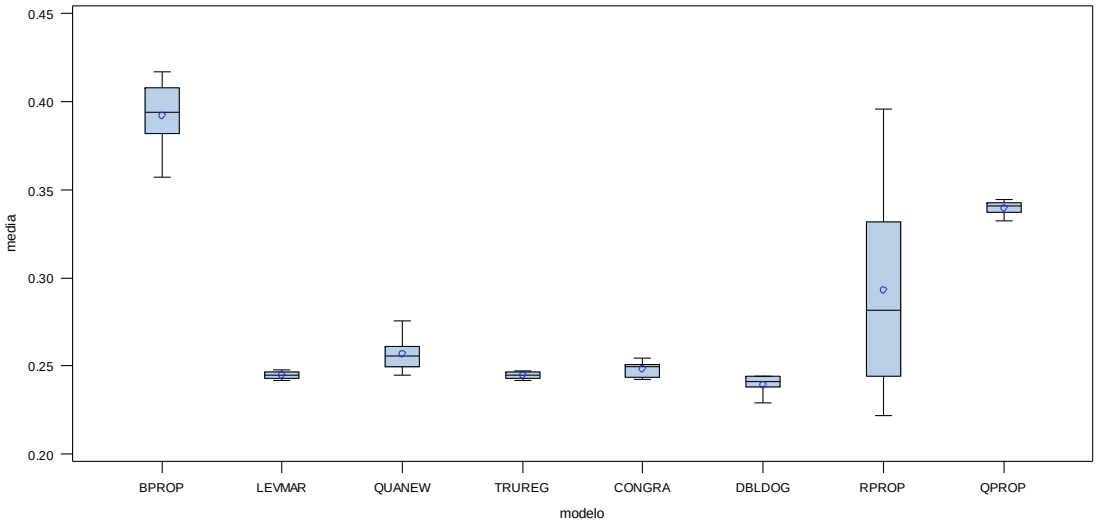
- Algoritmo de optimización

Después de decidir la función de activación, pasamos a decidir qué algoritmo de optimización obtiene mejores resultados. Los algoritmos que comprobaremos serán ocho, BPROP, LEVMAR, QUANEW, TRUREG, CONGRA, DBLDOG, RPROP y QPROP a través de la macro ***%algovalcruza***. Esta macro compara cada uno de los algoritmos a través de redes neuronales mediante validación cruzada.
Los resultados son los siguientes.

**Con 11 nodos ocultos.**




**Con 20 nodos ocultos.**




Los mejores algoritmos son LEVMAR, TRUREG, CONGRA y DBLDOG.

Con todo esto, ahora tenemos que comprobar a través de validación cruzada cuál es el mejor modelo de redes neuronales según cada uno de los aspectos anteriores que ha

mostrado mejor resultado. Es decir, debemos comprobar según los números de nodos que han mostrado mejores resultados (11 y 20), junto con la mejor función de activación (LIN) y junto con los mejores algoritmos de optimización (LEVMAR, TRUREG, CONGRA y DBLDOG), qué modelo ofrece mejor resultado.

De esta forma, diseñamos la siguiente tabla en la que se muestran los modelos de redes neuronales a comprobar.

| Modelo | Número de nodos | Función de activación | Algoritmo de optimización |
|---|---|---|---|
| 1 | 11 | LIN | LEVMAR |
| 2 | 11 | LIN | TRUREG |
| 3 | 11 | LIN | CONGRA |
| 4 | 11 | LIN | DBLDOG |
| 5 | 20 | LIN | LEVMAR |
| 6 | 20 | LIN | TRUREG |
| 7 | 20 | LIN | CONGRA |
| 8 | 20 | LIN | DBLDOG |

*Tabla 9.1.2.3*

Con la otra semilla actuamos exactamente de la misma forma, y la tabla en la que se muestran los modelos de redes neuronales a comprobar es la siguiente.

| Modelo | Número de nodos | Función de activación | Algoritmo de optimización |
|---|---|---|---|
| 1 | 15 | LIN | LEVMAR |
| 2 | 15 | LIN | TRUREG |
| 3 | 15 | LIN | CONGRA |
| 4 | 15 | LIN | DBLDOG |

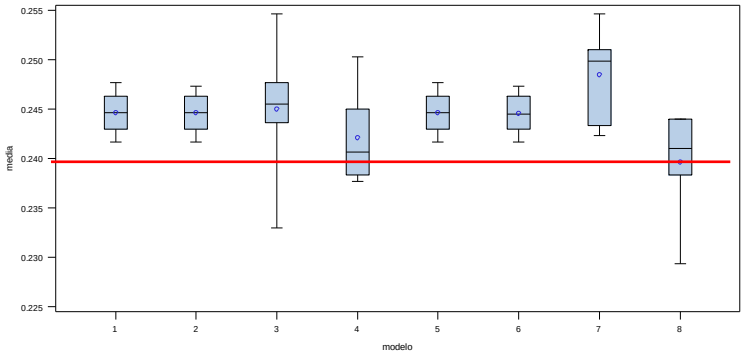
*Tabla 9.1.2.4*

Obtenemos el siguiente gráfico con los resultados de ambas semillas.

**Con la semilla original:** 12345 ~ 12350

El mejor modelo es el ocho con un ECM de 0.2396111.

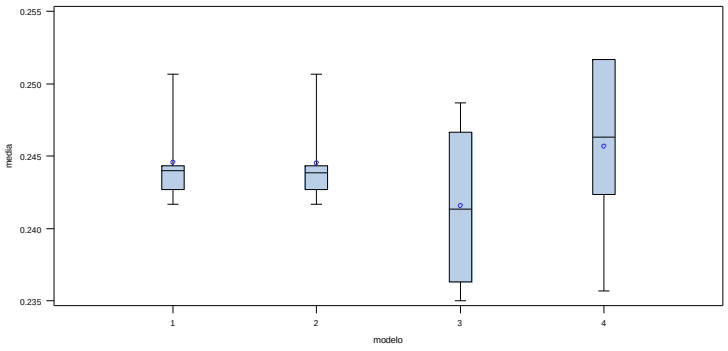
El mejor modelo de redes neuronales es aquel con 20 nodos, función de activación lineal y algoritmo de optimización DBLDOG.




**Tras cambiar la semilla:** 12340 ~ 12345

El mejor modelo es el tres con un ECM de 0.2415556.

El mejor modelo de redes neuronales es aquel con 15 nodos, función de activación lineal y algoritmo de optimización CONGRA.



TÉCNICAS BASADAS EN ÁRBOLES

Una vez realizado el análisis mediante redes neuronales, para intentar mejorar la clasificación conseguida en redes se prueban otros métodos de análisis basados en árboles. Estos métodos son tres, Bagging, Random Forest y Gradient Boosting.

### 9.1.3 BAGGING

Para realizar la técnica Bagging, utilizamos la macro ***%cruzadarandomforestbin*** la cual realiza validación cruzada repetida, utilizando el máximo número de variables (22 en este caso). Probamos modelos cambiando el tamaño mínimo de hoja final de 5 en 5 hasta un máximo de 50.

Los resultados son los siguientes.

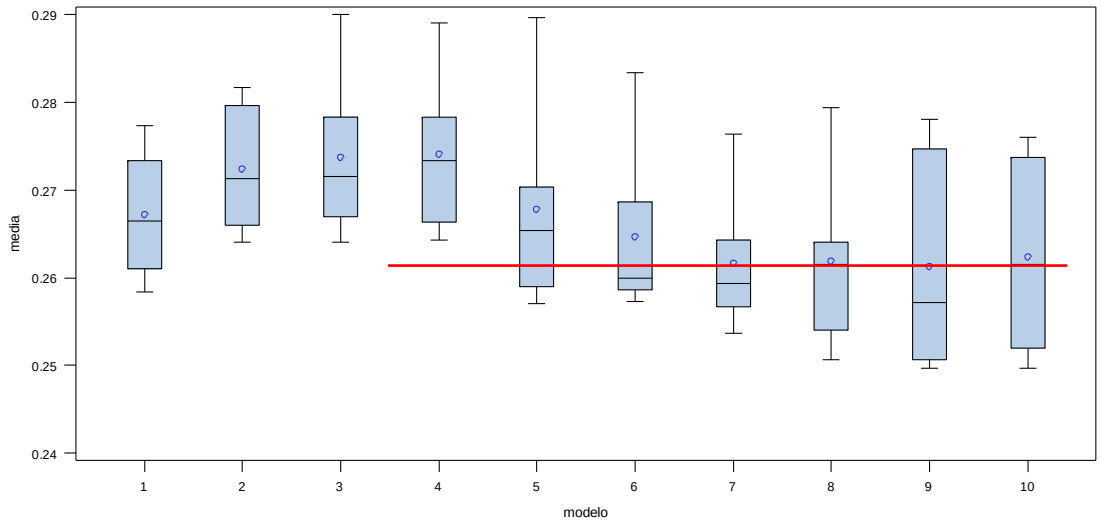



*Gráfico 9.1.3.1*

Vemos que el mejor modelo es el nueve, que es aquel con un tamaño mínimo de hoja final de 45.

Ahora probamos modelos con todas las variables, con un tamaño mínimo de hoja final de 45 y cambiando las divisiones máximas de un nodo de 2 en 2 hasta un máximo de 20.

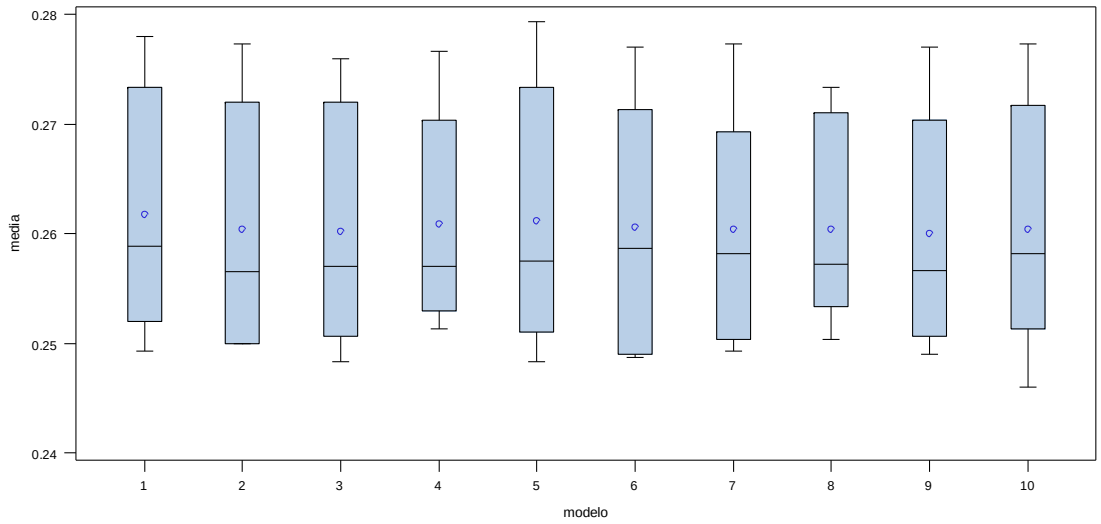



*Gráfico 9.1.3.2*

Gráficamente los resultados de los modelos son bastante similares, pero si analizamos sus valores numéricamente vemos que el modelo con mejores resultados es el modelo nueve (0.2600556), que es aquel con 18 divisiones máximas de un nodo.

Ahora probamos modelos con todas las variables, con un tamaño mínimo de hoja final de 45, 18 divisiones máximas de un nodo y cambiando la profundidad máxima de 2 en 2 hasta un máximo de 20.

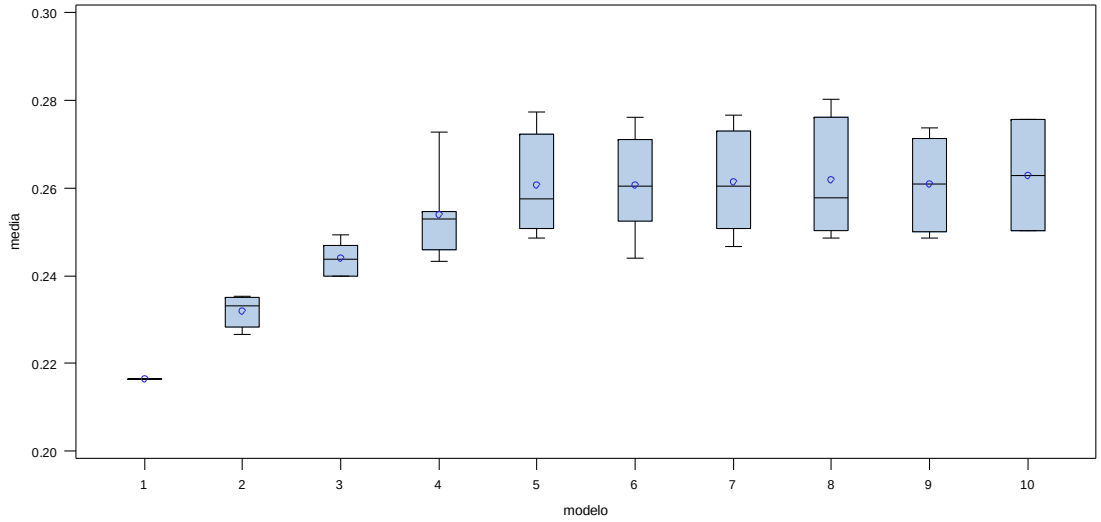



*Gráfico 9.1.3.3*

Vemos que el mejor modelo es el uno, que es aquel con una profundidad máxima de 2.

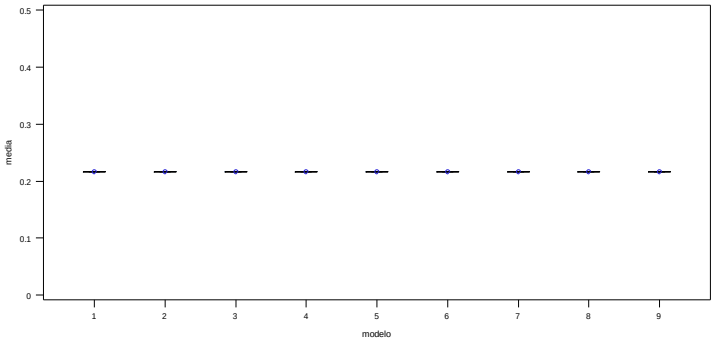
Por último, probamos modelos con todas las variables, con un tamaño mínimo de hoja final de 45, 18 divisiones máximas de un nodo, una profundidad máxima de 2 y cambiando el p-valor de 0.1 en 0.1 hasta un máximo de 0.9.

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el del p-valor.

**Con la semilla original:** 12345 ~ 12350

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.2163333) y desviación típica, así que elegimos el modelo uno que es aquel con un p-valor de 0.1.

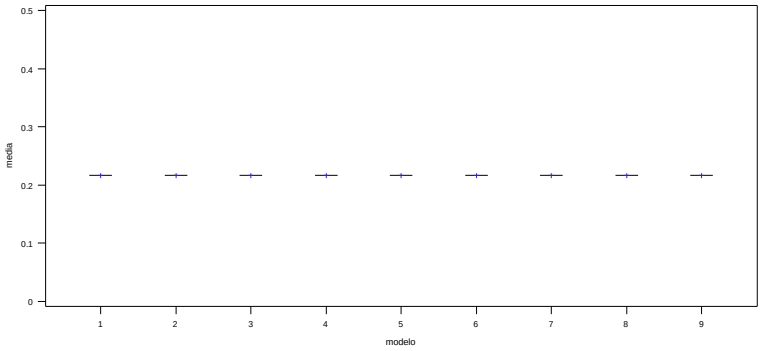
El mejor modelo mediante Bagging es aquel con todas las variables, con un tamaño mínimo de hoja final de 45, 18 divisiones máximas de un nodo, una profundidad máxima de 2 y un p-valor de 0.1.




**Tras cambiar la semilla:** 12340 ~ 12345

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.2163333) y desviación típica, así que elegimos el modelo uno que es aquel con un p-valor de 0.1.

El mejor modelo mediante Bagging es aquel con todas las variables, con un tamaño mínimo de hoja final de 45, 12 divisiones máximas de un nodo, una profundidad máxima de 2 y un p-valor de 0.1.




### 9.1.4 RANDOM FOREST

Para realizar la técnica Random Forest, utilizamos la macro ***%cruzadarandomforestbin*** la misma que utilizamos en el modelo anterior.

Inicialmente comprobamos cada modelo con un número de variables, pero sin llegar al número máximo ya que entonces estaríamos haciendo Bagging, con lo que comparamos los modelos desde teniendo una única variable hasta con veintiún variables.

Los resultados son los siguientes.

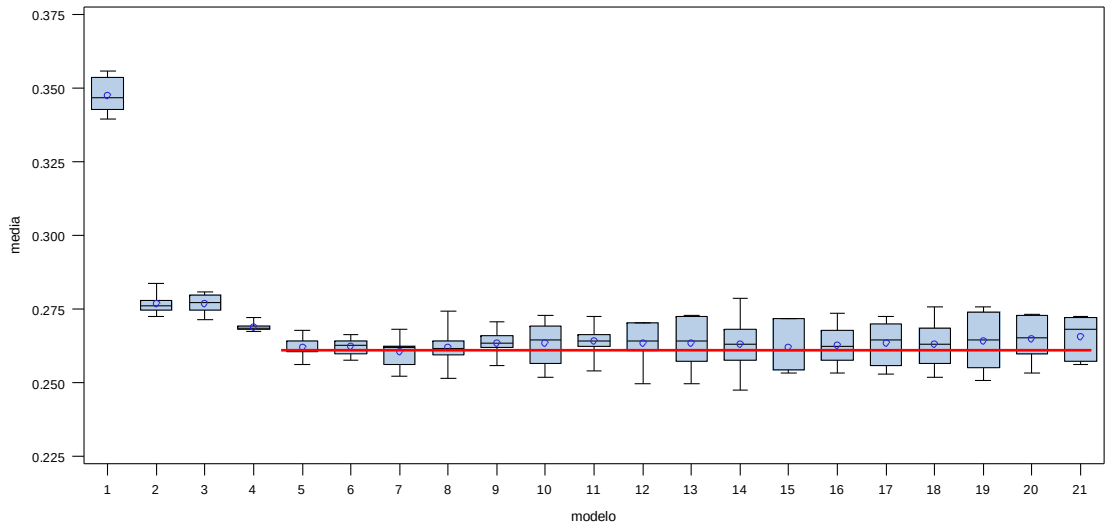



*Gráfico 9.1.4.1*

Como mejor funciona es con siete o quince. Como gráficamente son muy similares, comprobamos sus valores numéricamente y vemos que el modelo con siete variables tiene una media de 0.2605000 y el modelo con quince variables tiene una media de 0.2621111, así que nos quedamos con el modelo con siete variables.

Ahora probamos modelos con 7 variables y cambiando el tamaño mínimo de hoja final de 5 en 5 hasta un máximo de 50.

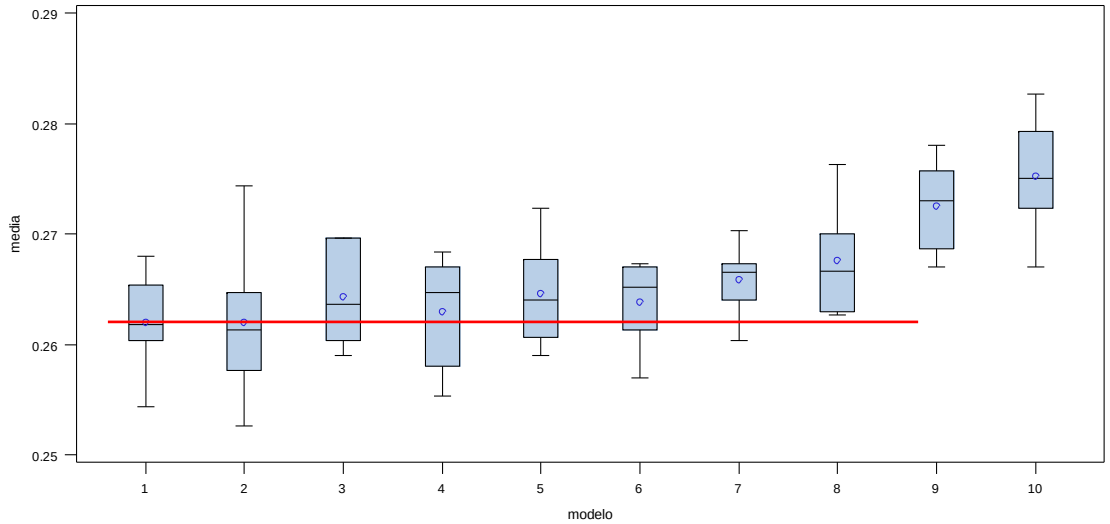



*Gráfico 9.1.4.2*

Gráficamente los valores son muy similares entre los modelos 1 y 2, pero fijándonos en su valor numérico descubrimos que el modelo que menor ECM tiene es el modelo 1, el de un tamaño mínimo de hoja final de 5, con un 0.2619444.

Ahora probamos modelos con 7 variables, con un tamaño mínimo de hoja final de 5 y cambiando las divisiones máximas de un nodo de 2 en 2 hasta un máximo de 20.

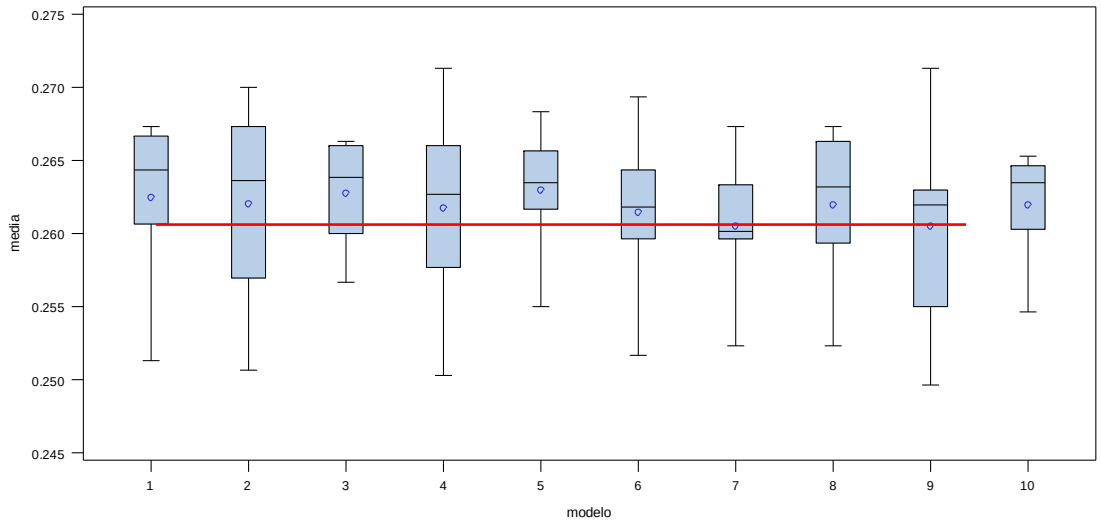



*Gráfico 9.1.4.3*

Gráficamente los resultados son muy similares entre el modelo 7 y el 9, así que comparamos sus valores numéricamente y vemos que los dos modelos tienen la misma media, pero el modelo 7 tiene menor desviación típica, así que el mejor modelo es el 7, que es aquel con un número de 14 divisiones máximas por un nodo.

Ahora probamos modelos con 7 variables, con un tamaño mínimo de hoja final de 5, 14 divisiones máximas de un nodo y cambiando la profundidad máxima de 2 en 2 hasta un máximo de 20.

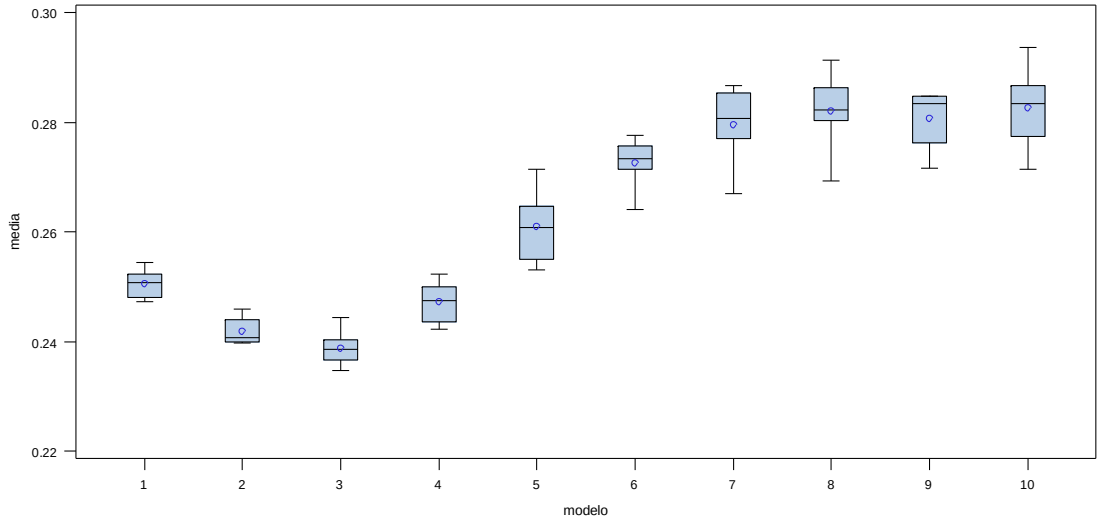



*Gráfico 9.1.4.4*

Vemos que el mejor modelo es el tres, que es aquel con una profundidad máxima de 6.

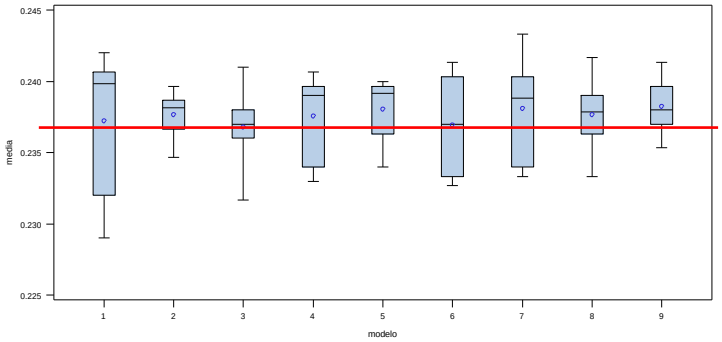
Por último, probamos modelos con 7 variables, con un tamaño mínimo de hoja final de 5, 14 divisiones máximas de un nodo, una profundidad máxima de 6 y cambiando el p-valor de 0.1 en 0.1 hasta un máximo de 0.9.

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el del p-valor.

**Con la semilla original:** 12345 ~ 12350

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que el mejor modelo es el 3 (0.2367778), así que elegimos el modelo tres que es aquel con un p-valor de 0.3.

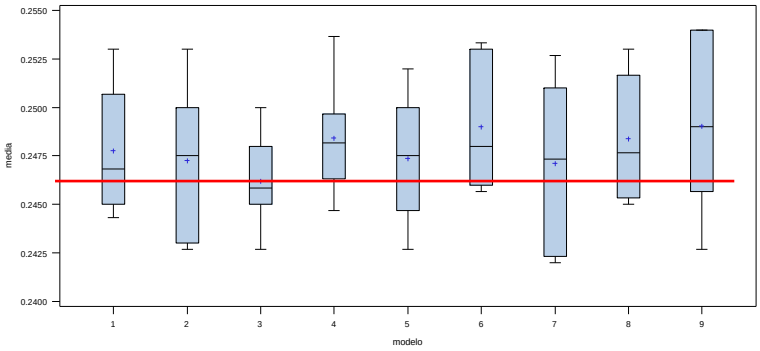
Por lo tanto, el mejor modelo mediante Random Forest es aquel con 7 variables, con un tamaño mínimo de hoja final de 5, 14 divisiones máximas de un nodo, una profundidad máxima de 6 y un p-valor de 0.3.




**Tras cambiar la semilla:** 12340 ~ 12345

El mejor modelo es el 3, con un p-valor de 0.3 (0.2465000).

Por lo tanto, el mejor modelo mediante Random Forest es aquel con 5 variables, con un tamaño mínimo de hoja final de 20, 16 divisiones máximas de un nodo, una profundidad máxima de 6 y un p-valor de 0.3.




### 9.1.5 GRADIENT BOOSTING

Para realizar la técnica Gradient Boosting, utilizamos la macro ***%cruzadatreeboostbin*** la cual realiza validación cruzada repetida.

Probamos modelos cambiando el shrink (constante v. de regularización) desde 0.05 a 0.1 y luego de 0.1 en 0.1 hasta 0.9.

Los resultados son los siguientes.

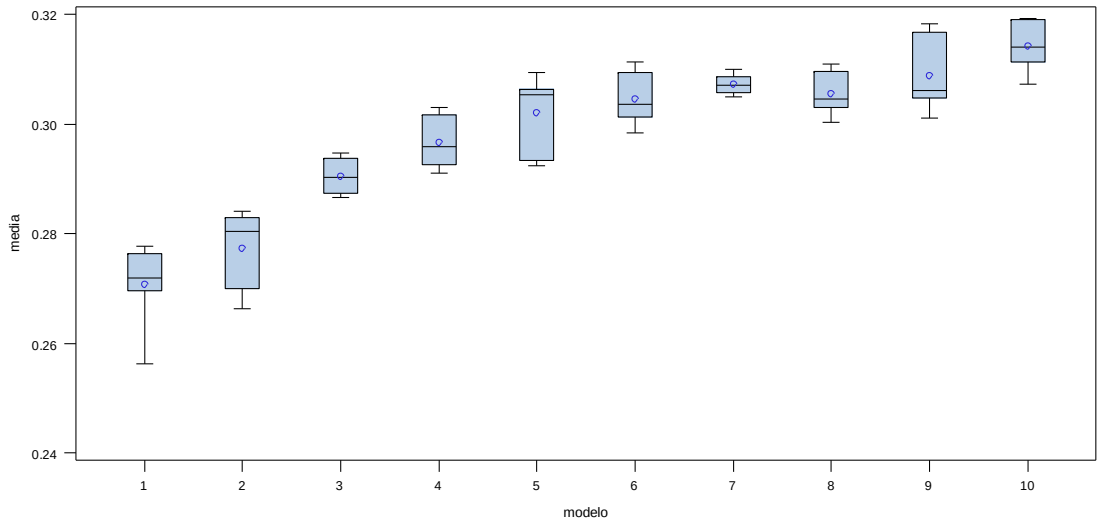



*Gráfico 9.1.5.1*

Vemos que el mejor modelo es el uno, que es aquel con un shrink de 0.05.

Ahora probamos modelos con un shrink de 0.05 y cambiando el tamaño mínimo de hoja final de 10 en 10 hasta 160.

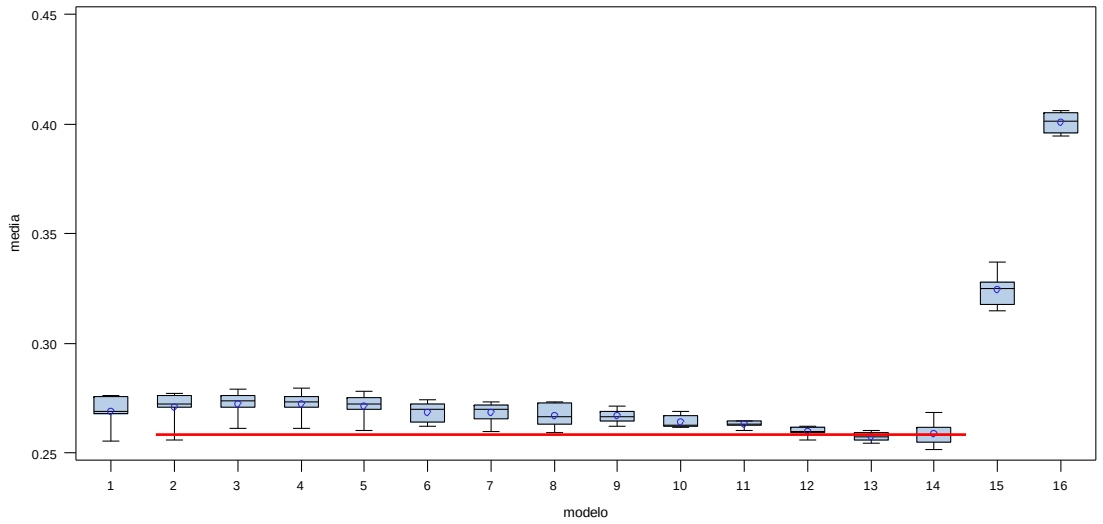



*Gráfico 9.1.5.2*

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que de todos los modelos, el que menor media tiene es el trece (0.2573889), así que elegimos el modelo trece que es aquel con un tamaño mínimo de hoja final de 130.

Ahora probamos modelos con un shrink de 0.05, con un tamaño mínimo de hoja final de 130 y con divisiones máximas en un nodo que van de 2 a 10 de uno en uno.

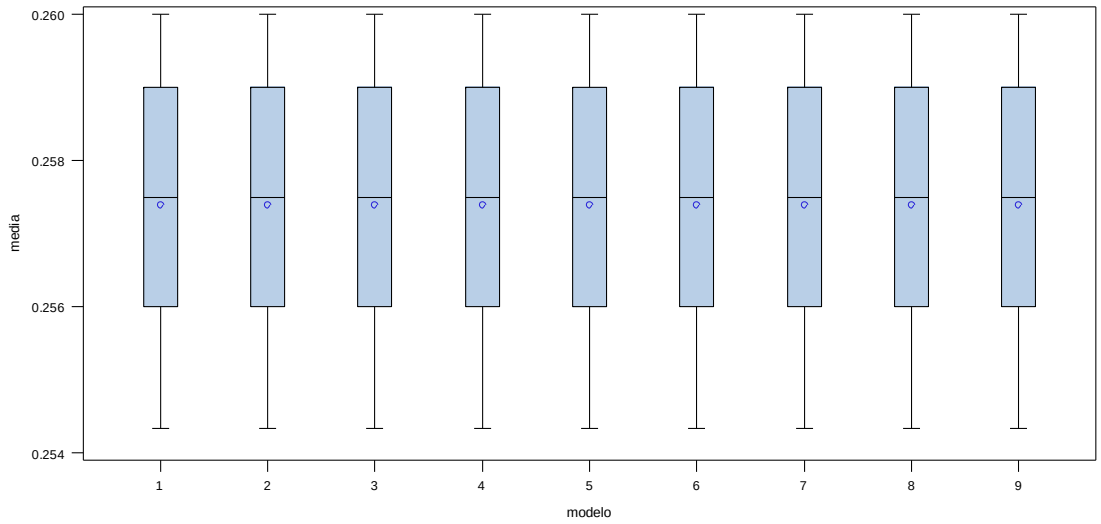



*Gráfico 9.1.5.3*

Gráficamente los resultados son idénticos, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.2573889), así que elegimos el modelo nueve que es aquel con diez divisiones máximas.

Ahora probamos modelos con un shrink de 0.05, con un tamaño mínimo de hoja final de 130, con 10 divisiones máximas en un nodo y con una profundidad máxima que va de 2 en 2 hasta 20.

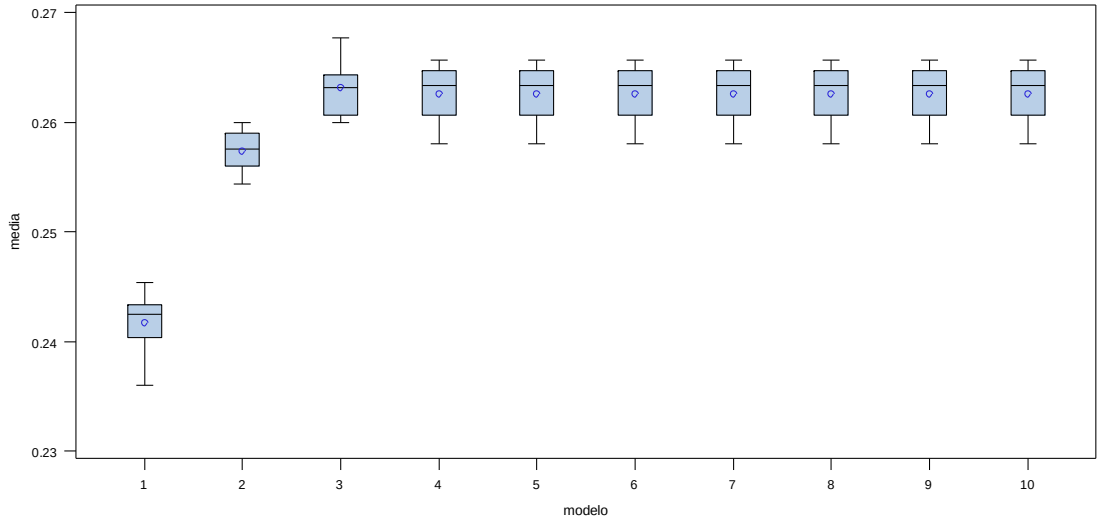



*Gráfico 9.1.5.4*

Vemos que el mejor modelo es el uno, que es aquel con una profundidad máxima de 2.

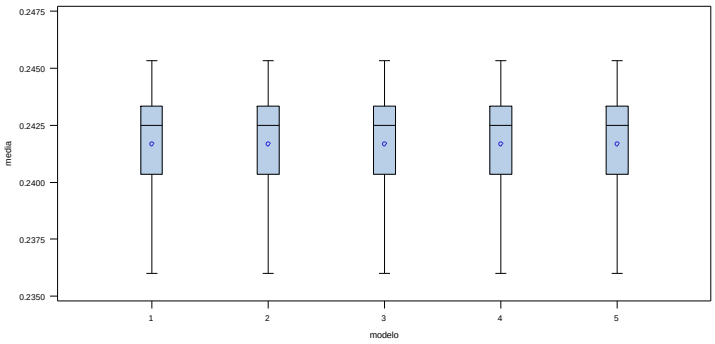
Por último, probamos modelos con un shrink de 0.05, con un tamaño mínimo de hoja final de 130, con 10 divisiones máximas en un nodo, con una profundidad máxima de 2 y con un número mínimo de observaciones para dividir un nodo que va de 5 en 5 hasta 25.

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el número mínimo de observaciones para dividir un nodo.

**Con la semilla original:** 12345 ~ 12350

Todos los modelos tienen la misma media (0.2416667) y desviación típica, así que elegimos el modelo cuatro que es aquel con 20 observaciones para dividir un nodo ya que es un número considerable de observaciones.

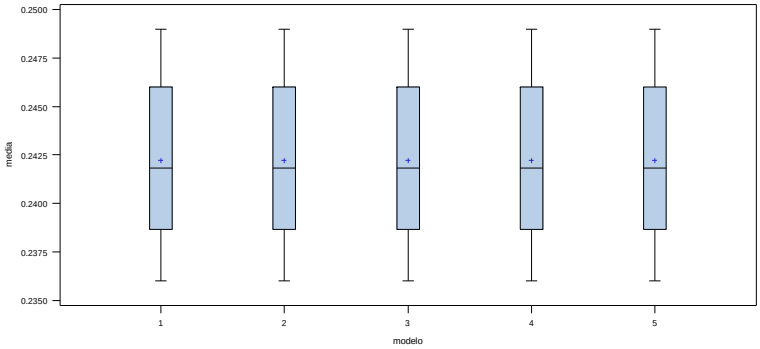
De esta forma, el mejor modelo mediante Gradient Boosting es aquel con un shrink de 0.05, con un tamaño mínimo de hoja final de 130, con 10 divisiones máximas en un nodo, con una profundidad máxima de 2 y con 20 observaciones para dividir un nodo.




**Tras cambiar la semilla:** 12340 ~ 12345

Todos los modelos tienen la misma media (0.2422222) y desviación típica, así que elegimos el modelo cuatro que es aquel con 20 observaciones para dividir un nodo ya que es un número considerable de observaciones.

De esta forma, el mejor modelo mediante Gradient Boosting es aquel con un shrink de 0.05, con un tamaño mínimo de hoja final de 130, con 10 divisiones máximas en un nodo, con una profundidad máxima de 2 y con 20 observaciones para dividir un nodo.




### 9.1.6 SVM

Para realizar la técnica SVM, utilizamos la macro ***%cruzadaSVMbin*** la cual realiza validación cruzada repetida.

Empezamos probando modelos cambiando el kernel entre “linear”, “polynom” y “RBF” manteniendo en todos los modelos el C (parámetro que controla el “soft margin”) en 10. En el modelo con kernel=polynom hacemos dos modelos, uno con k_par=2 y otro con k_par=3 (el 2 o 3 indica el grado del polinomio, si 2 o 3, cuanto más alto más complejo), y en el modelo con kernel=RBF, k_par lo dejamos con la opción gamma.

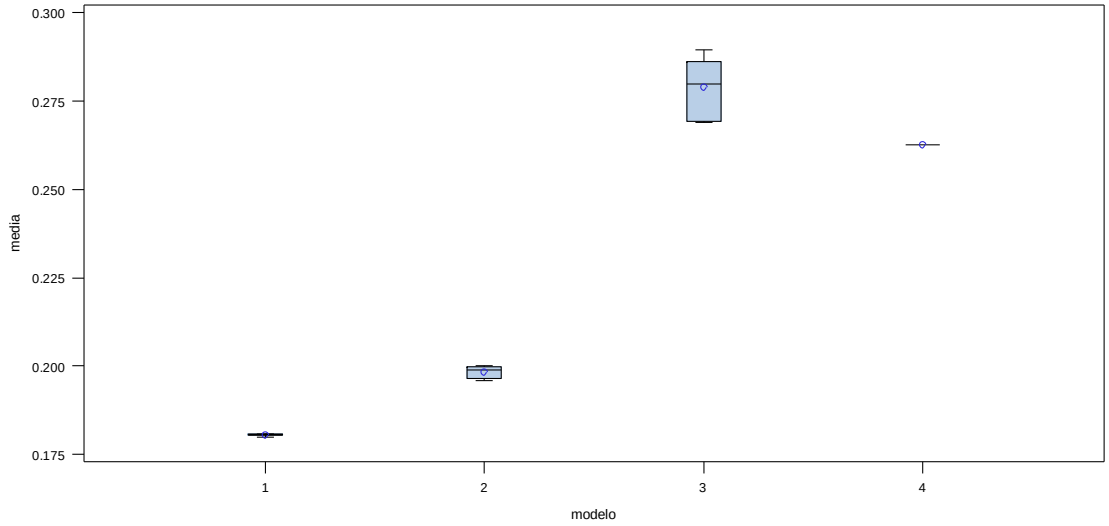



*Gráfico 9.1.6.1*

El mejor modelo es el uno, que es aquel con kernel=linear y C=10.

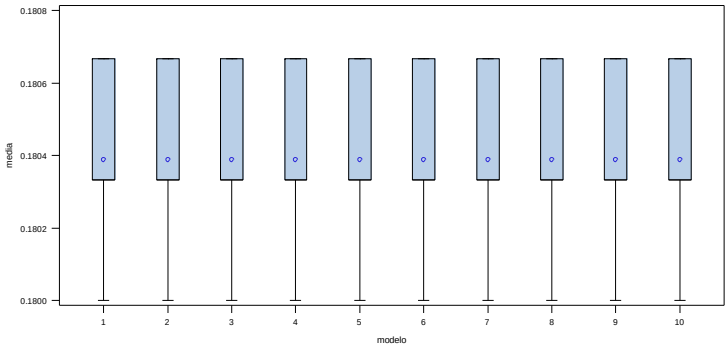
Probamos este mismo modelo pero con un C menor, con uno que vaya de 1 a 10 de uno en uno (cuanto C más grande resultará un modelo más grande y ajustado con menos sesgo y más varianza, y cuanto C más pequeño resultará un modelo más pequeño y simple con más sesgo y menos varianza).

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el tamaño de C.

**Con la semilla original:** 12345 ~ 12350

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.1803889) y desviación típica. De esta forma, elegimos el modelo dos ya que tiene un C menor y sería un modelo más simple, aunque sin ser el más simple de todos.

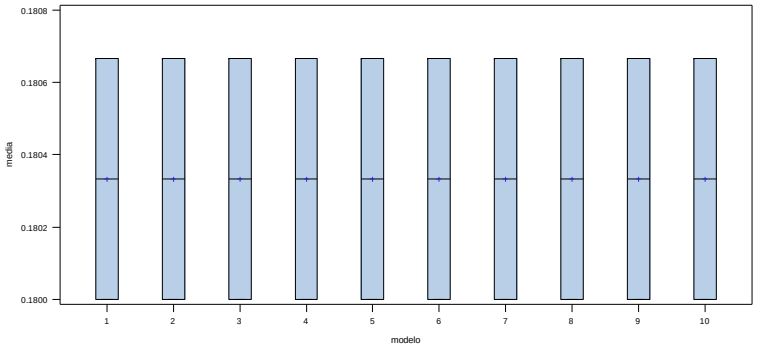
El mejor modelo mediante SVM es aquel con kernel=linear y C=2.




**Tras cambiar la semilla:** 12340 ~ 12345

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.1803333) y desviación típica. De esta forma, elegimos el modelo dos ya que tiene un C menor y sería un modelo más simple, aunque sin ser el más simple de todos.

El mejor modelo mediante SVM es aquel con kernel=linear y C=2.



## 9.2 VARIABLES WOE

Una vez construidos los modelos con las variables binarias procedemos a construir los modelos con las variables WOE.

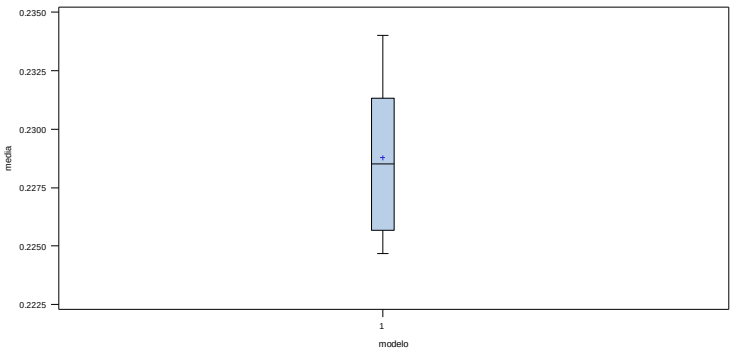
Procederemos de la misma forma que con las variables binarias a la hora de construir los modelos, pero esta vez simplemente utilizando este otro tipo de variables, por ese motivo de cada técnica únicamente mostraremos los gráficos en los que se muestra la comparación final de los modelos (los gráficos se encuentran en el anexo).

### 9.2.1 REGRESIÓN LOGÍSTICA

Para la regresión logística utilizamos la macro ***%cruzadalogistica*** la cual hace validación cruzada logística para variables dependientes binarias. Los resultados con cada una de las semillas son los siguientes.

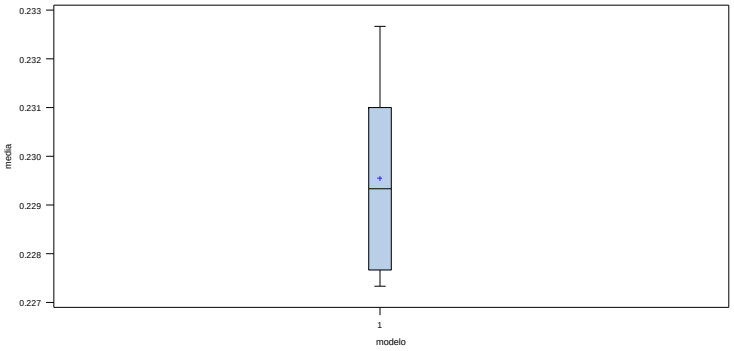
**Con la semilla original:** 12345 ~ 12350

**Media:** 0.2287778




**Tras cambiar la semilla:** 12340 ~ 12345

**Media:** 0.2295556




### 9.2.2 REDES NEURONALES

La determinación del mejor modelo de redes neuronales se llevará a cabo mediante un proceso de "ensayo y error" donde iremos añadiendo distintos aspectos de la red neuronal como número de nodos, función de activación, algoritmo, etc. E iremos variando cada uno de esos aspectos para comprobar cómo evolucionan los distintos modelos.

- Número de nodos

De la misma forma que con las variables binarias, construimos la tabla con la que establecer un número adecuado de nodos con el que partir, a través de la siguiente ecuación.

$$h(k+1)+h+1=\frac{nº\ observaciones}{nº\ de\ observaciones\ por\ parámetro}$$

Al no haber cambiado el número de observaciones, la tabla es la misma que la anterior.

| Observaciones por parámetro | Ecuación | h |
|---|---|---|
| 15 | $h(8+1)+h+1=\frac{3000}{15}\rightarrow 10h+1=\frac{3000}{15}$ | $h=19$ |
| 20 | $h(8+1)+h+1=\frac{3000}{20}\rightarrow 10h+1=\frac{3000}{20}$ | $h=14$ |
| 25 | $h(8+1)+h+1=\frac{3000}{25}\rightarrow 10h+1=\frac{3000}{25}$ | $h=11$ |

*Tabla 9.2.2.1*

Utilizamos la macro ***%repito*** en la que comprobamos con cada número de nodos obtenido en la tabla anterior y a través del error cuadrático medio, qué número de nodos muestran mejores resultados, y los resultados indican que a mayor número de nodos, menor error cuadrático medio. Esto tiene sentido ya que a un mayor número de nodos mejor se podrá ajustar la red a los datos, pero hay que tener cuidado de no utilizar un número excesivo de nodos para evitar el sobreajuste.

Una vez encontramos que los posibles mejores números de nodos para construir las redes neuronales son con 11, 14 o 18 nodos, para comprobar cuál es el mejor de todos ellos utilizaremos las macros ***%nodosvalcruza***, ***%numeronodos*** y ***%variar***. La macro ***%nodosvalcurza*** utiliza unos parámetros fundamentales de la red neuronal mediante validación cruzada repetida, siendo este un sistema de gran potencia que permite evaluar la variabilidad del modelo.

Probamos seleccionando los nodos de uno en uno desde el 10 al 20 para que haya margen y el mejor modelo se consigue con el segundo número de nodos, con 11 nodos.

Ahora probamos con la macro ***%numeronodos***. Esta macro nos ofrece los siguientes valores con los que poder evaluar qué número de nodos obtiene los mejores resultados, porcentaje de verdaderos negativos, porcentaje de falsos negativos, porcentaje de verdaderos positivos, porcentaje de falsos positivos, sensibilidad, especificidad, tasa de fallos, tasa de aciertos, precisión y F_M $\left(F_M=\frac{2*sensibilidad*precisión}{sensibilidad+precisión}\right)$.

Como comparar estas diez tablas a la vez puede ser bastante complicado, nos vamos a centrar en los verdaderos negativos y positivos, sensibilidad, especificidad, tasa de aciertos y tasa de fallos. Descubrimos que los nodos que ofrecen mejores resultados son 10, 12 y 15.

Para terminar con la búsqueda de aquellos nodos que ofrecen mejores resultados, utilizamos por último la macro ***%variar***. Esta macro nos ofrece los mismos valores que

la macro anterior, así que decidimos centrarnos únicamente en los verdaderos negativos y positivos, sensibilidad, especificidad, tasa de aciertos y tasa de fallos. Descubrimos que los nodos que ofrecen mejores resultados son 10 y 12.

Así que como conclusión obtenemos que, por dos de los tres métodos de selección de número de nodos, los números de nodos que ofrece mejores resultados son 10 y 12 nodos.

- Función de activación

El siguiente paso es comprobar qué función de activación ofrece mejores resultados con el mejor número de nodos establecido anteriormente. Las funciones de activación que comprobaremos son TANH, LOG, ARC, LIN, SIN, SOF, GAU y TAN. Para ello utilizaremos la macro ***%activalcruza*** que compara cada función a través de una red neuronal con validación cruzada repetida.

Para comprobar cómo responde cada función de activación con cada número de nodos, creamos una tabla donde recogemos de forma gráfica las distintas funciones de activación con los distintos números de nodos para de esa forma poder determinar cuál ofrece mejores resultados.

Encontramos que la función de activación que obtiene un menor ECM en media es la función LIN. El funcionamiento de la función de activación lineal es con diferencia la que mejor respuesta obtiene.

- Algoritmo de optimización

Después de decidir la función de activación, pasamos a decidir qué algoritmo de optimización obtiene mejores resultados. Los algoritmos que comprobaremos serán ocho, BPROP, LEVMAR, QUANEW, TRUREG, CONGRA, DBLDOG, RPROP y QPROP a través de la macro ***%algovalcruza***. Esta macro compara cada uno de los algoritmos a través de redes neuronales con validación cruzada.

Los mejores algoritmos son LEVMAR, QUANEW, TRUREG, CONGRA y DBLDOG.

Con todo esto, ahora tenemos que comprobar a través de la validación cruzada cuál es el mejor modelo de redes neuronales según cada uno de los aspectos anteriores que ha mostrado mejor resultado. Es decir, debemos comprobar según el número de nodos que ha mostrado mejores resultados (10 y 12), junto con la mejor función de activación (LIN) y junto con los mejores algoritmos de optimización (LEVMAR, QUANEW, TRUREG, CONGRA y DBLDOG), qué modelo ofrece mejor resultado.

De esta forma, diseñamos la siguiente tabla en la que se muestran los modelos de redes neuronales a comprobar.

| Modelo | Número de nodos | Función de activación | Algoritmo de optimización |
|---|---|---|---|
| 1 | 10 | LIN | LEVMAR |
| 2 | 10 | LIN | QUANEW |
| 3 | 10 | LIN | TRUREG |
| 4 | 10 | LIN | CONGRA |
| 5 | 10 | LIN | DBLDOG |
| 6 | 12 | LIN | LEVMAR |
| 7 | 12 | LIN | QUANEW |
| 8 | 12 | LIN | TRUREG |
| 9 | 12 | LIN | CONGRA |
| 10 | 12 | LIN | DBLDOG |

*Tabla 9.2.2.2*

Con la otra semilla actuamos exactamente de la misma forma, y la tabla en la que se muestran los modelos de redes neuronales a comprobar es la siguiente.

| Modelo | Número de nodos | Función de activación | Algoritmo de optimización |
|---|---|---|---|
| 1 | 10 | LIN | LEVMAR |
| 2 | 10 | LIN | QUANEW |
| 3 | 10 | LIN | TRUREG |
| 4 | 10 | LIN | CONGRA |
| 5 | 10 | LIN | DBLDOG |

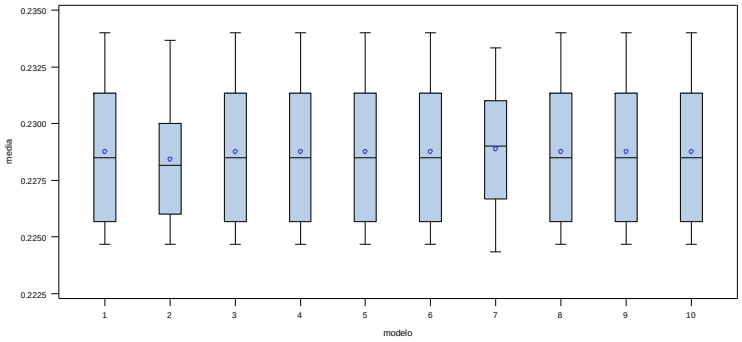
*Tabla 9.2.2.3*

Obtenemos el siguiente gráfico con los resultados de ambas semillas.

**Con la semilla original:** 12345 ~ 12350

En este caso, aunque vemos que los modelos tienen valores similares, numéricamente el mejor modelo en redes neuronales es el modelo 2 con un ECM de 0.2284444.

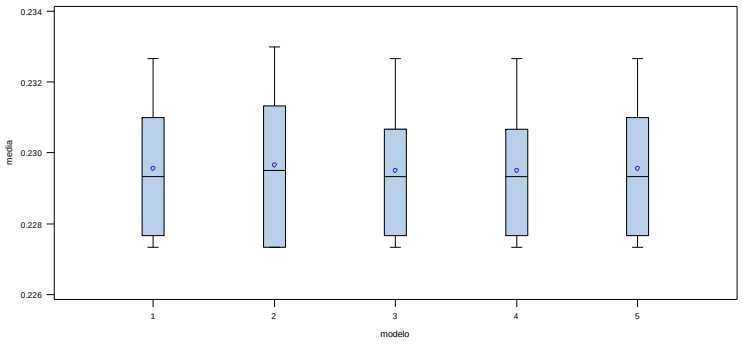
El mejor modelo de redes neuronales es aquel con 10 nodos, función de activación lineal y algoritmo de optimización QUANEW.




**Tras cambiar la semilla:** 12340 ~ 12345

El mejor modelo es el tres o cuatro con un ECM de 0.2295000.

Luego el mejor modelo de redes neuronales es aquel con 10 nodos, función de activación lineal y algoritmo de optimización TRUTEG o CONGRA.




## TÉCNICAS BASADAS EN ÁRBOLES

Una vez realizado el análisis mediante redes neuronales, para intentar mejorar la clasificación conseguida en redes se prueban otros métodos de análisis basados en árboles. Estos métodos son tres, Bagging, Random Forest y Gradient Boosting.

### 9.2.3 BAGGING

Para realizar la técnica Bagging, utilizamos la macro ***%cruzadarandomforestbin*** la cual realiza validación cruzada repetida, utilizando el máximo número de variables (8 en este caso). Probamos modelos cambiando el tamaño mínimo de hoja final de 5 en 5 hasta un máximo de 50. Hallamos que el mejor modelo es el diez, que es aquel con un tamaño mínimo de hoja final de 50.

Ahora probamos modelos con todas las variables, con un tamaño mínimo de hoja final de 50 y cambiando las divisiones máximas de un nodo de 2 en 2 hasta un máximo de 20. Los resultados de los modelos son bastante similares, pero si analizamos sus valores numéricamente vemos que el modelo con mejores resultados es el modelo uno, que es aquel con 2 divisiones máximas de un nodo.

Ahora probamos modelos con todas las variables, con un tamaño mínimo de hoja final de 50, 2 divisiones máximas de un nodo y cambiando la profundidad máxima de 2 en 2 hasta un máximo de 20. El mejor modelo es el uno, que es aquel con una profundidad máxima de 2.

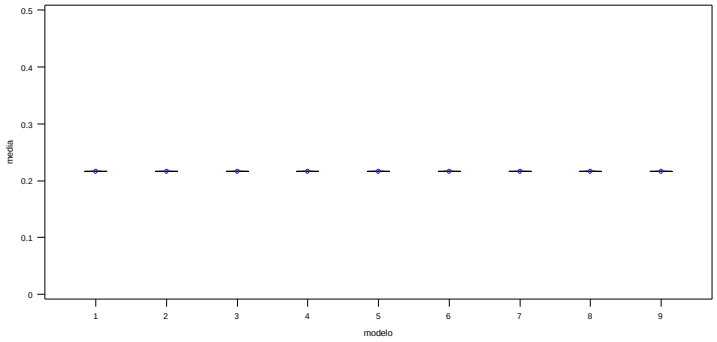
Por último, probamos modelos con todas las variables, con un tamaño mínimo de hoja final de 50, 2 divisiones máximas de un nodo, una profundidad máxima de 2 y cambiando el p-valor de 0.1 en 0.1 hasta un máximo de 0.9.

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el del p-valor.

**Con la semilla original:** 12345 ~ 12350

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.2163333) y desviación típica, así que elegimos el modelo uno que es aquel con un p-valor de 0.1.

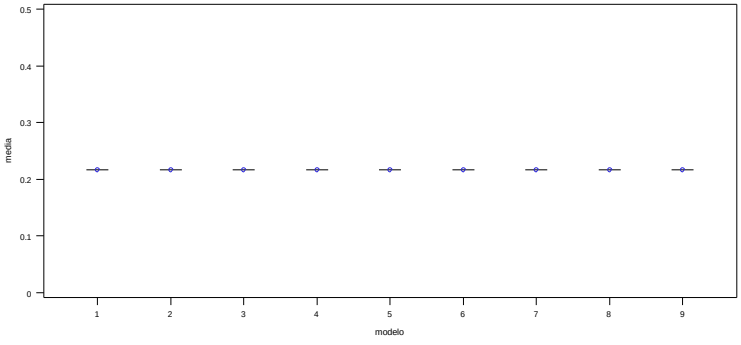
Luego el mejor modelo mediante Bagging es aquel con todas las variables, con un tamaño mínimo de hoja final de 50, 2 divisiones máximas de un nodo, una profundidad máxima de 2 y un p-valor de 0.1.




**Tras cambiar la semilla:** 12340 ~ 12345

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.2163333) y desviación típica, así que elegimos el modelo uno que es aquel con un p-valor de 0.1.

Luego el mejor modelo mediante Bagging es aquel con todas las variables, con un tamaño mínimo de hoja final de 50, 2 divisiones máximas de un nodo, una profundidad máxima de 2 y un p-valor de 0.1.



### 9.2.4 RANDOM FOREST

Para realizar la técnica Random Forest, utilizamos la macro ***%cruzadarandomforestbin*** la misma que utilizamos en el modelo anterior.

Inicialmente comprobamos cada modelo con un número de variables, pero sin llegar al número máximo ya que entonces estaríamos haciendo Bagging, con lo que comparamos los modelos desde teniendo una única variable hasta con siete variables.

Como mejor funciona es con dos o tres variables. Como gráficamente son muy similares, comprobamos sus valores numéricamente y vemos que el modelo con dos variables tiene una media de 0.2730000 y el modelo con tres variables tiene una media de 0.2728333, así que nos quedamos con el modelo con tres variables.

Ahora probamos modelos con 3 variables y cambiando el tamaño mínimo de hoja final de 5 en 5 hasta un máximo de 50. Descubrimos que el mejor modelo es el uno, que es aquel con un tamaño mínimo de hoja final de 5.

Ahora probamos modelos con 3 variables, con un tamaño mínimo de hoja final de 5 y cambiando las divisiones máximas de un nodo de 2 en 2 hasta un máximo de 20. Los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que el mejor modelo es el 10, que es aquel con un número de 20 divisiones máximas por un nodo.

Ahora probamos modelos con 3 variables, con un tamaño mínimo de hoja final de 5, 20 divisiones máximas de un nodo y cambiando la profundidad máxima de 2 en 2 hasta un máximo de 20. El mejor modelo es el dos, que es aquel con una profundidad máxima de 4.

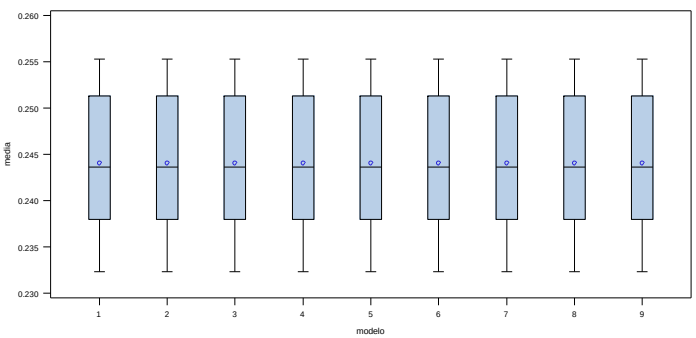
Por último, probamos modelos con 3 variables, con un tamaño mínimo de hoja final de 5, 20 divisiones máximas de un nodo, una profundidad máxima de 4 y cambiando el p valor de 0.1 en 0.1 hasta un máximo de 0.9.

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el del p-valor.

**Con la semilla original:** 12345 ~ 12350

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.2440556) y desviación típica, así que elegimos el modelo uno que es aquel con un p-valor de 0.1.

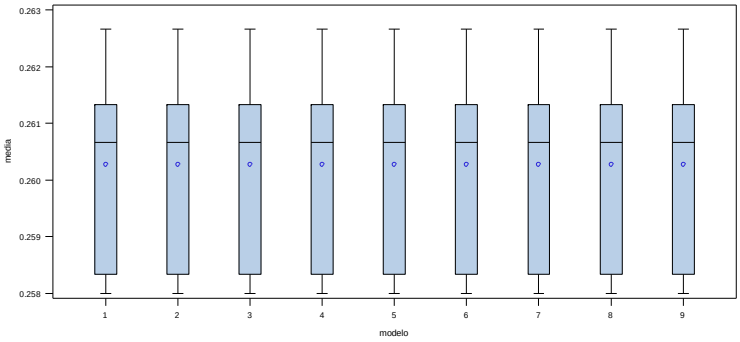
Por lo tanto, el mejor modelo mediante Random Forest es aquel con 3 variables, con un tamaño mínimo de hoja final de 5, 20 divisiones máximas de un nodo, una profundidad máxima de 4 y un p-valor de 0.1.



**Tras cambiar la semilla:** 12340 ~ 12345

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que todos los modelos tienen la misma media (0.2602778) y desviación típica, así que elegimos el modelo uno que es aquel con un p-valor de 0.1.

Por lo tanto, el mejor modelo mediante Random Forest es aquel con 2 variables, con un tamaño mínimo de hoja final de 5, 20 divisiones máximas de un nodo, una profundidad máxima de 6 y un p-valor de 0.1.




### 9.2.5 GRADIENT BOOSTING

Para realizar la técnica Gradient Boosting, utilizamos la macro ***%cruzadatreeboostbin*** la cual realiza validación cruzada repetida.

Probamos modelos cambiando el shrink (constante v. de regularización) desde 0.05, 0.1 y de 0.1 en 0.1 hasta 0.9. Hallamos que el mejor modelo es el uno, que es aquel con un shrink de 0.05.

Ahora probamos modelos con un shrink de 0.05 y cambiando el tamaño mínimo de hoja final de 10 en 10 hasta 160. Los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que de todos los modelos, el que menor media tiene es el catorce (0.2681111), así que elegimos el modelo catorce que es aquel con un tamaño mínimo de hoja final de 140.

Ahora probamos modelos con un shrink de 0.05, con un tamaño mínimo de hoja final de 140 y con divisiones máximas en un nodo que van de 2 a 10 de uno en uno. Los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que de todos los modelos, el que menor media tiene es el cuatro (0.2663333), así que elegimos el modelo cuatro que es aquel con cinco divisiones máximas.

Ahora probamos modelos con un shrink de 0.05, con un tamaño mínimo de hoja final de 140, con 5 divisiones máximas en un nodo y con una profundidad máxima que va de 2 en 2 hasta 20. El mejor modelo es el uno, que es aquel con una profundidad máxima de 2.

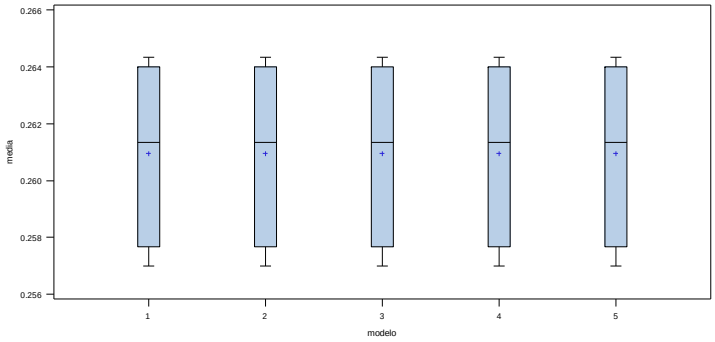
Por último, probamos modelos con un shrink de 0.05, con un tamaño mínimo de hoja final de 140, con 5 divisiones máximas en un nodo, con una profundidad máxima de 2 y con un número mínimo de observaciones para dividir un nodo que va de 5 en 5 hasta 25.

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el número mínimo de observaciones para dividir un nodo.

**Con la semilla original:** 12345 ~ 12350

Todos los modelos tienen la misma media (0.2609444) y desviación típica, así que elegimos el modelo cuatro que es aquel con 20 observaciones para dividir un nodo ya que es un número considerable de observaciones.

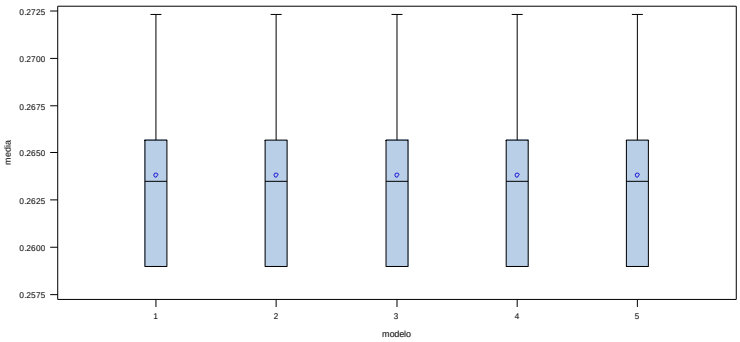
De esta forma, el mejor modelo mediante Gradient Boosting es aquel con un shrink de 0.05, con un tamaño mínimo de hoja final de 140, con 5 divisiones máximas en un nodo, con una profundidad máxima de 2 y con 20 observaciones para dividir un nodo.




**Tras cambiar la semilla:** 12340 ~ 12345

Todos los modelos tienen la misma media (0.2638333) y desviación típica, así que elegimos el modelo cuatro que es aquel con 20 observaciones para dividir un nodo ya que es un número considerable de observaciones.

De esta forma, el mejor modelo mediante Gradient Boosting es aquel con un shrink de 0.05, con un tamaño mínimo de hoja final de 150, con 10 divisiones máximas en un nodo, con una profundidad máxima de 2 y con 20 observaciones para dividir un nodo.




### 9.2.6 SVM

Para realizar la técnica SVM, utilizamos la macro ***%cruzadaSVMbin*** la cual realiza validación cruzada repetida.

Empezamos probando modelos cambiando el kernel entre “linear”, “polynom” y “RBF” manteniendo en todos los modelos el C (parámetro que controla el “soft margin”) en 10.

En el modelo con kernel=polynom hacemos dos modelos, uno con k_par=2 y otro con k_par=3 (el 2 o 3 indica el grado del polinomio, si 2 o 3, cuanto más alto más complejo), y en el modelo con kernel=RBF, k_par lo dejamos con la opción gamma.

El mejor modelo es el uno, que es aquel con kernel=linear y C=10.

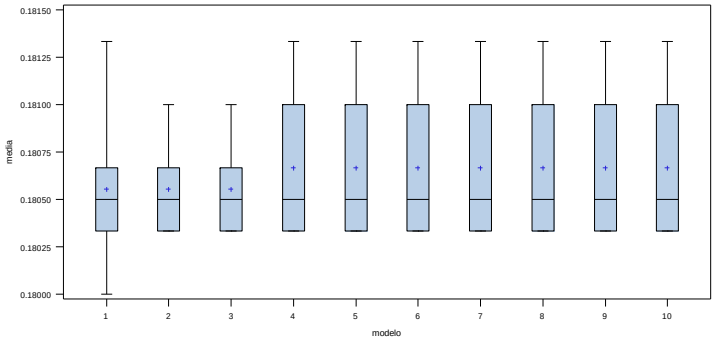
Probamos este mismo modelo pero con un C menor, con uno que vaya de 1 a 10 de uno en uno (cuanto C más grande resultará un modelo más grande y ajustado con menos sesgo y más varianza, y cuanto C más pequeño resultará un modelo más pequeño y simple con más sesgo y menos varianza).

Con la otra semilla actuamos exactamente de la misma forma, y construimos la siguiente tabla en la que mostramos los siguientes gráficos con los resultados de ambas semillas en el último paso a comprobar que es el tamaño de C.

**Con la semilla original:** 12345 ~ 12350

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que los modelos 1, 2 y 3 son los que menor media tienen (0.1805556), y dentro de ellos, los modelos 2 y 3 son los que tienen una menor desviación típica, así que elegimos el modelo dos ya que tiene un C menor y sería un modelo más simple.

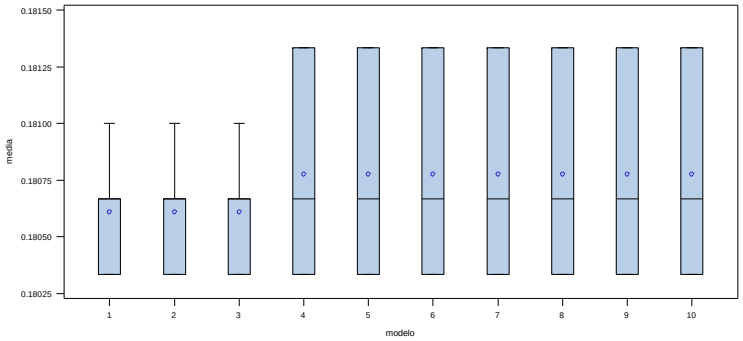
El mejor modelo mediante SVM es aquel con kernel=linear y C=2.




**Tras cambiar la semilla:** 12340 ~ 12345

Gráficamente los resultados son muy similares, así que comparamos sus valores numéricamente y vemos que los modelos 1, 2 y 3 tienen la misma media (0.1806111) y desviación típica. De esta forma, elegimos el modelo dos ya que tiene un C menor y sería un modelo más simple, aunque sin ser el más simple de todos.

El mejor modelo mediante SVM es aquel con kernel=linear y C=2.




# 10. HARDWARE Y SOFTWARE EMPLEADO

Para que se tenga en cuenta a la hora de la interpretación de los resultados del estudio, sobre todo en el ámbito de tiempos de procesamiento, al ser este un aspecto directamente relacionado con las capacidades del equipo se facilitan las siguientes especificaciones del mismo.

En el apartado de hardware, el equipo cuenta con un procesador Intel Core i5-4210U 1.70GHz 2.40GHz, 8 GB de memoria RAM, disco duro de 1TB de capacidad y sistema operativo Windows 10 de 64 bits instalado sobre un SSD de 250GB.

En el apartado de software, aparte de emplear el sistema operativo Windows 10, para el desarrollo del análisis se emplearán los programas SAS 9.4 y SAS Enterprise Miner Workstation 14.1.

# 11. RESULTADOS

A continuación se muestra cada uno de los modelos ganadores dentro de cada técnica estadística para cada semilla de datos, tanto por capacidad de predicción como por tiempo de ejecución.

## 11.1 SEGÚN LA CAPACIDAD DE PREDICCIÓN

Según la capacidad de predicción de cada modelo, basada en el ECM de cada uno de ellos, obtenemos los siguientes resultados.

### 11.1.1 CON LA PRIMERA SEMILLA DE DATOS

Con las observaciones obtenidas mediante la primera semilla logramos los siguientes resultados.

| | Variables Binarias | | Variables WOE | |
|---|---|---|---|---|
| | **Modelo** | **ECM** | **Modelo** | **ECM** |
| **Regresión Logística** | - Variables binarias | 0.2446111 | - Variables WOE | 0.2287778 |
| **Redes Neuronales** | - 20 nodos.<br>- Función de activación lineal.<br>- Algoritmo de optimización DBLDOG. | 0.2396111 | - 10 nodos.<br>- Función de activación lineal.<br>- Algoritmo de optimización QUANEW. | 0.2284444 |
| **Bagging** | - Todas las variables.<br>- Tamaño mínimo de hoja final de 45.<br>- 18 divisiones máximas de un nodo.<br>- Profundidad máxima de 2.<br>- P-valor de 0.1. | 0.2163333 | - Todas las variables.<br>- Tamaño mínimo de hoja final de 50.<br>- 2 divisiones máximas de un nodo.<br>- Profundidad máxima de 2.<br>- P-valor de 0.1. | 0.2163333 |
| **Random Forest** | - 7 variables.<br>- Tamaño mínimo de hoja final de 5.<br>- 14 divisiones máximas de un nodo.<br>- Profundidad máxima de 6.<br>- P-valor de 0.3. | 0.2367778 | - 3 variables.<br>- Tamaño mínimo de hoja final de 5.<br>- 20 divisiones máximas de un nodo.<br>- Profundidad máxima de 4.<br>- P-valor de 0.1. | 0.2440556 |
| **Gradient Boosting** | - Shrink de 0.05.<br>- Tamaño mínimo de hoja final de 130.<br>- 10 divisiones máximas en un nodo.<br>- Profundidad máxima de 2.<br>- 20 observaciones para dividir un nodo. | 0.2416667 | - Shrink de 0.05.<br>- Tamaño mínimo de hoja final de 140.<br>- 5 divisiones máximas en un nodo.<br>- Profundidad máxima de 2.<br>- 20 observaciones para dividir un nodo. | 0.2609444 |
| **SVM** | - Kernel = linear.<br>- C = 2 | 0.1803889 | - Kernel = linear.<br>- C = 2 | 0.1805556 |

*Tabla 11.1.1.1*

Los resultados obtenidos mediante redes neuronales son significativamente mejores que los obtenidos mediante regresión logística, tanto mediante las variables binarias como mediante las variables WOE, e incluso obtenemos mejores resultados con las variables WOE que con las binarias en los casos de la regresión logística y redes neuronales.

La técnica que mejores resultados ha obtenido ha sido la de Support Vector Machine, por encima de la regresión logística, las redes neuronales y las demás técnicas basadas en árboles. Pero en este caso sí se obtuvieron mejores resultados a través de las variables binarias que mediante las variables WOE, aunque estamos hablando de una diferencia de diezmilésimas.

### 11.1.2 CON LA SEGUNDA SEMILLA DE DATOS

Con las observaciones obtenidas mediante la segunda semilla logramos los siguientes resultados.

| | Variables Binarias | | Variables WOE | |
|---|---|---|---|---|
| | **Modelo** | **ECM** | **Modelo** | **ECM** |
| **Regresión Logística** | - Variables binarias | 0.2445000 | - Variables WOE | 0.2295556 |
| **Redes Neuronales** | - 15 nodos.<br>- Función de activación lineal.<br>- Algoritmo de optimización CONGRA. | 0.2415556 | - 10 nodos.<br>- Función de activación lineal.<br>- Algoritmo de optimización TRUTEG o CONGRA. | 0.2295000 |
| **Bagging** | - Todas las variables.<br>- Tamaño máximo de hoja de 45.<br>- 12 divisiones máximas en un nodo.<br>- Profundidad máxima de 2.<br>- P-valor de 0.1. | 0.2163333 | - Todas las variables.<br>- Tamaño máximo de hoja de 50.<br>- 2 divisiones máximas.<br>- Profundidad máxima de 2.<br>- P-valor de 0.1. | 0.2163333 |
| **Random Forest** | - 5 variables.<br>- Tamaño máximo de hoja de 20.<br>- 16 divisiones máximas en un nodo.<br>- Profundidad máxima de 6.<br>- P-valor de 0.3. | 0.2465000 | - 2 variables.<br>- Tamaño máximo de hoja de 5.<br>- 20 divisiones máximas en un nodo.<br>- Profundidad máxima de 6.<br>- P-valor de 0.1. | 0.2602778 |
| **Gradient Boosting** | - Shrink de 0.05.<br>- Tamaño mínimo de hoja final de 130.<br>- 10 divisiones máximas en un nodo.<br>- Profundidad máxima de 2.<br>- 20 observaciones para dividir un nodo. | 0.2422222 | - Shrink de 0.05.<br>- Tamaño mínimo de hoja final de 150.<br>- 10 divisiones máximas en un nodo.<br>- Profundidad máxima de 2.<br>- 20 observaciones para dividir un nodo. | 0.2638333 |
| **SVM** | - Kernel = linear.<br>- C = 2. | 0.1803333 | - Kernel = linear.<br>- C = 2 | 0.1806111 |

*Tabla 11.1.2.1*

Los resultados obtenidos son los mismos que mediante las observaciones de la primera semilla.

Los resultados de las redes neuronales son significativamente mejores que los obtenidos mediante la regresión logística, tanto para las variables binarias como para las variables WOE, e incluso obtenemos mejores resultados con las variables WOE que con las binarias en los casos de la regresión logística y redes neuronales.

Y la técnica que mejores resultados ha obtenido también ha sido la de Support Vector Machine, por encima de la regresión logística, las redes neuronales y las demás técnicas basadas en árboles. En este caso también se obtuvieron mejores resultados a través de las variables binarias que mediante las variables WOE, aunque de la misma forma seguimos hablando de una diferencia de diezmilésimas.

## 11.2 SEGÚN LOS TIEMPOS DE EJECUCIÓN

A continuación se muestra para cada uno de los modelos el tiempo de ejecución en segundos. Es importante aclarar que el tiempo que se muestra es únicamente del proceso final, de la ejecución y comparación últimas de los modelos de cada técnica. Esto es debido a que para la construcción de cada modelo no se siguen los mismos pasos en cada técnica, unos tienen más pasos que otros y no son todos iguales, por eso para que fuera una comparación lo más equilibrada y en igualdad de condiciones posible, estos tiempos únicamente son del proceso final.

### 11.2.1 CON LA PRIMERA SEMILLA DE DATOS

Los tiempos que se obtuvieron son los siguientes.

| | Variables Binarias | Variables WOE |
|---|---|---|
| | Tiempo de ejecución (segundos) | Tiempo de ejecución (segundos) |
| **Regresión Logística** | 7 | 6 |
| **Redes Neuronales** | 42969 los 8 modelos (11h 56’ 9’’) | 2852 los 10 modelos (47’ 32’’) |
| **Bagging** | 94 los 9 modelos (1’ 34’’) | 94 los 9 modelos (1’ 34’’) |
| **Random Forest** | 200 los 9 modelos (3’ 20’’) | 135 los 9 modelos (2’ 15’’) |
| **Gradient Boosting** | 307 los 5 modelos (5’ 7’’) | 232 los 5 modelos (3’ 52’’) |
| **SVM** | 112 los 10 modelos (1’ 52’’) | 93 los 10 modelos (1’ 33’’) |

*Tabla 11.2.1.1*

Vemos que aquellas técnicas con mejores resultados son Bagging y SVM, dependiendo de si es utilizando las variables binarias o las variables WOE.

### 11.2.2 CON LA SEGUNDA SEMILLA DE DATOS

Los tiempos que se obtuvieron son los siguientes.

| | Variables Binarias | Variables WOE |
|---|---|---|
| | Tiempo de ejecución (segundos) | Tiempo de ejecución (segundos) |
| **Regresión Logística** | 7 | 6 |
| **Redes Neuronales** | 19970 los 4 modelos (5h 32’ 50’’) | 1258 los 5 modelos (20’ 58’’) |
| **Bagging** | 96 los 9 modelos (1’ 36’’) | 97 los 9 modelos (1’ 37’’) |
| **Random Forest** | 150 los 9 modelos (2’ 30’’) | 223 los 9 modelos (3’ 43’’) |
| **Gradient Boosting** | 305 los 5 modelos (5’ 5’’) | 224 los 5 modelos (3’ 44’’) |
| **SVM** | 113 los 10 modelos (1’ 53’’) | 85 los 10 modelos (1’ 25’’) |

*Tabla 11.2.2.1*

Al igual que con la primera semilla de datos, vemos que aquellas técnicas con mejores resultados son Bagging y SVM, dependiendo de si es utilizando las variables binarias o las variables WOE.

# 12. INTERPRETACIÓN DE COEFICIENTES

A pesar de ser consideradas las técnicas de machine learning como cajas negras, sobre todo las redes neuronales, en este apartado vamos a analizar si es posible interpretar los efectos de cada una de las variables explicativas sobre la probabilidad de impago.

Para ello vamos a utilizar las predicciones sobre nuestro mejor modelo, en este caso el SVM utilizando variables binarias, dando a cada una de las variables independientes el valor medio de la muestra, y variando únicamente los valores de las categorías de aquella variable explicativa que queremos estudiar.

De esta forma lograremos encontrar el incremento o disminución de la probabilidad de que se dé un impago ($Y = 1$, DEFAULT = 1), en función de cada una de las categorías de cada una de las variables.

A continuación se muestra una tabla con las probabilidades de impago asociadas a cada una de las categorías de cada variable junto con su interpretación.

| Variable en estudio | Categoría de referencia | Valor de predicción / Diferencia con la categoría de referencia | Interpretación |
|---|---|---|---|
| **Menos de 26 años (AGE1)** *GRP_AGE1* | **AGE 4** *GRP_AGE4* | 0.48573 / 0,01041 | La probabilidad de sufrir impago teniendo menos de 26 años es de 0.485, hay un incremento del 0.01, es decir, un 1% con respecto a teniendo entre 35 y 46 años. |
| **Entre 26 y 30 años (AGE2)** *GRP_AGE2* | | 0.48634 / 0,01102 | La probabilidad de sufrir impago teniendo entre 26 y 30 años es de 0.486, hay un incremento del 0.011, es decir, un 1.1% con respecto a teniendo entre 35 y 46 años. |
| **Entre 30 y 35 años (AGE3)** *GRP_AGE3* | | 0.51248 / 0,03716 | La probabilidad de sufrir impago teniendo entre 30 y 35 años es de 0.512, hay un incremento del 0.037, es decir, un 3.7% con respecto a teniendo entre 35 y 46 años. |
| **Entre 35 y 46 años (AGE4)** *GRP_AGE4* | | 0.47532 | La probabilidad de sufrir impago teniendo entre 35 y 46 años es de 0.475. |
| **Más de 46 años (AGE5)** *GRP_AGE5* | | 0.54383 / 0,06851 | La probabilidad de sufrir impago teniendo más de 46 años es de 0.543, hay un incremento del 0.068, es decir, un 6.8% con respecto a teniendo entre 35 y 46 años. |

| | | | |
|---|---|---|---|
| **Importe del estado de la cuenta en septiembre de 2005 menor a 800.5 $ (BILL_AMT1)** *GRP_BILL_AMT11* | **BILL_AMT5** *GRP_BILL_AMT15* | 0.60489 / 0,0471 | La probabilidad de sufrir impago teniendo un importe del estado de la cuenta en septiembre menor a 800.5 $ es de 0.604, hay un incremento del 0.047, es decir, un 4.7% con respecto a teniendo un importe mayor a 52205.5 $. |
| **Importe del estado de la cuenta en septiembre de 2005 entre 800.5 y 9157.5 $ (BILL_AMT2)** *GRP_BILL_AMT12* | | 0.57468 / 0,01689 | La probabilidad de sufrir impago teniendo un importe del estado de la cuenta en septiembre de entre 800.5 y 9157.5 $ es de 0.574, hay un incremento del 0.016, es decir, un 1.6% con respecto a teniendo un importe mayor a 52205.5 $. |
| **Importe del estado de la cuenta en septiembre de 2005 entre 9157.5 y 37046 $ (BILL_AMT3)** *GRP_BILL_AMT13* | | 0.52334 / -0,03445 | La probabilidad de sufrir impago teniendo un importe del estado de la cuenta en septiembre de entre 9157.5 y 37046 $ es de 0.523, hay una disminución del 0.034, es decir, un 3.4% con respecto a teniendo un importe mayor a 52205.5 $. |
| **Importe del estado de la cuenta en septiembre de 2005 entre 37046 y 52205.5 $ (BILL_AMT4)** *GRP_BILL_AMT14* | | 0.48893 / -0,06886 | La probabilidad de sufrir impago teniendo un importe del estado de la cuenta en septiembre de entre 37046 y 52205.5 $ es de 0.488, hay una disminución del 0.068, es decir, un 6.8% con respecto a teniendo un importe mayor a 52205.5 $. |
| **Importe del estado de la cuenta en septiembre de 2005 mayor a 52205.5 $ (BILL_AMT5)** *GRP_BILL_AMT15* | | 0.55779 | La probabilidad de sufrir impago teniendo un importe del estado de la cuenta en septiembre mayor a 52205.5 $ es de 0.557. |

| | | | |
|---|---|---|---|
| **Logaritmo en base 10 a la cantidad de crédito otorgado, menor a 4.48 $ (LIMIT_BAL1)**<br>*GRP_LG10_LIMIT_BAL1* | **LIMIT_BAL4**<br>*GRP_LG10_LIMIT_BAL4* | 0.52138<br>/<br>0,16286 | La probabilidad de sufrir impago teniendo una logarítmica cantidad de crédito otorgado menor de 4.48 $ es de 0.521, hay un incremento del 0.162 es decir, un 16.2% con respecto a teniendo una cantidad de crédito de entre 5.15 y 5.49 $. |
| **Logaritmo en base 10 a la cantidad de crédito otorgado, de entre 4.48 y 4.9 $ (LIMIT_BAL2)**<br>*GRP_LG10_LIMIT_BAL2* | | 0.45884<br>/<br>0,10032 | La probabilidad de sufrir impago teniendo una logarítmica cantidad de crédito otorgado de entre 4.48 y 4.9 $ es de 0.458, hay un incremento del 0.1 es decir, un 10% con respecto a teniendo una cantidad de crédito de entre 5.15 y 5.49 $. |
| **Logaritmo en base 10 a la cantidad de crédito otorgado, de entre 4.9 y 5.15 $ (LIMIT_BAL3)**<br>*GRP_LG10_LIMIT_BAL3* | | 0.45550<br>/<br>0,09698 | La probabilidad de sufrir impago teniendo una logarítmica cantidad de crédito otorgado de entre 4.9 y 5.15 $ es de 0.455, hay un incremento del 0.096 es decir, un 9.6% con respecto a teniendo una cantidad de crédito de entre 5.15 y 5.49 $. |
| **Logaritmo en base 10 a la cantidad de crédito otorgado de entre 5.15 y 5.49 $ (LIMIT_BAL4)**<br>*GRP_LG10_LIMIT_BAL4* | | 0.35852 | La probabilidad de sufrir impago teniendo una logarítmica cantidad de crédito otorgado de entre 5.15 y 5.49 $ es de 0.358. |
| **Logaritmo en base 10 a la cantidad de crédito otorgado, mayor a 5.49 $ (LIMIT_BAL5)**<br>*GRP_LG10_LIMIT_BAL5* | | 0.35571<br>/<br>-0,00281 | La probabilidad de sufrir impago teniendo una logarítmica cantidad de crédito otorgado mayor a 5.49 $ es de 0.355, hay una disminución del 0.002 es decir, un 0.2% con respecto a teniendo una cantidad de crédito de entre 5.15 y 5.49 $. |

| | | | |
|---|---|---|---|
| **Importe del pago anterior en septiembre de 2005 menor a 316 $ (PAY_AMT1)**<br>*GRP_PAY_AMT11* | **PAY_AMT4**<br>*GRP_PAY_AMT14* | 0.58532<br>/<br>0,03763 | La probabilidad de sufrir impago teniendo un importe del pago anterior en septiembre menor a 316 $ es de 0.585, hay un incremento del 0.037, es decir, un 3.7% con respecto a teniendo un importe de entre 4309.5 y 10300 $. |
| **Importe del pago anterior en septiembre de 2005 entre 316 y 2000 $ (PAY_AMT2)**<br>*GRP_PAY_AMT12* | | 0.55750<br>/<br>0,00981 | La probabilidad de sufrir impago teniendo un importe del pago anterior en septiembre de entre 316 y 2000 $ es de 0.557, hay un incremento del 0.009, es decir, un 0.9% con respecto a teniendo un importe de entre 4309.5 y 10300 $. |
| **Importe del pago anterior en septiembre de 2005 entre 2000 y 4309.5 $ (PAY_AMT3)**<br>*GRP_PAY_AMT13* | | 0.57330<br>/<br>0,02561 | La probabilidad de sufrir impago teniendo un importe del pago anterior en septiembre de entre 2000 y 4309.5 $ es de 0.573, hay un incremento del 0.025, es decir, un 2.5% con respecto a teniendo un importe de entre 4309.5 y 10300 $. |
| **Importe del pago anterior en septiembre de 2005 entre 4309.5 y 10300 $ (PAY_AMT4)**<br>*GRP_PAY_AMT14* | | 0.54769 | La probabilidad de sufrir impago teniendo un importe del pago anterior en septiembre de entre 4309.5 y 10300 $ es de 0.547. |
| **Importe del pago anterior en septiembre de 2005 mayor a 10300 $ (GRP_PAY_AMT5)**<br>*GRP_PAY_AMT15* | | 0.45593<br>/<br>-0,09176 | La probabilidad de sufrir impago teniendo un importe del pago anterior en septiembre mayor a 103000 es de 0.455, hay una disminución del 0.091, es decir, un 9.1% con respecto a teniendo un importe de entre 4309.5 y 10300 $. |

| | | | |
|---|---|---|---|
| **Nivel de educación desconocida, escolar u otro (EDUCATION1)**<br>*GRP_EDUCATION1* | **EDUCATION3**<br>*GRP_EDUCATION3* | 0.52869<br>/<br>0,0164 | La probabilidad de sufrir impago teniendo una educación desconocida, escolar u otro es de 0.528, hay un incremento del 0.016, es decir, un 1.6% con respecto a teniendo una educación universitaria. |
| **Nivel de educación secundaria (EDUCATION2)**<br>*GRP_EDUCATION2* | | 0.52478<br>/<br>0,01249 | La probabilidad de sufrir impago teniendo una educación desconocida, escolar u otro es de 0.524, hay un incremento del 0.012, es decir, un 1.2% con respecto a teniendo una educación universitaria. |
| **Nivel de educación universitaria (EDUCATION3)**<br>*GRP_EDUCATION3* | | 0.51229 | La probabilidad de sufrir impago teniendo una educación universitaria es de 0.512. |

| | | | |
|---|---|---|---|
| **Estado de reembolso en septiembre de 2005 como "No consumido", "Pagado debidamente" o "Crédito revolvente" (PAY_1)**<br>*GRP_PAY_11* | **PAY_1**<br>*GRP_PAY_11* | 0.15329 | La probabilidad de sufrir impago un estado de reembolso en septiembre como "No consumido", "Pagado debidamente" o "Crédito revolvente" es de 0.153. |
| **Estado de reembolso en septiembre de 2005 como "Pago atrasado por un mes" (PAY_2)**<br>*GRP_PAY_12* | | 0.27322<br>/<br>0,11993 | La probabilidad de sufrir impago un estado de reembolso en septiembre como "Pago atrasado por un mes" es de 0.273, hay un incremento de 0.119, es decir, un 11.9% con respecto a "No consumido", "Pagado debidamente" o "Crédito revolvente". |
| **Estado de reembolso en septiembre de 2005 como "Pago atrasado por más de un mes" (PAY_3)**<br>*GRP_PAY_13* | | 0.76718<br>/<br>0,61389 | La probabilidad de sufrir impago un estado de reembolso en septiembre como "Pago atrasado por más de un mes" es de 0.767, hay un incremento de 0.613, es decir, un 61.3% con respecto a "No consumido", "Pagado debidamente" o "Crédito revolvente". |

| | | | |
|---|---|---|---|
| **Estado marital "Casado" o "Divorciado" (MARRIAGE1)**<br>*GRP_MARRIAGE1* | | 0.50325 | La probabilidad de sufrir impago estando casado o divorciado es de 0.503. |
| **Estado marital "Soltero" u "Otro" (MARRIAGE2)**<br>*GRP_MARRIAGE2* | | 0.53801 | La probabilidad de sufrir impago estando soltero u otro divorciado es de 0.538. |

| | | | |
|---|---|---|---|
| **Mujer (SEX1)**<br>*GRP_SEX1* | | 0.51795 | La probabilidad de sufrir impago siendo mujer es de 0.517. |
| **Hombre (SEX2)**<br>*GRP_SEX2* | | 0.52336 | La probabilidad de sufrir impago siendo mujer es de 0.523. |

A través de estas tablas descubrimos los aspectos más relevantes con relación a resultar DEFAULT, frente a los valores medios del resto de las variables. Estos aspectos han sido los siguientes:

- Disponer de un importe alto en el estado de la cuenta el mes antes, de más de 9157.5 $, reduce la probabilidad de resultar DEFAULT. Es decir, tener una cuantiosa cantidad de dinero en la cuenta, ya que eso suele indicar que la persona tiene un alto nivel adquisitivo.

- Otorgar un crédito de alto nivel reduce la probabilidad de resultar DEFAULT, su transformación en logaritmo en base 10 es de 5.49, con lo que sería de más de un cuarto de millón. Esos créditos de tan alto valor se les concederán únicamente a los mejores clientes que serán los mejores pagadores.

- Un importe del pago anterior cuantioso, de más de 10300 $ también reduce la probabilidad de resultar DEFAULT. Al tener pagos anteriores considerables, es más probable que sean personas con un alto nivel adquisitivo las que realicen ese tipo de pagos.

- Toda educación inferior a la universitaria aumenta la probabilidad de impago, eso es normal ya que la educación siempre suele ser un gran condicionante.

- Si el estado del reembolso el mes anterior es que se ha atrasado en algún pago, eso siempre aumenta la probabilidad de DEFAULT, lo que es frecuente ya que si lo hizo una vez, puede volver a hacerlo.

- Los solteros tienen más probabilidad de resultar DEFAULT que los casados o divorciados. Esto es común por que los solteros habitualmente por su edad se encuentran en una situación económica más precaria que las personas casadas o divorciadas.

- Los hombres tienen más probabilidad que las mujeres de resultar DEFAULT, un 0.5% más probable.

Los resultados que se han obtenido para cada variable han sido céteris páribus, manteniendo todas las demás variables constantes en sus valores medios.

Es importante remarcar que estos resultados no dejan de ser un primer intento de encontrar una interpretación a los modelos de machine learning en relación al efecto que cada una de las variables puede efectuar sobre la probabilidad de DEFAULT. Luego estas interpretaciones deben tomarse con cautela, ya que no hemos podido realizar un análisis de la significatividad de estos resultados. Queda pendiente profundizar en esta línea para encontrar, por ejemplo, la mejor manera de contrastar si dichos efectos son estadísticamente significativos o no.

# 13. CONCLUSIÓN

Una vez finalizado nuestro estudio de investigación podemos concluir los siguientes de aspectos.

En primer lugar, los resultados obtenidos en este estudio acorde a los resultados de estudios anteriores, han demostrado que a la hora de predecir una variable objetivo binaria, nuevas técnicas de machine learning, frente a la tradicional técnica de regresión logística obtienen mejores resultados en lo que a capacidad predictora se refiere.

En segundo lugar, de las diferentes técnicas de machine learning que hemos analizado, la mejor ha sido con diferencia, Support Vector Machine. A continuación se muestra una tabla en la que poder observar la sensibilidad y especificidad obtenidas mediante este modelo, por medio de las variables binarias.

| Frecuencia<br>Porcentaje<br>Pct fila<br>Pct col | Tabla de _I_ por DEFAULT | | |
|---|---|---|---|
| | DEFAULT | | |
| _I_ | 0 | 1 | Total |
| 0 | 498<br>74.55<br>83.00<br>94.86 | 102<br>15.27<br>17.00<br>71.33 | 600<br>89.82 |
| 1 | 27<br>4.04<br>39.71<br>5.14 | 41<br>6.14<br>60.29<br>28.67 | 68<br>10.18 |
| Total | 525<br>78.59 | 143<br>21.41 | 668<br>100.00 |

En esta tabla se puede observar que el porcentaje de verdaderos positivos ha sido del 29% y el porcentaje de verdaderos negativos ha sido del 95%.

También nos encontramos con un 71% de falsos negativos y un 5% de falsos positivos, lo cual puede sonar alarmante, pero no hay que olvidar que éste ha sido el modelo con menor ECM de todos los que hemos analizado.

En tercer lugar, nuestros resultados muestran que, a costa de una reducida disminución en la capacidad de predicción de los modelos, los tiempos de ahorro en ejecución mediante las variables WOE son sustancialmente significativas frente a las habituales variables binarias.

En resumen, con lo analizado en este y anteriores estudios, podríamos concluir que parece una apuesta segura y que no sería desacertado, sugerirles a las entidades financieras que evolucionasen, dentro de las técnicas de análisis para la calificación de riesgo en la concesión de créditos, a técnicas de machine learning más modernas. Técnicas como el Support Vector Machine, con la que, junto con el uso de variables tipo WOE, lograrían tanto un ahorro en tiempo de ejecución como una mejor predicción.

# BIBLIOGRAFÍA

[1] https://www.wellsfargo.com/es/financial-education/credit-management/calculate-credit-score/
*WELLS FARGO – "Cómo se calcula su puntuación de crédito" (2019)*

[2] https://www.uv.es/revispsi/articulos3.98/pitarque.pdf
*Redes neurales vs modelos estadísticos: Simulaciones sobre tareas de predicción y clasificación.* Alfonso Pitarque, Juan Francisco Roy y Juan Carlos Ruiz. Universitat de València. Psicológica (1998).

[3] https://www.sas.com/es_es/insights/analytics/machine-learning.html
*Aprendizaje automático. Qué es y por qué es importante. SAS (2019)*

[4] https://www.bravent.net/las-claves-del-machine-learning-y-las-redes-neuronales
*LAS CLAVES DEL MACHINE LEARNING Y LAS REDES NEURONALES. Bravent IT consulting company. (18 marzo, 2019)*

[5] https://www.paraisodigital.org/internet/aprendizaje-basado-en-arboles-de-decision.html
*Aprendizaje basado en árboles de decisión. Internet y Tecnologías de la Información (2018)*

[6] http://svm.michalhaltuf.cz/wp-content/uploads/MichalHaltufMastersThesis2014.pdf
*Support Vector Machines for Credit Scoring. Master's thesis by Michal Haltuf.* University of Economics in Prague (2014)

[7] https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset
Default of Credit Card Clients Dataset. Default Payments of Credit Card Clients in Taiwan from 2005. (Actualizado en 2016)

https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#
*Default of credit card clients Data Set. Machine Learning Repository UCI (2009)*

[8] https://support.sas.com/documentation/onlinedoc/miner/emhp123/emhpref.pdf
*SAS® Enterprise Miner™ High-Performance Data Mining Node Reference for SAS® 9.4 (2013)*

[9] https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html
*WEIGHT OF EVIDENCE (WOE) AND INFORMATION VALUE EXPLAINED. Deepanshu Bhalla. Listen Data Make Your Data Tell A Story (2015)*

[10]
https://documentation.sas.com/?docsetId=emref&docsetTarget=p1uhmtoprigyvkn147i1tw9e2ax0.htm&docsetVersion=14.3&locale=en#n1gtgy816n39dnn15btkzoih67hk
*SAS® Enterprise Miner™ 14.3: Reference Help. HP Forest Node (Actualizado en agosto de 2017)*

Siddiqi, N. (2006): Credit Risk Scorecards. Devoloping and implementing Intelligent Credit Scoring. J Wiley & Sons

Anderson, R( 2007) *The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation* . Oxford University Press

Mays,E and Niall Lynas (2011) Credit Scoring for Risk Managers: The Handbook for Lenders.Createspace (ISBN13: 9781450578967)

Trueck, S, & Rachev, Svetlozar (2009): Rating Based Modeling of Credit Risk. Theory and Application of Migration Matrices. Elsevier

# ANEXO

## Gráficos de los modelos de predicción mediante variables WOE (sólo con la primera semilla)

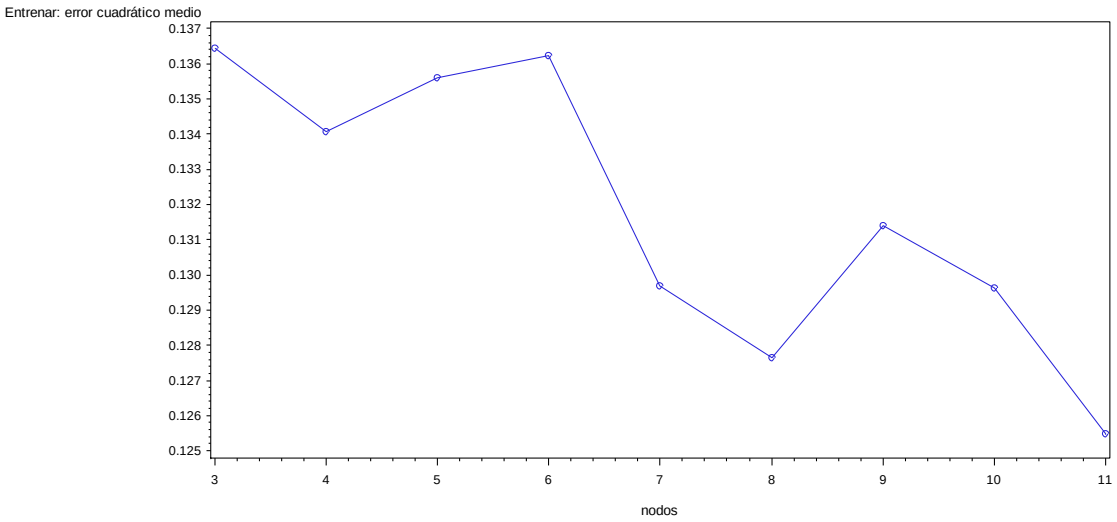
## Redes Neuronales

- Número de nodos

Con la macro ***%repito***:

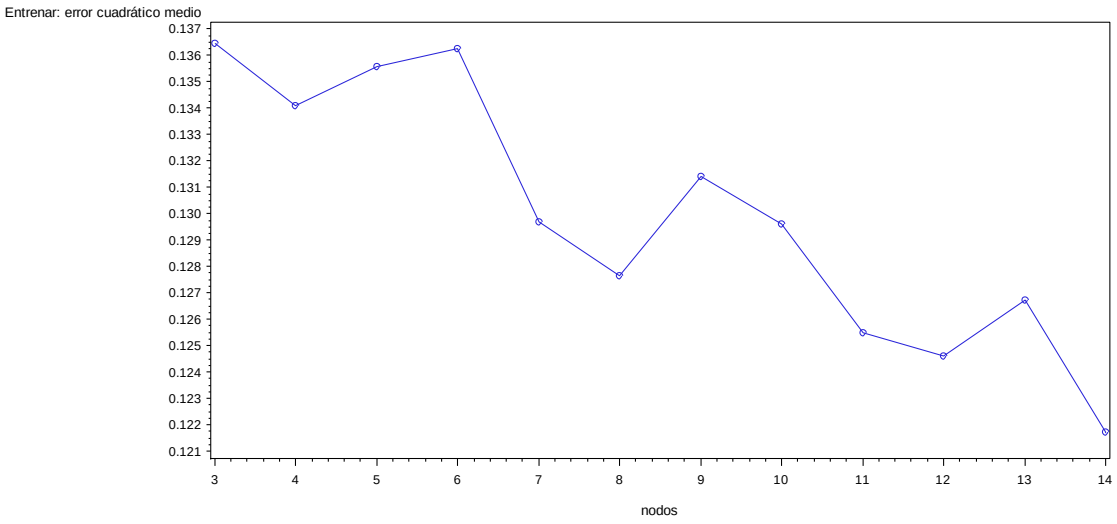
**Entre 3 y 11 nodos.**

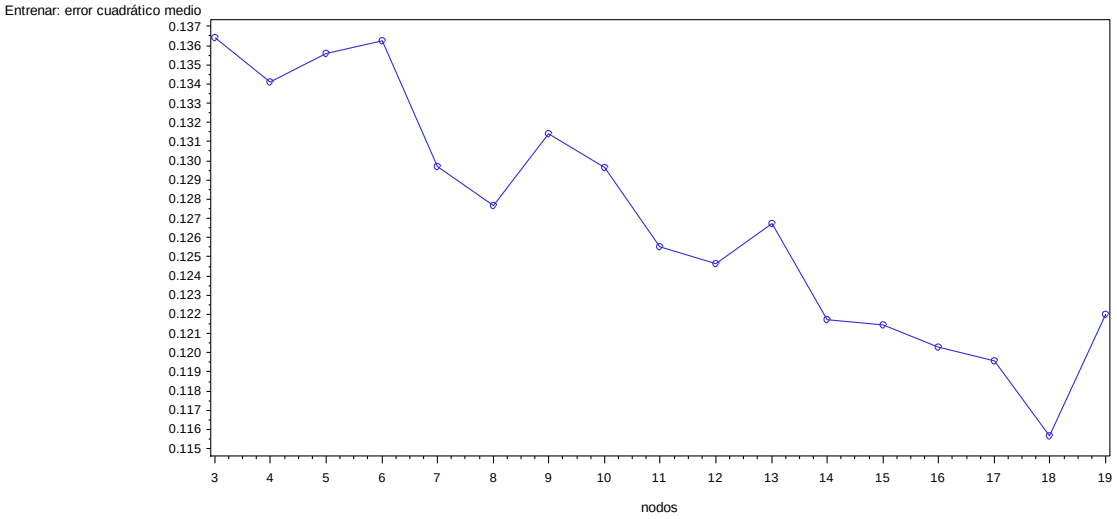
El mejor es el 11.




**Entre 3 y 14 nodos.**

El mejor es el nodo 14.

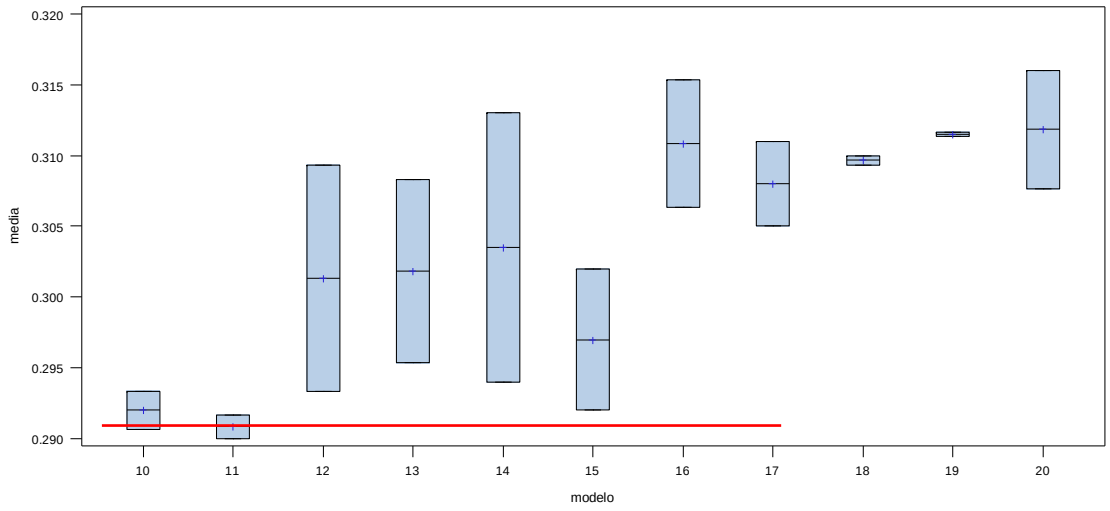




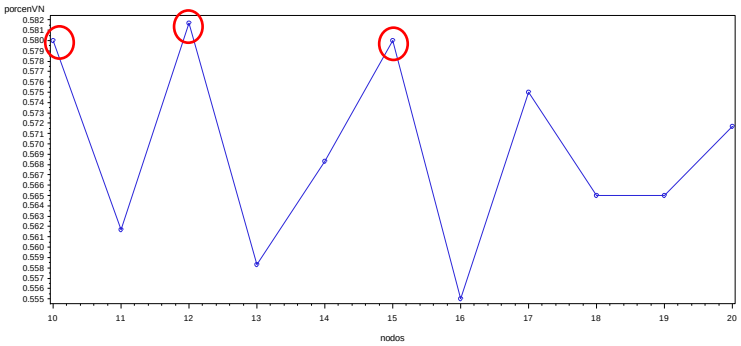

**Entre 3 y 19 nodos.**

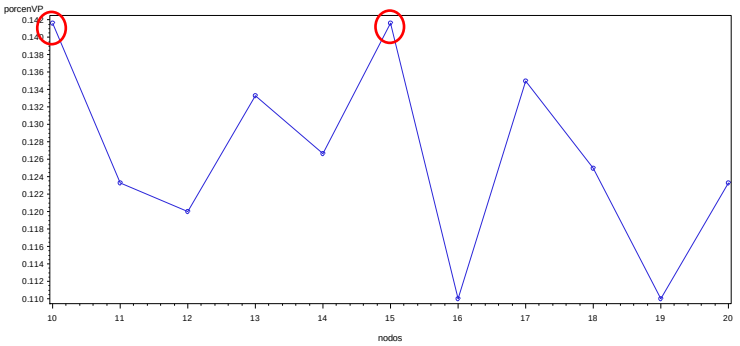
El mejor es el nodo 18.

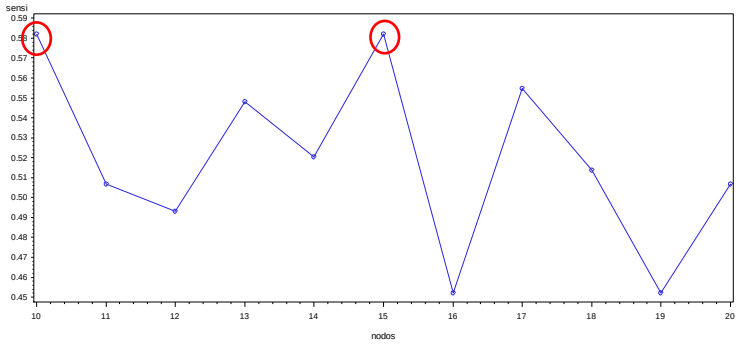
Con la macro ***%nodosvalcurza***:

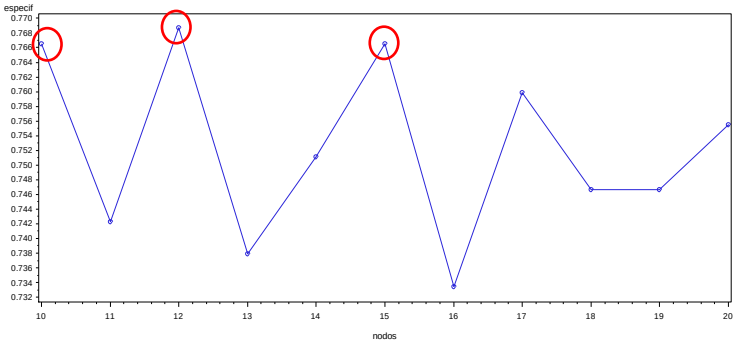


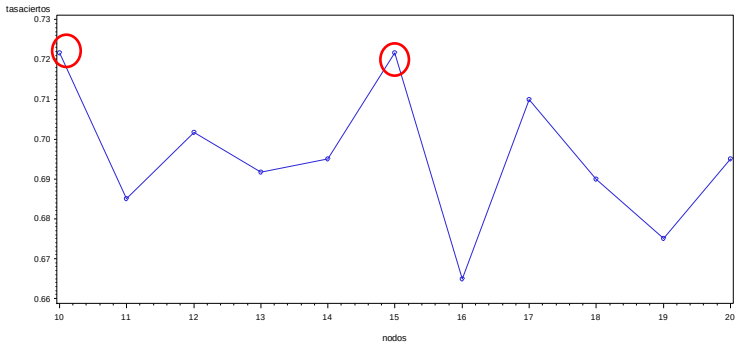
Con la macro ***%numeronodos***:

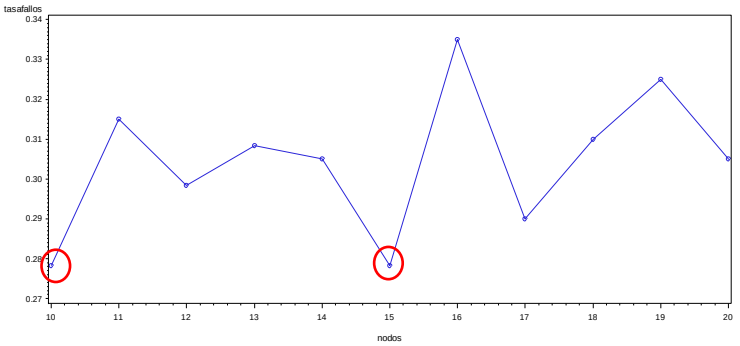


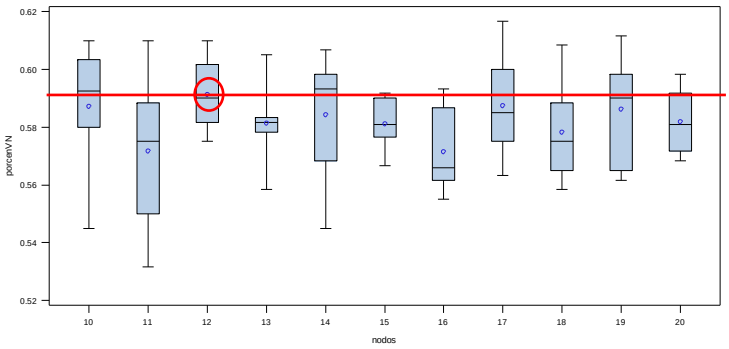



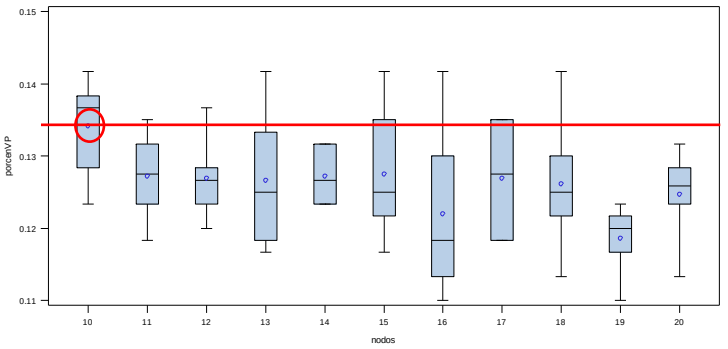



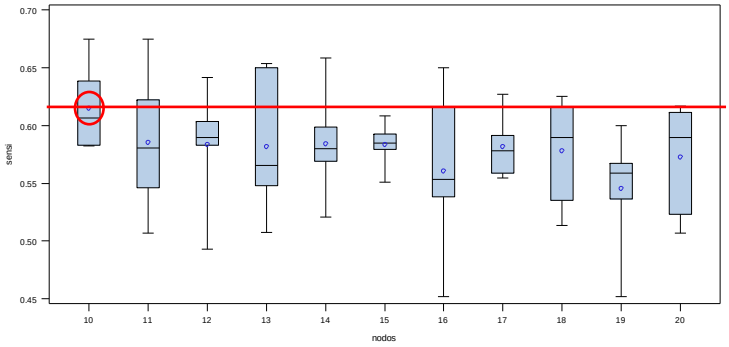



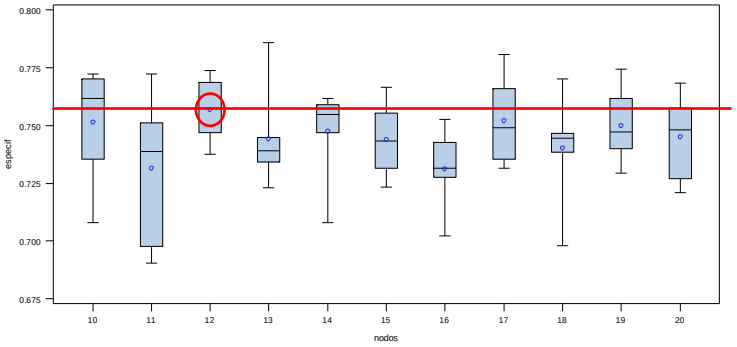



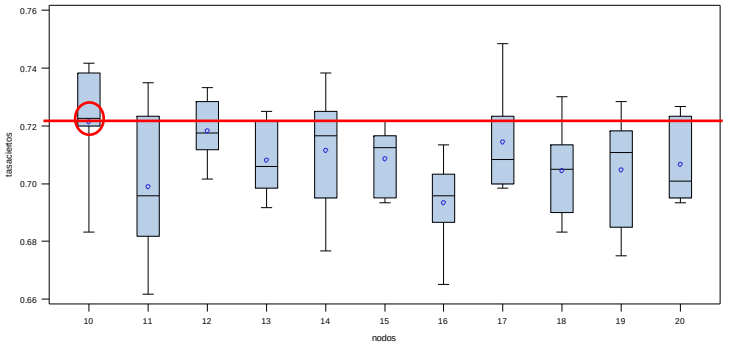
Con la macro ***%variar***:

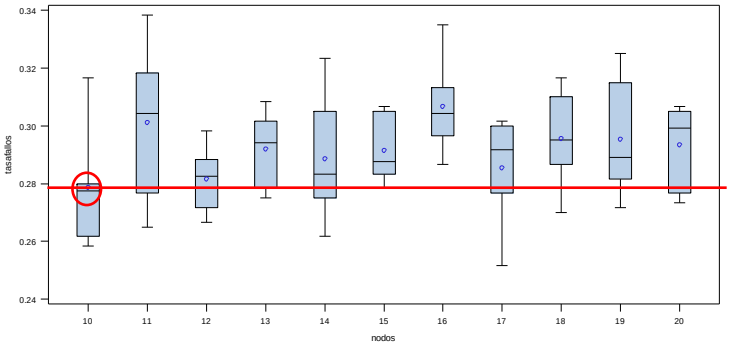


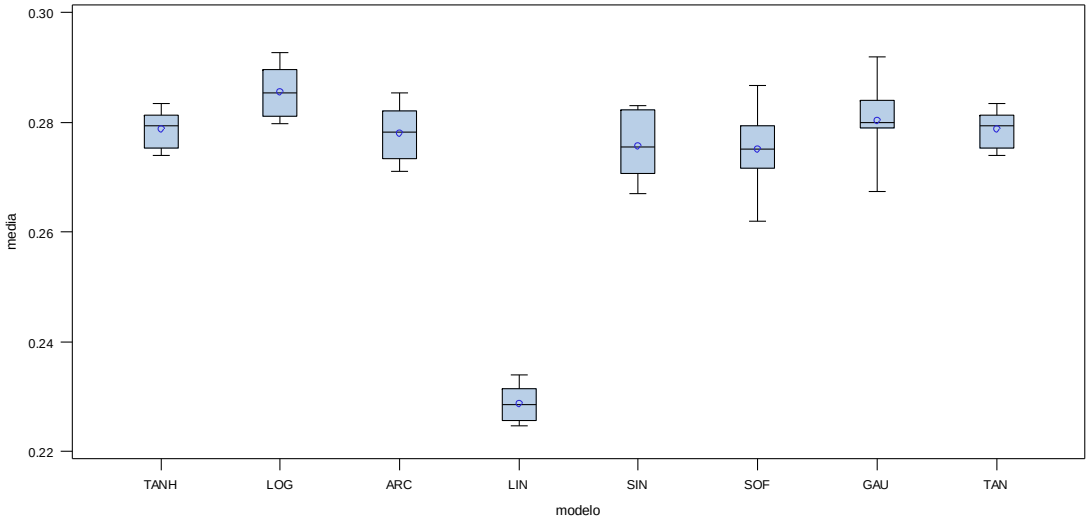



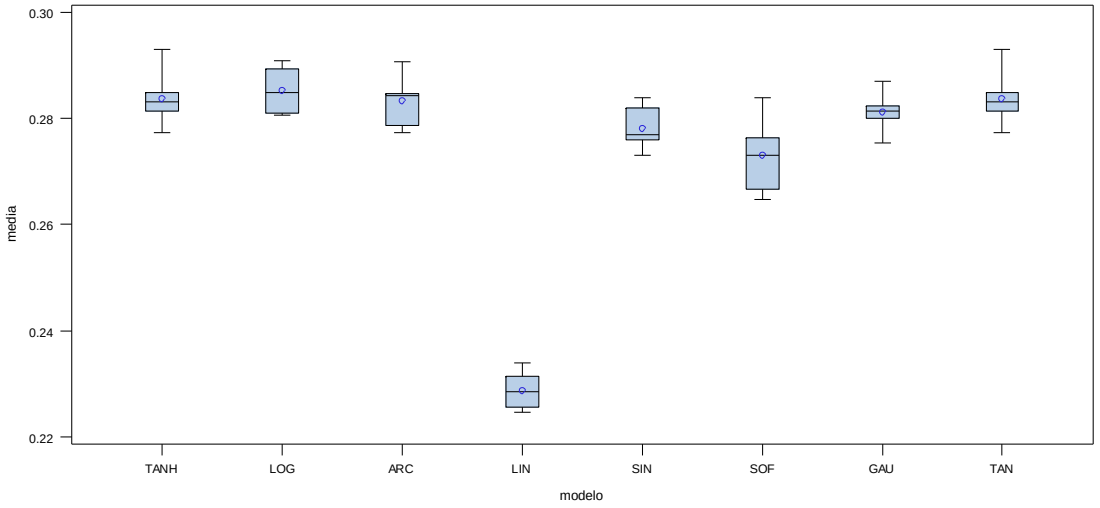
- Función de activación

Con la macro ***%activalcruza***:

**Con 10 nodos ocultos.**



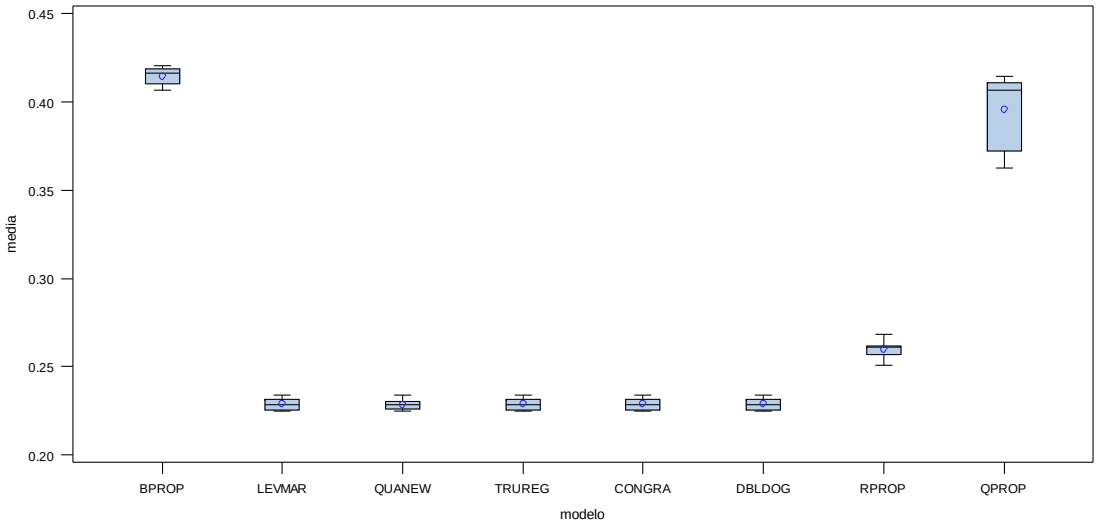

**Con 12 nodos ocultos.**



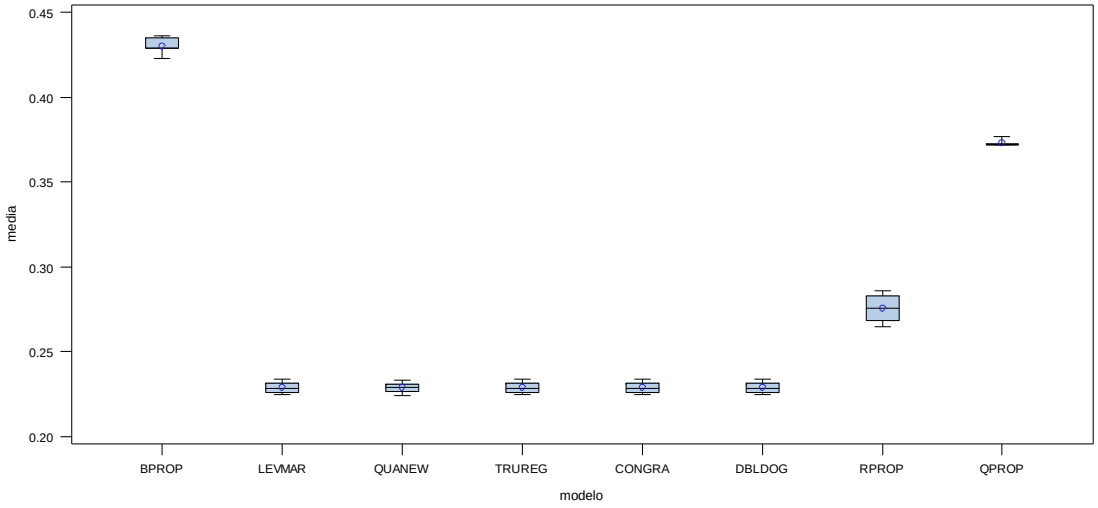
- Algoritmo de optimización

Con la macro ***%algovalcruza***:

**Con 10 nodos ocultos.**



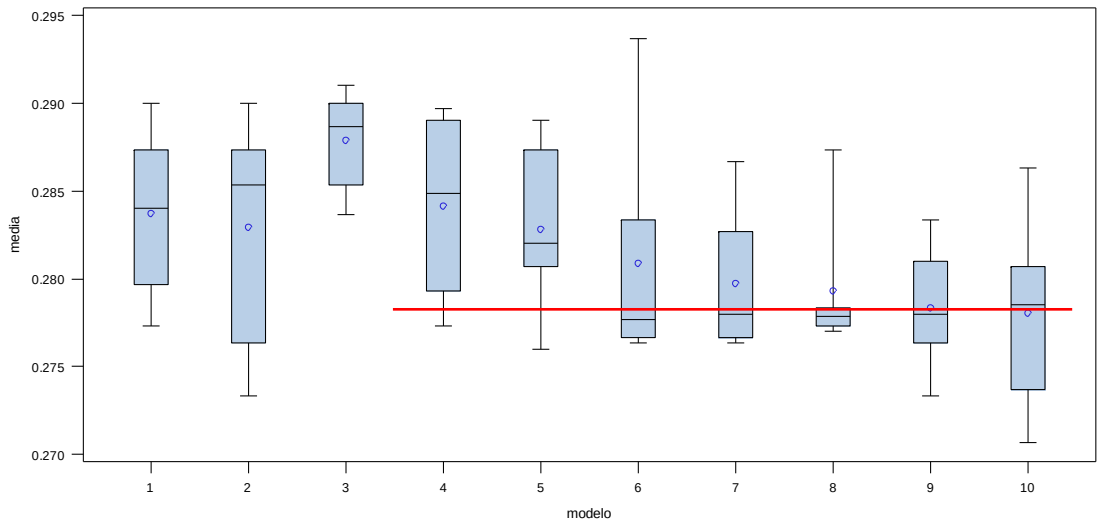

**Con 12 nodos ocultos.**



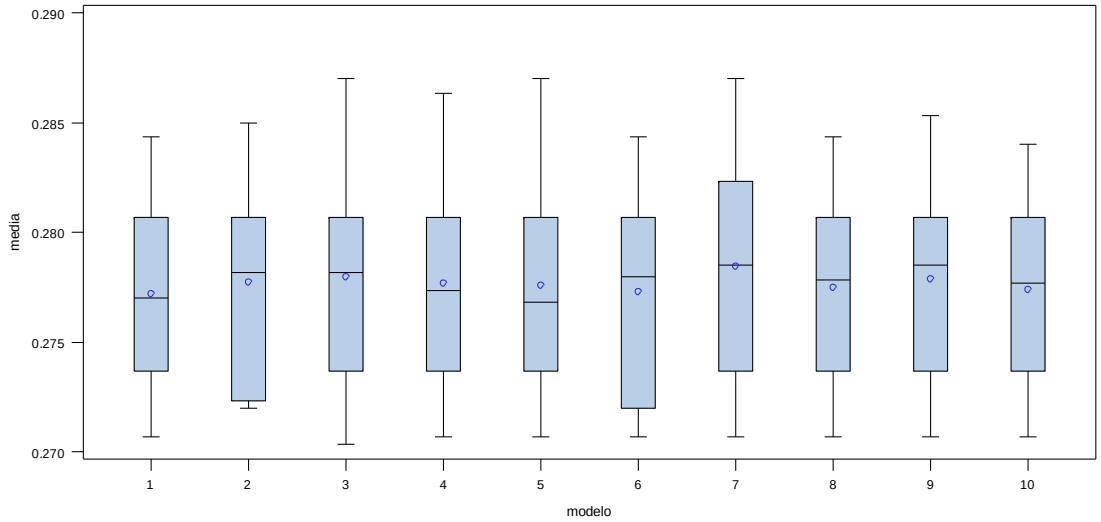
## Bagging

Con la macro ***%cruzadarandomforestbin***:

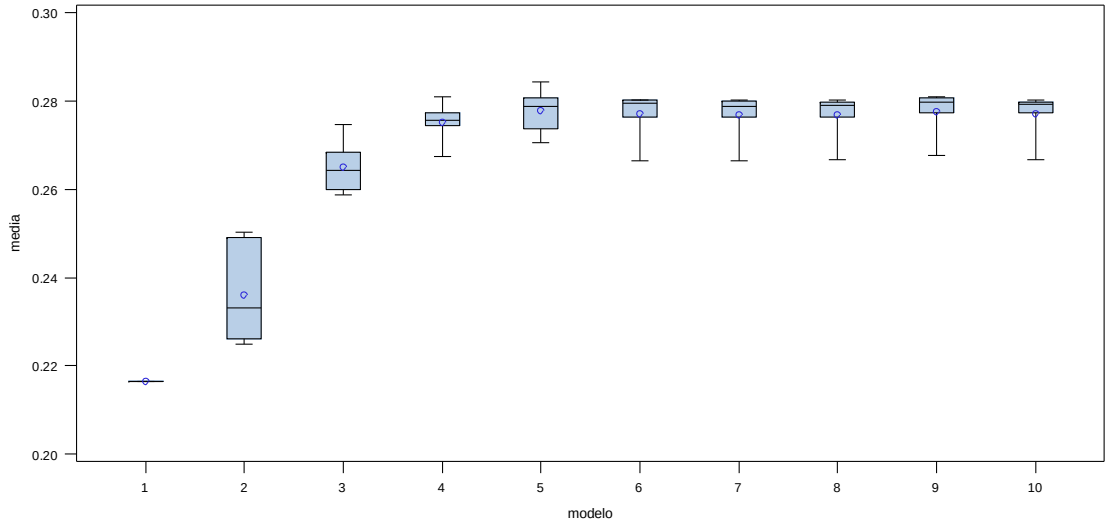
- Comprobando el tamaño mínimo de hoja final.




- Comprobando las divisiones máximas en un nodo.



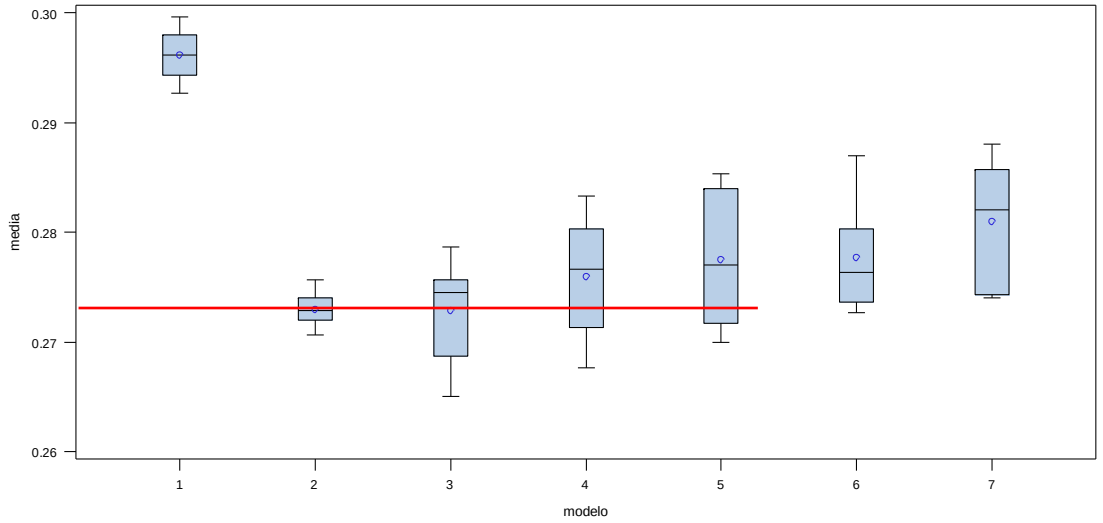
- Comprobando la profundidad máxima.



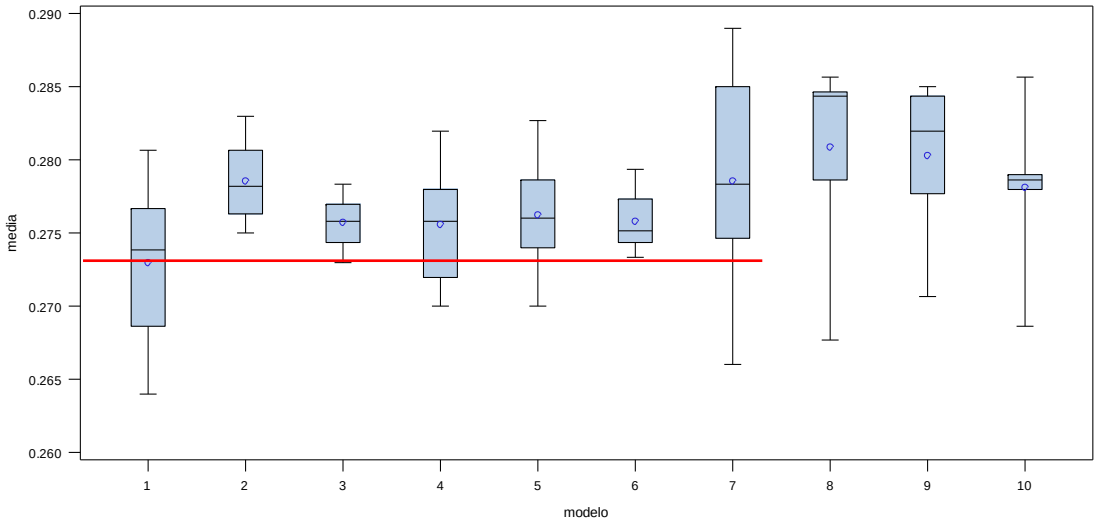

## Random Forest

Con la macro ***%cruzadarandomforestbin***:

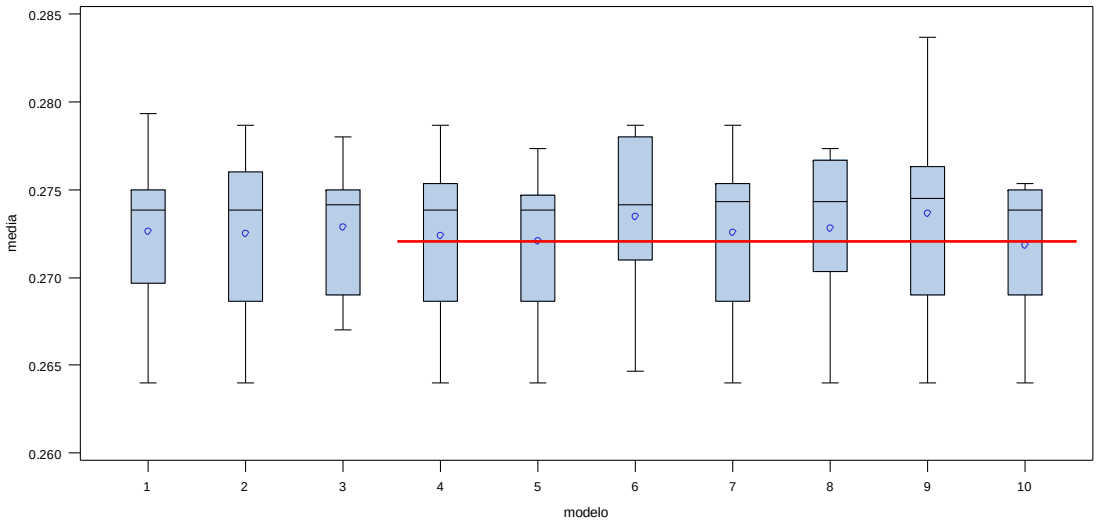
- Comprobando el número de variables.



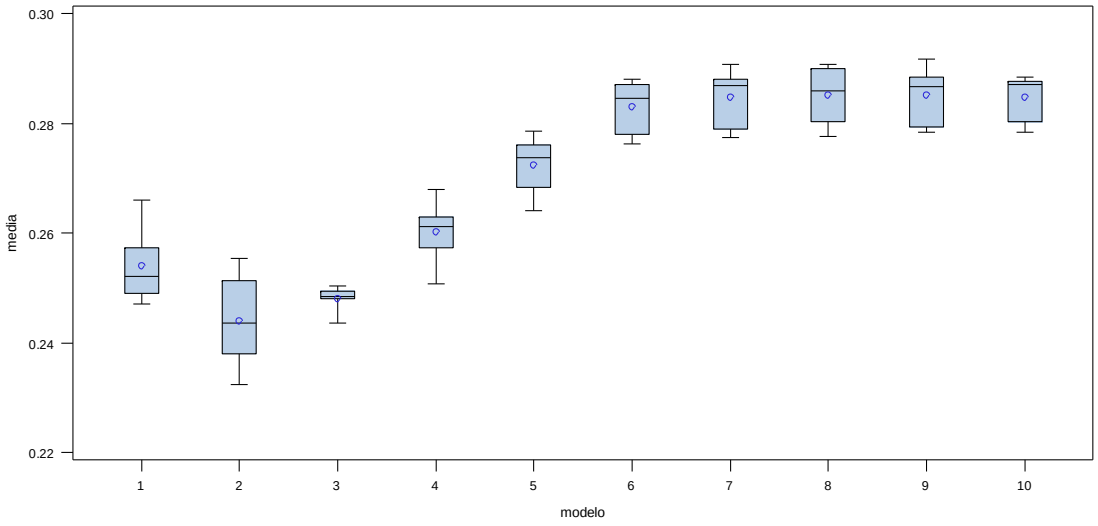
- Comprobando el tamaño mínimo de hoja final.




- Comprobando las divisiones máximas en un nodo.



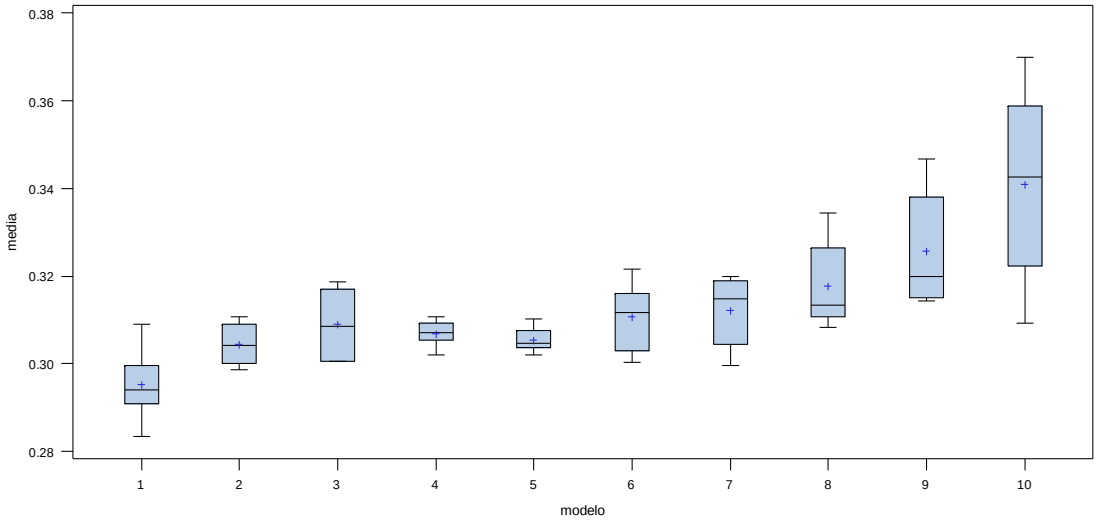
- Comprobando la profundidad máxima.



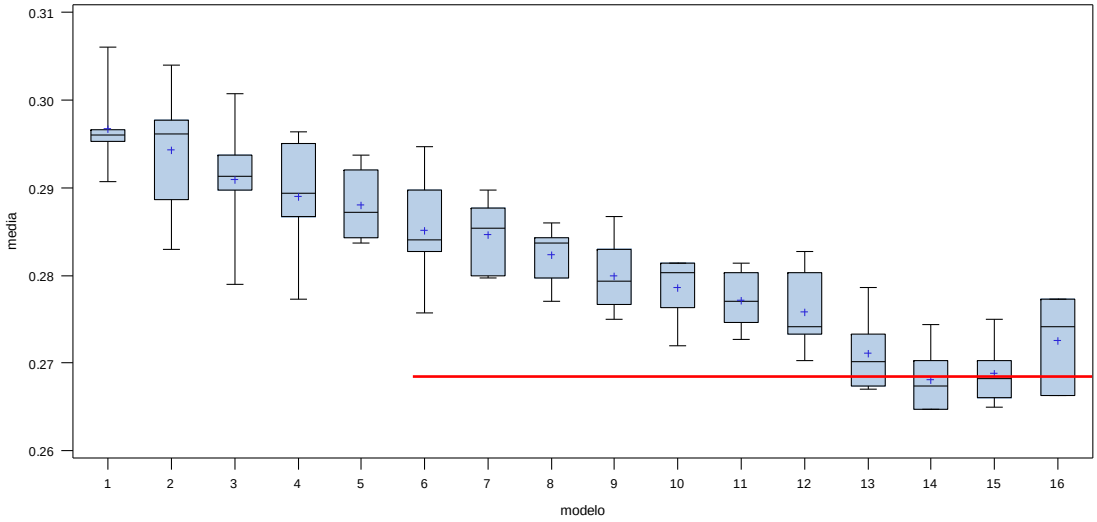

## Gradient Boosting

Con la macro ***%cruzadatreeboostbin***:

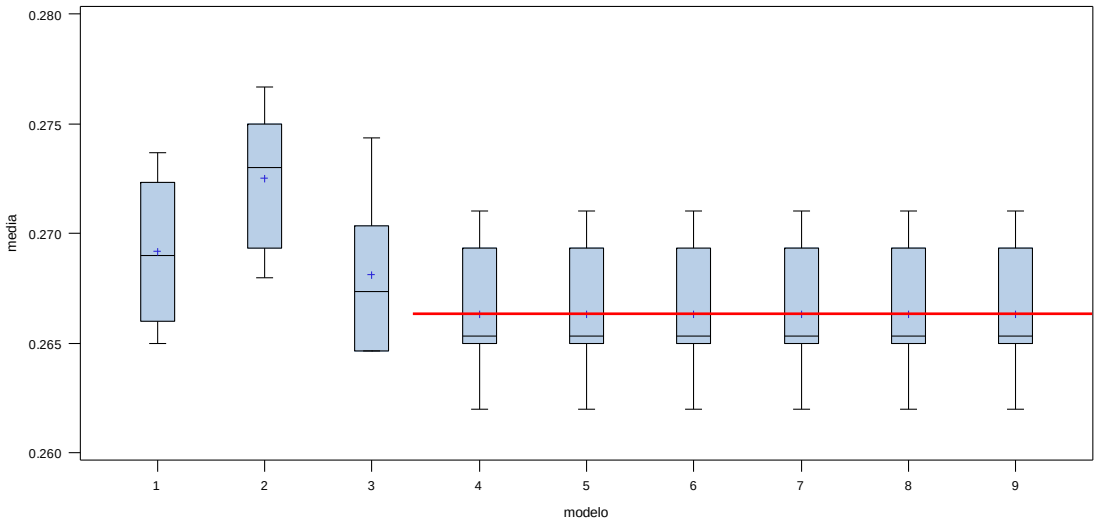
- Comprobando el shrink.



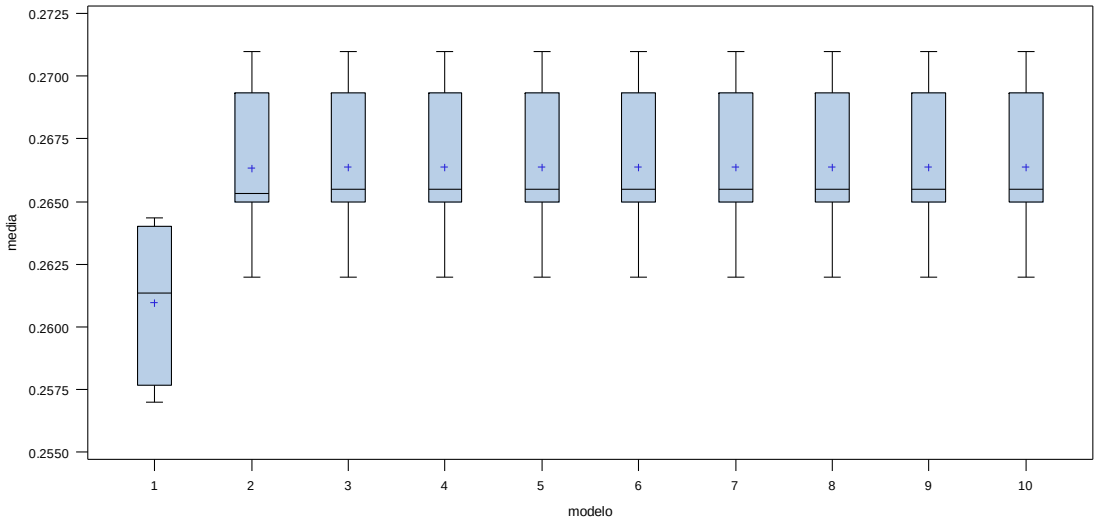
- Comprobando el tamaño mínimo de hoja final.




- Comprobando las divisiones máximas en un nodo.



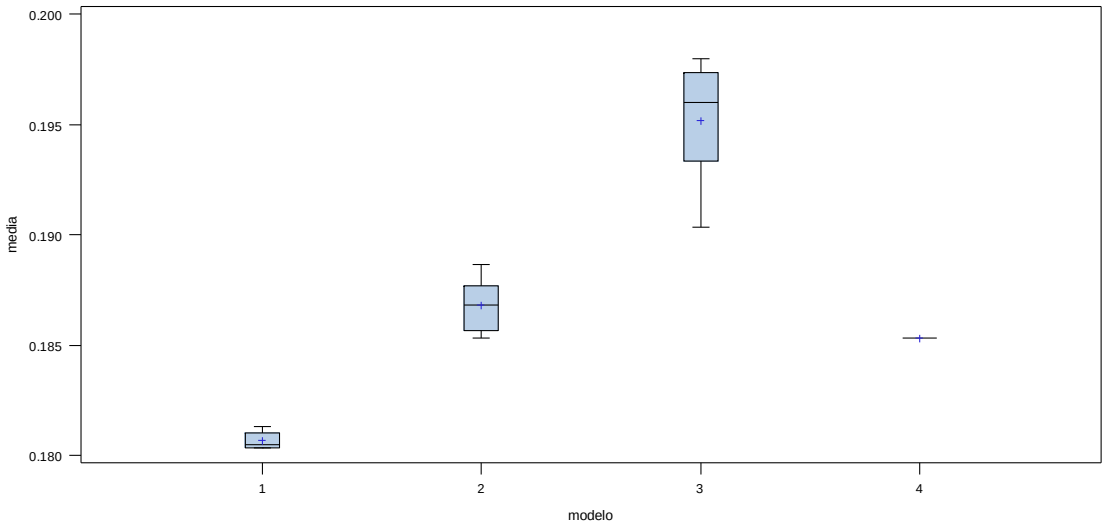
- Comprobando la profundidad máxima.




## SVM

Con la macro ***%cruzadaSVMbin***:

- Comprobando el kernel.



Anexo digital en el que se encuentra el material con el que se ha realizado el total de la investigación

https://bit.ly/2kb3eEW