FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2020/2021

Trabajo de Fin de Máster

TÍTULO: Factores determinantes en la generación de accidentes de tráfico en la ciudad de Barcelona

Alumno: María Otero Alonso

Tutor: Aída Calviño Martínez

Julio de 2021

UNIVERSIDAD COMPLUTENSE
MADRID

Índice

Índice de figuras y tablas

Índice de figuras

Índice de tablas

1. Introducción y objetivos

1.1. Introducción

Los accidentes de tráfico son una de las grandes lacras de la sociedad moderna. Según datos oficiales de la Dirección General de Tráfico (DGT), en el año 2020, 870 personas perdieron la vida en las carreteras españolas mientras que la cifra asciende a 3.463 si hablamos de hospitalizados.

Esta cifra, a pesar de ser traumática, representa el mejor dato registrado de la historia, debido principalmente a las restricciones de movilidad derivadas de la crisis de la COVID-19 (DGT, 2021). Esto hace que no se pueda bajar la guardia ya que, con toda probabilidad, la finalización de cualquier restricción y la vuelta a la normalidad dispararán esas cifras de nuevo en 2021.

El coste humanitario es altísimo y no se puede cuantificar el sufrimiento físico y/o mental de implicados y familiares. Sin embargo, el problema no reside únicamente ahí. Además de las secuelas de quienes los han sufrido, es necesario cuantificar los costes económicos implícitos asociados a los accidentes y que se conforman de cuantías:

- Administrativas, que son aquellas incurridas por policía o juzgados.
- De servicios, como son los importes derivados de los servicios médicos.
- De materiales, como los vehículos o reparación de carreteras.
- De productividad, debido a bajas o incapacitación laboral.
- Otros.

Estos costes se estiman oscilan entre 13.000 y 17.000 millones de euros cada año solo para el territorio español (Calabresi, 1970).

Si bien desde los organismos públicos se han hecho grandes esfuerzos poniendo en marcha campañas de sensibilización o endureciendo las leyes contra las infracciones de tráfico, estas medidas, aunque han reducido enormemente el número de fallecidos, desde 3.993 en el año 2003 (RTVE.es/Servimedia, 2017) hasta la actual cifra de 870 personas, no han sido suficientes para eliminar por completo esta catástrofe que son los accidentes.

Las nuevas tecnologías están incorporando avances; los vehículos están empezando a estar equipados con cámaras y sensores que actúan como visor humano y que, combinado con inteligencia artificial, traerán la llegada del coche autónomo y, con ella, la desaparición de la inmensa mayoría de los accidentes de tráfico. Esto ocurrirá porque el 90% de los accidentes son producidos por el factor humano, debido a innumerables características como la distracción, el cansancio o la falta de habilidad que hacen que, ante supuestos imprevistos, no se reaccione con el debido tiempo, y, por ende, no se pueda evitar la colisión (Jurgen, 2013).

Si bien es cierto que esos cambios traerán una reducción drástica de muertes en carreteras, esta metamorfosis no se hará efectiva en el corto plazo. Se necesita una

transición que, con alta probabilidad, tardará más de una década en hacerse realidad. La sociedad no puede permitirse seguir con esta deriva de accidentes durante este periodo, por lo que es necesario, tanto desde organismos públicos como desde la propia sociedad, implantar medidas que logren evitar, en la medida de lo posible, la producción de estos sucesos fatales.

La Comisión Europea ha lanzado un nuevo paquete de medidas de seguridad vial (2021-2030) donde incluye por una parte la necesidad de incorporar estos Sistemas Avanzados de Asistencia a la Conducción (ADAS, por sus siglas en inglés) en vehículos, así como una mejora de la evaluación y el mantenimiento de las carreteras. Bruselas estima que únicamente haciendo hincapié en la mejora del estado de las vías podrían evitarse unas 3.200 víctimas mortales anuales en toda la Unión Europea (Confederación Nacional de Autoescuelas [CNAE], 2018).

La pregunta que surge ahora es si existe un marco de actuación a corto plazo desde organismos, entidades y universidades que ayuden a paliar este desastre.

1.2. Motivación y objetivos

1.2.1. Motivación del trabajo

Ante los problemas previamente descritos y viendo la cantidad de datos existentes, surge la idea de realizar un estudio que ayude a identificar factores que influyan tanto positiva como negativamente en la generación de accidentes, con la finalidad de aportar valor que pueda ayudar a mitigar esas colisiones.

Resulta interesante realizar este análisis para conocer qué es aquello que genera situaciones conflictivas y qué acciones podrían llevarse a cabo. Podría decirse que las ciudades están “vivas”, es decir, se encuentran en constante cambio, por lo que, ante análisis externos, podrían tomar acción para tratar de subsanar este peligro.

En caso de inacción por parte de las ciudades en los puntos ya existentes, este estudio podría considerarse a la hora de plantear posibles proyectos urbanísticos o modificaciones viales dentro de la misma ciudad.

1.2.2. Objetivos del trabajo

El objetivo general que se plantea, dada la problemática a la que se ha aludido en la introducción, es determinar si una determinada área es más o menos peligrosa en función de una serie de características viarias y de densidad de peatones.

Este trabajo intenta contribuir a ese objetivo genérico proponiendo el uso de distintas técnicas de predicción que nos permitan, además, cumplir los siguientes objetivos específicos:

- Detectar los factores de riesgo que favorezcan la generación de accidentes viarios y sean susceptibles de ser modificados para reducir su peligrosidad.
- Obtener información sobre qué variables de manera conjunta son las que más relación tienen con las colisiones.

- Corroborar la premisa previa que indica que la alta afluencia de peatón se traduce en una mayor peligrosidad.

Se espera que la puesta en marcha de este trabajo permita alcanzar unas conclusiones que puedan ser de utilidad y den sentido a la realización del mismo.

1.3. Contexto

Para la realización de este trabajo, vamos a focalizarnos en la ciudad de Barcelona, puesto que dispone de una amplia red de información de acceso libre a disponibilidad de cualquier usuario.

Partimos del conocimiento del punto exacto donde se han dado los accidentes de tráfico en la última década y, mediante la creación de un mapa de calor utilizando esos datos (Figura 1), vemos cómo que hay zonas más proclives a que esto ocurra (puntos calientes o puntos negros).

Figura 1. Mapa de calor de accidentes en la ciudad de Barcelona (izquierda) y en el centro (derecha).

El centro de la ciudad presenta los puntos más conflictivos, lo que indica que, como se puede intuir, los accidentes ocurren en las zonas más concurridas, tanto de tráfico como de peatones.

Si vemos la densidad de habitantes por barrio (Figura 2), vemos como, por lo general, los barrios con mayor tasa de población se encuentran también en el centro, aunque la relación no se aprecia excesivamente significativa.

Esto conduce a pensar que no es únicamente la densidad de habitantes aquello que genera más riesgo, sino que es la afluencia que tiene una calle en base al número de peatones.

Es necesario determinar esta variable en base a la vitalidad que tienen las calles en cuanto a número de comercios, nivel de ocio, viviendas y edificios que aglomeren a gran cantidad de personas tales como centros educatios o sanitarios, entre otros.

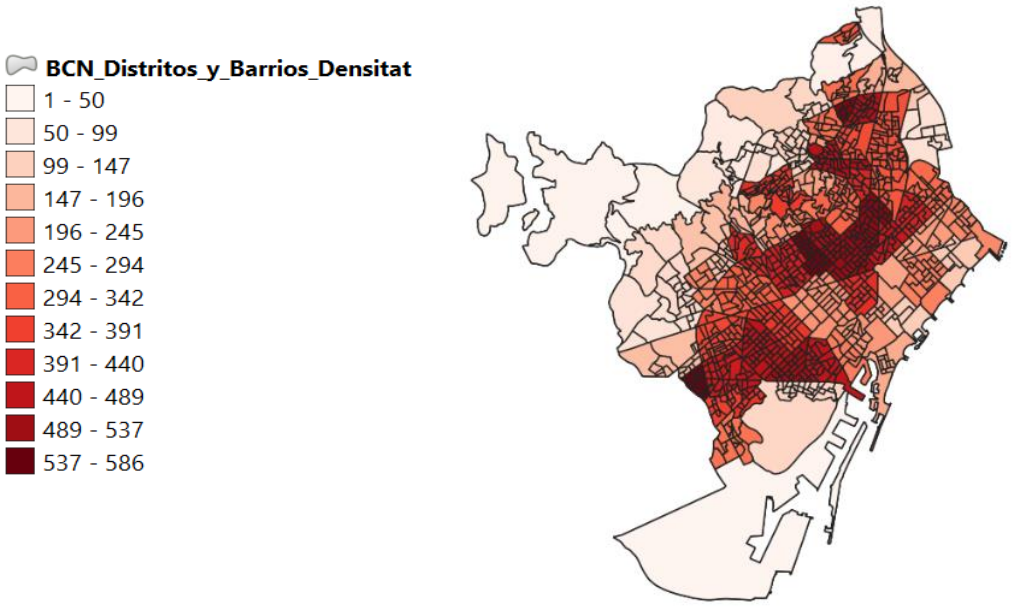


Figura 2. Densidad de habitantes por distritos y barrios.

Por otra parte, también será conveniente analizar aquellas variables relativas a las características de la vía como son la velocidad máxima, el sentido, el número de carriles, la señalización u obstáculos visuales que puedan existir como puede ser la arboleda viaria.

Tanto las variables relacionadas con la densidad de peatones, como aquellas relativas a la vía deben arrojar algo de luz sobre las causas que hacen más probable que un accidente ocurra.

2. Metodología

2.1. Descripción de la metodología SEMMA

En los últimos tiempos y, especialmente en la última década, la capacidad de almacenamiento de datos ha experimentado un desarrollo exponencial. Este hecho, de la mano de extraordinarios desarrollos computacionales y aparición de nuevas técnicas de aprendizaje han conseguido desarrollar el área de minería de datos, que consiste en extraer información relevante de fuentes de datos de gran volumen (Rodríguez Montequín, Álvarez Cabal, Mesa Fernández, & González Valdés).

Son varias las metodologías más utilizadas a la hora de realizar proyectos. En este trabajo, se utilizará el programa SAS Enterprise Miner donde se seguirá la metodología SEMMA, que descansa en la ejecución de las 5 fases siguientes (Calviño, 2019):

- *Sample*: Ante bases de datos de gran volumen, se precisa realizar una muestra que represente el total de los datos y pueda ser procesada.
- *Explore*: Se ha de realizar una exploración inicial de datos que detecte información útil como relaciones y tendencias.
- *Modify*: Es conveniente modificar los datos mediante transformación y selección de variables.
- *Model*: Fase de modelización.
- *Assess*: Evaluar la calidad y comparar los modelos obtenidos en la fase previa.

2.2. Transformación y selección de variables

Con el método "mejor" del nodo "Selección de variables" de SAS Enterprise Miner, se realizan las transformaciones óptimas de todas las variables de intervalo, a excepción de la variable aleatoria, para que nos permita comparar. Las variables de clase, por su parte, verán sus niveles agrupados con el nodo "Selección de variables", de manera que se reduzcan niveles perdiendo el mínimo poder predictivo.

Se mantendrán tanto las variables input originales como las transformadas para después llevar a cabo diferentes técnicas de selección de variables, donde de escogerán diferentes grupos a analizar.

Para la selección de variables, vamos a usar en el programa SAS Base la macro "Randomselectlog" (Portela, Explicación de las principales macros para redes neuronales, 2019) aplicando una regresión paso a paso o *stepwise*, que es un método que va añadiendo o eliminado variables del modelo y comprobando si se produce una mejora en el modelo para determinar un conjunto óptimo de variables.

El set resultante es el que queda tras no encontrar ninguna otra mejora al incluir variables o suprimirlas. El criterio seleccionado es criterio de información bayesiano (BIC, por sus siglas en inglés), ya que es más restrictivo que el El criterio de información de Akaike (AIC, por sus siglas en inglés).

Esta macro realiza el modelo paso a paso repetidas veces, sortendo diferentes porcentajes de datos train para cada paso. El resultado es una tabla de frecuencia que incorpora las variables de los modelos que han sido seleccionados en cada iteración.

Otra técnica estadística que utilizaremos con el programa SAS Enterprise Miner para realizar el proceso de selección, es el análisis Clúster, que agrupa variables intentando conseguir homogéneidad dentro de los grupos y diferencia entre los grupos (Fernández, 2011). Se forman clústeres automáticamente, aunque manualmente se extraerán de ellos aquellas aquellas variables cuyo R2 no llegue a 0,5.

Vamos ahora a hacer, grupos de variables siguiendo la técnica de clúster de variables. Se reduce el número de variables mediante agrupación de variables de intervalo, que conforman la mayoría del set. Se forman varios clústeres y de sacan de cada uno de ellos aquellas variables cuyo R^2 no sea de al menos, de 0,5.

2.3. Validación cruzada repetida y comparación de modelos

Todos los modelos que se lleven a cabo a lo largo de este estudio se realizarán con validación cruzada repetida de 4 grupos. Esto garantiza que los resultados obtenidos sean independientes de la partición de datos en entrenamiento y prueba.

Eso se consigue realizando diferentes subconjuntos de entrenamiento y test (para este estudio 80% y 20%, respectivamente) y realizando tantos modelos como grupos determinemos (4). Los datos de evaluación son diferentes para cada grupo, de manera que los modelos resultantes se entrenan y se evalúan usando grupos de datos complementarios. En este estudio usaremos validación cruzada repetida.

Una vez realizados, los compararemos entre ellos bien en términos de *accuracy* o exactitud, o bien en términos de tasa de fallos. La exactitud es el índice que marca qué cantidad de predicciones positivas fueron realmente positivas y que se halla a través de la siguiente fórmula:

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN} \qquad Sensibilidad = \frac{VP}{VP + FN} \qquad Especificidad = \frac{VN}{FP + VN}$$

Siendo VP Verdaderos Positivos, VN Verdaderos Negativos, FP Falsos Positivos y FN Falsos Negativos.

La tasa de fallos es complementaria a la exactitud por lo que se puede calcular como 1 – exactitud.

Otra medida que usaremos para realizar la comparación de modelos es el área bajo la curva ROC (AUC, por sus siglas en inglés). La curva ROC (*Receiver Operating Characteristic*) muestra el rendimiento bajo todos los umbrales de clasificación y lo hace comparando la sensibilidad frente al complementario de la especificidad para puntos de corte diferentes. El área bajo la curva ROC, por su parte, mide la calidad de las predicciones independientemente del umbral de clasificación y de la escala. Toma valores en un rango de 0 a 1, siendo este último el objetivo deseable, puesto que representa que todas las predicciones realizadas son correctas.

2.4. Descripción de algoritmos a utilizar

2.4.1. Modelos de regresión logística

En 1961 se origina el modelo de regresión logística, que viene de la mano de Confield, Gordon y Smith, y que trata de explicar una variable cualitativa mediante otras explicativas. Este es un algoritmo supervisado que trata de explicar la probabilidad (p) de que presente el efecto estudiado, por lo que tiene una doble función; explicativa y predictiva (Fiuza Pérez & Rodríguez Pérez, 2000).

A medida que el volumen y complejidad de los datos ha ido incrementado, y la necesidad de predecir ha superado a la de explicar, surgen nuevos modelos que dejan a la comprensión lógica de los modelos estadísticos tradicionales a un lado para dar paso a la precisión (Portela, Machine Learning. Introducción, 2019). Veremos en los siguientes apartados este tipo de algoritmos y cómo superan a los tradicionales.

2.4.2. Modelos de redes neuronales binarias

Las redes neuronales artificiales conceptualmente se derivan del procesamiento que de la información que se lleva a cabo las neuronas en los sistemas nerviosos biológicos. Estas adquieren conocimiento mediante la experiencia que se almacena dando pesos relativos a las conexiones interneuronales (o nodos).

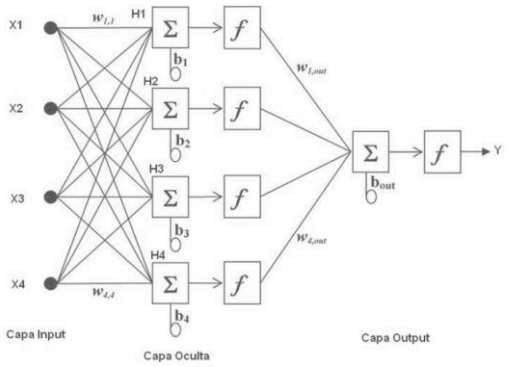


Figura 3. *Esquema de Red Neuronal* (Portela, Machine Learning. Introducción, 2019).

Las capas de entrada (imput) se conectan con las ocultas a través de una función de combinación que generalmente es la lineal (Figura 3). Después se aplica a cada nodo oculto una función de activación f, que suele ser la tangente hiperbólica (Figura 4). Para finalizar, se aplicaría combinación desde la capa oculta a la capa output.

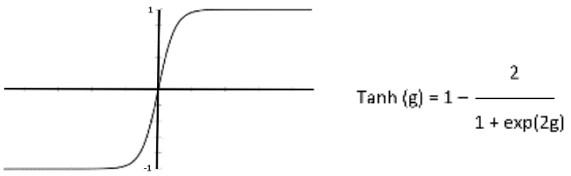


Figura 4. *Tangente hiperbólica.*

Los pesos wij son parámetros que requieren estimarse. Es necesario realizar un estudio de híper parámetros con el objetivo de no sobre ajustar.

Entre sus principales ventajas destaca la gran plasticidad y capacidad de adaptarse a situaciones cambiantes, detectar relaciones no lineales y tener una baja sensibilidad a información anómala (Izaurieta & Saavedra, 2000).

2.4.3. Modelos basados en árboles

Los árboles de decisión son algoritmos iterativos que consisten en dividir los datos según los intervalos de las variables independientes. Cuentan con grandes ventajas con son la gran potencia descriptiva ya que son capaces de detectar relaciones no lineales, la detección automática de puntos de corte o la gran interpretabilidad de los datos, si comparamos con otras técnicas utilizadas.

No obstante, los árboles también tienen una serie de limitaciones, como es su escasa robustez y gran inestabilidad, que se debe a la gran sensibilidad que tiene a los cambios. Otra desventaja es su escasa capacidad predictiva y gran variabilidad, alto sesgo y alta varianza (Portela, Bagging, Random Forest, Gradient Boosting, 2019). Para relajar estas limitaciones, surge la idea de realizar ensamblados de árboles que detallaremos a continuación.

Bagging

La primera aproximación llega de la mano de Brieman, en 1996, quien detalla esta técnica de combinación de árboles como “una forma relativamente fácil de mejorar un método existente... Lo que se pierde con los árboles es una simple y estructura interpretable. Lo que se gana es una mayor precisión” (Breiman, 1996).

El proceso es el siguiente: Se seleccionan muestras de tamaño N con reemplazamiento (en una primera instancia, después surgió la alternativa de no hacer la selección sin reemplazamiento) y se generan árboles que predicen los datos test. La media de las predicciones de cada árbol es el resultado de Bagging (Portela, Bagging, Random Forest, Gradient Boosting, 2019).

Los parámetros a configurar en este caso son los siguientes:

- Tamaño mínimo de nodos finales
- Número de árboles o iteraciones
- Tamaño de cada muestra
- Reemplazamiento o no
- Máxima profundidad de cada árbol
- Número máximo de ramificaciones
- P-valor

Random Forest

También desarrollado por Breiman, en 2001, este algoritmo es una importante herramienta de análisis de datos dada su gran capacidad de adaptación y gran potencial al tratar conjuntos de datos limitados, así como una buena precisión incluso con características de alta dimensión y estructuras de datos complejas (Scornet, Biau, & Vert, 2015).

Este algoritmo de bosque aleatorio sigue la misma filosofía, aunque mejora al anterior al sortear, además de observaciones, variables y así eliminar la problemática de árbol dominante que se general al tener variables con un nivel de importancia mayor. El resultado será un conjunto de árboles muy diferentes entre sí (Portela, Bagging, Random Forest, Gradient Boosting, 2019).

Los parámetros a configurar son los mismos que en *Bagging*, añadiendo el siguiente:

- Número de variables a sortear en cada nodo

Gradient Boosting Machine (GBM)

Se trata de un algoritmo iterativo bastante agresivo en el que se irán actualizando las predicciones de las observaciones de manera que se halle un residuo y se construye un árbol que los prediga para después actualizar el valor de la predicción en la dirección de decrecimiento de ese residuo a través de una constante de regularización que habrá que configurar. Así el valor predicho, se va acercando al valor real, lo que minimiza el error.

Es conveniente realizar un estudio previo para evitar el sobreajuste, lo que se llevará a cabo en R. Por otra parte, hay que tener en cuenta posibles valores atípicos o variables sin influencia en la objetivo, puesto que este algoritmo es sensible a ellas (Portela, Bagging, Random Forest, Gradient Boosting, 2019). Parámetros a ajustar:

- Constante de regularización o velocidad a la que cambian los valores de las predicciones.
- distintos tamaños mínimos de nodos finales
- Tamaño mínimo de nodos finales
- Número de árboles o iteraciones
- Tamaño de cada muestra
- Máxima profundidad de cada árbol
- Número máximo de ramificaciones

Este algoritmo cuenta con una versión mejorada; el *Extreme Gradient Boosting* (XGBOOST), que incluye penalización de las hojas. Veremos cómo supera los resultados del GMB más adelante con el programa RStudio.

2.4.4. Support Vector Machine (SVM)

"La máquina de vectores de soporte fue introducida por primera vez por Vapnik en 1992 como una serie armonía de conceptos superiores en el campo del

reconocimiento de patrones. SVM es un método de aprendizaje automático que funciona según el principio de minimización de riesgos estructurales (SRM) con el objetivo de encontrar el mejor hiperplano que separe las dos clases en la entrada espacio" (Nugroho, Witarto, & Handoko, 2003).

Tiene una filosofía, por lo tanto, geométrica que aborda el problema de separación de una manera sencilla. Entre sus virtudes destaca su gran flexibilidad, ya que tiene versión puramente lineal (que podría competir con la función logística) y otras no lineales. Destaca también los pocos parámetros que utiliza.

No obstante, igual que en *Random Forest*, es altamente sensible a *outliers*, datos faltantes o variables que no aportan nada a la dependiente. La principal desventaja personalmente observada es la gran exigencia computacional que requiere. Los parámetros que han de configurarse son los siguientes:

- Kernel lineal:
  - Parámetro C de penalización
- Kernel ponimonial:
  - Parámetro C de penalización
  - Grado del polinomio
  - Escala
- Radial Basis Function
  - Parámetro C de penalización
  - Sigma

2.4.5. Técnicas de ensamblado

Estas técnicas surgen de combinación de algoritmos surgen porque cada uno de ellos se basa en una filosofía diferente, por ejemplo, los árboles buscan regiones mientras que *Support Vector Machine* se basa en técnicas geométricas. Es por esta razón que se puede plantear que unos corrijan o complementen a otros, para que cada uno de ellos participe con lo mejor de sí mismos.

Con los resultados de los diferentes algoritmos utilizados se realiza un promedio de las predicciones (aunque también estas se pueden convertir predicciones en variables input de otro modelo) lo que hace que se logre reducir la variabilidad de los algoritmos.

3. Creación del conjunto de datos

El proceso de recopilación de información y creación de la base de datos se ha llevado a cabo utilizando el programa QGIS, que es un Sistema de Información Geográfica de código abierto, que nos va a permitir visualizar, modificar y crear datos geoposicionados.

Partimos de una capa de datos vectorial (capa matriz), suministrada por una empresa externa, donde aparecen geolocalizadas todas las vías de la ciudad de Barcelona tramificadas de forma desigual en 59.530 tramos que tienen una longitud media de unos 36 metros (el 80% tiene esa longitud). La separación viene dada generalmente por intersecciones entre calles.

La idea es construir una base de datos en la que las observaciones sean cada uno de los tramos mencionados anteriormente. Se incluirá en cada uno de ellos toda la información posible en cuanto a número de accidentes, elementos viarios y densidad de peatones. Esta última consistirá en un indicador derivado de diferentes variables sobre la aglomeración de personas en una calle, como es el número de hospitales, terrazas o museos, entre otras. Estas conformarán la variable "Vitalidad" que se basa en el supuesto de que cuanto mayor sea esta, mayor será el número de personas.

3.1. Instituto Cartográfico y Geológico de Cataluña (ICGC)

De esta fuente (ICGC, 2020) se obtienen las secciones censales de Cataluña. Es necesario crear una nueva capa únicamente con el municipio 080193, perteneciente a Barcelona, que utilitzaremos, entre otros usos, para cortar todas las capas que incluyan información más allá de la ciudad que estamos analizando y así trabajar únicamente con la información estrictamente necesaria.

3.2. Open Data BCN

La página web del ayuntamiento de Barcelona, Open Data BCN (Open Data BCN, 2020), deja a disposición del público una gran cantidad de datos relevantes para este trabajo. La información proviene de tres tipos de capas: de puntos, vectoriales y de polígonos.

Los datos derivados de esta fuente tienen unas coordenadas que generalmente pertenecen al sistema ETRS89/UTM zona 31N y que permiten la visualización en QGIS del punto exacto donde aparecen, así como su asociación con el tramo de vía más próximo de la capa matriz.

Esto se realiza mediante un análisis de vecino más cercano, que se lleva a cabo agregando el complemento "NNJOIN" a QGIS. Este paso se realiza individualmente entre cada conjunto de datos descargados y la capa maestra y da lugar a una nueva capa que guarda la información de ambas en un archivo Shape File.

Existen dos maneras de unir las capas, explicadas a continuación, dependiendo de la naturaleza de los datos: las capas de puntos se unirán a la capa matriz mediante la primera forma mientras que las capas de líneas y polígonos lo harán con la segunda.

3.2.1. Primera forma: Añadiendo a las nuevas variables la capa matriz

Tenemos las distintas capas a las cuales añadimos la capa matriz de tramos. Una vez realizado esto, se hace un conteo y se añade a la capa matriz el número de eventos de la capa analizada generando así nuevas variables. Esto se lleva a cabo con las capas de puntos. Vamos a ver a continuación los distintos tipos de capa que siguen este procedimiento ordenadas en tres grupos según la información que recojan:

Capas de accidentes

- *Número de colisiones en la ciudad de Barcelona (2010-2020):*

Con una base de datos creada a partir de diferentes archivos relativos a la accidentalidad en la ciudad de Barcelona, se halla el número de colisiones producidas en el periodo 2010-2020. Estos datos nos permitirán construir tanto los mapas de calor de puntos calientes como la variable objetivo. Los archivos son los siguientes:

  - Accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona.
  - Accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona según tipología.
  - Descripción de la causalidad de los accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona.
  - Personas involucradas en accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona.
  - Vehículos implicados en accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona.

Se limpian las series para eliminar duplicados, tanto por número de expediente como aquellos que, aun teniendo diferente número de expediente, coinciden en mes, día, hora, coordenadas, y descripción, ya que se entiende que se trata del mismo accidente. Una vez depurados, nos quedan 98.470 accidentes. Subimos la capa a QGIS y representamos los accidentes (Figura 5).

Figura 5. *Colisiones en la ciudad de Barcelona (2010-2020).*

Capas de características de la vía

- Arboleda Viaria.
- Aparcamientos Varios.
- Semáforos.
- Inventario de señalización horizontal:

Los datos que recoge este archivo son especialmente importantes ya que se va a extraer diferente tipo de información según el tipo de señal (Departament de Gestió de la Mobilitat de l'Ajuntament de Barcelona, 2017).

  - Ceda: R-1
  - STOP: R-2
  - Zona_Escolar: B-46. Camaras_y_radares: B-42, B-42ª, B-55ª y B-55b.
  - Curvas_peligrosas: P-13, P-13b, P-14ª y P-14b

Para tratar correctamente estas señales, es necesario realizar un *buffer* o, lo que es lo mismo, aumentamos el punto donde se ubica la señal en un círculo con un radio de 50 metros, en el caso de la zona escolar y en cámaras y radares, y de 10 metros para curvas peligrosas, ya que se entiende que estas son zonas sensibles que afectan no únicamente a un punto, sino a una determinada área.

Capas de vitalidad

- Bibliotecas y museos.
- Censo comercial.
- Centros de Día.
- Centros de servicios sociales.
- Cines, teatros y auditorios.
- Comedores sociales.
- Enseñanza infantil.
- Enseñanza no reglada.
- Espacios de música y copas.
- Espacios de participación ciudadana.
- Espacios infantiles.
- Farmacias.
- Hospitales de atención primaria.
- Hoteles.
- Instalaciones deportivas.
- Lugares de culto.
- Mercados municipales.
- Mercados y ferias de la calle.
- Otros alojamientos.
- Parques y jardines.
- Pensiones.
- Residencias de mayores.
- Restaurantes.
- Terrazas.

3.2.2. Segunda forma: Añadiendo a la capa matriz las nuevas variables

Capas de características de la vía

Las siguientes variables se hallan utilizando igualmente el *plug-in* "NNJOIN" pero, esta vez, añadiendo primero la capa de los tramos, puesto las otras capas son vectoriales y de polígonos. Necesitamos saber la información para cada tramo y no cuál de ellos está más cerca del polígono o vector, ya que estos engloban muchos tramos.

- Zonas 30 (capa de polígonos).
- Corredores Bici (capa de vértices).
- Carril Bici (capa de vértices).

- Vías Ciclables (capa de vértices).
- Laterales cortados al tráfico (capa de vértices).

Los archivos GeoJson que tenemos para las cuatro últimas capas relacionadas los carriles que permiten bicicletas, se muestran en líneas de vértices, al igual que lo hace la capa principal maestra, donde se está añadiendo toda la información. Esto genera un problema ya que las líneas de ambas difieren ligeramente, luego la distancia entre ellas no aparecerá como 0 en la mayoría de las ocasiones, por lo que podría caerse en el error clasificar una vía como no relacionada con la bici, cuando en realidad sí lo es.

Con el objetivo de evitar esto, se añadirá un *buffer* a la capa maestra de 8 metros (de manera que abarque todo el carril) para que su línea sea más gruesa y, por tanto, la unión de capas no genere sesgo de distancias.

Capas de vitalidad

- Densidad:

El último archivo que se descarga de Open Data BCN es la densidad de población por distritos y barrios para el año 2019 y que, aunque no dispone de coordenadas, sí aparecen los nombres y códigos de los distritos y barrios, por lo que unimos la información con los de división territorial hallada en el punto 2.1. Utilizaremos la variable "Densitat (hab/ha)", cuyo cálculo es el número de habitantes entre la superficie.

3.3. OpenStreetMaps

A través de esta página web (OSM, 2021) se descarga un archivo Shape File de varias capas para todo el territorio nacional, por lo que es necesario reducir la cantidad de datos a la ciudad de Barcelona. Para ello, se torna vital la capa de secciones censales obtenida en el punto 2.1. ya que con ella cortaremos el archivo limitándolo a la ciudad de Barcelona.

Tabla 1: Resumen de capas y atributos a extraer de los archivos de OSM

Tipo de Capa	*Descripción de la capa*	*Nombre de atributos*
Puntos	*Points of Interest*	Attraction / Monument / Castle / Archaeological-ruins / Viewpoint
		University/School/College
Líneas	*Roads*	Pedestrian only streets / Living streets / Footpaths/motorway/primary/secondary/tertiary…
		Speed
		One way street
	Railway	Railway
Polígonos	*Water*	Beaches

A partir de ahora se van a tratar datos derivados de OpenStreetMaps usando de referencia el índice que acompaña a los mismos (Ramm, 2019) y que se resume en la

tabla 1. De cada capa obtendremos las variables que aparecen en la última columna y que posteriormente tendremos que unir a la capa matriz una a una aplicando Vecino más próximo.

Capas de características de la vía

- Puente.
- Sentido_via (one way street).
- Tipo de vía (creada por los atributos: pedestrian only streets, living streets, footpaths, motorway, primary, secondary, tertiary...).
- Túnel.
- Velocidad de la vía (Speed).

Las capas puente y túnel no aparecen en la tabla puesto que no tienen un nombre de atributo en la capa "Roads" sino que aparecen como información adicional.

- Bicicletas:

Una vez tenemos las cuatro capas relativas al uso de la bicicleta (Corredores Bici, Carril Bici, Vías Ciclables y Laterales cortados al tráfico) más la de velocidad, creamos una única capa donde unificamos la información en 5 niveles:

  - Vía no acondicionada: aquella sin señalización ni acondicionamiento específico para bicicletas, vía ciclable o de <30km/hora.
  - Vía amigable con la bicicleta, bien por algún tipo de acondicionamiento o bien por velocidad.
  - Carril bici, aquellos carriles adicionales específicos para bicicletas.
  - Corredores bici.
  - Carriles bici protegido: aquellos cuyos laterales están cerrados al tráfico.

- Intersecciones Vía Férrea – Carretera

Con la finalidad de hallar los peligrosos pasos a nivel, se hallan las intersecciones de la capa "Carreteras" con "Railway". Para ello hay que filtrar las variables de manera que Túnel es falso (variable dicotómica T/F) para que no se mezclen los niveles.

- Intersecciones en Carretera

Esta capa pretende obtener información sobre el riesgo inherente en los cruces de carreteras dependiendo de las diferencias de velocidad de las mismas (Tabla 2). Para conseguirlo, se cruza la capa de tramos con ella misma y vemos dónde se producen las intersecciones.

Hay que tener en cuenta que los tramos son menores que las calles, por lo que habrá más intersecciones que cruces, aunque la velocidad será la misma y el riesgo será considerado, por tanto, muy bajo. Todos los cruces, en cambio, sí están representados. Si en un tramo hay más de una intersección, se considerará la de mayor riesgo.

Tabla 2: Riesgo de un cruce en función de la diferencia de velocidad

Niveles de Velocidad	*Diferencia entre tramo 1 y tramo 2 en km/h*	*Riesgo del Cruce*
<30	0	Muy bajo
40	10	Bajo
50	20	Medio
60	30	Medio
70	40	Alto
80	50	Alto
90	60	Extremo
100	70	Extremo

Capas de características de la vía

- Archaeological-ruins.
- Attraction.
- Beaches.
- Castle.
- College.
- Monument.
- Railway.
- School.
- University.
- Viewpoint.

Cabe mencionar que la unión de todas estas capas con la matriz de tramos sigue el mismo procedimiento que el mencionado en los puntos anteriores, dependiendo del tipo de capa que se trate (puntos, líneas o polígonos).

3.4. *Centro Nacional de Información Geográfica (CNIG)*

Capas de características de la vía

- Pendiente:

El objetivo de este punto es hallar la pendiente en grados del terrero para determinar si aquellos puntos con mayor diferencia de altitud afectan al riesgo de que se produzca un accidente.

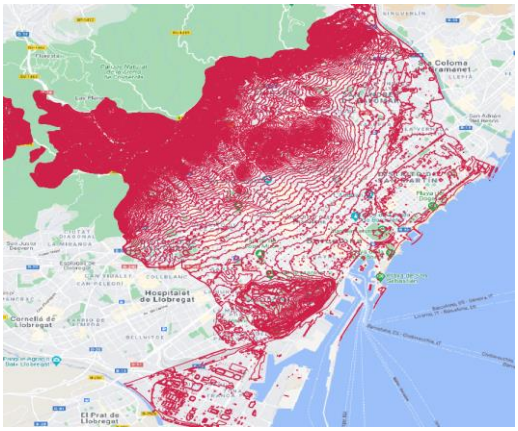


Figura 6. Curvas de nivel de la ciudad de Barcelona.

Para ello, es necesario extraer un modelo digital del terreno, que se extrae de la página del Centro Nacional de Información Geográfica (CNIG, 2021), desde donde adquirimos los modelos digitales del terreno 2ª Cobertura (2015-Actualidad), con paso de malla de 2 metros, pertenecientes a la ciudad de Barcelona. Unimos esos modelos y, una vez cargados, el algoritmo "Curvas de Nivel" que facilita QGIS nos permite obtener estas curvas cuya representación se puede visualizar en la figura 6.

Usamos estas curvas de nivel para realizar una interpolación espacial y así conseguir una continuidad de la topografía. En concreto, utilizaremos una interpolación de red irregular triangulada (TIN por sus siglas en inglés) formando triángulos a partir de los datos de los que partimos.

Cortamos el ráster TIN con la capa división administrativa del municipio de Barcelona (080193) que hemos hallado anteriormente y ahora, vamos a generar un mapa de pendientes en porcentaje, que se muestra en la Figura 7.

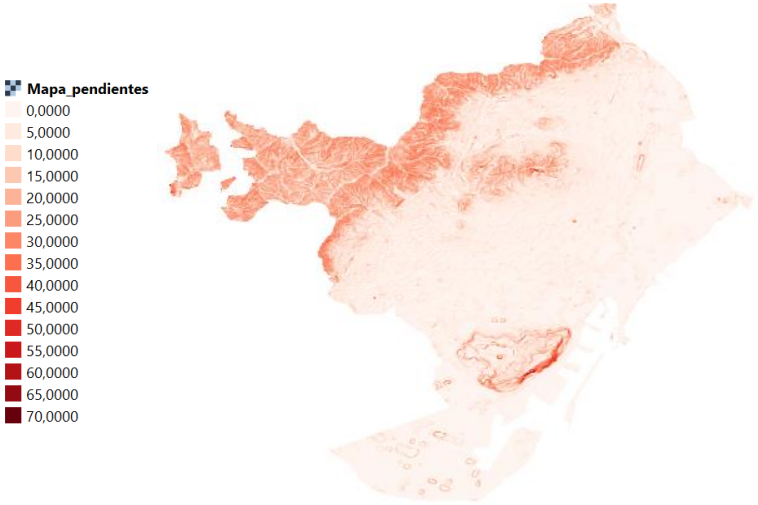


Figura 7. Mapa de pendientes de la ciudad de Barcelona.

Para hallar la pendiente de cada tramo de vía se procede a utilizar el *Plugin Point Sampling Tool*, que es un instrumento que halla valores de capa rásters (en nuestro caso, la capa de pendientes) a través de una capa de puntos. No contamos con estos puntos, por lo que se realiza una capa de intersecciones entre nuestra capa matriz y las curvas de nivel.

Una vez hallados dichos puntos, se procede a utilizar la herramienta previamente mencionada mezclando el ráster con la capa de puntos, de manera que se cree una nueva capa denominada "Pendiente_en_intersecciones", que únicamente muestra la pendiente de cada punto.

El siguiente punto sería mezclar mediante "JoinNN" esta capa recién creada con principal de tramos, con el objetivo de ver las diferencias de nivel en cada tramo. Como sabemos la pendiente exacta en cada punto, vemos la diferencia de pendientes. De cada tramo se coge el punto con mayor pendiente.

Capas de vitalidad

- Vitalidad:

Ya tenemos toda la información disponible en cuanto a vivacidad de las calles y, consiguientemente, una aproximación de la densidad de peatones. Pasamos, por lo tanto, a crear la nueva variable "Vitalidad". Para ello, tendremos dos variables: "Densidad", hallada previamente y "Puntuación", explicada a continuación:

Puntuación: Para obtener esta variable, se asigna una puntuación a las variables que hasta ahora se han encontrado dentro de "Capas de Vitalidad", en función de lo que se ha considerado que genera más o menos capacidad de atracción de personas.

- Capacidad alta (1 observación = 3 puntos): Se consideran los centros que generan aglomeraciones como mercados y ferias de calle, aunque también focos de población vulnerable, como es el caso de los espacios de copas, por existir posibles peatones etílicos.
- Capacidad media (1 observación = 2 puntos): Núcleos que congregan población considerada no vulnerable, como son los restaurantes.
- Capacidad baja (1 observación = 1 punto): establecimientos y comercios pequeños y medianos, tales como farmacias y actividades inmobiliarias.

El sumatorio de todos estos puntos crea la variable "Puntuación".

A partir de ahora se considera la importancia que estas dos variables pueden tener sobre la afluencia de peatones en las calles y, para determinar esto, realizamos una normalización estándar y asignamos unos pesos del 75% a la variable "Puntuación" y del resto a la variable "Densidad".

Se considera que la primera es más importante porque tenemos información por tramo mientras que la densidad de población está calculada por barrios, por lo que la información no es tan precisa.

3.5. *Tratamiento final de variables*

Para la obtención de variables se ha ido siguiendo un riguroso proceso de limpieza de las mismas, eliminando duplicados y valores ilógicos. Esto hace que el proceso de limpieza que realizaremos a continuación en SAS Base sea haga más ameno.

Por su parte, las variables con valores no representativos, por tener un gran número de ceros, pasan a unirse por tipología, como aparece en la tabla 3, que se encuentra en la página siguiente.

Dado que la base de datos tiene un total de 49.529 tramos, se consideran no representativos aquellas variables con un nivel de unos que no supere el 2%.

Tabla 3: Agrupación de variables no significativas

VARIABLE ORIGINAL	*VARIABLE AGRUPADA*
HOTELES	***Hoteles_pensiones_otros_alojamientos***
PENSIONES	
OTROS_ALOJAMIENTOS	
BIBLIOTECAS_Y_MUSEOS	***Bibliotecas_museos_cines_teatros_y_auditorios***
CINES_TEATROS_Y_AUDITORIOS	
ENSEÑANZA_INFANTIL	***Enseñanza***
ENSEÑANZA_NO_REGLADA	
UNIVERSITY	
SCHOOL/COLLEGE/IDIOMAS	
RESIDENCIAS_MAYORES	***Residencias_y_centros_de_mayores***
CENTROS_DE_DIA_DE_MAYORES	
MERCADOS_Y_FERIAS_CALLE	***Mercados_y_ferias_calle_y_mercados_municipales***
MERCADOS_MUNICIPALES	
ESPACIOS_INFANTILES	***Espacios_de_participacion_ciudadana***
COMEDORES_SOCIALES	
CENTROS_CIVICOS	
LUDOTECAS	
CENTROS_DE_BARRIO	
CENTROS_Y_ESPACIOS PARA JÓVENES	
CENTROS_DE_SERVICIOS_SOCIALES	
ATTRACTION	***Atracciones_turisticas***
MONUMENT	
CASTLE	
ARCHAEOLOGICAL/RUINS	
VIEWPOINT	
INSTALACIONES_DEPORTIVAS	***Otros_Comercios***
AUTOMOCION	
EQUIPAMIENTOS_CULTURALES_Y_RECREATIVOS	
ACTIVIDADES_INMOBILIARIAS	
MANTENIMIENTO_LIMPIEZA_Y_PRODUCCION	
OTROS	

Por último, aquellas variables que, una vez realizado el conteo y unidas a la capa matriz de tramos, toman valores de 0 y 1 (variables dicotómicas), es decir, que en cada tramo o bien no existe ningún registro de la variable que estemos analizando en ese momento, o, como máximo existe uno, se tratarán de distinta manera.

En lugar de hacer el análisis de vecino más cercano determinando qué tramo de vía tienen más cerano las observaciones de las distintas variables, volveremos a realizar todos los análisis al revés, de manera que obtendremos las distancias medias de cada tramo a cada observación a analizar.

4. Preparación de los datos

4.1. Exploración de datos

La cantidad de datos que se van a tratar no precisa de extracción de una muestra, por lo que se pasa directamente a la fase de exploración.

4.1.1. Variable objetivo:

Partimos de una variable continua que representa el número de accidentes que ha ocurrido en un determinado tramo de vía en la ciudad de Barcelona en últimos 10 años. Se pretende hacer un estudio que determine la peligrosidad de la vía en función de diversos factores, por lo que se considera necesario modificar la variable dependiente.

Se parte del número de accidentes por tramo y se halla el número de accidente por metro cuadrado primero, utilizando las variables "num_accidentes" y "meters" para después calcular el número de colisiones por 100 metros cuadrados.

En lugar de predecir el número de accidentes per se, se observa en qué tramos se han producido más accidentes de los que "correspondería". En el periodo analizado, ha habido un total de 98.470 accidentes y, teniendo en cuenta que hay 2.650.625 metros de calle, si estos se hubieran repartido de manera equitativa por toda la ciudad, tendría que haber habido 0,037 accidentes.

Una vez hallado esto, multiplicamos por 100 para obtener el número de accidentes por 100 metros cuadrados y redondeamos la serie. Se considera, por tanto, que los tramos con valores iguales o superiores a 4 (redondeo derivado de 3,7) pasarán a calificarse como tramos de gran incidencia de accidentes, mientras que aquellos que tengan cifras inferiores a ese número, se considerarán de poca incidencia de accidentes.

La variable objetivo pasa a ser una binaria que queda desbalanceada y que toma el valor 0 en tramos con escasa incidencia de accidentes y el valor 1 en trayectos con gran incidencia (Figura 8):

Rol de los datos	Nombre de la variable	Rol	Nivel	Número de ocurrencias	Porcentaje
TRAIN	Incidencia_Accidente_por_metro	TARGET	0	40284	81.3342
TRAIN	Incidencia_Accidente_por_metro	TARGET	1	9245	18.6658

Figura 8. Estructura de la variable objetivo.

Si estudia el siguiente histograma (Figura 9), vemos que en casi un 80% de los metros se producen, o bien menos accidentes de los que "deberían", o bien los que "corresponderían".

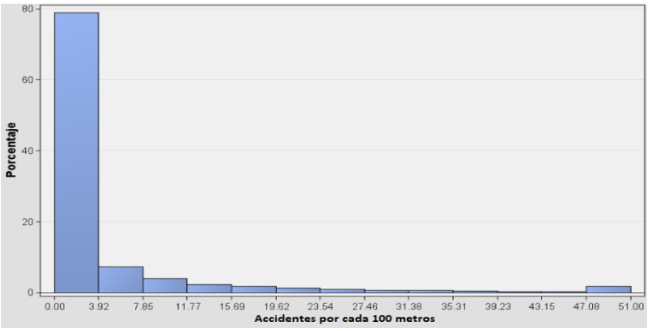


Figura 9. Histograma de accidentes por cada 100 metros.

En el gráfico se aprecia cómo es muy común que haya hasta 5 accidentes por cada 100 metros en los últimos 10 años en las calles de Barcelona. Es necesario resaltar que los datos que toman valores superiores a 50 están agrupados en el valor 51 y es por esta razón que aparecen representados de forma más significativa que los tramos anteriores.

4.1.2. Variables explicativas:

El objetivo es averiguar qué características de las calles hacen que un tramo de vía sea más o menos peligrosa. Para ello, utilizando el programa SAS Enterprise Miner, se va a proceder a la realización de un estudio del conjunto de datos y así encontrar relaciones entre las variables input y la objetivo. El primer paso es importar los datos y asignarles el rol correspondiente. Tenemos un conjunto de datos que tiene 49.529 observaciones y 57 variables de las cuales una de ellas es identificativa, 4 son binarias, incluyendo la objetivo, 8 nominales y 44 de intervalo.

En la tabla 4 aparecen todas las variables donde se especifican sus roles y su tipología, así como los niveles que las conforman:

Tabla 4: Variables que conforman el conjunto de datos; rol, tipo y niveles.

VARIABLE	*ROL*	*TIPO*	*NIVELES*
ACCIDENTES_POR_100_METROS	Input	Intervalo	
APARCAMIENTO	Input	Intervalo	
APARCAMIENTO_BICIS	Input	Intervalo	
APARCAMIENTO_COCHES	Input	Intervalo	
APARCAMIENTO_MOTOS	Input	Intervalo	
APARCAMIENTO_OTROS	Input	Intervalo	
ARBOLEDA_VIARIA	Input	Intervalo	
BICICLETAS	Input	Clase	Carril Bici, Carril bici protegido, Corredores Bici, Vía ciclable y <30 km/h, Vía ciclista y Vía no acondicionada
COTIDIANO_ALIMENTARIO	Input	Intervalo	
COTIDIANO_NO_ALIMENTARIO	Input	Intervalo	
DISTANCIA_ATRACCIONES_TURISTICAS	Input	Intervalo	
DISTANCIA_BIBLIOTECAS_MUSEOS_CINES_TEATROS_Y_AUDITORIOS	Input	Intervalo	
DISTANCIA_CURVA_PELIGROSA	Input	Intervalo	
DISTANCIA_ENSEÑANZA	Input	Intervalo	
DISTANCIA_ENSEÑANZA_INFANTIL	Input	Intervalo	
DISTANCIA_ESPACIOS_DE_MUSICA_Y_COPAS	Input	Intervalo	
DISTANCIA_ESPACIOS_DE_PARTICIPACION_CIUDADANA	Input	Intervalo	
DISTANCIA_HOSPITALES_DE_ATENCION_PRIMARIA	Input	Intervalo	
DISTANCIA_HOTELES_PENSIONES_OTROS_ALOJAMIENTOS	Input	Intervalo	
DISTANCIA_LUGARES_DE_CULTO	Input	Intervalo	

DISTANCIA_MERCADOS_MUNICIPALES	Input	Intervalo		
DISTANCIA_MERCADOS_Y_FERIAS_CALLE	Input	Intervalo		
DISTANCIA_MERCADOS_Y_FERIAS_CALLE_Y_MERCADOS_MUNICIPALES	Input	Intervalo		
METERS	Input	Intervalo		
DISTANCIA_PARQUES_Y_JARDINES	Input	Intervalo		
DISTANCIA_PLAYA	Input	Intervalo		
DISTANCIA_RADARES_Y_CAMARAS	Input	Intervalo		
DISTANCIA_RESIDENCIAS_Y_CENTROS_DE_DIA	Input	Intervalo		
DISTANCIA_ROUNDABOUT	Input	Intervalo		
EQUIPAMIENTO_PERSONAL	Input	Intervalo		
FARMACIAS	Input	Intervalo		
FINANCIERAS_Y_ASEGURADORAS	Input	Intervalo		
INCIDENCIA_ACCIDENTE_POR_METRO	Objetivo	Binaria	0 y 1	
INTERSECCION_DIFF_VELOCIDAD	Input	Clase	0, 10, 20, 30, 40, 50, 60 y 70	
INVENTARI_SEMAFORS	Input	Intervalo		
MENAJE_HOGAR	Input	Intervalo		
NOM_BARRI	Input	Clase	Nombre de los barrios de Barcelona	
NOM_DISTRICTE	Input	Clase	Ciutat Vella, Eixample, Gracia, Horta-Guinardo, Les Corts, Nou Barris, Sant Andreu, Sant Marti, Sants-Montjuic y Sarria - Sant Gervasi	
NUM_CEDA	Input	Intervalo		
NUM_CEDA_STOP	Input	Intervalo		
NUM_STOP	Input	Intervalo		
OCIO_Y_CULTURA_PEQUEÑOS	Input	Intervalo		
OTROS_CENTROS_PEQUEÑOS_ENSEÑANZA	Input	Intervalo		
OTROS_COMERCIOS	Input	Intervalo		
PENDIENTE	Input	Intervalo		
PUENTE_O_TUNEL	Input	Binaria	2	N y Y
RAILWAY_IN_STREET_INTERSECCIONES_VIA_FERREA_CARRETERA	Input	Binaria	2	N y Y
REPARACIONES	Input	Intervalo		
RESTAURANTS	Input	Intervalo		
RIESGO_INTERSECCION	Input	Clase	5	Muy bajo, Medio, Alto, Bajo y Extremo
SENTIDO_VIA	Input	Binaria	2	1 y 2
STREET_NAME	Input	Clase		Nombre de las calles de Barcelona
TERRAZAS	Input	Intervalo		
TIPO_DE_VIA	Input	Clase	16	living_street, motorway, path, pedestrian, primary, residential, secondary, service, tertiary, track, track_grade1, track_grade2,track_grade3, track_grade4, track_grade5 y trunk
VELOCIDAD_VIA	Input	Clase	8	30, 40, 50, 60, 70, 80, 90 y 100
VITALIDAD	Input	Intervalo		

4.2. Modificación de datos

4.2.1. Tratamiento de datos ausentes y detección de errores:

En cuanto a las variables de clase, no se encuentran datos ausentes (Anexo I, Tabla A), aunque se detectan una gran cantidad de niveles que tendrá que reducirse en el siguiente paso. Tampoco se encuentran anomalías.

Si se tienen en cuenta las variables de intervalo, tampoco existen ausentes en este caso (Anexo I, Tabla B) y, esto se debe al exhaustivo trabajo previo en el proceso de obtención de las variables. Sí existe, sin embargo, un valor erróneo en la variable metros, donde se corrigen los metros poniendo un mínimo de 0 y un máximo de 1.600 metros.

4.2.2. Modificación de variables:

Una vez analizada la posible presencia de errores con datos interpretables, procedemos a relativizar las variables continúas dividiendo entre la variable “Meters”, con la finalidad de tener en cuenta la longitud de la calle obteniendo el número de observaciones por metro y que no se generen valoraciones erróneas (la variable vitalidad ya está tratada). No se tratan aquellas variables continuas que muestran la

distancias a determinados puntos y cuya obtención se explica en los dos últimos párrafod del punto 2.5. (Tratamiento final de variables).

Se eliminan ahora del modelo las variables "Accidentes_por_100_metros", "Meters" y "Street_name". Las dos primeras se suprimen puesto que ya han sido utilizadas previamente para la consecución de una tercera. En el caso del nombre de la calle, esta queda eliminada debido al gran número de clases existente. Nos queda entonces un conjunto de datos configurado de la manera que muestra la Figura 10:

Rol	Nivel de medida	Número de ocurrencias
ID	INTERVAL	1
INPUT	BINARY	3
INPUT	INTERVAL	42
INPUT	NOMINAL	7
REJECTED	INTERVAL	2
REJECTED	NOMINAL	1
TARGET	BINARY	1

Figura 10. Resumen de variables.

El siguiente paso es reducir niveles de las variables de clase de manera automática con el nodo "Selección de variables", de manera que se pierda el mínimo poder predictivo, para hacer así que todas las clases sean significativamente diferentes. Se comprueba después que los valores resultantes representen al menos un porcentaje del 2% sobre el total de datos (49.529).

El programa agrupa las clases de todas las variables nominales del conjunto de datos. Se reducen todas las variables nominales; un ejemplo sería "Tipo_de_via", que pasa de tener 16 niveles a tener únicamente 5.

4.2.3. Tratamiento de datos atípicos:

La asimetría y curtosis son indicadores del tipo de distribución que tienen las variables. Todas ellas, a excepción de "Distancia_Playa", tienen valores de asimetría que no se encuentran en el intervalo que va desde -1 a 1, por lo que estas no siguen una distribución normal. Emplearemos el método Desviación Absoluta Mediana (MAD) para la detección de atípicos en aquellas que no tengan mediana igual a 0, donde se utilizará el método de percentiles extremos. Para la variable "Distancia_Playa" se indicará Desviación Estándar.

Se pasa a estudiar si el número de atípicos encontrado es realmente un valor *outlier* o si, por el contrario, estamos ante distribuciones muy asimétricas. Para ello observamos si el porcentaje de estos supera el 5% (2.476) del total de datos y se ha comprobardo que no es así, por lo que estos pasan a ser datos faltantes.

Al resto de las variables, se les asigna el método NINGUNO, ya que no tienen datos atípicos. La tabla 5 contiene información sobre datos faltantes que se han originado como consecuencia de pasar los valores no atípicos a *missings*.

Tabla 5: Estadísticos de variables de intervalo.

Variable	*Media*	*Desviación estándar*	*No ausente*	*Ausente*
REP_FARMACIAS	0.000133	0.00092	49525	4
REP_NUM_CEDA	0.000864	0.003447	49337	192
REP_NUM_STOP	0.000156	0.001236	49295	234
REP_PENDIENTE	0.011902	0.030482	49281	248
REP_TERRAZAS	0.000572	0.00237	49387	142
REP_VITALIDAD	-0.00049	0.002985	49047	482

Como se aprecia en la tabla 5, el porcentaje de ausentes es muy bajo por lo que no se elimina ninguna variable. Aun así, vamos a estudiar los datos faltantes por filas mediante la creación de la variable "numMissing", que cuente los ausentes por observación.

Variable	Rol	Media	Desviación estándar	No ausente	Ausente	Mínimo	Mediana	Máximo	Asimetría	Curtosis
numMissing	INPUT	7.043687	8.942256	132807	0	1	1	4	1.03536	-0.84955

Figura 11. Estadísticos de la variable "NumMissing".

La nueva variable tiene un máximo de 4 valores faltantes por fila (Figura 11). Como el total de variables con el que contamos es de 51, no se descarta ninguna de ellas; los valores *missings* no son representativos.

Se pasan por tanto a imputar los valores faltantes mediante un método de imputación que asigna valores siguiendo las distribuciones de las variables. Posteriormente se comprueba que ya no falta ningún dato.

4.2.4. Transformación de variables:

Con el método "mejor" del nodo "Selección de variables" de SAS Enterprise Miner, se realiza una transformación logarítmica en "Distancia_residencias_y_centros_de_mayores" mientras que el resto de las variables se transforma mediante el agrupamiento óptimo, es decir, estas se discretizan para maximizar la relación con la variable objetivo.

Se mantienen tanto las variables input originales como las transformadas para después llevar a cabo diferentes técnicas de selección de variables. Es necesario recordar que las variables de clase ya se han transformado anteriormente con el método selección de variables. El número de variables que tenemos ahora ha aumentado considerablemente. Por último y con el objetivo de reducir la ralentización de escritura de código, se pasa a renombrar las variables con letras del abecedario (Anexo II).

4.2.5. Análisis de la relación de las variables input con la variable objetivo

Imprimimos el gráfico de la V de Cramer que toma valores entre 0 y 1 con el objetivo de analizar las relaciones entre factores.

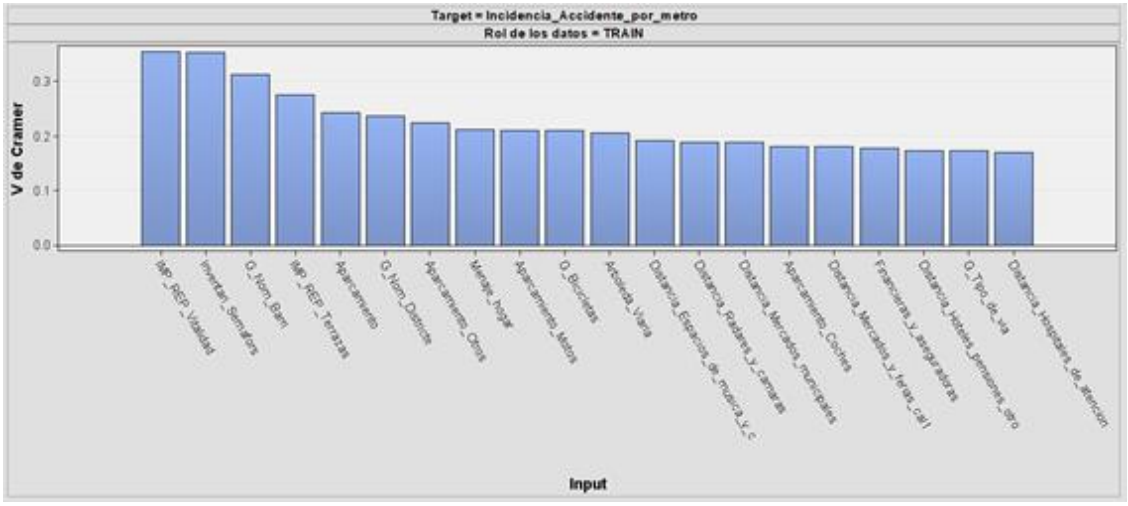


Figura 12. *Relación entre las variables explicativas sin transformar y la objetivo. V de Cramer.*

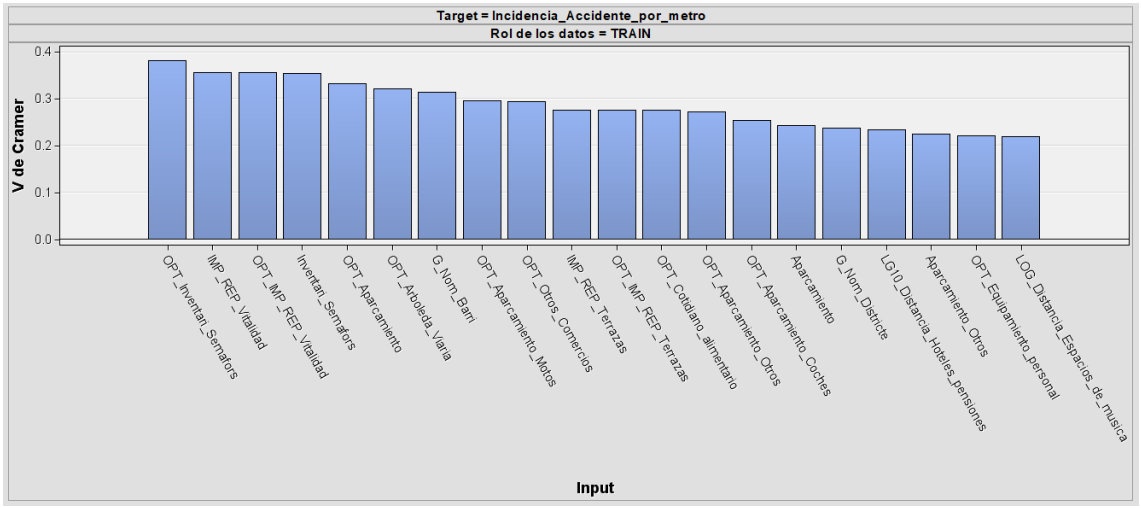


Figura 13. *Relación entre todas las variables explicativas y la objetivo. V de Cramer.*

En ambas figuras muestran relaciones no muy fuertes entre variables input y la objetivo, puesto que la mayoría está en un valor menor a 0.3. La variable que más correlación presenta es "OPT_Inventari_Semafors" seguida de "IMP_REP_Vitalidad" que rozan la cifra de 0,4.

4.3. Selección de variables

Se siguen varios caminos a la hora de realizar la selección de variables que veremos a continuación:

4.3.1. Camino I: Todas las variables sin transformar

Se usan todas las variables que no hayan sufrido ninguna transformación, por lo que tenemos las mismas que al principio, cuando hemos cargado los datos.

4.3.2. *Camino II: Todas las variables transformadas y sin transformar con selección de variables (BIC)*

En este camino, se ursará la macro "Randomselectlog" (Portela, Explicación de las principales macros para redes neuronales, 2019) y aplicando el criterio BIC. Se procede a realizar una búsqueda analizando 255 veces el criterio, a fin de ver si existen conjuntos de variables que se repiten o no.

La salida de SAS (Figura 14) indica que, de todas esas iteraciones llevadas a cabo, únicamente dos conjuntos de variables se han repetido (obs1 y obs2), por lo que crearemos dos nuevos conjuntos de variables (grupo 2 y grupo 3).

Obs	efecto	COUNT	PERCENT
1	as at au bf bj bk bl bm bn bp br bt bw by bz ca cb cc cf cg cj ck cl cn co cq cr ct g ac ag aj al am bv d k n r t u v w y	2	0.49875
2	as at au bf bj bk bl bm bn bp br bt bw by bz cb cc cf cg cj ck cl cn co cq cr ct g ac al an ax bv d k r t u v w y	2	0.49875

Figura 14. *Tabla de frecuencias de los modelos. "Randomselectlog" siguiendo el camino II.*

4.3.3. *Camino III: Clúster de variables*

Vamos ahora a hacer con SAS Enterprise Miner, grupos de variables siguiendo la técnica de clúster de variables. Se reduce el número de variables mediante agrupación de variables de intervalo, que conforman la mayoría del set (Figura 15). Se forman varios clústeres y de sacan de cada uno de ellos aquellas variables cuyo R^2 no sea de al menos, de 0,5.

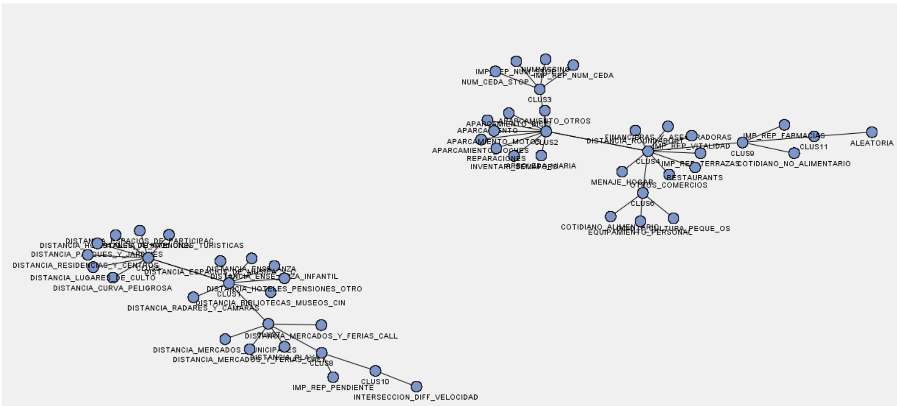


Figura 15. *Clúster de variables.*

4.3.4. *Camino IV: Clúster de variables, transformación posterior y selección (BIC)*

Una vez tenemos el conjunto que nos ha dado los clústeres, se procede a realizar una transformación de dichas variables. Tanto las variables del clúster como las nuevas transformadas, usando la macro "Randomselectlog", pasan ahora el filtro del proceso de selección BIC en SAS Base y obteniendo los resultados de la Figura 16.

Sistema SAS

Obs	efecto	COUNT	PERCENT
1	ar as au bf bj bk bl bp bq bt by cb cc cf cg cj cl cn co ct g x w an t u ay d ac	7	1.74564
2	ar as au bf bj bk bl bp bq bt by cb cc cf cg cj cl cn co ct g x w ag an t s u ay d ac	6	1.49626
3	ar as au bf bj bk bl bp bq bt by cb cc cf cg cj cl cn co ct g x w an t s u ay d ac	5	1.24688

Figura 16. *Tabla de frecuencias de los modelos. "Randomselectlog" siguiendo el camino IV.*

De las 250 realizaciones del método BIC llevadas a cabo, solo un conjunto de variables se repite 7 veces, siendo 6 las veces el número de veces que se repite el segundo grupo y 5 las veces que lo hace el tercero. La mayoría de las variables de estos tres grupos coincide a excepción de una o dos. Estudiaremos el primer set y después iremos añadiendo o reduciendo variables para hacer estudios de diferentes conjuntos.

Los grupos que emergen del Camino 4 tienen básicamente la misma configuración, por lo que, en el estudio posterior de híper parámetros, únicamente se estudiará el grupo 5, y sus resultados se aplicarán a los grupos 6 y 7.

4.3.5. Camino V: Selección manual de variables basada en exploración inicial

Este grupo se configura con las 20 variables más importantes del gráfico V de Cramer que no han sufrido transformación y que aparecen en el punto 4.2.5 (Figura X).

Encontramos un resumen de los caminos seguidos y los grupos de variables obtenidos en la Tabla 6.

Tabla 6. *Selección de variables: caminos y grupos.*

GRUPOS	CAMINOS	DESCRIPCIÓN CAMINOS		*VBLE DE CLASE*	*VBLE DE INTERVALO*	*TOTAL VBLE*
GRUPO 1	Camino I	Todas las variables sin transformar		10	42	52
GRUPO 2	Camino II	Todas las variables transformadas y sin transformar con selección de variables (BIC)	Obs 1	29	15	44
GRUPO 3			Obs 2	28	13	41
GRUPO 4	*Camino III*	Clúster de variables		10	25	35
GRUPO 5	*Camino IV*	Clúster de variables, transformación posterior y selección (BIC)	Obs 1	20	8	28
GRUPO 6			Obs 2	20	10	30
GRUPO 7			Obs 2	20	9	29
GRUPO 8	*Camino V*	Selección manual de variables basada en exploración inicial		4	16	20

Las variables que se incluyen en cada grupo se encuentran mencionadas en el Anexo III. Grupos de Variables.

Teniendo los diferentes conjuntos de variables con los que se trabajará, el siguiente paso es avanzar hacia la realización de modelos predictivos.

5. Construcción de modelos en SAS Base

Se procede a realizar un modelo de regresión logística por cada grupo de

5.1. Modelos de regresión logística

Se procede a realizar un modelo de regresión logística por cada grupo de variables y los resultados se comparan en el siguiente gráfico, donde se muestra la tasa de fallos:

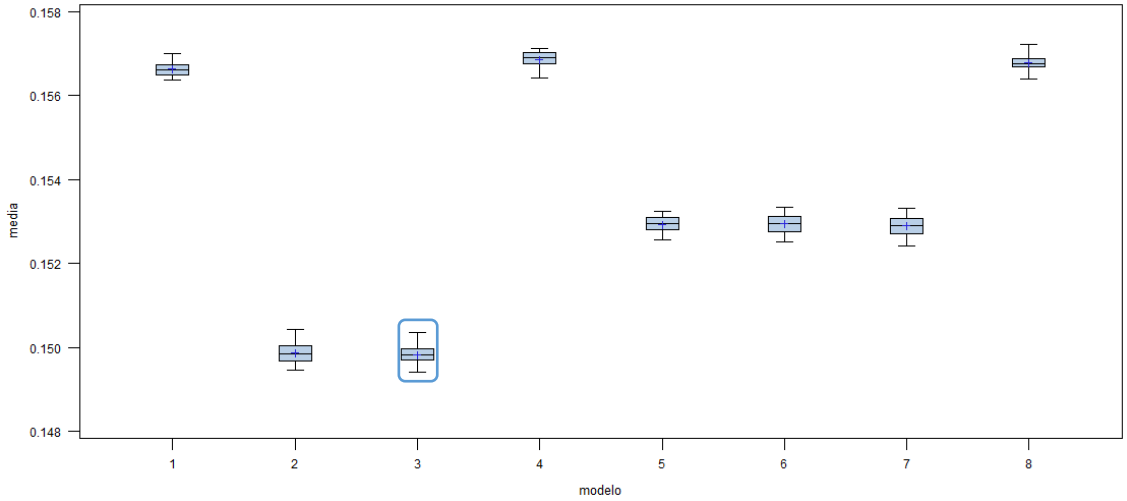


Figura 17. Tasa de fallos de modelos de regresión logística realizados en SAS Base.

El modelo conformado por las variables del grupo 3 es el que mejor resultado presenta en cuanto a tasa de fallos.

5.2. Modelos de redes neuronales binarias

Antes de pasar a la realización de los modelos se realiza un estudio de híper parámetros para cada grupo de variables detectado con anterioridad. Estos estudios únicamente se mostrarán para el primer conjunto, pudiendo encontrarse el resto en los diferentes anexos de este trabajo.

Determinación del número parámetros y número de nodos óptimo:

Previo a la realización de modelos de redes neuronales, se pasa ahora a analizar el número máximo de nodos que se puede usar considerando las características del conjunto objeto de estudio, con la finalidad de evitar la sobre-parametrización. Para ello, es necesario saber que la variable objetivo es una variable dicotómica que tiene un total de 49.529 observaciones. La clase minoritaria (gran incidencia de accidentes), supone un 18,67 % de la misma, y cuenta con 9.245 observaciones.

Estimaremos sobre este número de observaciones de la clase minoritaria el número de nodos máximo utilizando la siguiente fórmula (Portela, Construcción del modelo y primeros ejemplos, 2019):

Número máximo de parámetros = h (k + 1) + h + 1
h= número de nodos ocultos, k=número de nodos inputs, premisa: mínimo 30 obs / parámetro

Partiendo de la base que asume que es necesario contar con un número mínimo de 30 observaciones por parámetro, calculamos un máximo de 308 parámetros (9.245 observaciones de la clase minoritaria entre 30).

Dividiendo ese número máximo de parámetros (308) entre el total de imput que tiene cada grupo de variables y redondeando el número que resulta, se obtiene el número máximo de nodos (H) que no podremos superar si no queremos sobre ajustar.

Tabla 7. *Número máximo de nodos posibles para evitar sobre ajustar*

Grupos de variables	*Total de Input**	*H (nº de nodos máximo)*
Primer grupo	83	4
Segundo grupo	80	4
Tercer grupo	77	4
Cuarto grupo	49	6
Quinto grupo	60	5
Sexto grupo	62	5
Séptimo grupo	61	5
Octavo grupo	34	9

El número total de inputs de cada grupo se deriva de la suma de los que tienen las distintas variables. Cada variable de intervalo es un input, mientras que las variables nominales representan tantos inputs como niveles – 1 tienen. Los niveles inputs que suponen cada variable se pueden observar en el Anexo IV.

Este algoritmo se configurará con una capa input de entrada y una única capa oculta que contará con un número de nodos a determinar. Ambas capas se relacionan mediante unos pesos que concretan la importancia de cada input. Conociendo el número de nodos máximo posible que evita el sobre ajuste, se estudia ahora, en términos de tasa de fallos, qué cantidad de estos garantiza un mejor resultado para cada grupo de variables, utilizando en una primera instancia el algoritmo de optimización Levmar y, después, el Back Propagation:

Grupo 1:

a. Algoritmo de optimización Levmar

Usar 3 nodos asegura mayor exactitud y un menor error que utilizar cualquier otro número de estos utilizando Levmar (Figura 18). Back propagation, sin embargo, invita a usar un número de nodos mayor (4) para la obtención de mejores resultados.

Si los comparamos entre ellos, vemos como el algoritmo de optimización Levmar presenta una menor tasa de fallos (0.146 frente a 0.150 de BPROP). Además, para conseguir esto, necesita un menor número de nodos.

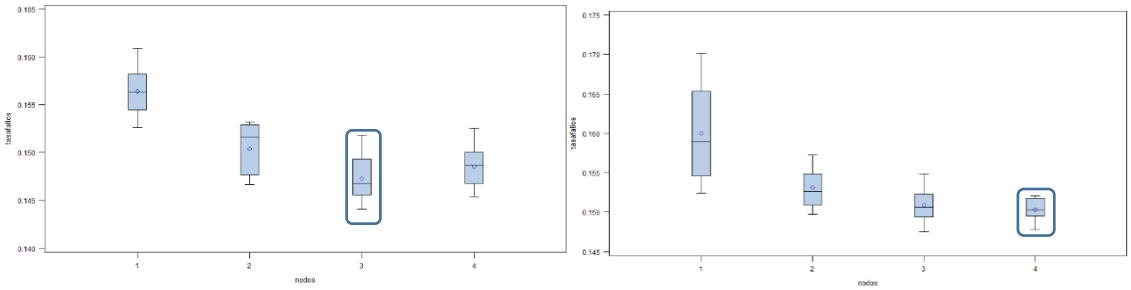


Figura 18. Tasa de fallos resultantes de utilizar Levmar (Izq.) y BPROP (Der.) con variables del Grupo 1.

El siguiente paso es realizar un estudio de *early stopping* y obtener el punto a partir del cual aumentos de iteraciones no se traduzcan a mejoras en términos de error. En concreto, se intenta hacer mínima la función de error Entropía (Figura 19) utilizando los algoritmos de optimización Levmar y Back Propagation para los distintos grupos de variables para un número de nodos previamente determinado.

$$Error = \sum_{i=1}^{n} [y_i \cdot log(\frac{y_i}{f(x_i)}) + (1 - y_i) \cdot log(\frac{1 - y_i}{1 - f(x_i)})]$$

Figura 19. Función de Entropía (Portela, 2020).

Estudio de early stopping

Grupo 1:

a. Algoritmo de optimización Levmar y 3 nodos

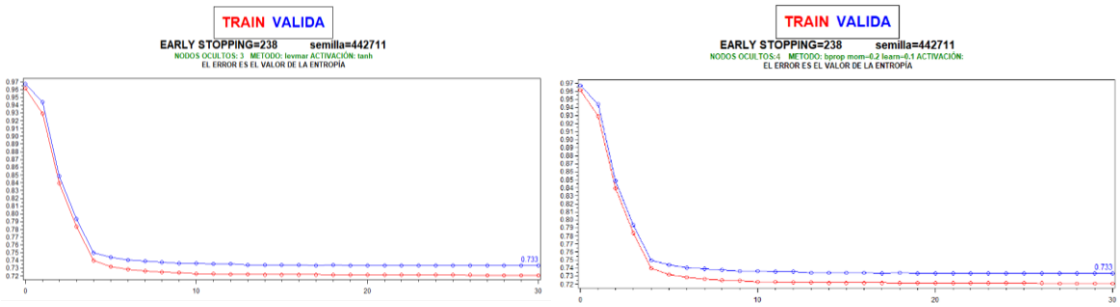


Figura 20. *Early stopping* utilizando 3 nodos con Levmar (Izq.) y 4 con BPROP (Der.) y Grupo 1.

En la gráfica se observa que configurar más de 5 repeticiones del algoritmo Levmar crearía un modelo más extenso, pero con iguales resultados. En caso de usar *Back Propagation*, el número de iteraciones ideal es también de 5.

La salida del programa recomienda, para ambos algoritmos, un número mayor de 238, que es el punto donde se minimiza el error, aunque mirando la Figura 20, a simple vista se observa que no es necesario un número tan grande.

Se resumen los híper parámetros del algoritmo redes neuronales binarias en la siguiente tabla (véase este mismo estudio realizado para las variables del grupo 1 para los demás grupos en el Anexo 5, en el subapartado Redes Neuronales).

Tabla 8. *Resumen de nodos e iteraciones a utilizar en Redes Neuronales Binarias*

Grupos de variables	**LEVMAR**		**BPROP**	
	Número de nodos	Número de iteraciones	Número de nodos	Número de iteraciones
1	3	5	3	5
2	2	7	2	100
3	2	8	2	120
4	3	9	4	33
5, 6, 7	2	6	2	100
8	3	6	7	30

Conseguida la información de esta tabla, ya podemos pasar a realizar los modelos. Para cada grupo de variables, se configurarán un total de 10 donde se configuran el número de nodos óptimo para cada uno de ellos. Se realizan primero utilizando *early stopping* y después se llevan a cabo los mismos modelos sin usarlo. También se utilizan distintas combinaciones de *momentum* y *learning rate* para BPROP y se juega con las funciones de activación tangente hiperbólica y lineal.

Tabla 9. *Resumen de híper parámetros a utilizar en Redes para cada grupo de variables.*

MODELO	*Nº DE NODOS*	*EARLY STOPPING*	*ALGORITMO DE OPTIMIZACIÓN*	*FUNCIÓN DE ACTIVACIÓN*	*MOMENTUM*	*LEARNING RATE*
1	Nº óptimo	No	levmar	tanh		
2	Nº óptimo	Nº óptimo	levmar	tanh		
3	Nº óptimo	No	bprop	tanh	0.2	0.1
4	Nº óptimo	Nº óptimo	bprop	tanh	0.2	0.1
5	Nº óptimo	No	bprop	tanh	0.1	0.2
6	Nº óptimo	Nº óptimo	bprop	tanh	0.1	0.2
7	Nº óptimo	No	bprop	tanh	0.3	0.1
8	Nº óptimo	Nº óptimo	bprop	tanh	0.3	0.1
9	Nº óptimo	No	levmar	LIN		
10	Nº óptimo	Nº óptimo	levmar	LIN		

Comparativa de modelos por grupo

Grupo 1:

La figura 21 muestra cómo el modelo 9, compuesto por las variables del primer grupo y una configuración que utiliza 3 nodos, sin *early stopping*, y el algoritmo de optimización Levmar con la tangente hiperbólica como función de activación, es el que menor tasa de fallos presenta, lo que lo lleva directamente a la comparativa de mejor modelo de redes neuronales (véase el resto de los modelos ganadores por grupos en el Anexo 5, subapartado Redes Neuronales).

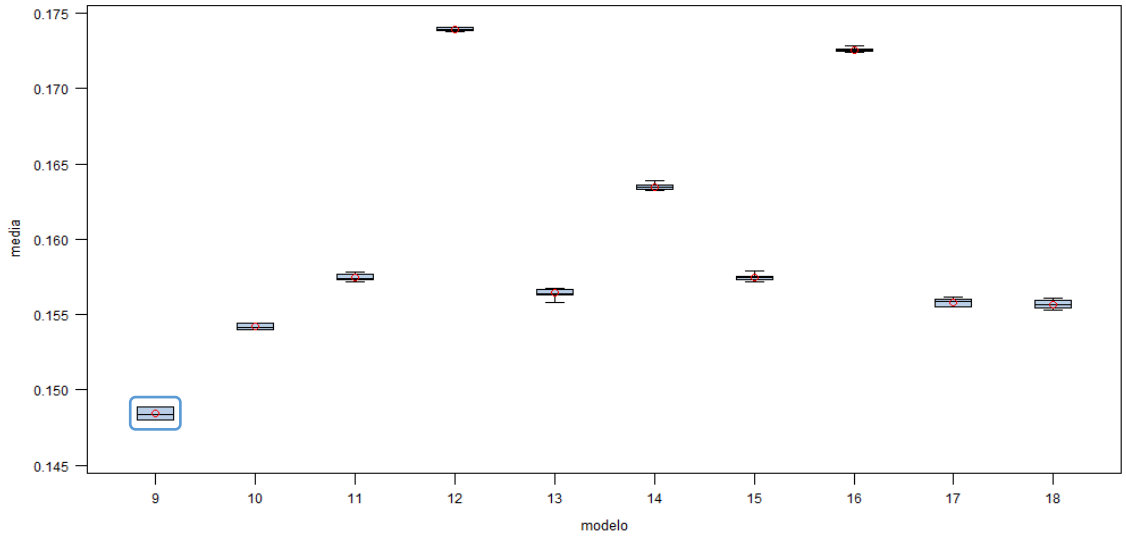


Figura 21. *Tasa de fallos de modelos de Redes Neuronales Binarias con las variables del Grupo 1.*

Comparativa de mejores modelos de Redes Neuronales Binarias

Pasamos ahora a descubrir qué modelo de todos los estudiados en redes es el que presenta una menor tasa de fallos:

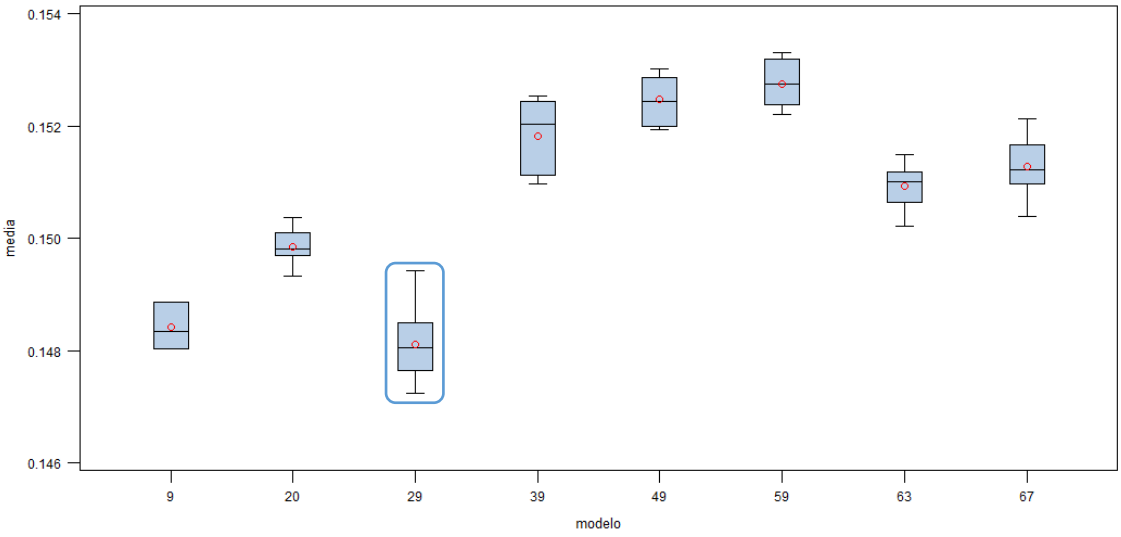


Figura 22. *Comparativa de los mejores modelos de cada grupo usando Redes Neuronales Binarias*

El modelo resaltado en la Figura 22, que está configurado con las variables del grupo 3, y tiene una tasa de fallos que ronda el valor 0,148. El número de nodos usado es el que hemos estudiado previamente, sin *early stopping* y el algoritmo de optimización es Levmar con la tangente hiperbólica como función de activación.

Es necesario relatar que todos los modelos ganadores de cada grupo están formados con la misma configuración que el ganador, a excepción del modelo 20, que sí tiene *early stopping* (Tabla 10). Se puede decir, por tanto, que es la mejor combinación estudiada.

Tabla 10. *Híper parámetros de los mejores modelos de cada grupo en Redes.*

MODELO	*Nº DE NODOS*	*EARLY STOPPING*	*ALGORITMO DE OPTIMIZACIÓN*	*FUNCIÓN DE ACTIVACIÓN*	*MOMENTUM*	*LEARNING RATE*
1	Nº óptimo	No	levmar	tanh		
2	Nº óptimo	Nº óptimo	levmar	tanh		
3	Nº óptimo	No	bprop	tanh	0.2	0.1
4	Nº óptimo	Nº óptimo	bprop	tanh	0.2	0.1
5	Nº óptimo	No	bprop	tanh	0.1	0.2
6	Nº óptimo	Nº óptimo	bprop	tanh	0.1	0.2
7	Nº óptimo	No	bprop	tanh	0.3	0.1
8	Nº óptimo	Nº óptimo	bprop	tanh	0.3	0.1
9	Nº óptimo	No	levmar	LIN		
10	Nº óptimo	Nº óptimo	levmar	LIN		

5.3. Modelos basados en árboles

Dentro de los modelos conformados por árboles estudiaremos *Bagging* y *Random Forest* y Gradient Boosting Machine.

5.3.1. Bagging

Para la realización de modelos utilizando el algoritmo basado en árboles *Bagging*, planteamos un modelo muy agresivo como primera aproximación, con un tamaño de la hoja de 5, profundidad máxima de 10 y un p-valor de 0,1, para después ir relajando las condiciones con la finalidad de evitar el sobre ajuste.

Para todos los modelos, se fija una ramificación de 4 así como un porcentaje de muestra del 80% y 200 iteraciones, ya que se entienden como suficientes para estabilizar el árbol. Vamos a jugar con un tamaño mínimo de nodos finales de 5, 10, 15 y 20, una profundidad máxima de 5, 10 y 20, así como un p-valor de 0,1 y 0,05. En concreto, para cada grupo de variables, se van a probar los modelos que aparecen en la tabla 10.

Tabla 11. *Resumen de híper parámetros a utilizar en Bagging para cada grupo de variables.*

MODELOS	*ITERACIONES*	*VARIABLES*	*% MUESTRA*	*MÁXIMO RAMAS*	*MÍNIMO TAMAÑO HOJA*	*MÁXIMA PROFUNDIDAD*	*P-VALOR*
1	200	Total	0,8	4	5	10	0,1
2	200	Total	0,8	4	5	5	0,1
3	200	Total	0,8	4	10	10	0,1
4	200	Total	0,8	4	10	5	0,1
5	200	Total	0,8	4	15	5	0,1
6	200	Total	0,8	4	15	10	0,1
7	200	Total	0,8	4	5	20	0,1
8	200	Total	0,8	4	5	20	0,1
9	200	Total	0,8	4	5	10	0,05
10	200	Total	0,8	4	5	5	0,05
11	200	Total	0,8	4	10	10	0,05
12	200	Total	0,8	4	10	5	0,05
13	200	Total	0,8	4	15	5	0,05
14	200	Total	0,8	4	15	10	0,05
15	200	Total	0,8	4	5	20	0,05
16	200	Total	0,8	4	5	20	0,05

Como en casos anteriores y para no sobrecargar de gráficos al lector, se realizará el análisis para las variables del grupo 1 mientras que el estudio del resto de grupos podrá visualizarse en el anexo (Anexo 5, subapartado Bagging).

Grupo 1:

Aunque existen similitudes en cuanto a los resultados obtenidos en tasa de fallos, se elige como mejor modelo el 78, ya que su media es ligeramente menor a la del resto (Figura 23). Este tiene una configuración estricta, ya que cuenta con todas las variables del grupo 1, un tamaño de la hoja y profundidad máxima de 5, así como un p-valor de 0.1.

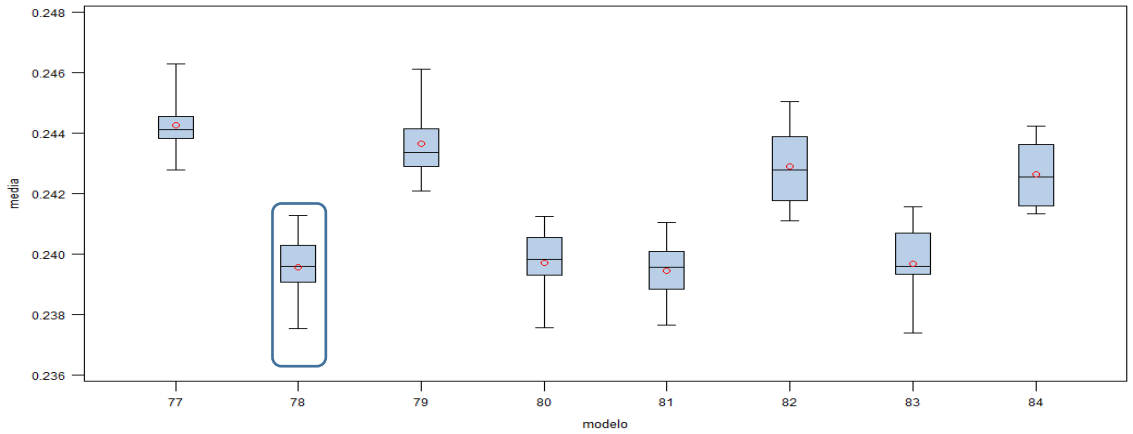


Figura 23. *Tasa de fallos de modelos Bagging con las variables del Grupo 1 y p-valor igual a 0,1*.

Pasamos a analizar ahora los mismos modelos, pero con un p-valor de 0,05:

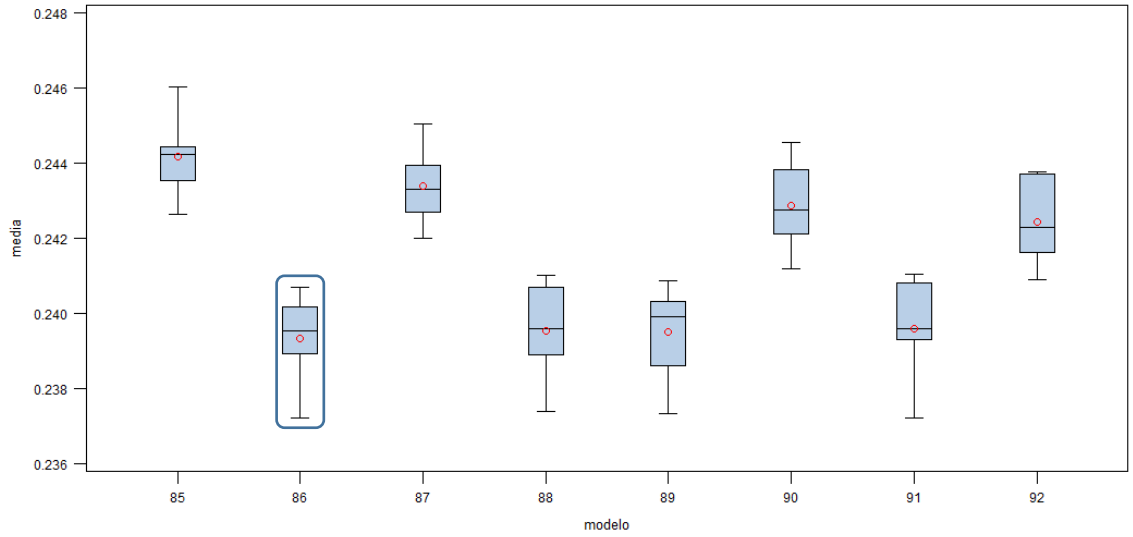


Figura 24. *Tasa de fallos de modelos Bagging con las variables del Grupo 1 y p-valor igual a 0,05.*

Se pueden apreciar unos patrones de comportamiento similares a los que se seguían con un p-valor mayor, lógicamente porque son prácticamente los mismos modelos. El mejor modelo, en este caso, es el equivalente al 78, pero con un p-valor de 0,05 (véase este mismo estudio para los diferentes grupos de variables en el anexo Anexo 5, subapartado Bagging).

Comparativa de mejores modelos Bagging

Los modelos ganadores de los diferentes grupos de variables tienen la siguiente configuración que aparece en la Tabla 12.

Tabla 12. *Híper parámetros de los mejores modelos de cada grupo en Bagging.*

MODELOS	*ITERACIONES*	*VARIABLES*	*% MUESTRA*	*MÁXIMO RAMAS*	*MÍNIMO TAMAÑO HOJA*	*MÁXIMA PROFUNDIDAD*	*P-VALOR*
78	200	Grupo 1	0,8	4	5	5	0,1
86	200	Grupo 1	0,8	4	5	5	0,05
99	200	Grupo 2	0,8	4	20	5	0,1
105	200	Grupo 2	0,8	4	15	5	0,05
112	200	Grupo 3	0,8	4	10	5	0,1
123	200	Grupo 3	0,8	4	20	5	0,05
131	200	Grupo 4	0,8	4	20	5	0,1
139	200	Grupo 4	0,8	4	20	5	0,05
144	200	Grupo 5	0,8	4	10	5	0,1
150	200	Grupo 5	0,8	4	5	5	0,05
159	200	Grupo 6	0,8	4	15	5	0,1
168	200	Grupo 7	0,8	4	20	5	0,1
176	200	Grupo 8	0,8	4	10	5	0,1
187	200	Grupo 8	0,8	4	20	5	0,05

La máxima profundidad coincide para todos ellos, luego profundidades de 10 no resultan buenas en términos de error. En ocasiones, cambiar el p-valor es idóneo mientras que en otras empeora los resultados. El tamaño de la hoja idóneo, por su parte, varía en función de las variables que se introduzcan.

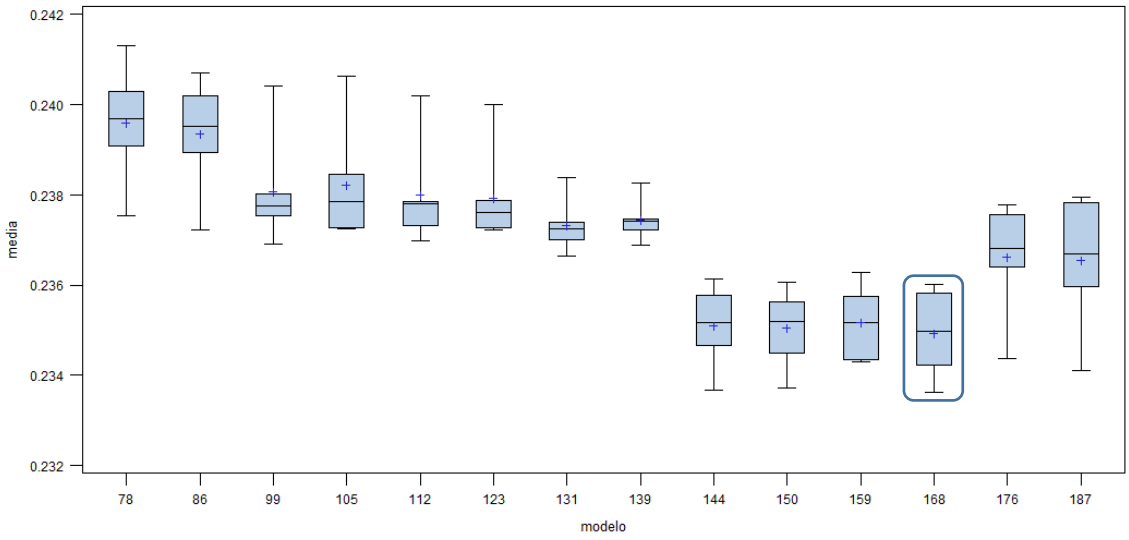


Figura 25. *Comparativa de los mejores modelos de cada grupo usando Bagging.*

El modelo que aparece resaltado en la Figura 25, que es el que se configura con las variables del grupo 7, una profundidad máxima de 5 con tamaño mínimo de la hoja de 20 y un p-valor de 0,1, es el ganador de todos los modelos estudiados en *Bagging*.

Los resultados que se presentan mediante la realización de este algoritmo son peores que los obtenidos en Logística y Redes Neuronales Binarias.

5.3.2. Random Forest

Siguiendo los pasos que hemos llevado a cabo en *Bagging* y se propone un modelo más agresivo con un tamaño de hoja de 7, máxima profundidad de 10 y un p-valor de 0,1. Vamos a ir modificando esos valores de manera que fluctúen de la siguiente manera: tamaño mínimo de nodos finales de 7 y 15, profundidad máxima de 5 y 10 y p-valor de 0,1 y 0,05.

Para todos los modelos, se fija una ramificación de 4 para todos los modelos, así como un porcentaje de muestra del 80% y 200 iteraciones, ya que se entienden como suficientes para estabilizar el árbol.

El algortimo de bosque aleatorio precisa de la configuración de un parámetro adicional; el número de variables a sortear en cada iteración. Este valor, en el estudio realizado se ha determinado como 8, con el que intentamos protegernos del sobreajuste, y 17 para intentar reducir un posible sesgo (Tabla 13).

Tabla 13. *Resumen de híper parámetros a utilizar en RF para cada grupo de variables.*

MODELOS	*ITERACIONES*	*VARIABLES*	*% MUESTRA*	*MÁXIMO RAMAS*	*MÍNIMO TAMAÑO HOJA*	*MÁXIMA PROFUNDIDAD*	*P-VALOR*
1	200	8	0,8	4	7	10	0,1
2	200	8	0,8	4	7	5	0,1
3	200	8	0,8	4	15	10	0,1
4	200	8	0,8	4	15	5	0,1
5	200	17	0,8	4	7	5	0,1
6	200	17	0,8	4	7	10	0,1
7	200	17	0,8	4	15	5	0,1
8	200	17	0,8	4	15	10	0,1
9	200	8	0,8	4	7	10	0,05
10	200	8	0,8	4	7	5	0,05
11	200	8	0,8	4	15	10	0,05
12	200	8	0,8	4	15	5	0,05
13	200	17	0,8	4	7	5	0,05
14	200	17	0,8	4	7	10	0,05
15	200	17	0,8	4	15	5	0,05
16	200	17	0,8	4	15	10	0,05

Grupo 1:

Los modelos 190 y 192 (Figura 26) presentan resultados bastantes similares, aunque al ampliar, se decide escoger el segundo de ellos porque, aunque su media de fallos es ligeramente peor, tiene menor variabilidad.

Este modelo tiene una configuración estricta, ya que cuenta con todas las variables del grupo 1, un tamaño de la hoja de 15 y profundidad máxima de 5, así como un p-valor de 0.1.

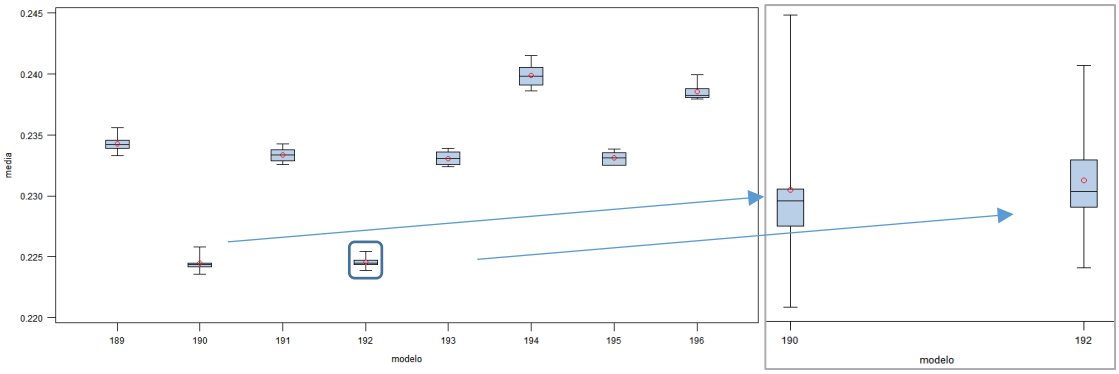


Figura 26. *Tasa de fallos de modelos RF con las variables del Grupo 1 y p-valor igual a 0,1*.

Pasamos a analizar ahora los mismos modelos, pero con un p-valor de 0,05:

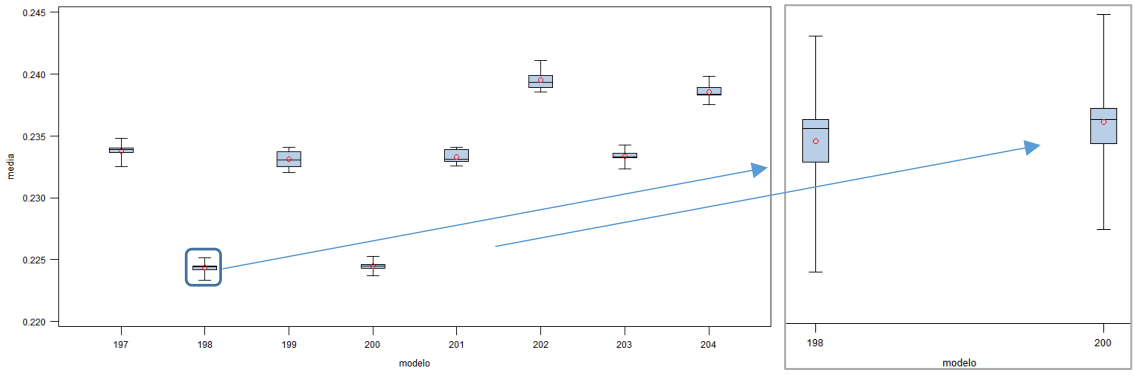


Figura 27. *Tasa de fallos de modelos RF con las variables del Grupo 1 y p-valor igual a 0,05*.

Las gráficas muestran esquemas similares al tratarse de los mismos modelos, pero con p-valores diferentes. Por lo tanto, para este último caso segundo modelo y el cuarto, son también buenos competidores.

Finalmente, se elige el 198, que es el que tiene un tamaño de hoja de 7 y una profundidad máxima de 5, con un p-valor de 0,05.

El resto de las gráficas que determinan el mejor modelo del resto de grupos se encuentra en el anexo (Anexo 5, subapartado Random Forest).

Comparativa de mejores modelos Random Forest

Los modelos ganadores de los diferentes grupos de variables tienen la configuración que se muestra en la Tabla 14.

Al igual que pasaba con *Bagging*, la mejor profundidad del árbol es de tamaño 5. Los resultados modificando p-valor y tamaño de la hoja actúan de manera diferente dependiendo del grupo de variables al que se apliquen. Aunque se prueba sortear las variables bien con 8 o con 17, todos los modelos ganadores sorteaban 8 variables.

Tabla 14. *Híper parámetros de los mejores modelos de cada grupo en RF.*

MODELOS	*ITERACIONES*	*VARIABLES*	*% MUESTRA*	*MÁXIMO RAMAS*	*MÍNIMO TAMAÑO HOJA*	*MÁXIMA PROFUNDIDAD*	*P-VALOR*
192	200	8 del Grupo 1	0,8	4	15	5	0,1
198	200	8 del Grupo 1	0,8	4	7	5	0, 05
208	200	8 del Grupo 2	0,8	4	15	5	0,1
216	200	8 del Grupo 2	0,8	4	15	5	0, 05
224	200	8 del Grupo 3	0,8	4	15	5	0,1
232	200	8 del Grupo 3	0,8	4	15	5	0, 05
240	200	8 del Grupo 4	0,8	4	15	5	0,1
248	200	8 del Grupo 4	0,8	4	15	5	0, 05
254	200	8 del Grupo 5	0,8	4	7	5	0,1
262	200	8 del Grupo 5	0,8	4	7	5	0, 05
269	200	8 del Grupo 6	0,8	4	7	5	0,1
273	200	8 del Grupo 7	0,8	4	7	5	0,1
278	200	8 del Grupo 8	0,8	4	7	5	0,1
288	200	8 del Grupo 8	0,8	4	15	5	0, 05

Se muestran estos modelos en el siguiente gráfico para comparar sus resultados:

Tasa de fallos:

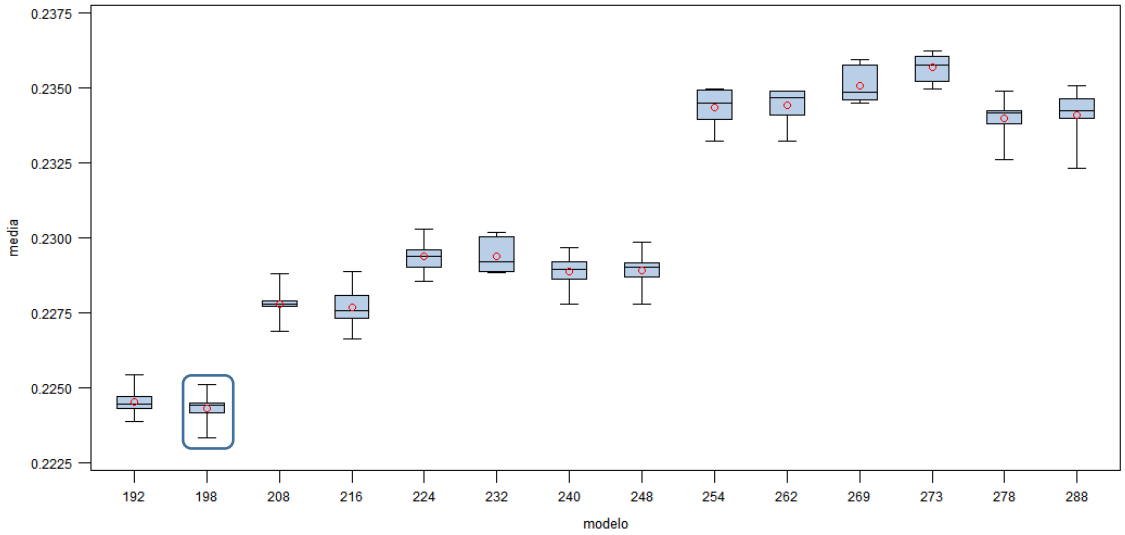


Figura 28. *Comparativa de los mejores modelos de cada grupo usando RF.*

El modelo 198, conformado por las variables del grupo 1, con profundidad máxima de 5, tamaño de la hoja 7 y un p-valor de 0.05, es el que menor tasa de fallos presenta (Figura 28), por lo que es el candidato de *Random Forest* a la comparación final. Los resultados siguen siendo peores que los que obteníamos en Redes y Logística.

5.3.3. Gradient Boosting Machine

Para intentar obtener el mejor modelo posible mediante este algoritmo se seguirá la siguiente estrategia: se parte de los mejores modelos para cada grupo alcanzados en *Random Forest* (Tabla 14) en cuanto a número de observaciones y tamaño mínimo de la hoja final. A partir de ahí se analizan los primeros modelos modificando el

número de iteraciones y la constante de regularización, que modificaremos desde 0,001 hasta 1, probando así, desde modelos más lentos y toscos hasta otros más rápidos, que tengan una variación acelerada de la predicción de la observación. Se tiene en cuenta que menor valor de *shrinkage* o velocidad de cambio requiere un mayor número de iteraciones.

Del mejor modelo obtenido, se modifica el número de iteraciones añadiendo y reduciendo en 100 el número de árboles y, del ganador de esta fase, se tocará el máximo de profundidad, de manera que se estudie una profundidad de 2, 3, 4, y 5 para todos los modelos. Una vez hallado el mejor hasta ahora, modificamos tanto el mínimo número de observaciones de la variable categórica (mincatsize) como el mínimo número de observaciones para dividir un nodo (minobs).

Grupo 1:

Partimos del modelo 198 que tiene un tamaño de la hoja de 7 observaciones, un máximo de profundidad de 5 y 4 particiones por cada nodo y vamos a ir modificándolo según se ha comentado en el punto anterior y se aprecia en la Tabla 15.

Tabla 15. *Primer resumen de híper parámetros a utilizar en GBM en el Grupo 1.*

MODELOS	*LEAFSIZE*	*ITERACIONES*	*SHRINK*	*MÁXIMO RAMAS*	*MÁXIMA PROFUNDIDAD*	*MINCATSIZE*	*MINOBS*
293	7	400	0,001	4	5	15	20
294	7	300	0,01	4	5	15	20
295	7	200	0,05	4	5	15	20
296	7	100	0,1	4	5	15	20
297	7	50	1	4	5	15	20

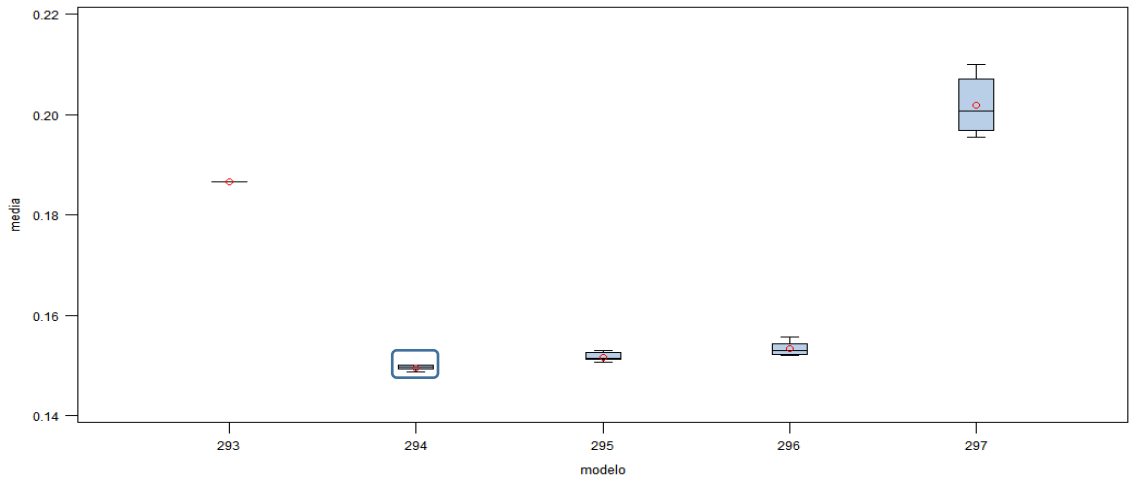


Figura 29. *Tasa de fallos de los primeros modelos GBM con las variables del Grupo 1.*

El mejor modelo mostrado en la figura 29 es el 294, que tendría una constante de regularización de 0,01 y 300 árboles. Vamos ahora a investigar más modelos similares por si pudiésemos mejorarlo cambiando el número de árboles a 200 (modelo 298) y a 400 (modelo 299).

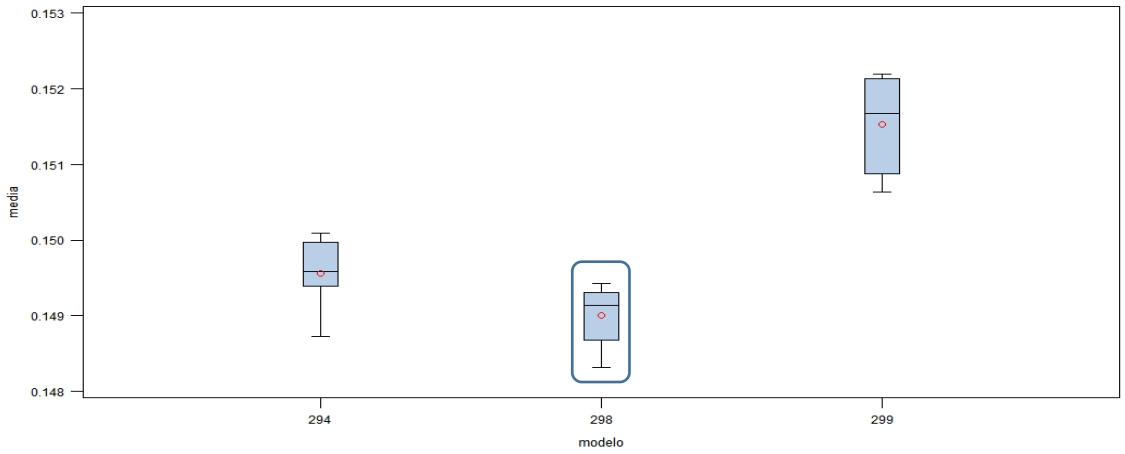


Figura 30. *Tasa de fallos de los segundos modelos GBM con las variables del Grupo 1.*

El ganador, hasta el momento, es aquel que cuenta con 400 iteraciones (Figura 30). Modificamos la profundidad del árbol a 3 (modelo 300), a 4 (modelo 301) y a 6 (modelo 302).

Tasa de fallos:

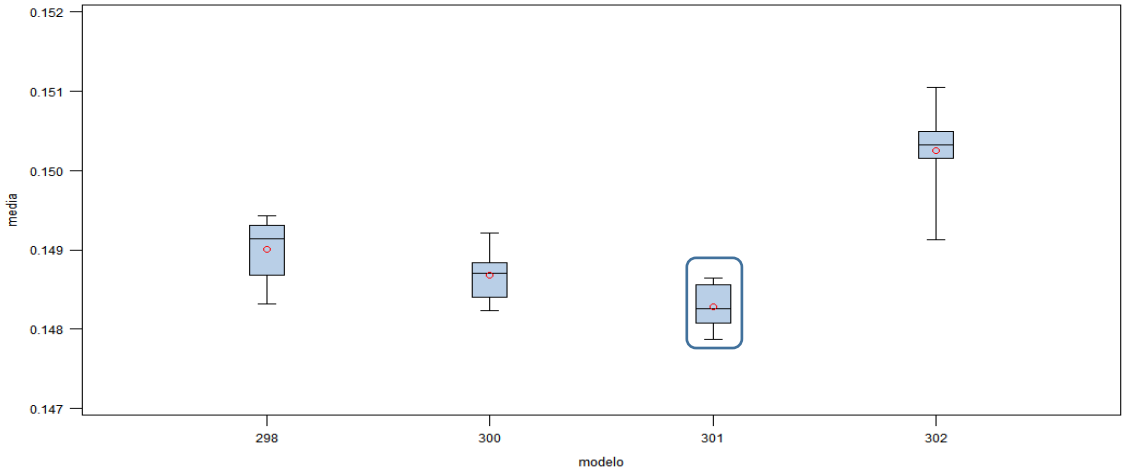


Figura 31. *Tasa de fallos de los terceros modelos GBM con las variables del Grupo 1.*

La profundidad óptima es 3 (Tabla 31), por lo que es la que configuraremos en el paso final de modificación del parámetro mincatsize y minobs: modelo 303, con 10 y 15, respectivamente y modelo 304, con 20 y 25.

Para este primer grupo de variables, el modelo ganador es el 301 (Figura 32), que es el que se compone de los siguientes elementos: leafsize igual a 7, 200 iteraciones, shrinkage de 0,01, máxbranch de 4 y máxima profundidad también de 4 con un mincatsize de 15 y minobs de 20.

Si nos fijamos en la escala, la mejora no es significativa por lo que no se pasará a realizar esta última fase de modificación de parámetros para los siguientes grupos de variables (véase Anexo 5, subapartado Gradient Boosting Machine).

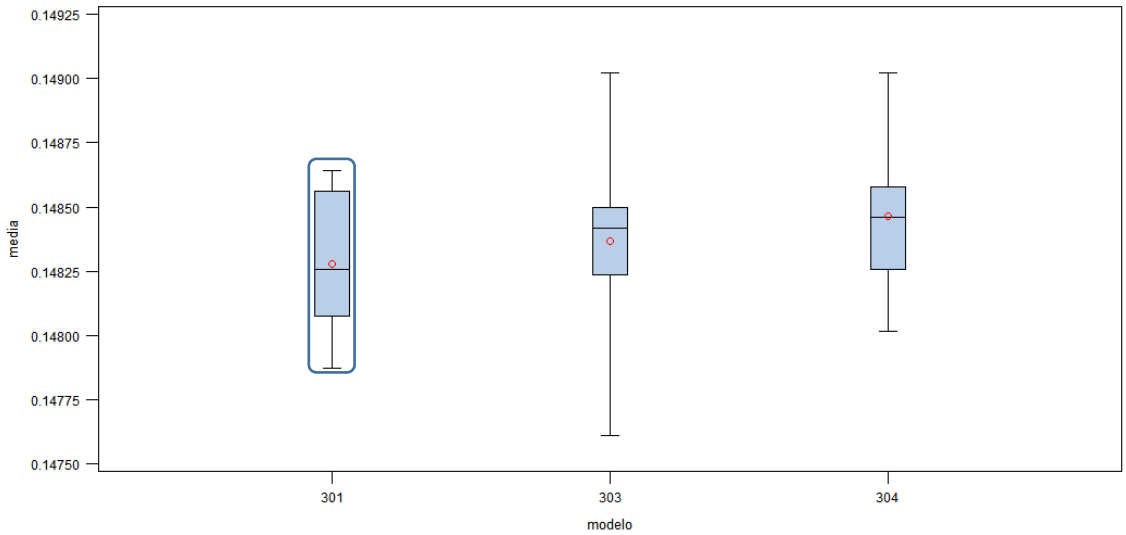


Figura 32. *Tasa de fallos de los cuartos modelos GBM con las variables del Grupo 1.*

Si nos fijamos en la escala, la mejora no es significativa por lo que no se pasará a realizar esta última fase de modificación de parámetros para los siguientes grupos de variables (véase Anexo 5, subapartado Gradient Boosting Machine).

Comparativa de mejores modelos Gradient Boosting Machine

Se procede a realizar la comparación de los modelos resultantes de cada grupo de variables que siguen la configuración de la tabla 17.

En la figura 33, se observa que el modelo 301 de GBM está conformado por las variables del grupo 1, un tamaño de hoja de 7, una constante de regularización de 0,01, un mínimo número de observaciones de la variable categórica (mincatsize) de 15 y un mínimo número de observaciones para dividir un nodo de 20.

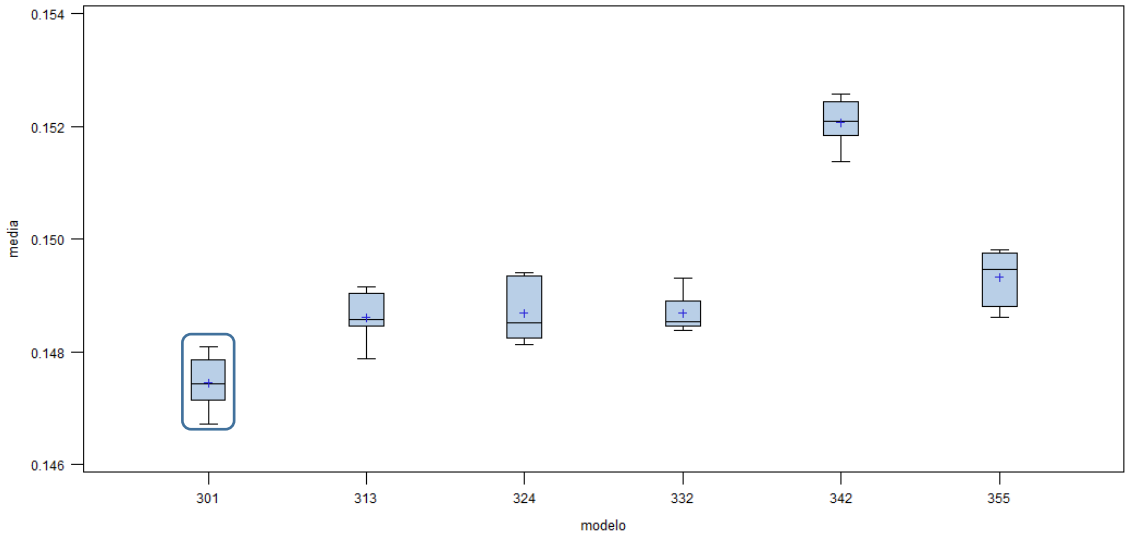


Figura 33. *Comparativa de los mejores modelos de cada grupo usando GBM.*

Tabla 17. *Híper parámetros de los mejores modelos de cada grupo en GBM.*

MODELOS	*VARIABLE*	*LEAFSIZE*	*ITERACIONES*	*SHRINK*	*MÁXIMO RAMAS*	*MÁXIMA PROFUNDIDAD*	*MINCATSIZE*	*MINOBS*
301	Grupo 1	7	200	0,01	4	4	15	20
313	Grupo 2	15	200	0,01	4	4	15	20
324	Grupo 3	15	200	0,01	4	6	15	20
332	Grupo 4	15	200	0,01	4	3	15	20
342	Grupo 1	15	200	0,01	4	3	15	20
355	Grupo 8	15	200	0,01	4	4	15	20

5.4. Support Vector Machine

Pasamos ahora a la realización del algoritmo supervisado de máquinas de vectores de soporte para las variables del grupo 1.

En una primera instancia, se configura el kernel lineal mediante el parámetro C de penalización. Este es un parámetro que puede acoger cifras muy variadas, con un rango desde 10^{-5} hasta 10^{5} aunque en este caso únicamente probamos once valores que van desde 0,001 hasta 10 (Tabla 18). Se eleva hasta ese número para tratar de reducir el posible sesgo que se pudiera dar.

Tabla 18. *Resumen de híper parámetros a utilizar en SVM Lineal en el Grupo 1.*

MODELOS	***356***	***357***	***358***	***359***	***360***	***361***	***362***	***363***	***364***	***365***	***366***
KERNEL	Lineal	Lineal	Lineal	Lineal	Lineal	Lineal	Lineal	Lineal	Lineal	Lineal	Lineal
C	0,001	0,01	0,05	0,1	0,2	0,5	1	2	5	10	100

Se procede a realizar una gráfica donde se intentan obtener patrones y agudizar la búsqueda para obtener el kernel donde el *accuracy* es más alto. En concreto, los algoritmos a estudiar serán los que se encuentran en la figura 34:

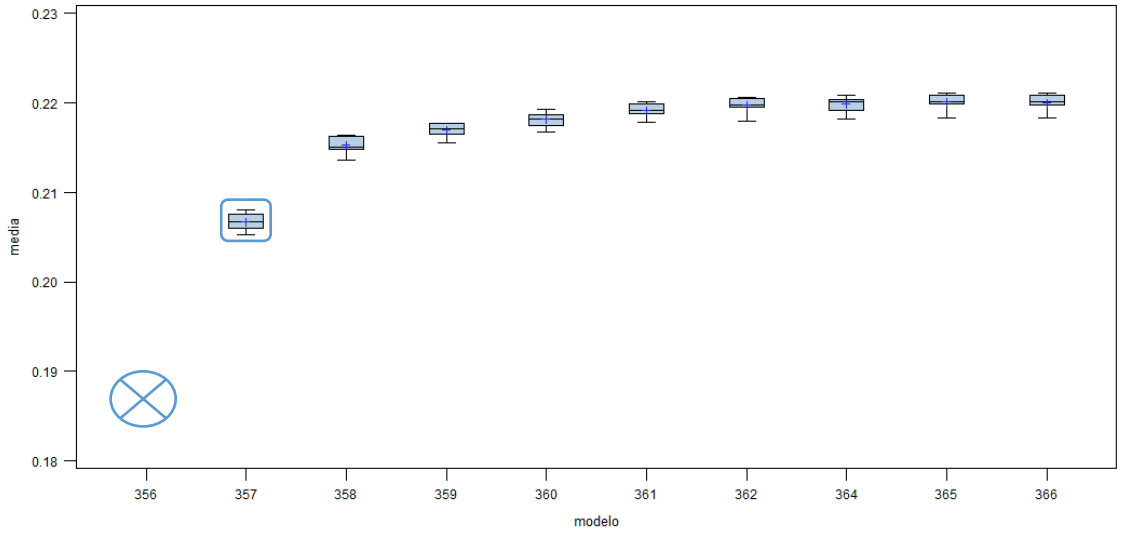


Figura 34. *Tasa de fallos de los modelos SVM lineales con las variables del Grupo 1.*

Se elige el modelo 357 debido a que el anterior, no es válido pues se debe a una configuración errónea de parámetros. Este modelo se obtiene introduciendo las variables del grupo 1 e introduciendo un parámetro de penalización de 0,01. Observamos que con un kernel pequeño se puede obtener una gran *accuracy.*

El segundo kernel que veremos es el polinomial, donde estableceremos un parámetro C exactamente igual kernel lineal y pondremos 2 y 3 grados al polinomio. Resumidamente, se realizarán los modelos configurados como aparece en la Tabla 19.

Tabla 19. *Resumen de híper parámetros a utilizar en SVM Polinomial en el Grupo 1.*

MODELOS	*KERNEL*	*C*	GRADOS DEL POLINOMIO	K PAR
367	Polinomial	0,001	2	1
368	Polinomial	0,01	2	1
369	Polinomial	0,05	2	1
370	Polinomial	0,1	2	1
371	Polinomial	0,5	2	1
372	Polinomial	0,001	3	1
373	Polinomial	0,01	3	1
374	Polinomial	0,05	3	1
375	Polinomial	0,1	3	1
376	Polinomial	0,5	3	1

No es posible usar SVM polinomial y RBF por limitaciones computacionales. Para solucionar esto, se carga una muestra estratificada y se trabaja con ella. Se tendrá en cuenta en los resultados al comparar los modelos SVM. El mejor modelo usando el kernel polinomial para las variables del grupo 1 (Figura 35), es el 372, que se configura con un valor de penalización de 0,001 y tres dimensiones.

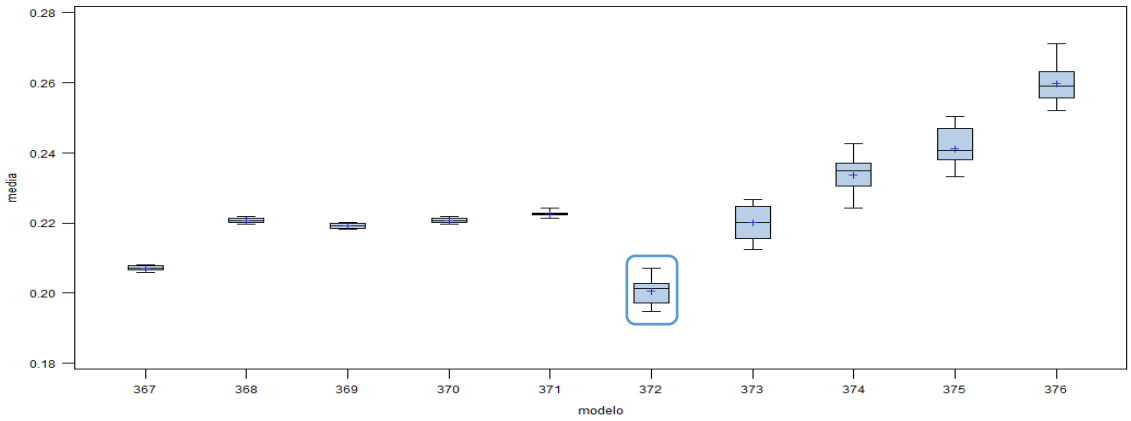


Figura 35. *Tasa de fallos de los modelos SVM polinomiales con las variables del Grupo 1.*

Por último, veremos modelos que derivan de la utilización del Kernel RBF donde se configurarán modelos que tomen los siguientes valores de la Tabla 20.

Aunque similares en resultados, en la Figura 36 se aprecia que el modelo 388 muestra una menor media en tasa de fallo por lo que se muestra como candidato del primer grupo de variables en RBF, de pasar a la comparativa SVM que se realizará seguidamente. Este modelo tiene un parámetro C de 10 y una sigma de 3.

Tabla 20. *Resumen de híper parámetros a utilizar en SVM RBF en el Grupo 1.*

MODELOS	*KERNEL*	*C*	GRADOS DEL POLINOMIO	K PAR
383	Polinomial	0,01	2	1
384	Polinomial	0,1	2	1
385	Polinomial	1	2	1
386	Polinomial	10	2	1
387	Polinomial	1	3	1
388	Polinomial	10	3	1

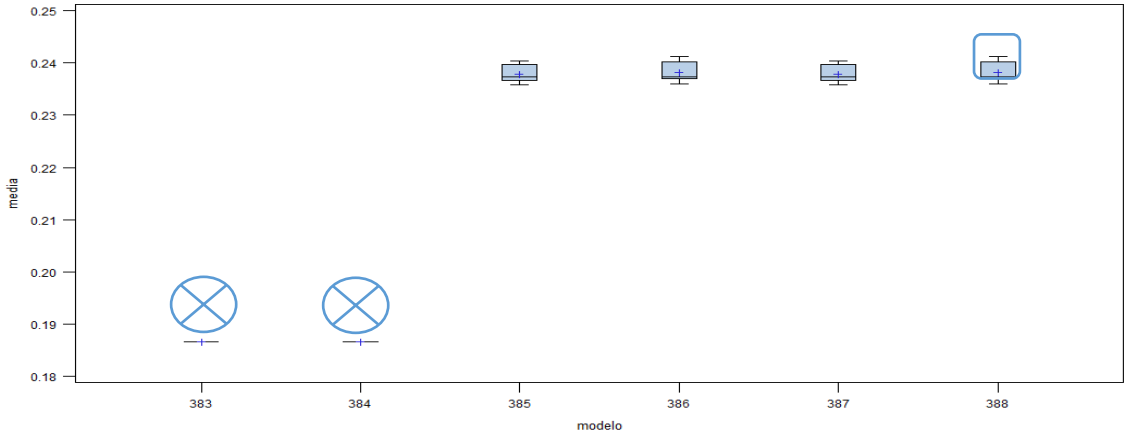


Figura 36. *Tasa de fallos de los modelos SVM RBF con las variables del Grupo 1.*

Comparativa de mejores modelos Support Vector Machine

Debido a la exigencia de procesamiento de este algoritmo y las limitaciones que esto conlleva, por esta vez y sin que sirva de precedente no se realizarán los mismos modelos para el resto de los grupos de variables. Se cogerán los mejores para cada kernel (lineal, polinomial y RBF) y se analizarán con los mismos parámetros para el resto de los grupos, dando lugar a tres modelos por grupo, que analizaremos en la Figura 37:

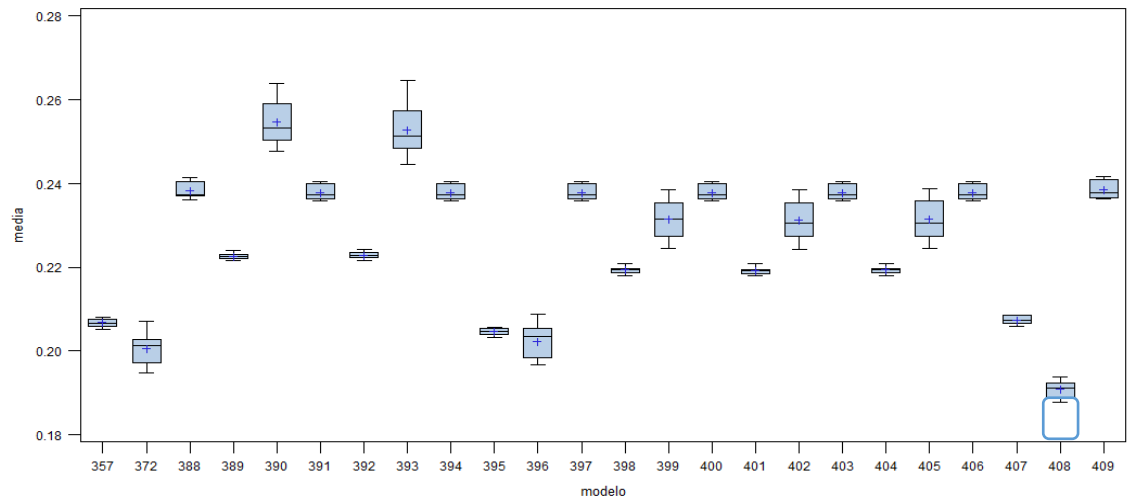


Figura 37. *Comparativa de los mejores modelos de cada grupo usando SVM.*

El modelo 408, que se configura con las variables de grupo 8 utiliza un kernel polinomial de 3 grados y un parámetro C de 0.001.

5.5. Comparativa final de mejores modelos

Hallados los modelos que presentaban una tasa de fallos más baja para cada algoritmo analizado, se pasa ahora a compararlos entre sí y ver cuál de ellos es el mejor de todos los estudiados con el programa SAS Base (Figura 38).

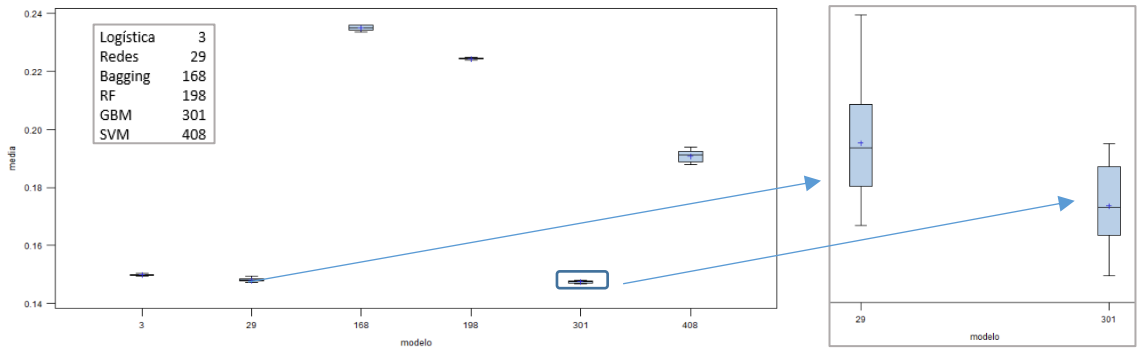


Figura 38. *Comparativa final de los mejores modelos de cada grupo en SAS Base.*

Los modelos que mejor funcionan para este conjunto de datos son la logística, las redes neuronales binarias, así como el compuesto por el algoritmo GBM. Todos ellos tienen bajo sesgo y baja varianza; las cualidades deseadas a la hora de predecir. Los tres presentan resultados similares, aunque al ampliar la imagen se aprecia que es el último de los tres, el 301 (GBM) el que menor tasa de fallos presenta.

Tanto el modelo de logística como el de redes están compuesto por variables del tercer grupo mientras que las variables dependientes del tercero son las del conjunto 1. Esto es entendible se tenemos en cuenta que los árboles de decisión, por lo que actúan mejor cuando se les da más opciones. Se comprueba la teoría que afirma que diferentes agrupaciones de variables pueden presentar similares conclusiones si se tratan de diferente manera.

Bagging y *Random Forest* no logran desafiar al resto de modelos. Tampoco lo hace *Support Vector Machine* ni con su versión lineal, que se supone que es competidor directo de la regresión logística. Se concluye que el modelo ganador es el mejor de *Gradient Boosting Machine*, que combina variables del grupo 1, tamaño mínimo de la hoja de 7, una constante de regularización de 0,01, mínimo número de observaciones de la variable categórica (mincatsize) de 15 y mínimo número de observaciones de 20.

Observamos en la salida de PROC MEANS de SAS como la media de fallos es de un valor menor a 0.15, con una desviación estándar relativamente baja. De hecho, los valores mínimo y máximo de la validación cruzada repetida son 0,1467 y 0,1481, respectivamente (Figura 39).

Procedimiento MEANS

Variable	N	Media	Dev std	Mínimo	Máximo
suma	6	0.5897958	0.0019682	0.5868886	0.5923800
media	6	0.1474489	0.000492060	0.1467221	0.1480950
semilla	6	13347.50	1.8708287	13345.00	13350.00
modelo	6	301.0000000	0	301.0000000	301.0000000

Figura 39. *Procedimiento Means del modelo ganador.*

6. Modelos realizados en RStudio

En este programa también se compararán todos los modelos con validación cruzada repetida de cuatro grupos, pero esta vez, mostrando los resultados tanto de área bajo la curva ROC como de tasa de fallos en gráficos de cajas que nos permitan ver a simple vista qué modelo de los que estudiaremos presenta mejores resultados.

Para la ejecución de estos modelos, se utilizará la librería "Caret" y para ello, es aconsejable tratar los datos previamente pasando variables categóricas a *dummies* y estandarizando aquellas continuas. Se incluyen *kdummies*-1 para sortear el problema de la multicolinealidad.

Previo a la realización de modelos de predicción en el programa R, se tunean parámetros para evitar la realización de un gran número de modelos sin criterio alguno que puedan estar lejos de obtener buenos resultados o generen problemas de sobreajuste.

Debido a la gran exigencia computacional que demanda la búsqueda de híper parámetros, se procede a realizar particiones y remuestreo estratificado, que únicamente se usa para hallar estos parámetros (los modelos se realizan con todos los datos).

6.1. Regresión logística

El primer algoritmo a utilizar, al igual que se ha realizado en SAS Base, es una regresión logística. Se realiza un modelo por cada grupo de variables, por lo que tenemos los siguientes ocho que se muestran en la Figura 40.

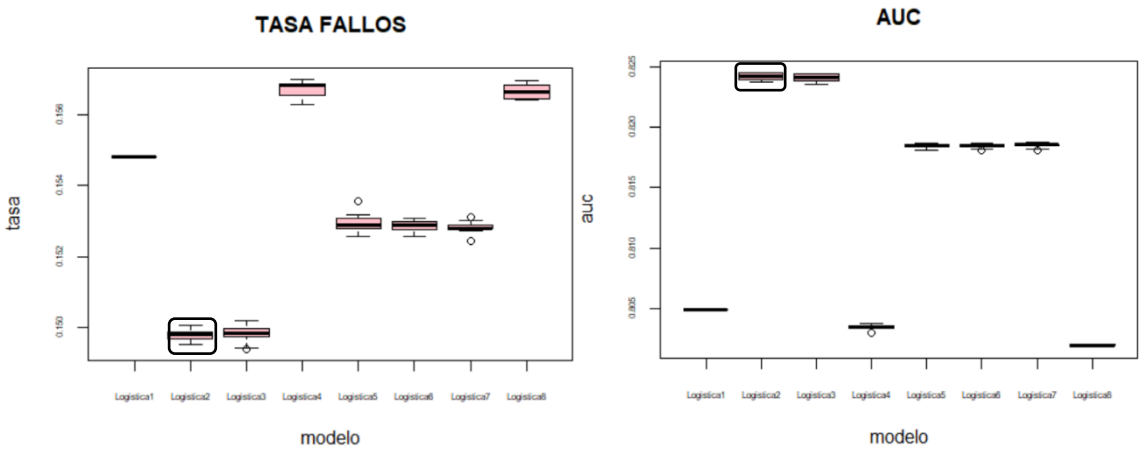


Figura 40. *Tasa de fallos y AUC en modelos de Regresión Logística.*

El modelo por excelencia, en este caso, es claramente Logística2, que se compone de las variables del grupo dos. Es el que muestra un AUC mayor (0,824) así como una menor tasa de fallos. Logística3, es un buen competidor, aunque muestra unos resultados ligeramente peores en ambos aspectos.

6.2. *Árboles*

Se tiene en cuenta este algoritmo básico, aunque a priori sabemos que proporcionará peores resultados que aquellos que combinan una gran cantidad de árboles, como son *Bagging* o *Random Forest*. Para la consecución de los siguientes resultados, se han llevado a cabo ocho modelos; uno por cada grupo de variables. Los analizamos en términos de varianza y sesgo en los siguientes gráficos (Figura 41):

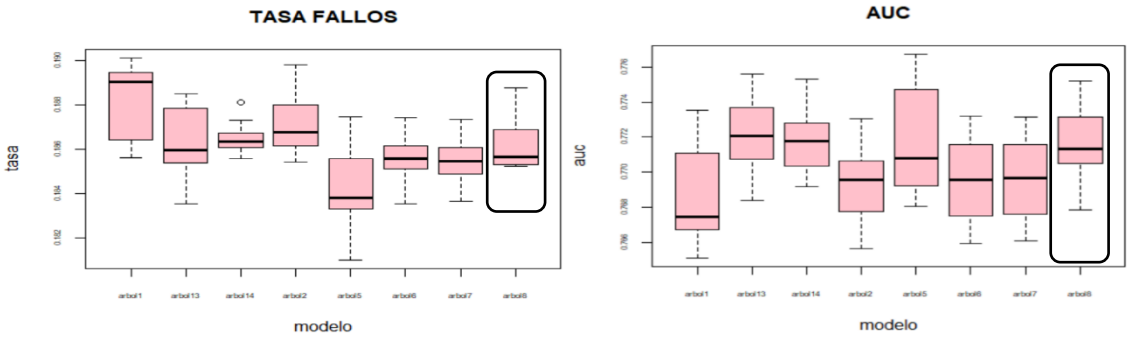


Figura 41. *Tasa de fallos y AUC en Árbol de Clasificación Binaria.*

Elegir un árbol óptimo observando los gráficos puede generar división de opiniones, ya que va a depender de lo que se busque. Destaca árbol8, compuesto por las variables del grupo 8, al encontrarse entre los que mayor área tienen bajo la curva ROC (*accuracy* = 0.8138081) así como una tasa de fallos no muy elevada. No es posible decantarse por el modelo 5 debido a su gran variabilidad. Árbol13 presenta resultados similares al seleccionado, aunque con mayor tasa de fallos.

6.3. *Redes neuronales*

Intentar encontrar la red neuronal binaria perfecta es más sencillo usando este programa que el anterior, ya que se puede refinar esa búsqueda mediante la configuración de los híper parámetros en las rejillas y evitar así probar innumerables modelos.

Para ello, se utiliza la librería CARET y, en concreto, la función AVNNET, que permite validación cruzada, así como comparar el mejor número de nodos posible (*size*), cuyo valor máximo será el determinado previamente en el análisis de nodos previamente realizado (véase página 31). También se puede ajustar la regularización (*decay*), que ayuda a evitar el sobreajuste y que determinaremos con valores que van desde el 0,001 hasta el 0,2.

Se realiza, por tanto, el estudio para los distintos grupos con los que contamos y la salida del programa, que nos indica con qué valores de los indicados se obtendría una mayor área bajo la curva ROC, especifica las siguientes cifras:

Grupo 1: `The final values used for the model were size = 4, decay = 0.2 and bag = FALSE.`
Grupo 2: `The final values used for the model were size = 3, decay = 0.1 and bag = FALSE.`
Grupo 3: `The final values used for the model were size = 4, decay = 0.2 and bag = FALSE.`
Grupo 4: `The final values used for the model were size = 4, decay = 0.2 and bag = FALSE.`
Grupos 5, 6 y 7*: `The final values used for the model were size = 4, decay = 0.1 and bag = FALSE.`
Grupo 8: `The final values used for the model were size = 4, decay = 0.2 and bag = FALSE.`

Configurando los parámetros que recomienda R Studio, surgen los siguientes modelos, que analizaremos en los dos siguientes gráficos:

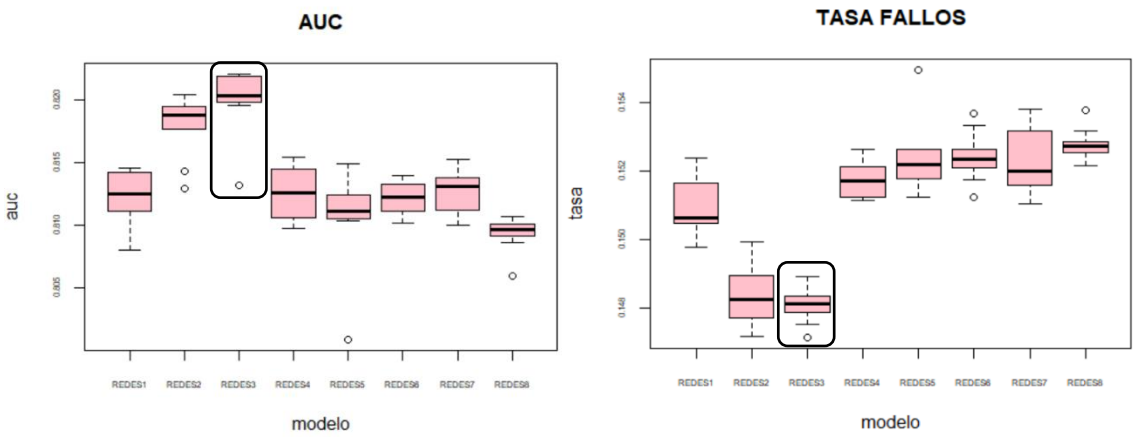


Figura 42. *Tasa de fallos y AUC en Redes Neuronales Binarias.*

La elección del modelo REDES3 no da lugar a dudas puesto que, aunque cuenta con un valor atípico muy por debajo del rango donde se encuentra en términos de exactitud, es el que cuenta tanto con un mayor AUC como con una menor tasa de fallos.

6.4. Bagging

Como se ha mencionado previamente, este algoritmo logra superar las debilidades de los árboles simples, que son el alto sesgo y la alta varianza.

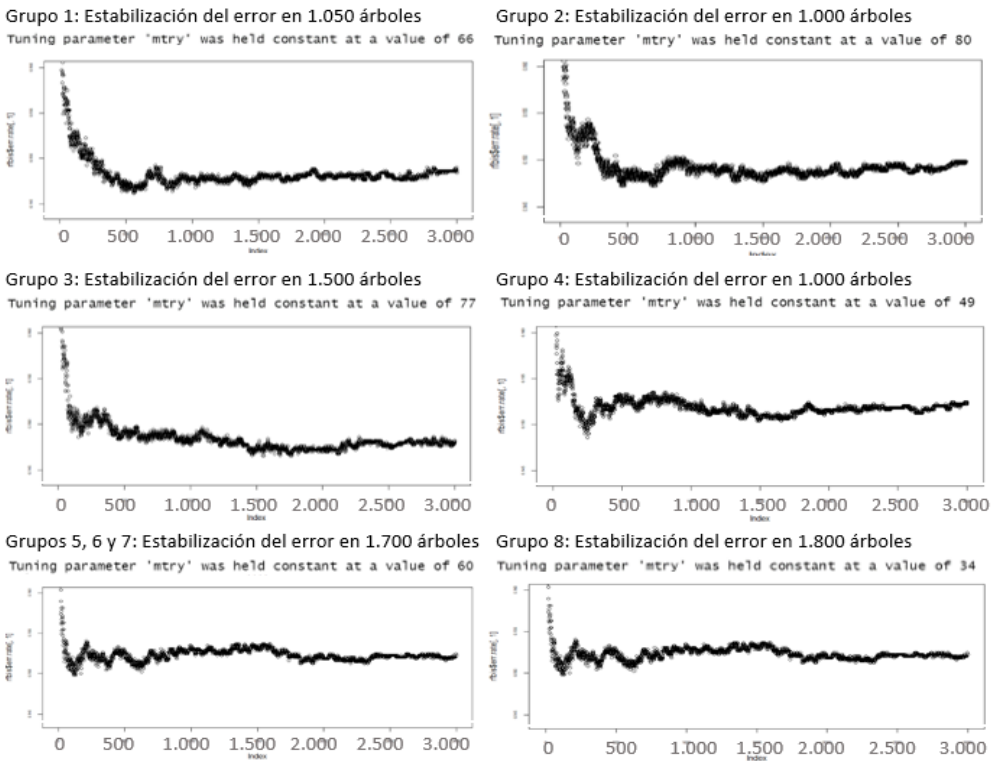


Figura 43. *Estudio de número de árboles óptimo en Bagging.*

Para ello, se realiza un estudio previo a la realización de los algoritmos con el método "rf" de librería CARET, donde se especifica el número de variables a sortear en cada nodo (*Mtry*) con el total de cada grupo, un tamaño mínimo de nodos finales (*nodesize*) fijo de 25 que evite el sobre ajuste y un tamaño de la muestra (*sampsize*) de 2.500.

Con esa configuración, se procede a determinar ahora el tamaño del modelo realizando un estudio sobre el número de árboles (*ntree*) a partir del cual, aumentos de iteraciones no generen mejoras en el resultado. Los resultados para las distintas categorías de variables, son los que aparecen en la Figura 43.

Conocido el número de árboles adecuado donde se estabiliza el error, se pasa a realizar los modelos: se realizan dos modelos por cada grupo de variables donde, para el primero, se utilizará el reemplazamiento a la hora de seleccionar la muestra (replace=TRUE) y, un segundo, donde esto no se hará (replace=FALSE).

BAGNº.1 = Utilizando parámetros óptimos y con reemplazamiento.
BAGNº.2 = Utilizando parámetros óptimos y sin reemplazamiento.

Como se ha ido haciendo hasta ahora, sacamos en dos gráficos los resultados:

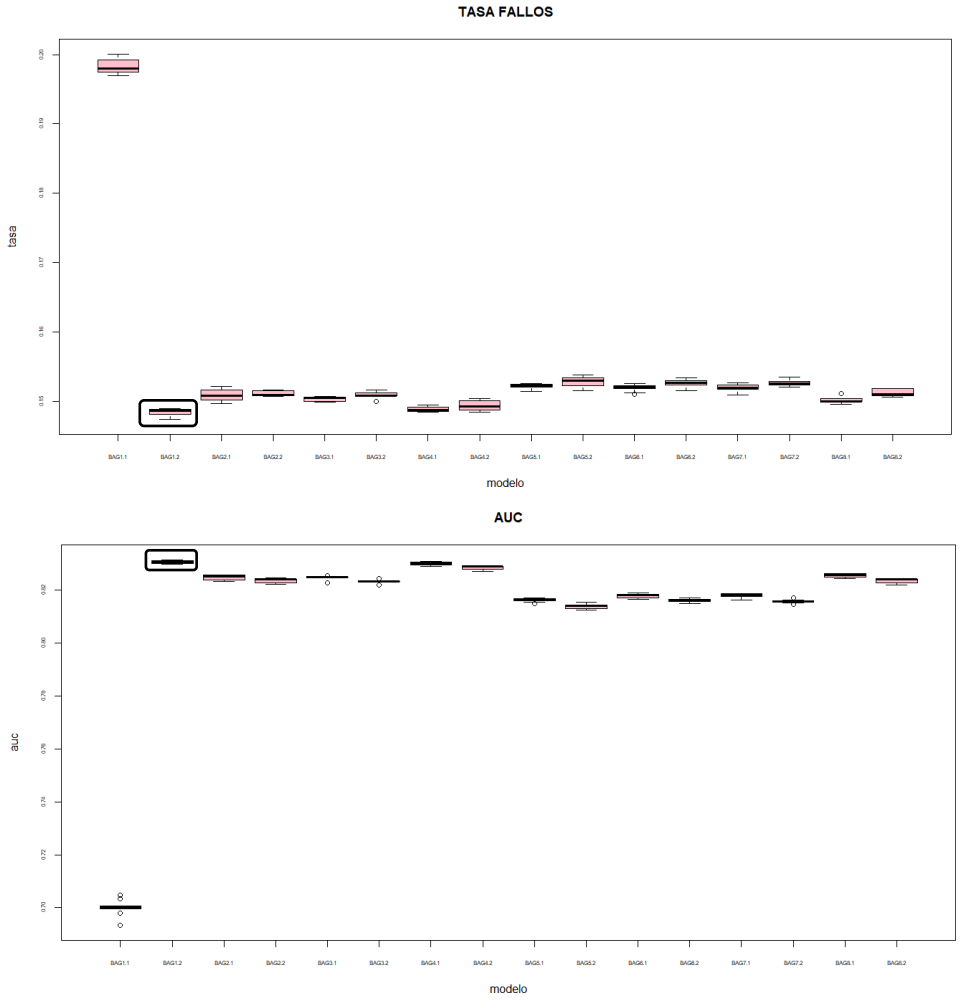


Figura 44. *Comparativa de los modelos de Bagging.*

El modelo *Bagging* elegido*, BAG1.2, que se* compone de las variables del primer grupo y se construye con un Mtry de 66 variables, 1.050 árboles y que no usa reemplazamiento al seleccionar la muestra, es el que mayor área bajo la curva ROC tiene (*accuracy*=0.8515839), así como una menor tasa de fallos.

6.5. *Random Forest*

El algoritmo *Random Forest* es una versión mejorada de *Bagging*, puesto que evita que se generen árboles semejantes cuando existen variables dominantes y, en lugar de esto, cree conjuntos muy diversos. Esto ocurre gracias a que el número de variables a incluir no es del total de variables.

Para ello, utilizando el método "rf" de la librería CARET, se configuran parámetros de la misma manera que con el algoritmo anterior (*nodesize* = 25 y *sampsize* = 2.500), aunque esta vez, no se fijará un número de variables determinado a incluir en el Mtry, sino que en la rejilla se introducirán todas las variables posibles desde 1 hasta N y se sorteará el número que nos recomiende el programa.

Los resultados son los siguientes (Figura 45):

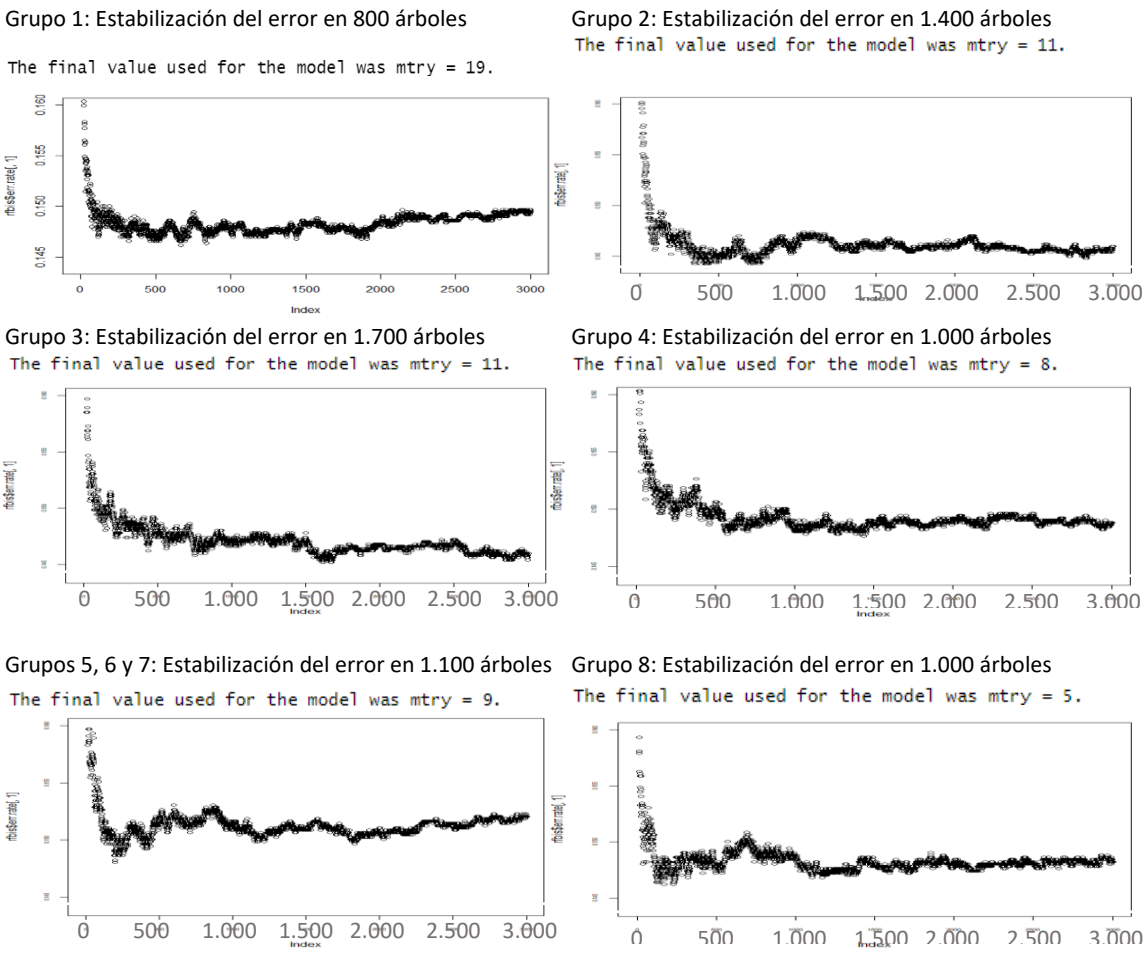


Figura 45. *Estudio de número de árboles óptimo en RF.*

Obtenidos los Mtry que mejor resultado presentan y analizado el número de árboles idóneo que no sobreajuste, pasamos a construir los modelos que, al igual que con *Bagging*, serán dos por cada grupo de variables; con y sin reemplazamiento:

RFNº.1 = Utilizando parámetros óptimos y con reemplazamiento.
RF2Nº.2 = Utilizando parámetros óptimos y sin reemplazamiento.

Comparamos los modelos:

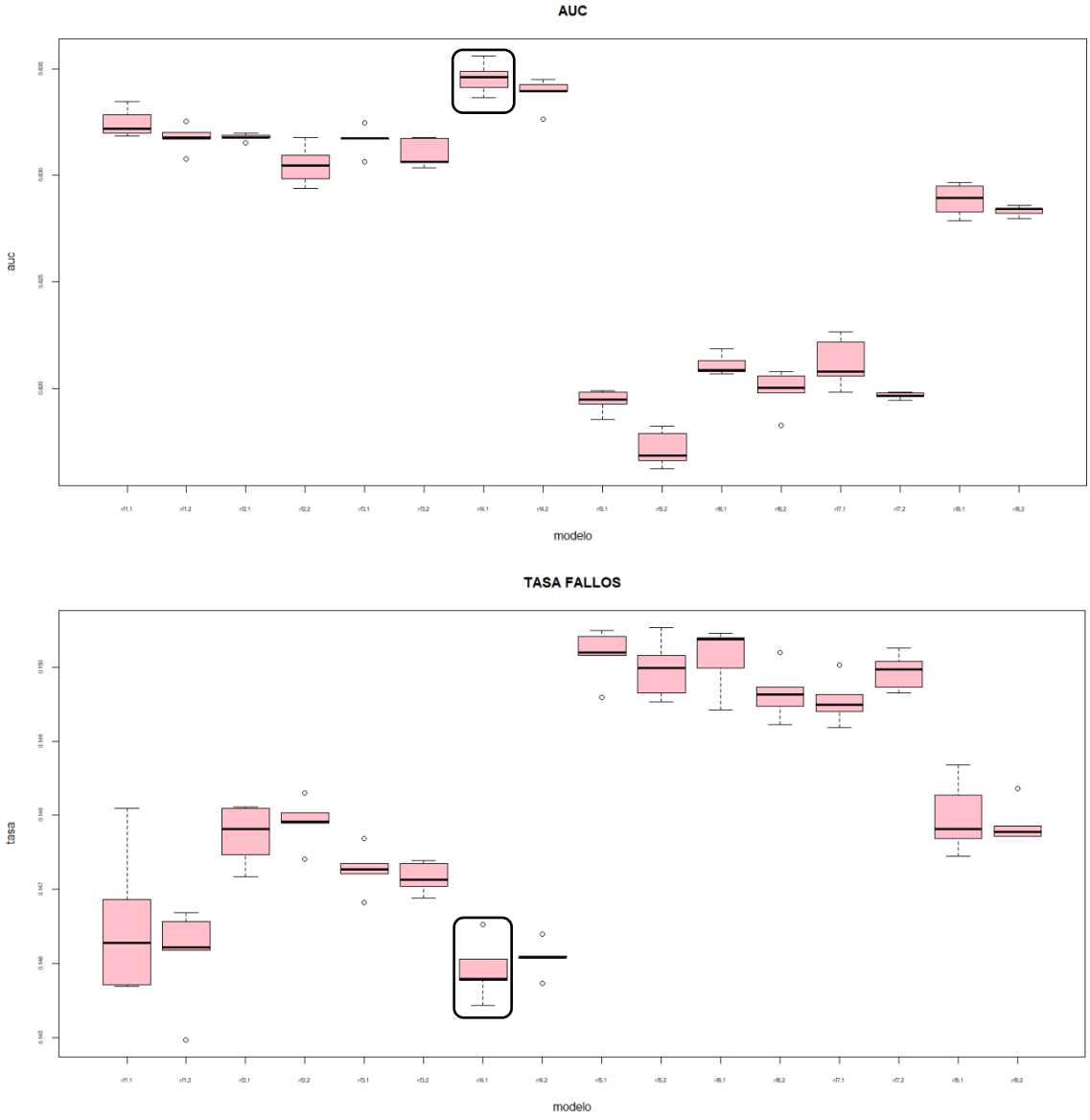


Figura 46. *Comparativa de los modelos de RF.*

El mejor de los modelos analizados en este sub-apartado el RF4.1 (Figura 46), formado por las variables del grupo 4, ya que logra un *accuracy* de 0.8540855, sorteando 8 variables y con 1.000 árboles; el *nodesize* es de 25 y el *sampsize* de 2.500.

6.6. *Gradient Boosting Machine (GBM)*

Este algoritmo, al igual que los dos anteriores, cuenta con las ventajas de los árboles, aunque los mejora dado a que es más agresivo cuando hablamos de reducción de error.

Para evitar sobre ajustar, se va a realizar un estudio previo mediante el método GBM de la librería CARET, tanto del parámetro de regularización (*shrinkage*) para diferentes números de árboles y distintos tamaños mínimos de nodos finales (*n.minobsnode*).

Se estudia, para los diferentes conjuntos de variables, que la constante de regularización varíe entre 0,001 hasta 1, para hallar la velocidad óptima a la que varían las predicciones. Se tiene en cuenta que a menor *shrinkage*, es necesario un mayor número de iteraciones, por lo que se introducen en la rejilla un tramo desde 100 a 3.000 árboles y un tamaño mínimo de nodos finales de 15 y 25.

Grupo 1:

Se valoran las interrelaciones entre híper parámetros y vemos si existen patrones (gráfico de la izquierda) y una vez sabemos ese punto, se especifican los parámetros y, mediante el uso de una rejilla con árboles que rondan el número que recomienda el programa, se realiza de nuevo un estudio de iteraciones más favorable (gráfico de la derecha).

```
The final values used for the model were n.trees = 500, interaction.depth = 2, shrinkage
 = 0.05 and n.minobsinnode = 15.
```

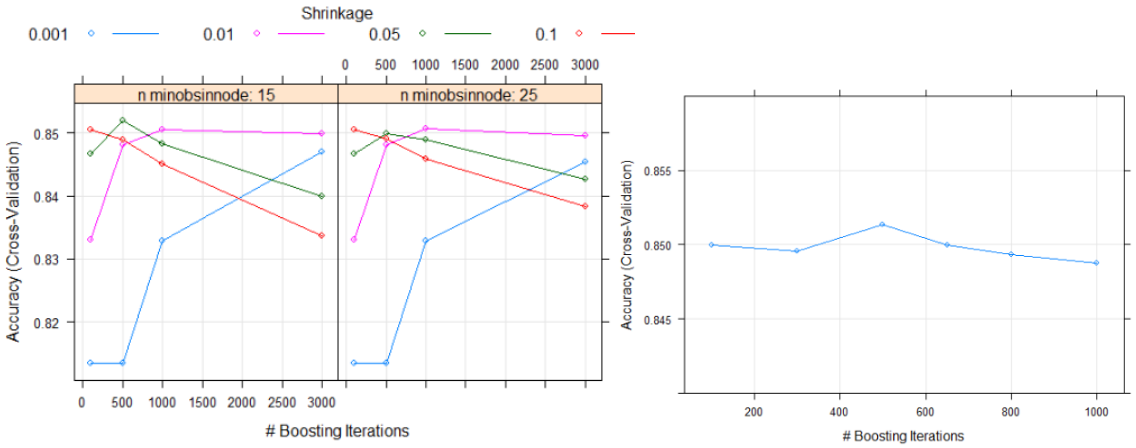


Figura 47. *Estudio de híper parámetros en GBM con variables del grupo 1.*

Se aprecia cómo, para este caso, el uso de más árboles no supone una mayor *accuracy* y que la velocidad de regularización óptima es de 0,05 ya que valores superiores a esta hacen que la exactitud disminuya. En concreto, el punto que nos recomienda R, donde el *accuracy* es máximo, es el siguiente el que tiene un número de árboles de 500, una constante de regularización de 0,05 y un tamaño mínimo de nodos de 15.

Mediante el estudio de *early stooping* (gráfico de la derecha), se comprueba que, efectivamente, el número óptimo de árboles donde la exactitud es mayor es de 500. El análisis del resto de grupos se podrá visualizar en el Anexo VI, subgrupo Gradient Boosting Machine.

Una vez obtenidos los parámetros que ofrecen mejores resultados de cada grupo de variables, pasamos a configurar los mismos y a compararlos mediante validación cruzada repetida de 4 grupos:

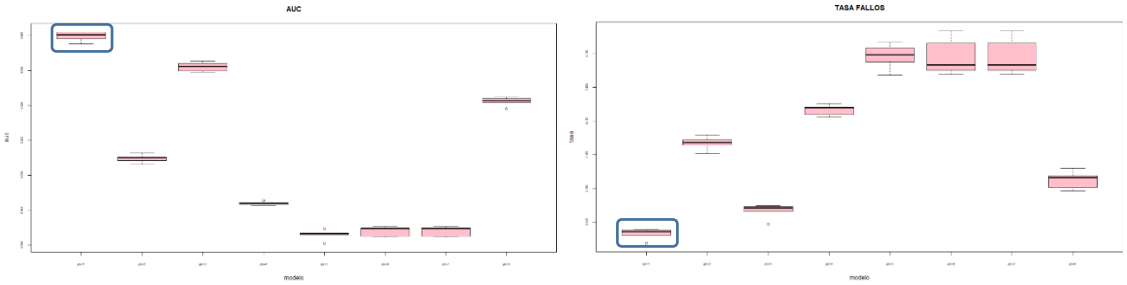


Figura 48. *Comparativa de los modelos de GBM.*

El modelo *Gradient Boosting Machine* que supera al resto según el criterio seguido, es el GBM1, ya que es el que mayor área bajo la curva ROC presenta (0.8513477) y menor tasa de fallos, además de ser el que menos variabilidad tiene. Está compuesto por las variables del primer grupo y ajustado con los siguientes parámetros: Velocidad de regularización de 0,05, tamaño mínimo de nodos de 15 y 500 iteraciones.

6.7. *Extreme Gradient Boosting Machine (XGBOOST)*

Este algoritmo es una versión mejorada del anterior ya que incluye penalización de las hojas. Al igual que se ha realizado en el caso anterior, lo primero que se llevará a cabo será un análisis a través del método "xgbTree" de la librería CARET, que permite la validación cruzada, para determinar la velocidad de cambio de la predicción (*shrinkage*), así como el número de árboles y el tamaño mínimo de nodos.

Grupo 1:

```
The final values used for the model were nrounds = 100, max_depth = 6, eta = 0.03, gamma =
 0, colsample_bytree = 1, min_child_weight = 20 and subsample = 1.
```

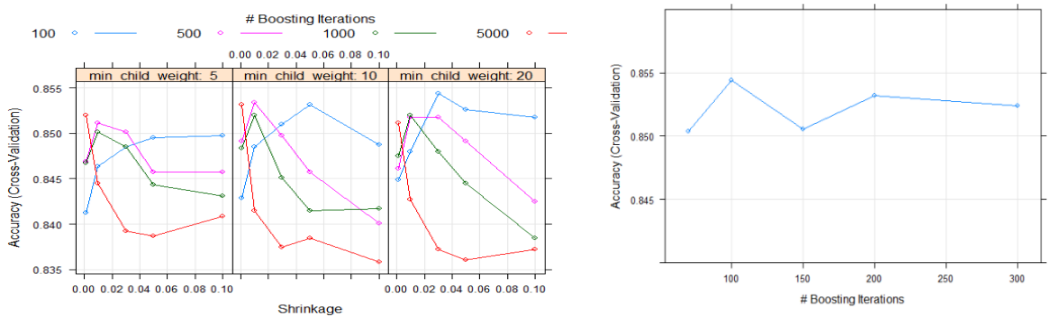


Figura 49. *Estudio de híper parámetros en XGBM con variables del grupo 1.*

Utilizamos los híper parámetros que recomienda el programa, que son los que maximizan el *accuracy*. El análisis del resto de grupos se podrá visualizar en el anexo VI.

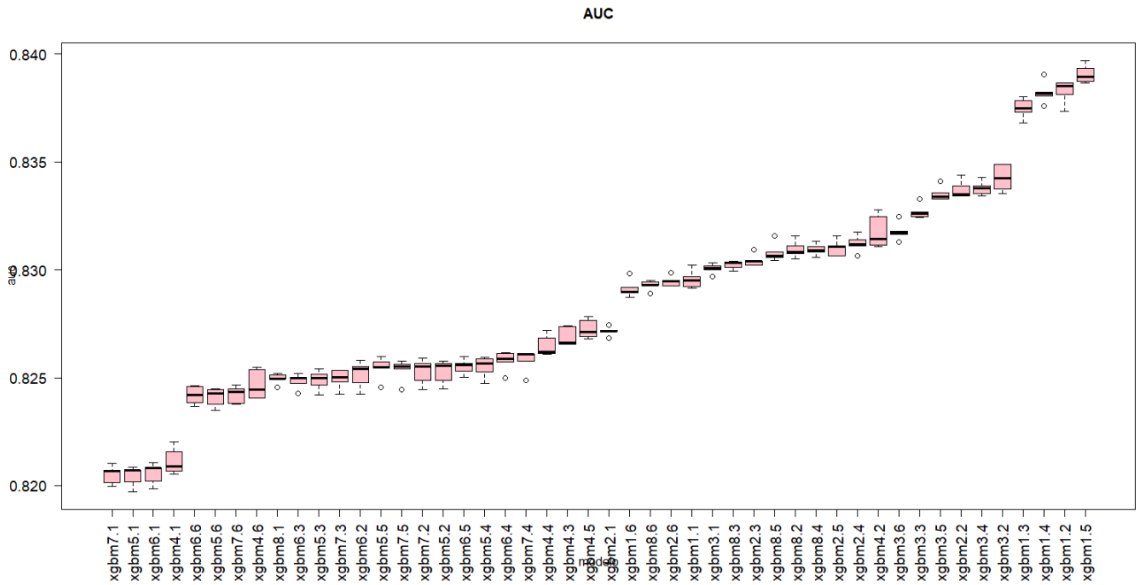
Una vez tenemos los híper parámetros que hacen óptimo al algoritmo en términos de maximización de *accuracy*, se procede a realizar varios modelos para grupo de variables. En concreto, primero se analiza el algoritmo configurándolo con aquellos valores obtenidos previamente. Seguidamente, se estudia otro XGBM añadiendo los parámetros óptimos hallados anteriormente en *Gradient Boosting Machine*.

Por otra parte, con el empeño de reducir la varianza, se analizarán otros tres modelos a los cuales se les especifiquen los mismos valores que en el primero de este conjunto, pero esta vez sorteando variables, de forma similar a lo que se hacía en *Random Forest*, sorteando variables, al estilo *Bagging*, o sorteando ambos puntos. Para esto, se tiene en cuenta que es necesario aumentar el número de árboles.

Finalmente, se crea un modelo donde se añade regularización. En resumen, se probarán los 6 siguientes modelos para cada grupo, de manera que tendremos un total de 48, derivados de hacerlos para los 8 conjuntos de variables que hemos venido estudiando hasta ahora.

XGMBNº.1: Configurando los parámetros óptimos según el estudio previamente realizado
XGMBNº.2: Se usan los parámetros óptimos del algoritmo anterior GBM para cada grupo de variables
XGMBNº.3: Parámetros óptimos del primer XGMB sorteando variables como en *Random Forest*
XGMBNº.4: Parámetros óptimos del primer XGMB sorteando muestra, como en *Bagging*
XGMBNº.5: Parámetros óptimos del primer XGMB sorteando variables y muestra
XGMBNº.6: Parámetros óptimos del primer XGMB añadiendo regularización

Representamos en un gráfico todos ellos y los ordenamos de menor a mayor para poder compararlos entre sí de una manera más sencilla y rápida:



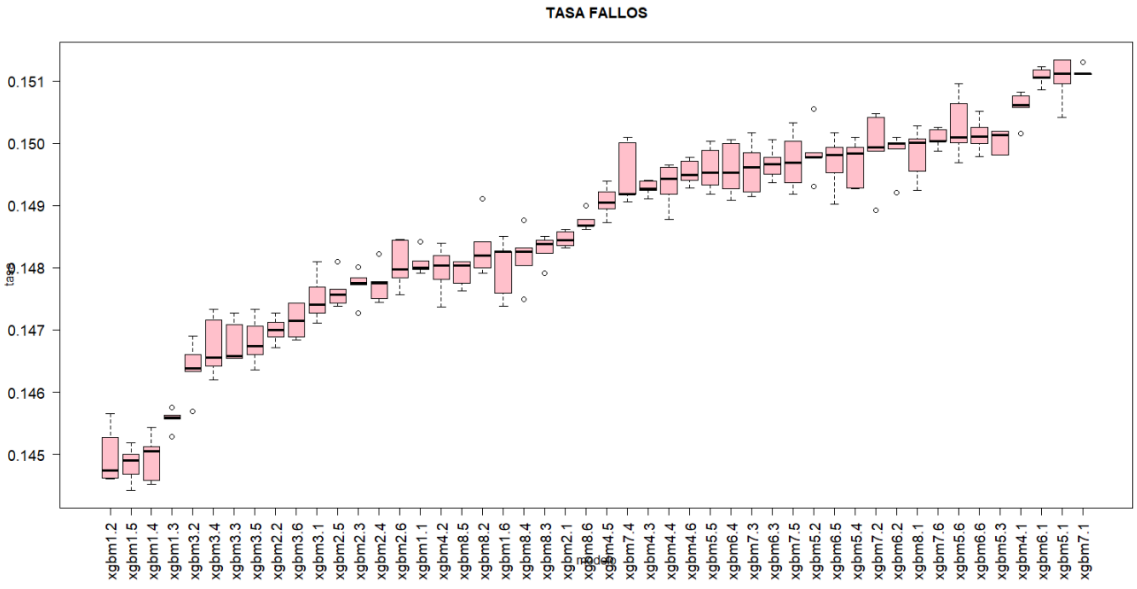


Figura 50. *Comparativa de los modelos de XGBM.*

Vemos los mejores comparados por separado: XGBM1.2, XGBM1.3, XGBM1.4 y XGBM1.5:

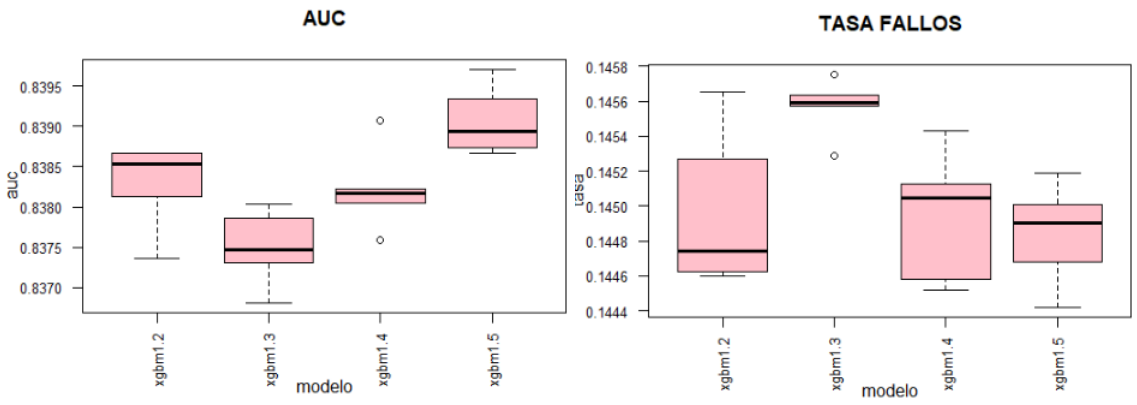


Figura 51. *Comparativa de los cuatro mejores modelos de XGBM.*

Se considera que el mejor modelo es el XGBM1.5 (Figura 51), que es el formado por el primer grupo de variables y configurado a través de parámetros óptimos de XGMB sorteando variables y muestra.

La elección de este modelo se debe a que además de ser el que mayor área bajo la curva ROC tiene, es uno de los que tiene menor tasa de fallo y menor variabilidad. El XGBM1.2 cuenta con una media de error menor, aunque el modelo es menos estable.

6.8. Support Vector Machine (SVM)

El último algoritmo que se va a utilizar en este trabajo es *Support Vector Machine*. Se trata de un algoritmo muy flexible gracias a que tiene tanto versiones lineales, que compiten con la versión logística como no lineales, que detecta otro tipo de relaciones.

Su objetivo es buscar separadores óptimos entre los puntos. Esta separación puede ser lineal o polinomial, donde se encuentran separaciones lineales en más de una dimensión, mediante el kernel polinomial o con el Kernel *Radial Basis Function*. Cuanto mayor es la dimensión, más agresivo será el modelo construido.

Como en ocasiones anteriores, previo a la realización del modelo, se buscarán los híper parámetros que logren buenos resultados para cada grupo de variables. Para ello, primero se estudia mediante el método "svmLinear" de Caret, el parámetro del Kernel lineal, configurando la rejilla con valores que van desde 0.001 hasta 50. Después comienza la búsqueda de los parámetros del SVM polinomial, con la misma rejilla del caso anterior y añadiendo grados al modelo (2 y 3). Finalmente, se configuran los parámetros del *SVM Radial Basis Function*; una constante C y sigma, que a mayor valor, mayor agresividad del modelo.

Grupo 1:

Se analiza primero el parámetro a determinar en el SVM lineal (Figura 52) mediante el siguiente gráfico, donde se observa que, para un nivel bajo de C (0,03) se obtiene un buen *accuracy* (0,84).

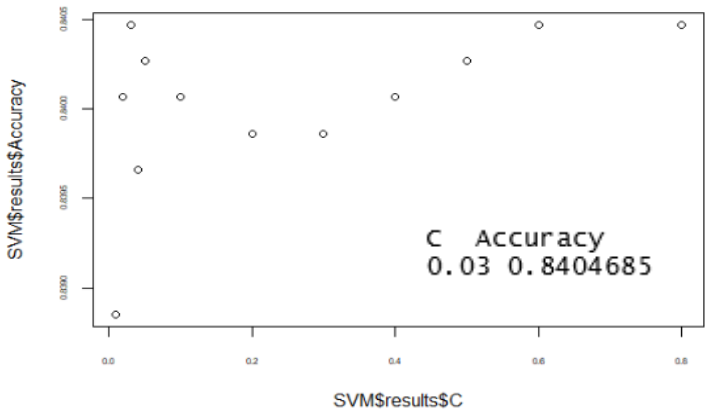


Figura 52. *Estudio de híper parámetros en SVL lineal con variables del grupo 1.*

Se analizan ahora los parámetros óptimos del SVM polinomial en el siguiente cuadro de la izquierda, así como los del SVM *Radial Basis Function* (cuadro de la derecha).

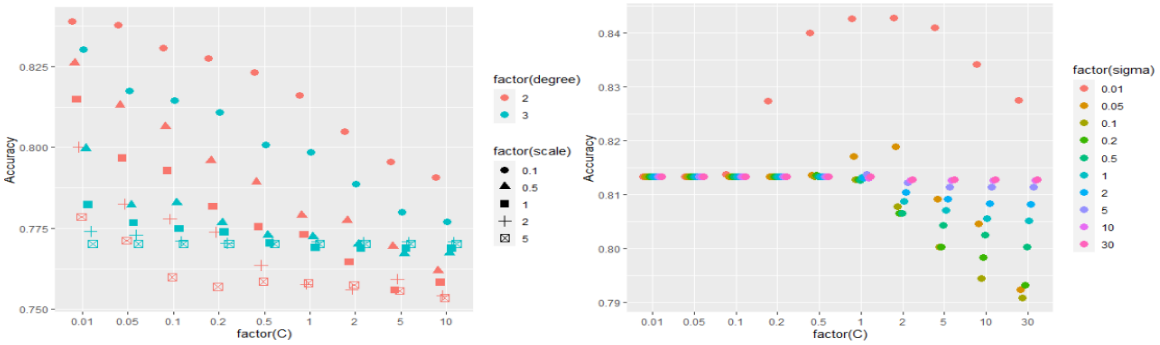


Figura 53. *Estudio de híper parámetros en SVL polinomial y RBF con variables del grupo 1.*

Con la imagen izquierda de la figura 53, analizamos patrones de comportamiento en términos de *accuracy* con varias dimensiones y diferente escala. Se aprecia como al usar dos dimensiones, una escala de 0,1 y configurando el factor C con 0.01 obtenemos el mejor dato posible de todas las combinaciones:

C	degree	scale	Accuracy	Kappa	AccuracySD	KappaSD
0.01	2	0.1	0.8390549	0.2838777	0.004727262	0.035128974

Estudiando los parámetros de kernel *Radial Basis Function* (cuadro de la derecha) vemos cómo ese punto óptimo se encuentra al fijar el factor sigma en 0.01 y el C en 2.

C	sigma	Accuracy	Kappa
2.00	0.01	0.8428918	0.3346733178

El análisis del resto de grupos se podrá visualizar en el anexo VI, en concreto, en el subapartado Support Vector Machine. Finalmente, representamos en dos gráficos los modelos que se derivan de incluir los diferentes parámetros hallados para los distintos grupos de variables:

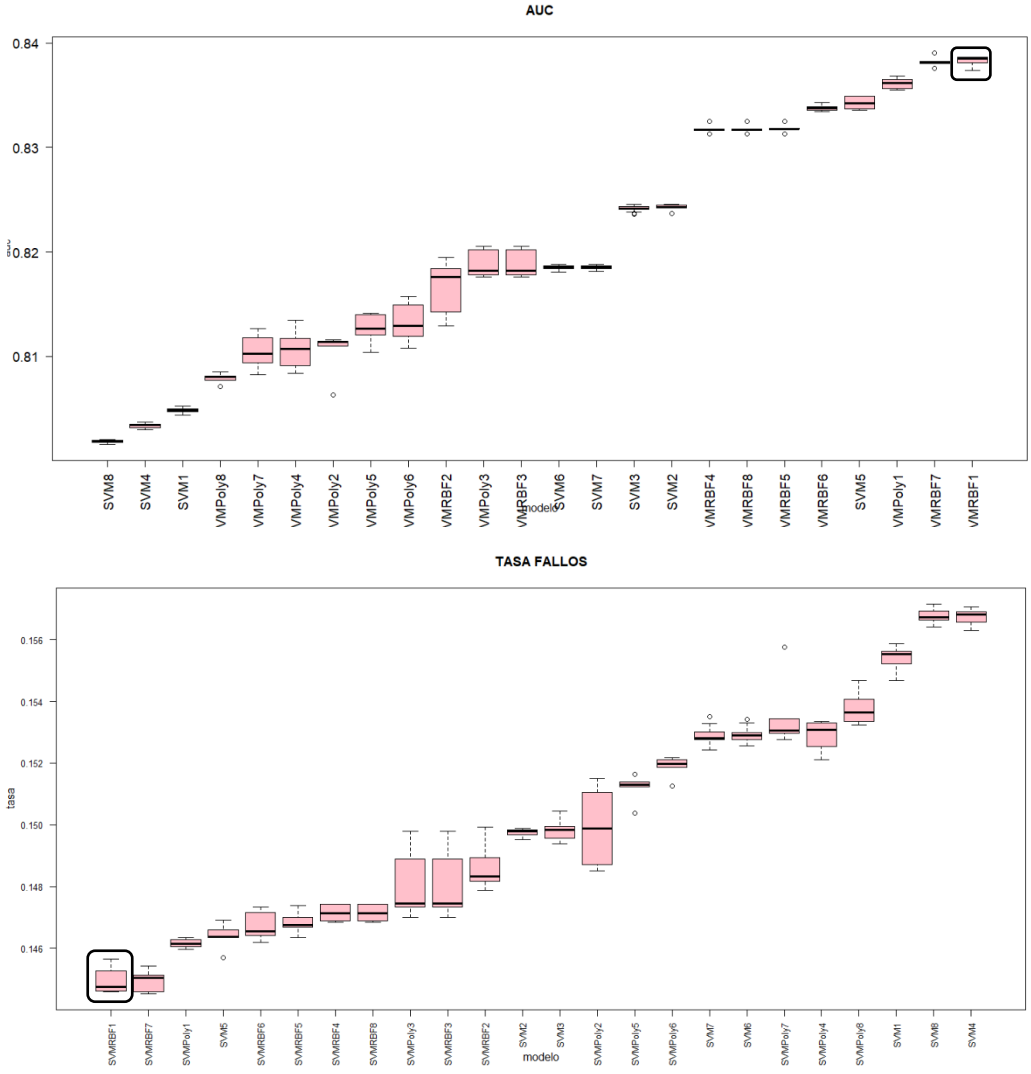


Figura 54. *Comparativa de los cuatro mejores modelos de SVM.*

De todos los modelos obtenidos mediante todas las variables de *Support Vector Machine* y que se pueden apreciar en la figura 54, destaca el VMRBF1, por ser el que tiene un mayor AUC y una menor tasa de fallos. Este se compone de las variables del grupo 1 usando el kernel RBF y los parámetros óptimos según el estudio previo (constante C=2 y sigma=0,01).

Un gran competidor es el modelo SVMRBF7 es un buen competidor ya que presenta resultados similares, aunque ligeramente peores tanto en fallos como en AUC.

6.9. Comparación de los mejores modelos

Realizados todos los modelos y detectado el mejor de cada grupo, se procede ahora a compararlos entre ellos utilizando validación cruzada repetida de 4 grupos y 5 repeticiones para determinar cuál de ellos es el ganador:

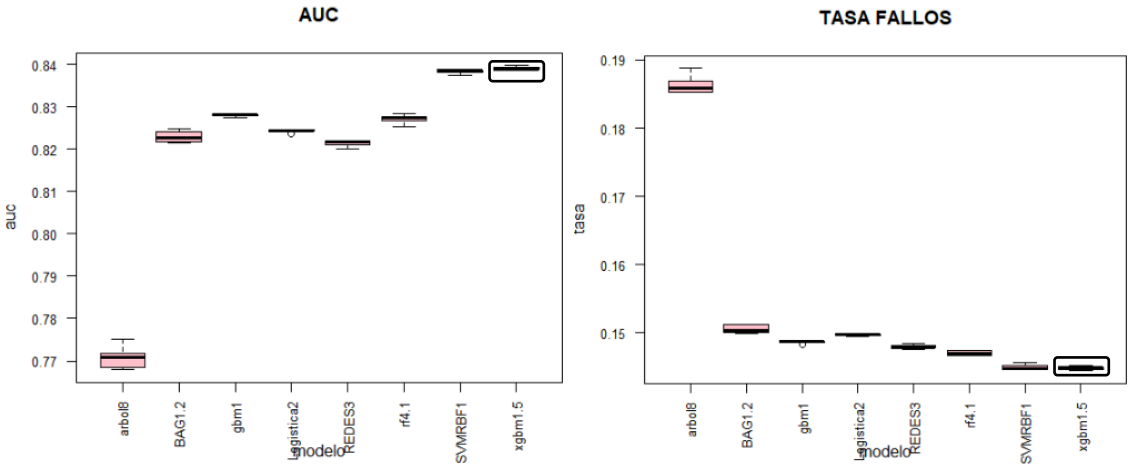


Figura 55. *Comparativa final de los mejores modelos de cada grupo en R Studio.*

Comparando los mejores modelos estudiados (Figura 55) llegamos a la conclusión, de que el modelo hallado usando el algoritmo *Extreme Gradient Boosting Machine* es el mejor modelo predictor, seguido muy de cerca, del encontrado usando *SVM Radial Basis*; ambos usando con las variables del grupo 1.

En concreto, el XGMB1.5 ganador, que muestra el mayor *accuracy* y una menor tasa de fallos (0,145789), es el que se consigue utilizando las variables del grupo 1, con un tamaño mínimo de la hoja de 20, 300 árboles, un *shrinkage* de 0,03, máxima profundidad de 6 y sorteo tanto de muestra como de variables.

Con estos gráficos también comprobamos, como previamente se conocía, que el modelo formado por un único árbol no presenta una gran área bajo la curva ROC además de tener una gran tasa de fallos y mayor variabilidad. Sin embargo, *Random Forest*, que combinación de un gran número de árboles, logra un tercer puesto en la comparación.

Con el objetivo de comprobar si estos modelos son estables, se van a probar con otras semillas consiguiendo el siguiente resultado:

Semilla 9813: XGBM1.5 sigue siendo el algoritmo dominante en ambos casos.

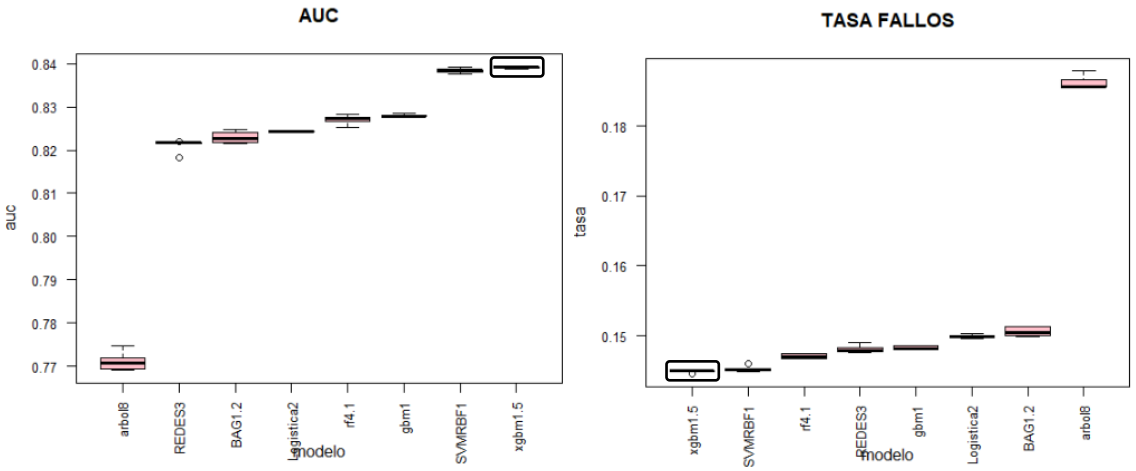


Figura 56. *Comparativa final de los mejores modelos de cada grupo en R Studio con distinta semilla 1.*

Semilla 5988: El modelo ganador sigue mostrando los mismos resultados.

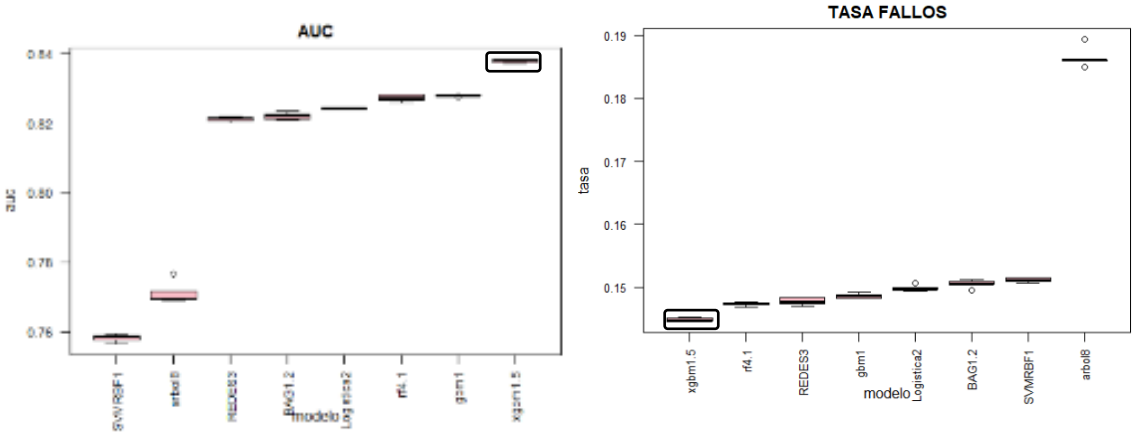


Figura 57. *Comparativa final de los mejores modelos de cada grupo en R Studio con distinta semilla 2.*

Se comprueba, por tanto, la estabilidad del modelo XGBM1.5, que presenta resultados similares con diferentes semillas.

Vamos a probar en este punto si existe alguna combinación de algoritmos que supere a este modelo.

6.10. *Técnicas de ensamblado*

En este caso, utilizaremos las medias de los resultados de cada modelo para realizar las técnicas de ensamblado. La configuración de los híper parámetros data de los mejores modelos hallados en la realización de este trabajo. Los resultados son los siguientes:

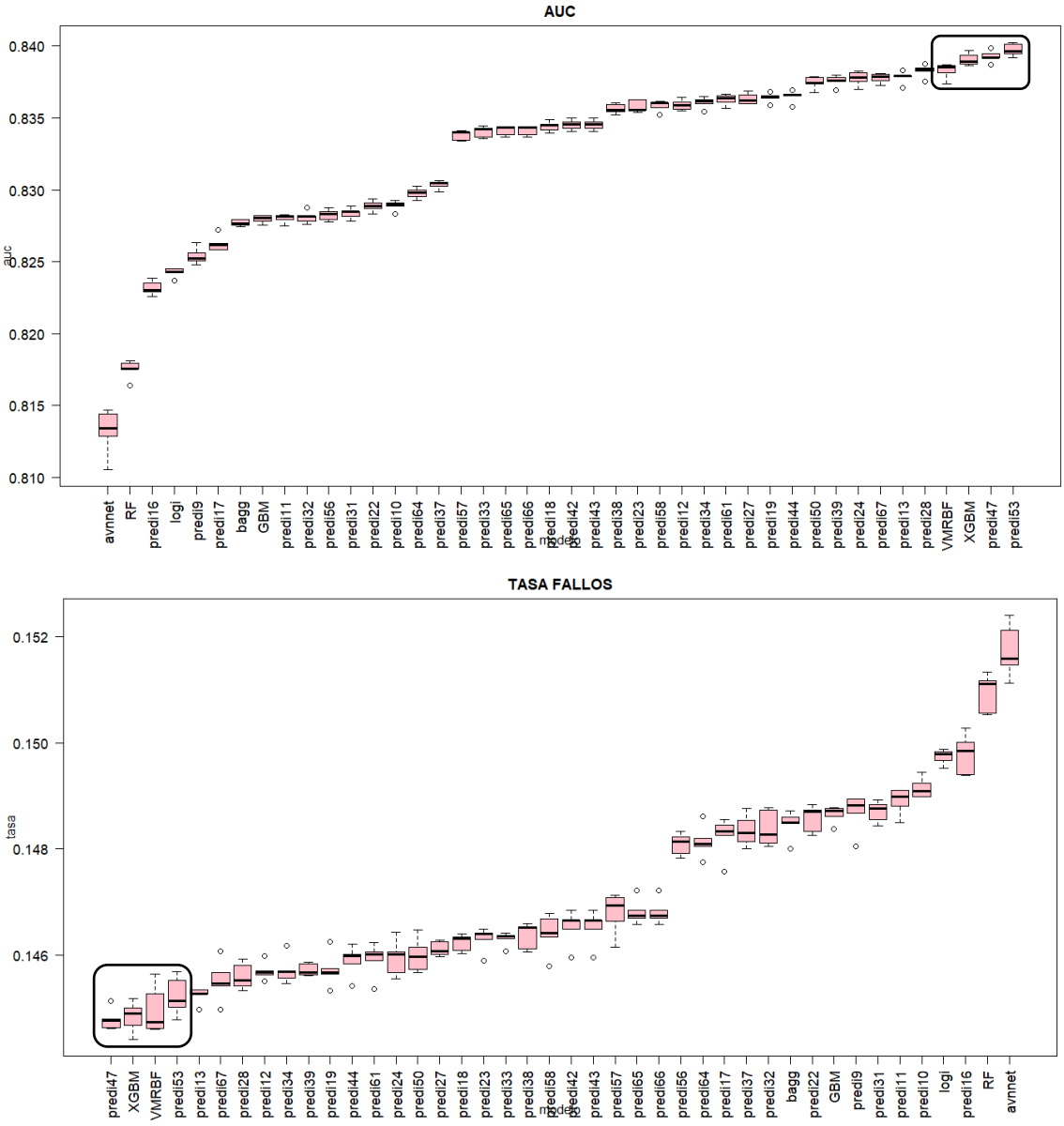


Figura 58. *Comparativa de los mejores modelos usando técnicas de ensamblado.*

Ambas gráficas muestran cómo los ensamblados superan ligeramente a los mejores algoritmos previos (Figura 58). En concreto, son los modelos predi53 y predi47 y los lideran el ranking en términos de AUC con áreas bajo la curva ROC de casi 0,84 puntos. Estos modelos se componen de la combinación de los siguientes algoritmos:

predi47<-logi + XGBM + SVMRBF
predi53<- RF+ XGBM + SVMRBF

Si nos fijamos en ellos, vemos cómo ambos unen el modelo XGMB y el SVMRBF, que son a su vez, los triunfadores cuando compiten por solitario. El primero de ellos se combina con la logística mientras que el segundo lo hace con *Random Forest*.

Finalmente, elegimos el modelo XGBM porque las diferencias tanto en AUC como en tasa de fallos no son suficientes para justificar la inclusión de diferentes grupos de

variables. El modelo resultante se complica en exceso si comparamos con la ganancia que proporciona.

Las conclusiones a las que se llega utilizando R Studio complementan el análisis realizado previamente en SAS Base.

6.11. Comparación mejor modelo de SAS Base y R Studio

Se muestran los resultados obtenidos por ambos en la Tabla 21:

Tabla 21. Comparación mejor modelo de SAS Base y RStudio en cuanto a tasa de fallos.

Nombre	*Algoritmo*	*Tasa de Fallos*
301	*Gradient Boosting Machine*	0,1474489
XGMB1.5	*Extreme Gradient Boosting Machine*	0,1457803

Teniendo en cuenta la tasa de fallos, optaremos por elegir el modelo XGBM1.5, realizado con el programa RStudio, y configurado con las variables del grupo 1. Este grupo se conforma con todas las variables sin transformar, recopiladas en el Anexo III.

Además, aparte de presentarse como un modelo estable que no varía sus resultados cuando se cambian las semillas, se realiza con un algoritmo de una relativamente baja demanda de procesamiento. Se configura de la siguiente manera:

```
grupos=4,sinicio=1234,repe=5,
min_child_weight=15,eta=0.05,nrounds=500,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1,
alpha=0,lambda=0,lambda_bias=0)
```

No obstante, se tiene en cuenta que esta comparación no es del todo acertada pues se está observando únicamente la tasa de fallos y no se ha realizado un estudio de curva ROC en SAS Base. Por otra parte, hay que tener en cuenta que ambos muestran unos datos muy similares, por lo que esta elección se toma de manera simbólica.

Una de las ventajas de tener un modelo ganador basado en árboles es que podemos obtener la importancia de las distintas variables, que indica cómo de útil fue la variable en la construcción de los árboles. En concreto, obtendremos la importancia relativa de las 20 variables más valiosas usando el algoritmo XGBM.

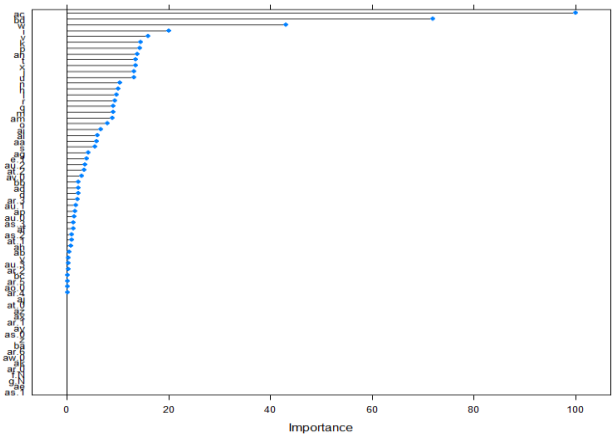


Figura 59: *Importancia relativa de las variables en XGBM.*

Tabla 22. Importancia relativa de las variables en XGBM.

RELATIVE IMPORTANCE		*VARIABLES*
100.000	ac	Inventari_Semafors
71.969	bd	IMP_REP_Vitalidad
42.979	w	Arboleda_Viaria
19.988	i	Distancia_Curva_peligrosa
15.947	v	Aparcamiento_Coches
14.419	k	Distancia_Hoteles_pensiones
14.314	p	Distancia_Lugares_de_culto
13.763	an	Distancia_Ense_anza
13.512	t	Distancia_Espacios_de_partic
13.469	x	Aparcamiento
13.136	j	Distancia_Espacios_de_musica
13.114	u	Distancia_Playa
10.392	n	Distancia_Parques_y_jardines
10.020	h	Distancia_Radares_y_camaras
9.716	l	Distancia_Hospitales_de_aten
9.373	r	Distancia_Mercados_municipal
9.051	q	Distancia_Mercados_y_ferias_
8.993	m	Distancia_Bibliotecas_museos
8.904	am	Distancia_Ense_anza_Infantil
7.947	o	Distancia_Residencias_y_cent

La inclusión de semáforos (Tabla 22), como a priori se podría pensar, es determinante a la hora de que se produzca o no una determinada colisión. Se trata de una variable vital del modelo, por lo que no puede ser obviada.

Suponiento que la importancia de la red semafórica es del 100%, la siguiente variable más valiosa sería la de vitalidad, seguida de la arboleda viaria, que podría aparecer en ese puesto debido a las pérdidas de visibilidad que ocasiona.

El resto de las variables, aunque menos significativas, no debe pasarse por alto, puesto que la combinación de todas ellas es lo que hace al modelo obtener buenos resultados.

7. Conclusiones y propuestas de mejora

7.1. Conclusiones

La catastrófica cifra de muertes y hospitalizaciones causadas por accidentes de tráfico, así como la poca expectativa de que estras cifras se reduzcan drásticamente en el corto plazo, han dado sentido a este trabajo. Por otra parte, la amplia disponibilidad de datos públicos ha impulsado su realización.

El estudio se basa sobre una base de datos creada mediante la combinación de varios archivos, que recopila información de 98.470 accidentes en la ciudad de Barcelona y que fueron registrados por la Guardia Urbana en el periodo 2010-2020. Se percibe, mediante el análisis visual de los mismo, una gran concentración de siniestros en el centro de la ciudad.

El objetivo es determinar si una determinada área es más o menos peligrosa en función de una serie de características viarias y de la densidad de peatones. Para lograrlo, se lleva a cabo un exhaustivo proceso de búsqueda de información, tratamiento de datos y, por último, creación de variables. Se cuenta con un total de 57 variables independientes.

Siguiendo la metodología SEMMA, una vez conseguida la base de datos, se inicia el proceso de depuración, modificación y transformación de datos y variables, y se da paso a la fase de modelización, que se lleva a cabo utilizando los programas SAS Base y RStudio.

Se realizan más de 400 modelos utilizando algoritmos con diferentes metodologías; regresión logística, redes neuronales, que se basan en el funcionamiento biológico de las neuronas, los basados en árboles (*Bagging, Random Forest, Gradient Boosting Machine o Extreme Gradient Boosting*) o de cultura geométrica, como es *Support Vector Machine*.

El objetivo general que se plantea, dada la problemática a la que se ha aludido en la introducción, es determinar si una determinada área es más o menos peligrosa en función de una serie de características viarias y de densidad de peatones, lo que se consigue con la obtención de un modelo realizado a través de *Extreme Gradient Boosting Machine*.

Gracias a la figura 59, que aparece en el punto anterior y que muestra la importancia de variables del modelo ganador, se logra la consecución del primer objetivo específico de conocer los factores determinantes en la generación de accidentes. Estos son, principalmente, la red semafórica seguida de la vitalidad y la arboleda viaria (Tabla 22). Se desconoce, sin embargo, si estos actúan como alicientes o como freno en la generación de estas colisiones.

Por otra parte, la variable vitalidad, que halla la densidad de viandantes, ocupa el segundo puesto es esa clasificación, lo que hace que se pueda cumplir el segundo

objetivo de confirmar la premisa inicial que suponía que una alta afluencia de peatones se traduce a una mayor peligrosidad. De forma visual también se aprecia fácilmente en la siguiente figura:

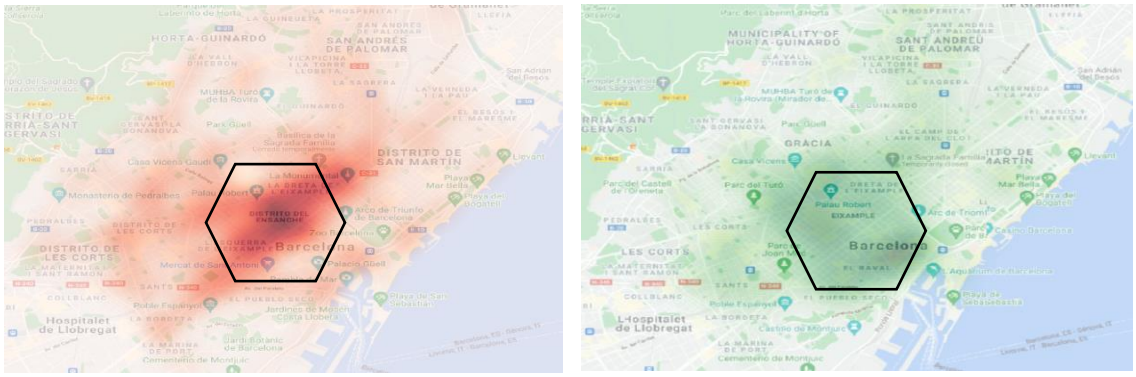


Figura 60: *Mapa de calor de accidentes de tráfico VS mapa de calor de vitalidad.*

Cumplidos los objetivos iniciales, podemos concluir que los resultados de este trabajo pueden ser de utilidad, lo que da sentido a la realización del mismo.

7.2. Propuestas de mejora

Para afinar el estudio habría que tener en cuenta variables adicionales que puedan resultar importantes y que hayan quedado fuera del modelo debido a inaccesibilidad pública. Este es el caso de datos como el ancho de la vía o de las aceras.

Por otra parte, sería conveniente utilizar la información más actualizada posible a fin de evitar caer en errores provocados por decalajes de tiempo entre datos y realidad.

Ambas líneas podrían solucionarse con la información que proporcionan sistemas tecnológicos que recopilen datos a tiempo real y que analicen el entorno. Un ejemplo podría ser analizar los datos obtenidos mediante sistemas avanzados de asistencia a la conducción (ADAS, por sus siglas en inglés) impulsado por la Comisión Europea.

Por lo tanto, incluyendo un mayor número de variables y más actualizadas, así como modelos espaciales, que tengan en cuenta la relación de los tramos con sus vecinos, modelos en sintonía con el desarrollado en este trabajo podrían tenerse en cuenta a la hora de realizar proyectos urbanísticos introduciendo datos y analizando el resultado en términos de peligrosidad.

8. Bibliografía

Breiman, L. (1996). Bagging Predictors. En L. Breiman, *Machine learning* (Vol. 24, págs. 123-140). Boston: Kluwer Academic Publishers.

Calabresi, G. (1970). *The Cost of Accidents: A Legal and Economic Analysis.* New Haven, ESTADOS UNIDOS: Yale University Press.

Calviño, A. (2019). *Técnicas y Metodología de la Minería de Datos.* Universidad Complutense de Madrid, Departamento de Estadística y Ciencia de los Datos.

CNIG. (17 de 05 de 2021). *Centro de Descargas Organismo Autónomo Centro Nacional de Información Geográfica.* Obtenido de Centro de Descargas Organismo Autónomo Centro Nacional de Información Geográfica: http://centrodedescargas.cnig.es/CentroDescargas/catalogo.do?Serie=LIDAR

Confederación Nacional de Autoescuelas [CNAE]. (24 de mayo de 2018). *Confederación Nacional de Autoescuelas*. Obtenido de Confederación Nacional de Autoescuelas: https://www.cnae.com/index.aspx/autoescuelas/noticias/la-dgt-de-acuerdo-con-bruselas-los-vehiculos-deben-llevar-de-serie-adas

Departament de Gestió de la Mobilitat de l'Ajuntament de Barcelona. (2017). *Manual de Senyalització Urbana per a la Ciutat de Barcelona.* Barcelona: IPS Vial, S.L.

DGT. (07 de enero de 2021). *Dirección General de Tráfico*. Recuperado el junio de 2021, de Dirección General de Tráfico: https://www.dgt.es/es/prensa/notas-de-prensa/2021/Los_accidentes_de_trafico_se_cobran_la_vida_de_870_personas_durante_el_ano_pasado.shtml

Fernández, S. d. (2011). *Análisis Conglomerados.* Universidad Autónoma de Madrid, Facultad de Ciencias Económicas y Empresariales., Madrid.

Fiuza Pérez, M. D., & Rodríguez Pérez, J. C. (2000). La regresión logística: una herramienta versátil. *Nefrología*, 477-565.

ICGC. (24 de noviembre de 2020). *Instituto Cartográfico y Geológico de Cataluña.* Obtenido de Instituto Cartográfico y Geológico de Cataluña: https://www.icgc.cat/Descarregues

Izaurieta, F., & Saavedra, C. (2000). *Redes Neuronales Artificiales.* Universidad de Concepción, Departamento de Física, Concepción.

Jurgen, R. K. (2013). *Autonomous Vehicles for Safer Driving.* Warrendale, ESTADOS UNIDOS: SAE International.

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine. En *Proceeding Indones. Sci. Meeting Cent.* Japón.

Open Data BCN. (2020). *Open Data BCN*. Obtenido de Open Data BCN: https://opendata-ajuntament.barcelona.cat/data/es/dataset

OSM. (2021). *OpenStreetMap.* Obtenido de OpenStreetMap: http://download.geofabrik.de/europe/spain.html

Portela, J. (2019). Universidad Complutense de Madrid, Departamento de Estadística e Investigación Operativa III, Madrid.

Portela, J. (2019). *Construcción del modelo y primeros ejemplos.* Universidad Complutense de Madrid, Departamento de Estadística e Investigación Operativa III, Madrid.

Portela, J. (2019). *Explicación de las principales macros para redes neuronales.* Universidad Complutense de Madrid, Departamento de Estadística e Investigación Operativa III, Madrid.

Portela, J. (2019). *Machine Learning. Introducción.* Universidad Complutense de Madrid, Departamento de Estadística e Investigación Operativa III, Madrid.

Ramm, F. (06 de noviembre de 2019). *Geofrabrik.* Obtenido de Geofrabrik: https://www.geofabrik.de/data/geofabrik-osm-gis-standard-0.7.pdf

Rodríguez Montequín, M. T., Álvarez Cabal, V., Mesa Fernández, J. M., & González Valdés, A. (s.f.). *Metodologías para la realización de proyectos de Data Mining.* Oviedo.

RTVE.es/Servimedia. (18 de diciembre de 2017). *La mortalidad por accidentes de tráfico repunta en 2017 por segundo año consecutivo*. Recuperado el 23 de diciembre de 2020, de Radio Televisión Española [RTVE]: https://www.rtve.es/noticias/20171218/mortalidad-accidentes-trafico-repunta-2017-segundo-ano-consecutivo/1648003.shtml

Scornet, E., Biau, G., & Vert, J.-P. (2015). Consistency of Random Forest. En *The Annals of Statistics* (Vol. 43(4), págs. 1716-1741).

Anexos

Anexo I. Exploración de datos

Tabla A. Estudio de valores ausentes y atípicos en variables de clase.

Nombre de la variable	Nº de niveles	Ausente	Moda	% moda	Moda2	% Moda2
Bicicletas	6	0	Vía ciclable y <30 km/h	77.09	Vía no acondicionada	11.84
Intersección diff velocidad	9	0	0	92.83	20	6.16
Nom Barri	73	0	el Poble Sec	9.69	la Marina del Prat Vermell	5.24
Nom Districte	10	0	Sants-Montjuic	21.62	Sant Marti	13.50
Otros centros pequeños enseñanza	2	0	0	97.27	1	2.73
Puente o túnel	2	0	N	98.06	Y	2.14
Railway in street Intersecciones	2	0	N	97.15	Y	2.85
Riesgo Intersección	6	0	Muy bajo	92.83	Medio	6.63
Sentido vía	2	0	1	53.99	2	46.01
Tipo de vía	16	0	Residential	34.62	Pedestrian	16.68
Velocidad vía	8	0	30	84.56	50	13.58

Tabla B. Estudio de valores ausentes y atípicos en variables de intervalo.

Variable	Media	No ausente	Au-sente	Mín	Mediana	Máx	Asimetría	Curtosis
Aparcamiento	1.271.316	49529	0	0	0	60	4.847.625	3.206.507
Aparcamiento Bicis	0.096166	49529	0	0	0	12	7.044.278	7.308.726
Aparcamiento Coches	0.543116	49529	0	0	0	35	5.219.666	3.874.255
Aparcamiento Motos	0.45749	49529	0	0	0	38	6.171.486	5.684.251
Aparcamiento Otros	0.17412	49529	0	0	0	18	6.279.219	5.757.024
Arboleda Viaria	3.180.319	49529	0	0	0	115	3.680.917	2.244.547
Cotidiano alimentario	0.158271	49529	0	0	0	231	6.780.339	7.172.482
Cotidiano no alimentario	0.052252	49529	0	0	0	14	9.900.219	2.317.867
Distancia Atracciones turisticas	4.938.712	49529	0	0	3.605.164	4.481.795	3.833.906	1.837.585
Distancia Bibliotecas museos cin	3.256.607	49529	0	0.031864	2.622.332	2.929.919	295.204	1.398.124
Distancia Curva peligrosa	1.382.634	49529	0	1.51E-11	1.198.659	5476.09	1.282.908	2.196.556
Distancia Enseñanza	2.159.413	49529	0	0	1.408.851	2.692.718	3.157.781	1.278.951
Distancia Ense-ñanza Infantil	2.469.756	49529	0	0	1.666.646	2.692.718	316.286	1.261.254

Distancia Espacios de música	8.587.636	49529	0	0.382294	5.878.743	5.479.835	2.324.628	6.815.798
Distancia Espacios de participac	250.939	49529	0	0	1.299.568	3.616.002	4.726.612	2.483.335
Distancia Hospitales de atención	5.324.481	49529	0	0.059596	3.998.221	4.055.663	3.493.109	153.716
Distancia Hoteles pensiones otro	5.388.518	49529	0	0.084684	303.775	4.014.382	2.058.668	4.493.352
Distancia Lugares de culto	3.095.727	49529	0	0.016109	1.914.444	3602.02	440.831	2.246.955
Distancia Mercados municipales	7.661.053	49529	0	0.411086	5.790.252	4.841.015	262.486	8.603.022
Distancia Mercados y ferias cal1	6.617.326	49529	0	0.209162	5.100.061	4.841.015	2.483.836	8.343.177
Distancia Mercados y ferias call	8.846.477	49529	0	0.209162	7.668.855	4902.43	157.021	430.389
Distancia Parques y jardines	3.482.233	49529	0	0.044002	2.216.188	4.089.337	4.474.662	2.258.497
Distancia Playa	5.167.499	49529	0	0	5.446.928	9.999.566	-0.18987	-0.903
Distancia Radares y cámaras	8.212.182	49529	0	0	6.732.148	5.040.134	1.673.793	4.566.339
Distancia Residencias y centros	5.182.942	49529	0	0	2.818.121	5.282.924	3.694.625	1.522.734
Distancia Roundabout	5.597.764	49529	0	0	497.323	4.252.378	2.130.456	1.000.544
Equipamiento personal	0.097922	49529	0	0	0	98	5.380.489	4.518.509
Farmacias	0.020655	49529	0	0	0	2	6.823.928	4.533.279
Financieras y aseguradoras	0.025076	49529	0	0	0	4	8.089.492	8.029.471
Inventari Semafors	2.255.991	49529	0	0	0	59	296.795	1.031.484
Menaje hogar	0.048012	49529	0	0	0	9	8.715.542	1.159.442
Meters	5.351.663	49529	0	2	32	**182214**	1.574.297	24895.32
NUM CEDA STOP	0.196935	49529	0	0	0	13	5.627.308	4.530.728
Num CEDA	0.156232	49529	0	0	0	13	6.205.231	5.523.468
Num STOP	0.040703	49529	0	0	0	8	1.025.913	1.324.347
Ocio y cultura pequeños	0.032183	49529	0	0	0	13	1.555.888	4.834.576
Otros Comercios	0.474853	49529	0	0	0	59	5.791.278	964.468
Pendiente	1.980.472	49529	0	0	0	5.400.727	3.257.215	1.165.034
Reparaciones	0.028367	49529	0	0	0	5	8.308.635	8.680.336
Restaurants	0.202366	49529	0	0	0	45	1.095.971	1.677.649
Terrazas	0.100991	49529	0	0	0	11	6.312.148	5.638.058
Vitalidad	-0.03648	49529	0	-0.58609	-0.1654	9.502.441	4.235.486	4.538.536

Anexo II. Glosario de nombres de las variables del conjunto de datos

NOMBRE VARIABLE	NUEVO NOMBRE	TIPO	NOMBRE VARIABLE	NUEVO NOMBRE	TIPO
APARCAMIENTO	x	Intervalo	LOG_DISTANCIA_RESIDENCIAS_Y_CENT	cd	Intervalo
APARCAMIENTO_BICIS	z	Intervalo	MENAJE_HOGAR	af	Intervalo
APARCAMIENTO_COCHES	y	Intervalo	NUM_CEDA_STOP	ad	Intervalo
APARCAMIENTO_MOTOS	aa	Intervalo	NUMMISSING	ax	Intervalo
APARCAMIENTO_OTROS	ab	Intervalo	OCIO_Y_CULTURA_PEQUE_OS	ap	Intervalo
ARBOLEDA_VIARIA	w	Intervalo	OPT_APARCAMIENTO	bf	Clase
COTIDIANO_ALIMENTARIO	ag	Intervalo	OPT_APARCAMIENTO_BICIS	bg	Clase
COTIDIANO_NO_ALIMENTARIO	ah	Intervalo	OPT_APARCAMIENTO_COCHES	bh	Clase
DISTANCIA_ATRACCIONES_TURISTICAS	v	Intervalo	OPT_APARCAMIENTO_MOTOS	bi	Clase
DISTANCIA_BIBLIOTECAS_MUSEOS_CIN	m	Intervalo	OPT_APARCAMIENTO_OTROS	bj	Clase
DISTANCIA_CURVA_PELIGROSA	i	Intervalo	OPT_ARBOLEDA_VIARIA	bk	Clase
DISTANCIA_ENSE_ANZA	an	Intervalo	OPT_COTIDIANO_ALIMENTARIO	bl	Clase
DISTANCIA_ENSE_ANZA_INFANTIL	am	Intervalo	OPT_COTIDIANO_NO_ALIMENTARIO	bm	Clase
DISTANCIA_ESPACIOS_DE_MUSICA_Y_C	j	Intervalo	OPT_DISTANCIA_ATRACCIONES_TURIST	bn	Clase
DISTANCIA_ESPACIOS_DE_PARTICIPAC	t	Intervalo	OPT_DISTANCIA_CURVA_PELIGROSA	bp	Clase
DISTANCIA_HOSPITALES_DE_ATENCION	l	Intervalo	OPT_DISTANCIA_ENSE_ANZA	bq	Clase
DISTANCIA_HOTELES_PENSIONES_OTRO	k	Intervalo	OPT_DISTANCIA_ENSE_ANZA_INFANTIL	br	Clase
DISTANCIA_LUGARES_DE_CULTO	p	Intervalo	OPT_DISTANCIA_ESPACIOS_DE_PARTIC	bt	Clase
DISTANCIA_MERCADOS_MUNICIPALES	r	Intervalo	OPT_DISTANCIA_HOSPITALES_DE_ATEN	bu	Clase
DISTANCIA_MERCADOS_Y_FERIAS_CAL1	s	Intervalo	OPT_DISTANCIA_LUGARES_DE_CULTO	bw	Clase
DISTANCIA_MERCADOS_Y_FERIAS_CALL	q	Intervalo	OPT_DISTANCIA_MERCADOS_MUNICIPAL	bx	Clase
DISTANCIA_PARQUES_Y_JARDINES	n	Intervalo	OPT_DISTANCIA_MERCADOS_Y_FERIAS_	by	Clase
DISTANCIA_PLAYA	u	Intervalo	OPT_DISTANCIA_MERCADOS_Y_FERIAS1	bz	Clase
DISTANCIA_RADARES_Y_CAMARAS	h	Intervalo	OPT_DISTANCIA_PARQUES_Y_JARDINES	ca	Clase
DISTANCIA_RESIDENCIAS_Y_CENTROS_	o	Intervalo	OPT_DISTANCIA_PLAYA	cb	Clase
DISTANCIA_ROUNDABOUT	c	Intervalo	OPT_DISTANCIA_RADARES_Y_CAMARAS	cc	Clase
EQUIPAMIENTO_PERSONAL	ai	Intervalo	OPT_DISTANCIA_ROUNDABOUT	ce	Clase
FINANCIERAS_Y_ASEGURADORAS	ak	Intervalo	OPT_EQUIPAMIENTO_PERSONAL	cf	Clase
G_BICICLETAS	at	Clase	OPT_FINANCIERAS_Y_ASEGURADORAS	cg	Clase
G_NOM_BARRI	ar	Clase	OPT_IMP_REP_NUM_CEDA	ch	Clase
G_NOM_DISTRICTE	as	Clase	OPT_IMP_REP_PENDIENTE	cj	Clase
G_RIESGO_INTERSECCION	aw	Clase	OPT_IMP_REP_TERRAZAS	ck	Clase
G_TIPO_DE_VIA	au	Clase	OPT_IMP_REP_VITALIDAD	cl	Clase
G_VELOCIDAD_VIA	av	Clase	OPT_INTERSECCION_DIFF_VELOCIDAD	cm	Clase
IMP_REP_FARMACIAS	ay	Intervalo	OPT_INVENTARI_SEMAFORS	cn	Clase
IMP_REP_NUM_CEDA	az	Intervalo	OPT_MENAJE_HOGAR	co	Clase
IMP_REP_NUM_STOP	ba	Intervalo	OPT_OCIO_Y_CULTURA_PEQUE_OS	cq	Clase
IMP_REP_PENDIENTE	bb	Intervalo	OPT_OTROS_COMERCIOS	cr	Clase
IMP_REP_TERRAZAS	bc	Intervalo	OPT_REPARACIONES	cs	Clase
IMP_REP_VITALIDAD	bd	Intervalo	OPT_RESTAURANTS	ct	Clase
INCIDENCIA_ACCIDENTE_POR_METRO	b	Objetivo	OTROS_CENTROS_PEQUE_OS_ENSE_ANZA	ao	Clase
INTERSECCION_DIFF_VELOCIDAD	d	Intervalo	OTROS_COMERCIOS	al	Intervalo
INV_IMP_REP_NUM_STOP	ci	Intervalo	PUENTE_O_TUNEL	f	Clase
INV_NUM_CEDA_STOP	cp	Intervalo	RAILWAY_IN_STREET_INTERSECCIONES	g	Clase
INVENTARI_SEMAFORS	ac	Intervalo	REPARACIONES	aj	Intervalo
JOIN_OBJEC	a	ID	RESTAURANTS	ae	Intervalo
LG10_DISTANCIA_HOTELES_PENSIONES	bv	Intervalo	SENTIDO_VIA	e	Clase
LOG_DISTANCIA_ESPACIOS_DE_MUSICA	bs	Intervalo	SQRT_DISTANCIA_BIBLIOTECAS_MUSEO	bo	Intervalo

Anexo III. Grupos de variables

Grupo 1:

Clase = ao	ar	as	at	au	av	aw	e	f	g	
Intervalo = aa	ab	ac	ad	ae	af	ag	ah	ai	aj	ak
al	am	an	ap	ax	ay	az	ba	bb	bc	bd
d	h	i	j	k	l	m	n	o	p	q
r	s	t	u	v	w	x	y	z		

Grupo 2:

Clase = as	at	au	bf	bj	bk	bl	bm	bn	bp	br
bt	bw	by	bz	ca	cb	cc	cf	cg	cj	ck
cl	cn	co	cq	cr	ct	g				
Intervalo = ag	aj	al	am	bv	c	d	k	n	r	t
u	v	w	y							

Grupo 3:

Clase = as	at	au	bf	bj	bk	bl	bm	bn	bp	br
bt	bw	by	bz	cb	cc	cf	cg	cj	ck	cl
cn	co	cq	cr	ct	g					
Intervalo = al	an	ax	bv	c	d	k	r	t	u	v
w	y									

Grupo 4:

Clase = ao	ar	as	at	au	av	aw	e	f	g	
Intervalo = ab	ac	ad	ae	af	ag	ai	aj	ak	an	ax
ay	ba	bb	bd	c	d	h	i	s	t	u
w	x	z								

Grupo 5:

Clase = ar	as	aw	bf	bj	bk	bl	bp	bq	bt	by
cb	cc	cf	cg	cj	cl	cn	co	ct	g	
Intervalo = ac	ai	an	d	t	u	w	x			

Grupo 6:

Clase = ar	as	aw	bf	bj	bk	bl	bp	bq	bt	by
cb	cc	cf	cg	cj	cl	cn	co	ct	g	
Intervalo = ac	ag	ai	an	d	s	t	u	w	x	

Grupo 7:

Clase = ar	as	aw	bf	bj	bk	bl	bp	bq	bt	by
cb	cc	cf	cg	cj	cl	cn	co	ct	g	
Intervalo = ac	ai	an	d	s	t	u	w	x		

Grupo 8:

Clase = ar	as	at	au							
Intervalo = aa	ab	ac	af	ak	bc	bd	h	j	k	l
r	s	w	x	y						

Anexo IV. Número de Inputs que representa cada variable

Nombre de variable de clase	ROL	Nº Niveles
G_Bicicletas	INPUT	4
G_Nom_Barri	INPUT	8
G_Nom_Districte	INPUT	5
G_Riesgo_Interseccion	INPUT	2
G_Tipo_de_via	INPUT	5
G_Velocidad_via	INPUT	2
OPT_Aparcamiento	INPUT	4
OPT_Aparcamiento_Bicis	INPUT	4
OPT_Aparcamiento_Coches	INPUT	4
OPT_Aparcamiento_Motos	INPUT	4
OPT_Aparcamiento_Otros	INPUT	3
OPT_Arboleda_Viaria	INPUT	4
OPT_Cotidiano_alimentario	INPUT	3
OPT_Cotidiano_no_alimentario	INPUT	2
OPT_Distancia_Atracciones_turist	INPUT	4
OPT_Distancia_Curva_peligrosa	INPUT	4
OPT_Distancia_Ense_anza	INPUT	4
OPT_Distancia_Ense_anza_Infantil	INPUT	4
OPT_Distancia_Espacios_de_partic	INPUT	4
OPT_Distancia_Hospitales_de_aten	INPUT	3
OPT_Distancia_Lugares_de_culto	INPUT	3
OPT_Distancia_Mercados_municipal	INPUT	4
OPT_Distancia_Mercados_y_ferias1	INPUT	3

Nombre de variable de clase	ROL	Nº Niveles
OPT_Distancia_Mercados_y_ferias_	INPUT	3
OPT_Distancia_Parques_y_jardines	INPUT	2
OPT_Distancia_Playa	INPUT	3
OPT_Distancia_Radares_y_camaras	INPUT	4
OPT_Distancia_Roundabout	INPUT	4
OPT_Equipamiento_personal	INPUT	2
OPT_Financieras_y_aseguradoras	INPUT	2
OPT_IMP_REP_Num_CEDA	INPUT	2
OPT_IMP_REP_Pendiente	INPUT	3
OPT_IMP_REP_Terrazas	INPUT	3
OPT_IMP_REP_Vitalidad	INPUT	4
OPT_Interseccion_diff_velocidad	INPUT	2
OPT_Inventari_Semafors	INPUT	4
OPT_Menaje_hogar	INPUT	3
OPT_Ocio_y_cultura_peque_os	INPUT	2
OPT_Otros_Comercios	INPUT	3
OPT_Reparaciones	INPUT	3
OPT_Restaurants	INPUT	2
Otros_centros_peque_os_ense_anza	INPUT	2
Puente_o_tunel	INPUT	2
Railway_in_street_Intersecciones	INPUT	2
Sentido_via	INPUT	2
Incidencia_Accidente_por_metro	TARGET	2

Anexo V. Estudio de los híper parámetros de los distintos grupos de variables en SAS Base

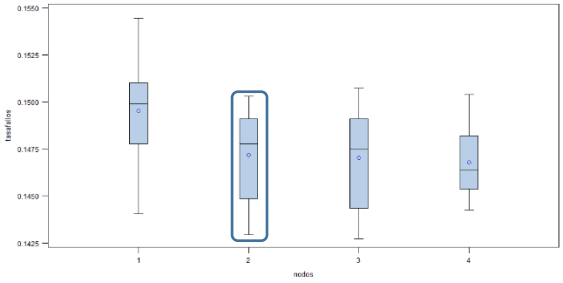
Redes Neuronales

Determinación número de nodos óptimo:

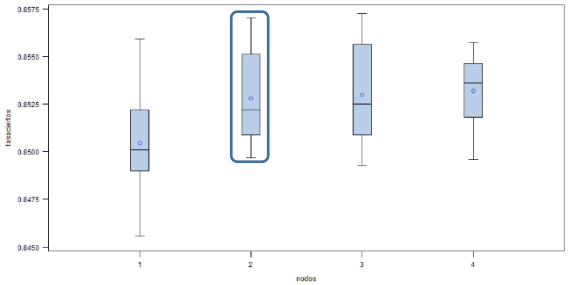
Grupo 2:

a. Algoritmo de optimización Levmar: 2 nodos

Tasa de aciertos

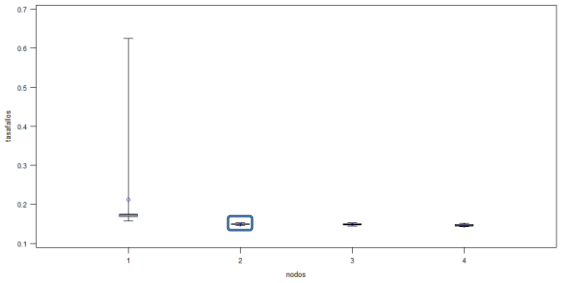


Tasa de fallos

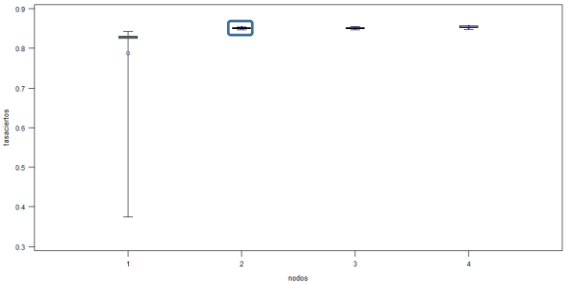


b. Algoritmo de optimización BPROP: 2 nodos

Tasa de aciertos



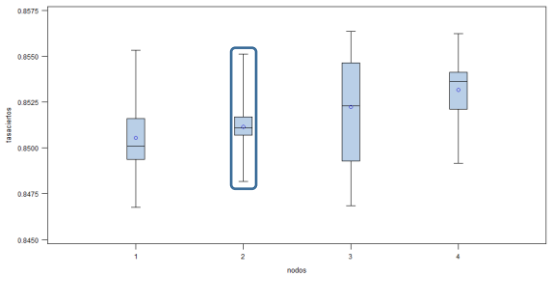
Tasa de fallos



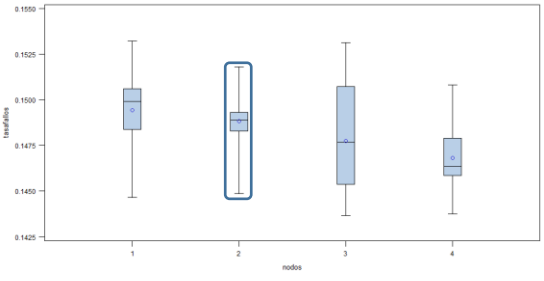
Grupo 3:

a. Algoritmo de optimización Levmar: 2 nodos

Tasa de aciertos

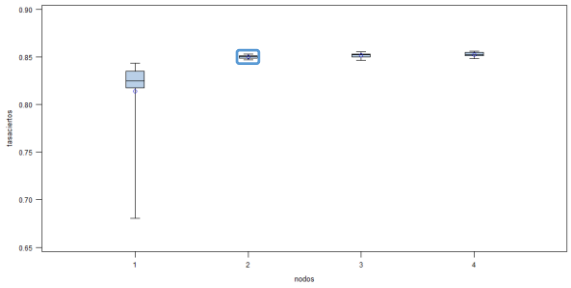


Tasa de fallos

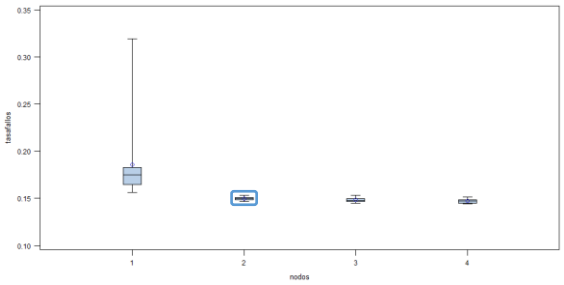


b. Algoritmo de optimización BPROP: 2 nodos

Tasa de aciertos



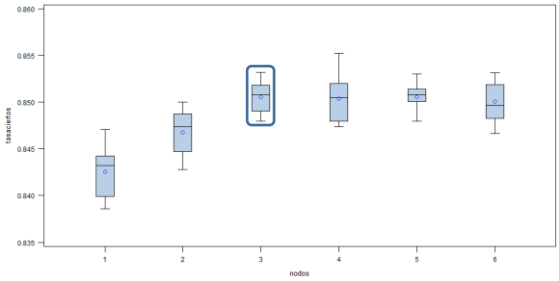
Tasa de fallos



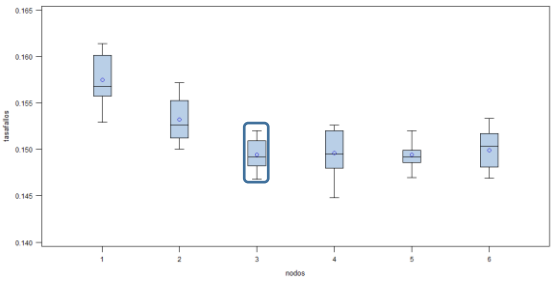
Grupo 4:

a. Algoritmo de optimización Levmar: 3 nodos

Tasa de aciertos

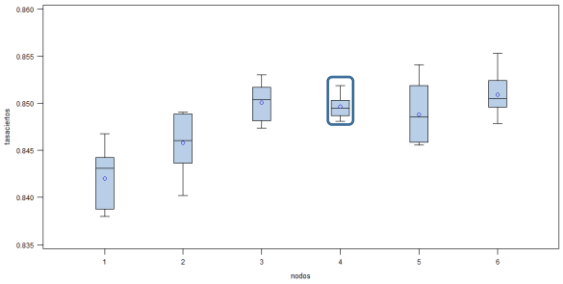


Tasa de fallos

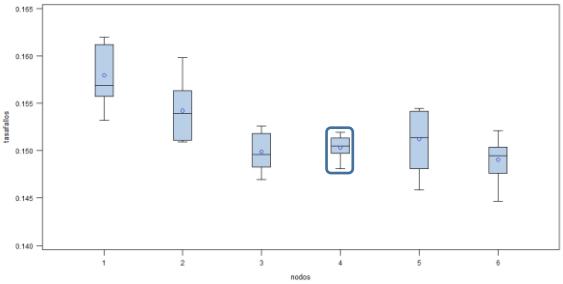


b. Algoritmo de optimización BPROP: 4 nodos

Tasa de aciertos



Tasa de fallos

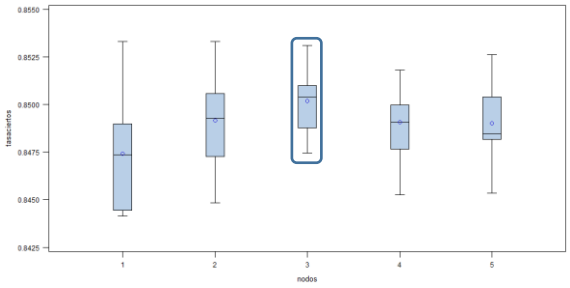


Grupos 5, 6 y 7:

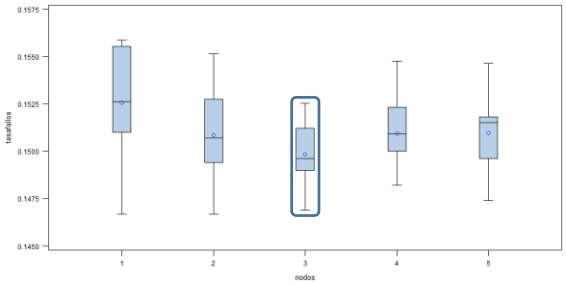
Algoritmo de optimización Levmar: 2 nodos (la elección se debe a que escala es muy pequeña y aunque se dan mejores resultados con otros nodos, no existen apenas mejoras si tenemos en cuenta cuanto se complicaría el modelo en cuanto a número de parámetros).

a. Algoritmo de optimización Levmar: 3 nodos

Tasa de aciertos

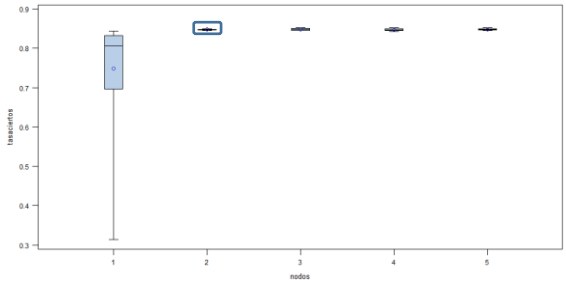


Tasa de fallos

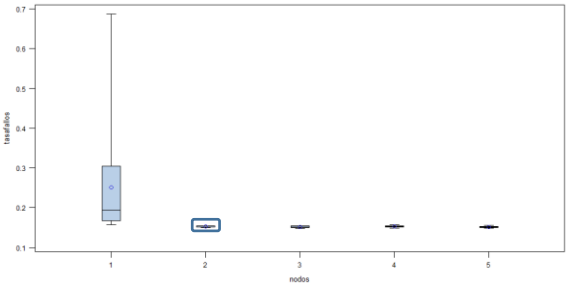


b. Algoritmo de optimización BPROP: 2 nodos

Tasa de aciertos



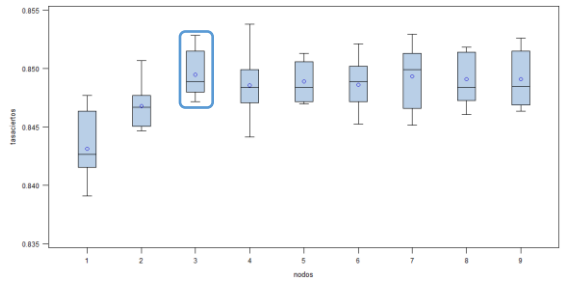
Tasa de fallos



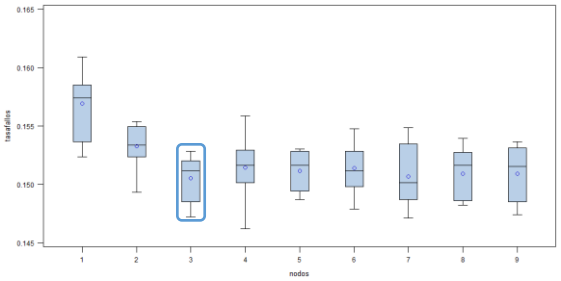
Grupo 8:

a. Algoritmo de optimización Levmar: 3 nodos

Tasa de aciertos

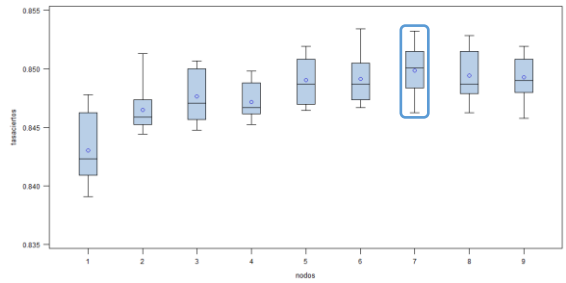


Tasa de fallos

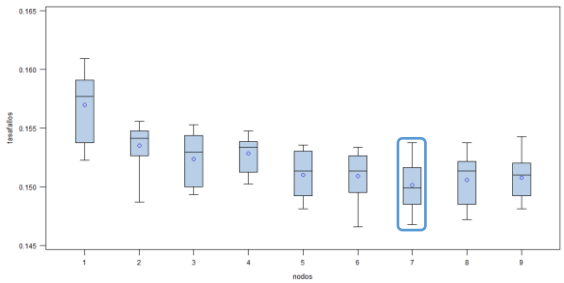


b. Algoritmo de optimización BPROP: 7 nodos

Tasa de aciertos



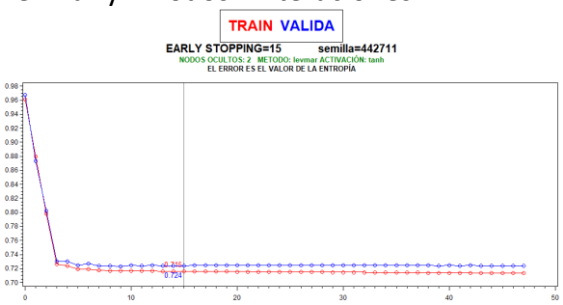
Tasa de fallos



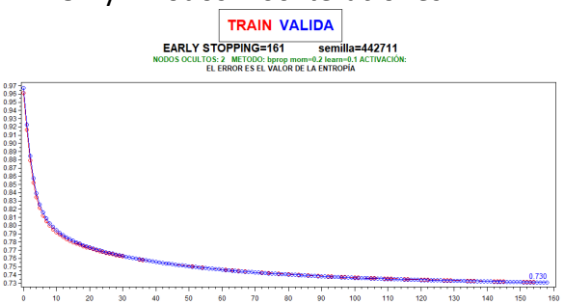
Estudio de ealy stopping:

Grupo 2:

Levmar y 2 nodos: 7 iteraciones

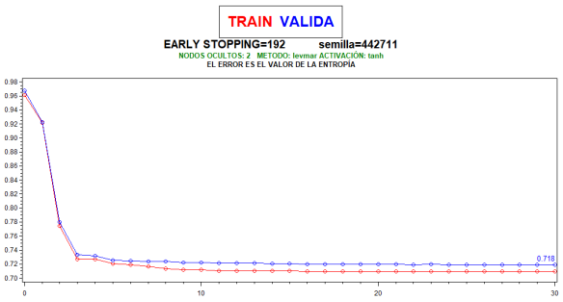


BPROP y 2 nodos: 100 iteraciones

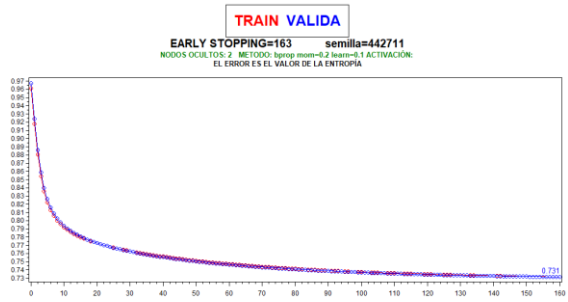


Grupo 3:

Levmar y 2 nodos: 8 iteraciones

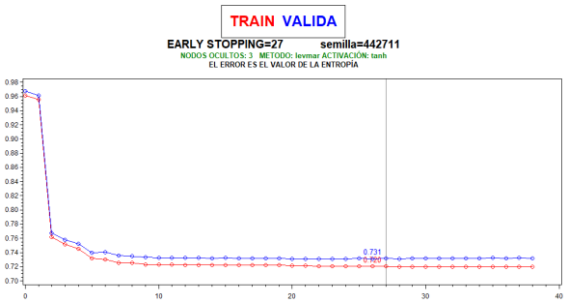


BPROP y 2 nodos: iteraciones 120



Grupo 4:

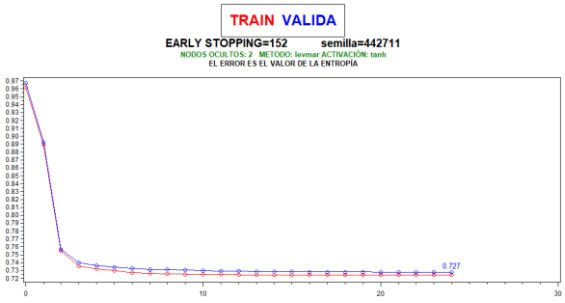
Levmar y 3 nodos: 9 iteraciones



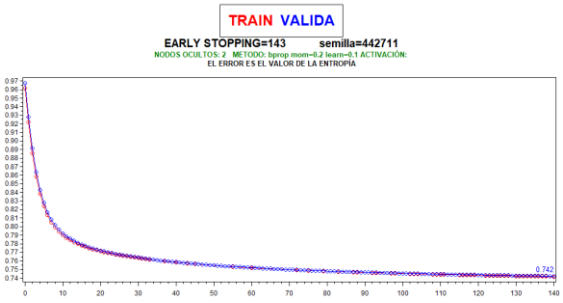
BPROP y 4 nodos: 33 iteraciones

Grupos 5, 6 y 7:

Levmar y 2 nodos: 6 iteraciones

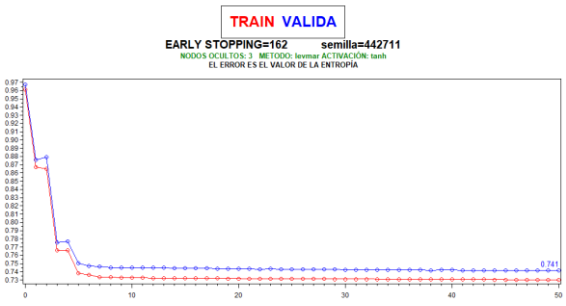


BPROP y 2 nodos: 100 iteraciones

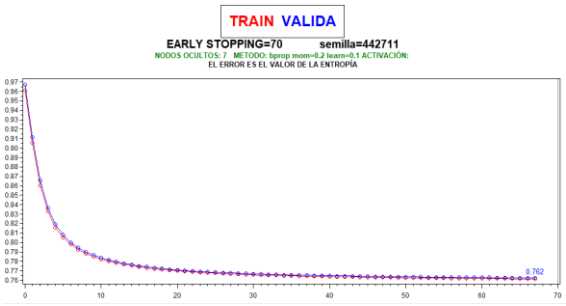


Grupo 8:

Levmar y 3 nodos: 6 iteraciones



BPROP y 7 nodos: 30 iteraciones

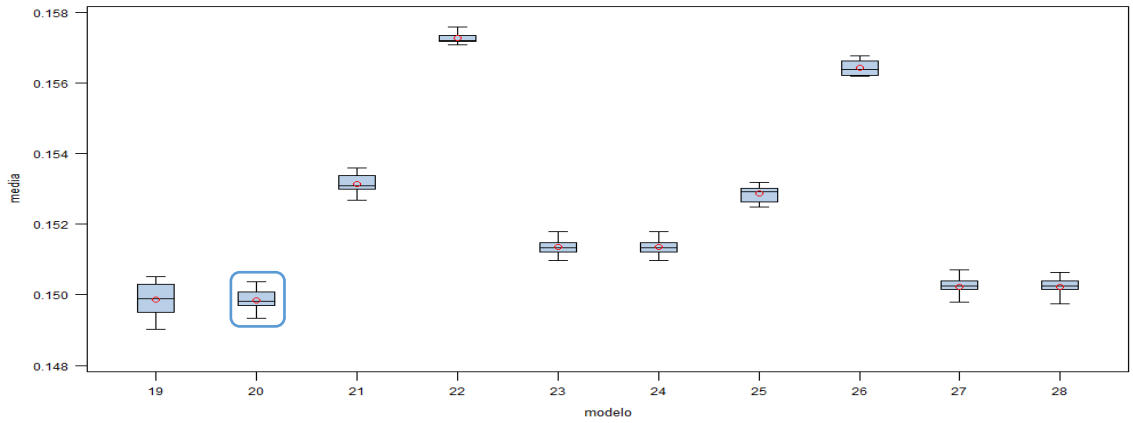


Configuración de modelos

Comparativa de modelos por grupo

Grupo 2:

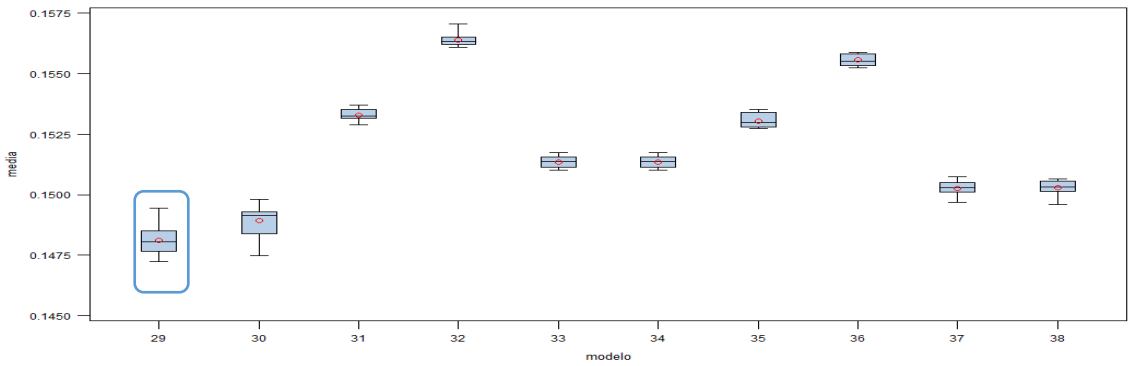
Tasa de fallos



El modelo ganador es el 20 para este grupo.

Grupo 3:

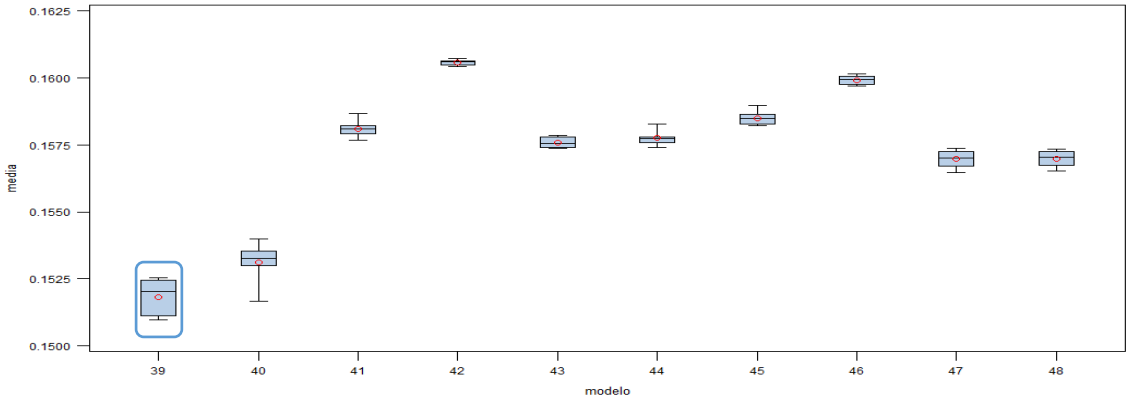
Tasa de fallos



El modelo ganador es el 29 para este grupo.

Grupo 4:

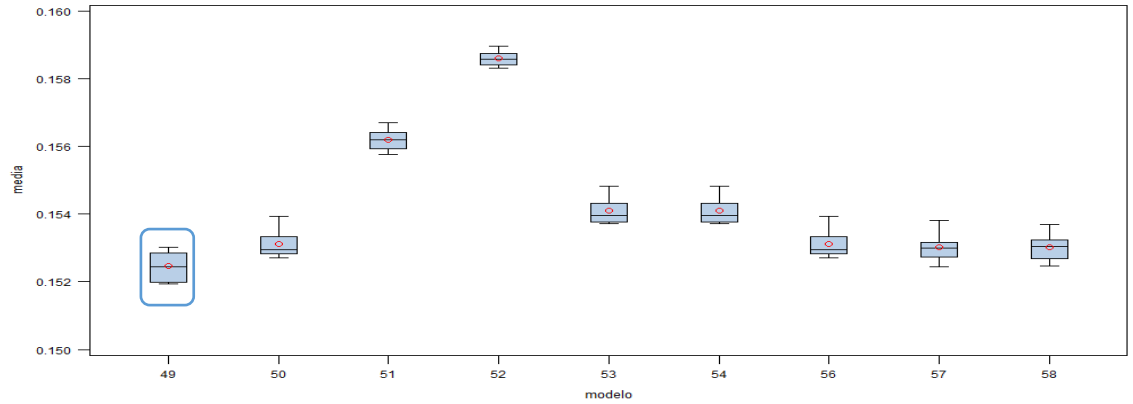
Tasa de fallos



El modelo ganador es el 39 para este grupo.

Grupo 5:

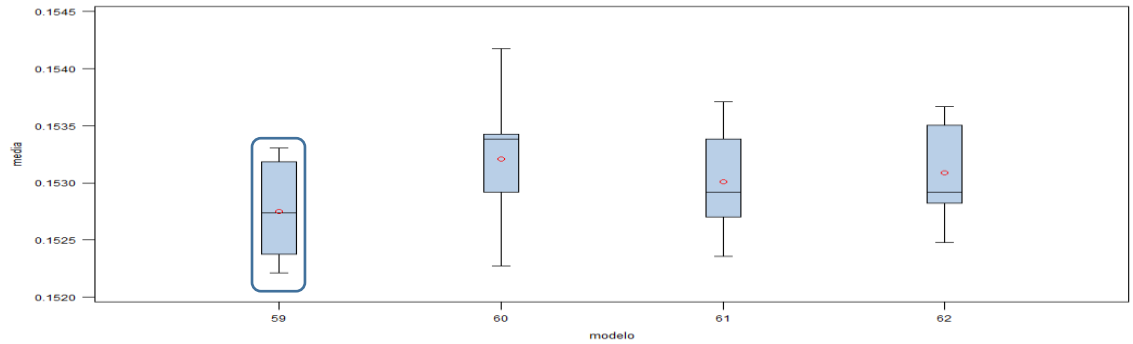
Tasa de fallos



El modelo ganador es el 49 para este grupo.

Grupo 6:

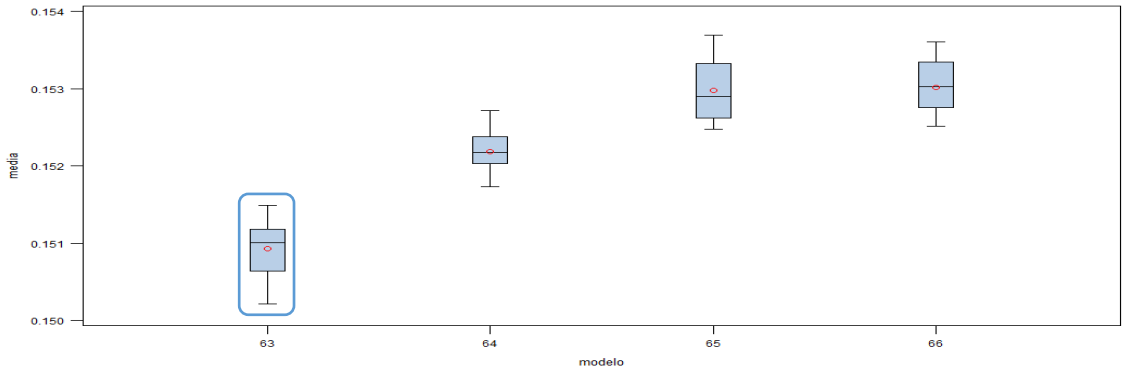
Tasa de fallos



El modelo ganador es el 59 para este grupo.

Grupo 7:

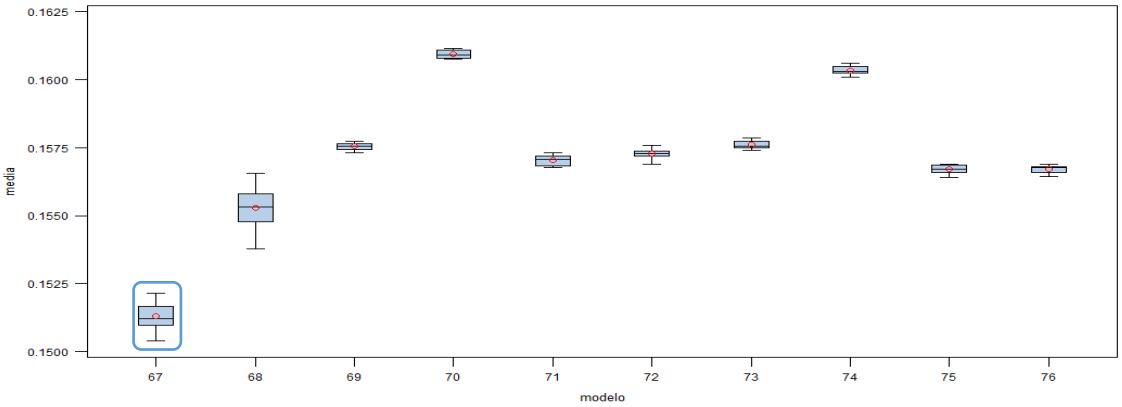
Tasa de fallos



El modelo ganador es el 63 para este grupo.

Grupo 8:

Tasa de fallos



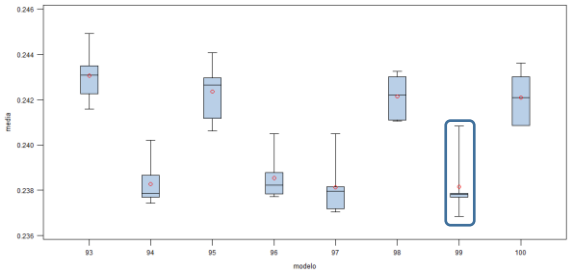
El modelo ganador es el 67 para este grupo.

Por lo general se aprecia que las mismas configuraciones de algoritmo para distintas variables muestran patrones similares. El algoritmo de optimización Levmar y función de activación tangente hiperbólica abanderan los mejores modelos en tasa de fallos para todos los grupos.

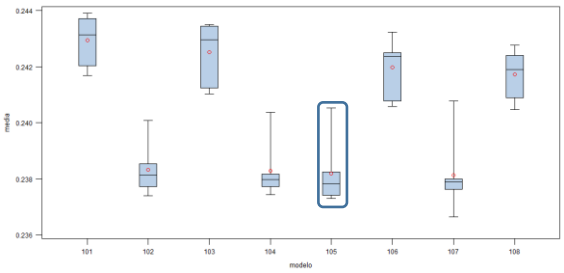
Bagging

Grupo 2:

P-valor: 0.1: modelo 99

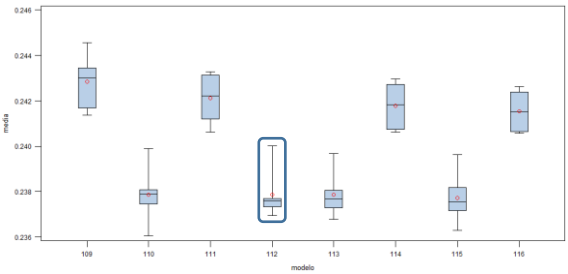


P-valor: 0.05: modelo 105

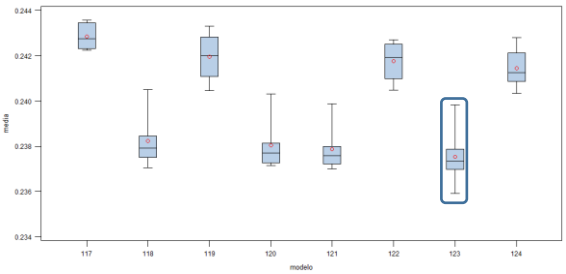


Grupo 3:

P-valor: 0.1: modelo 112

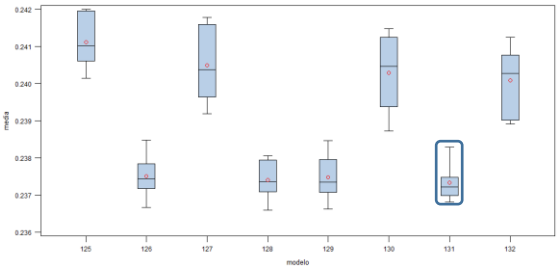


P-valor: 0.05: modelo 123

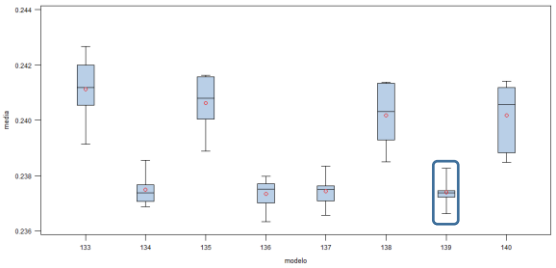


Grupo 4:

P-valor: 0.1: modelo 131

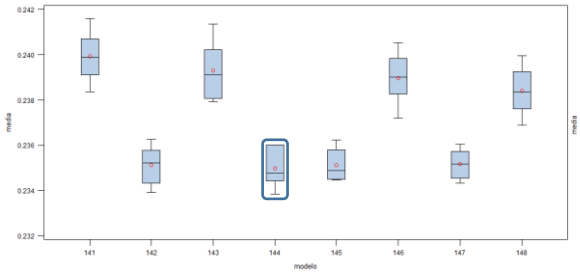


P-valor: 0.05: modelo 139

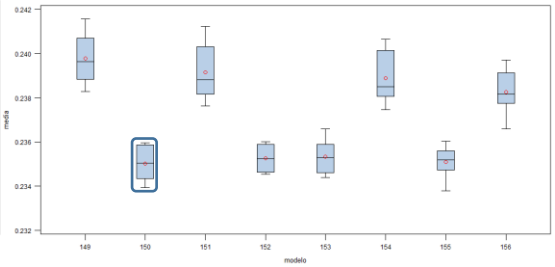


Grupo 5:

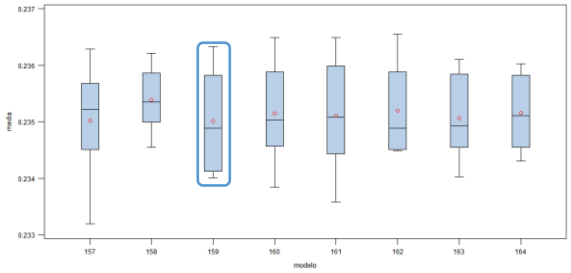
P-valor: 0.1: modelo 144



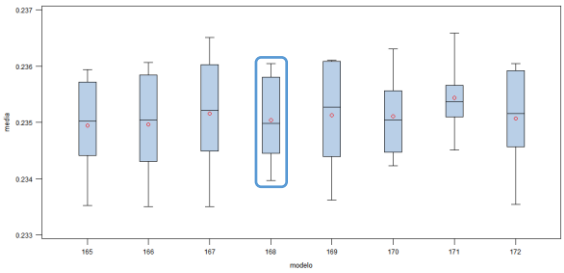
P-valor: 0.05: modelo 150



Grupo 6: Mejores modelos del grupo 5 con las variables del grupo 6: modelo 159

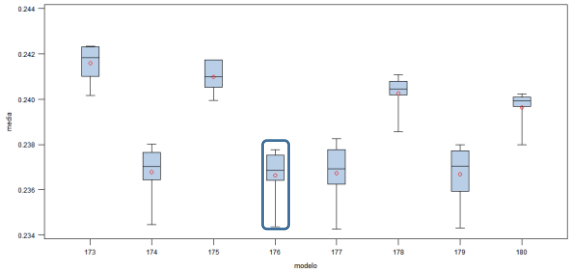


Grupo 7: Mejores modelos del grupo 5 con las variables del grupo 7: modelo 168

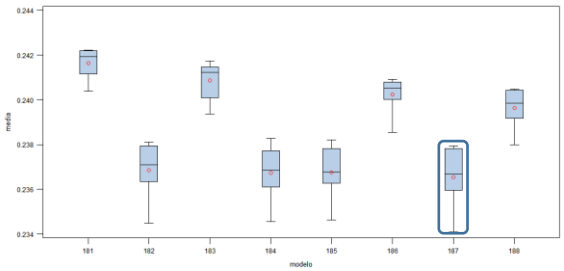


Grupo 8:

P-valor: 0.1: modelo 176



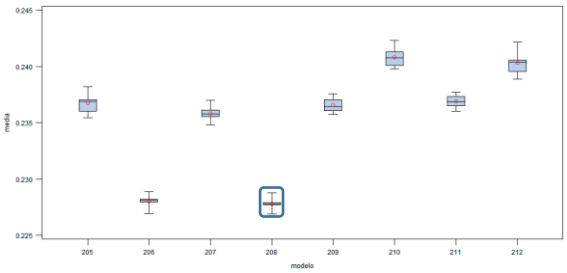
P-valor: 0.05: modelo 187



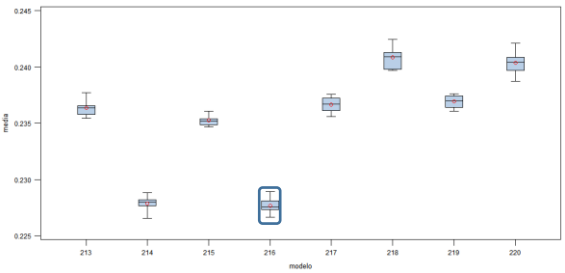
Random Forest

Grupo 2:

P-valor: 0.1: Modelo 208

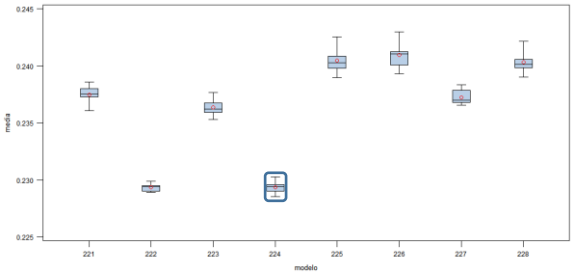


P-valor: 0.05: Modelo 216

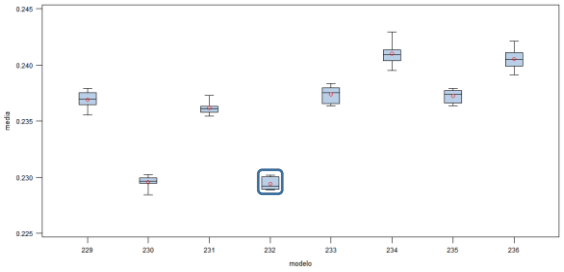


Grupo 3:

P-valor: 0.1: Modelo 224

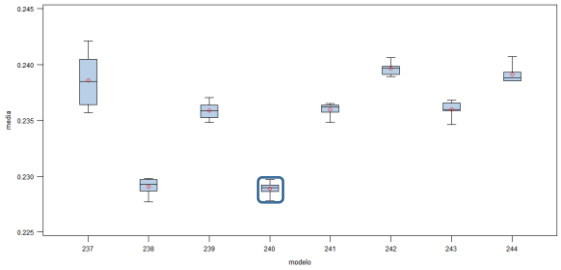


P-valor: 0.05: Modelo 232

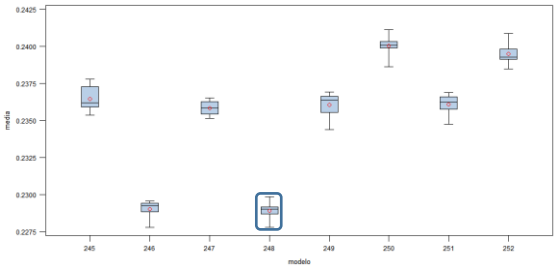


Grupo 4:

P-valor: 0.1: Modelo 240

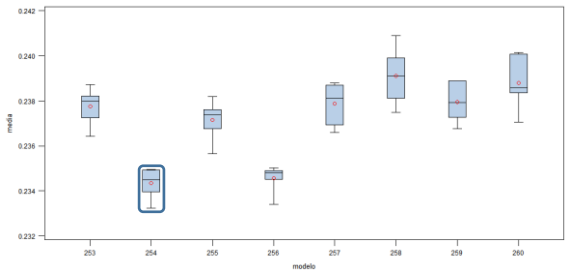


P-valor: 0.05: Modelo 248

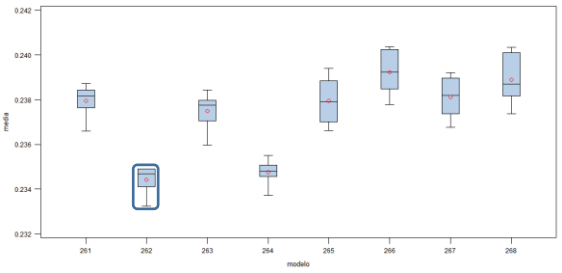


Grupo 5:

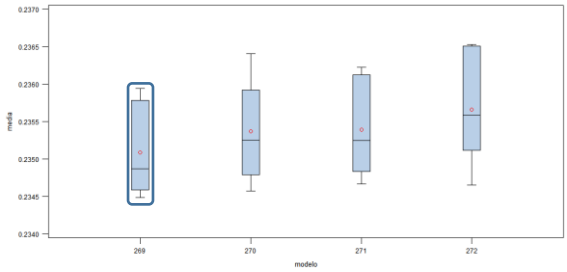
P-valor: 0.1: Modelo 254



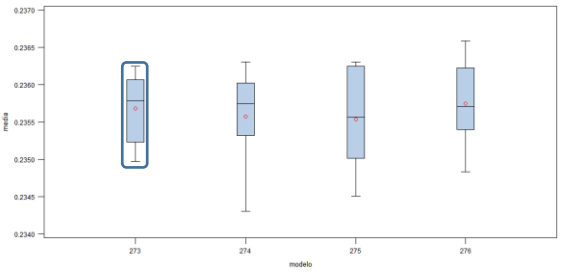
P-valor: 0.05: Modelo 262



Grupo 6: Mejores modelos del grupo 5 con las variables del grupo 6: Modelo 269

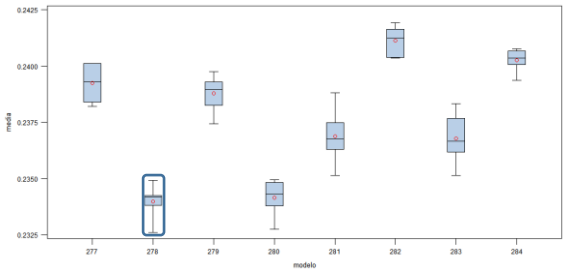


Grupo 7: Mejores modelos del grupo 5 con las variables del grupo 7: Modelo 273

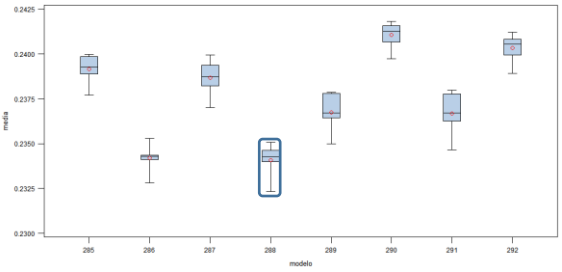


Grupo 8:

P-valor: 0.1: Modelo 278



P-valor: 0.05: Modelo 288

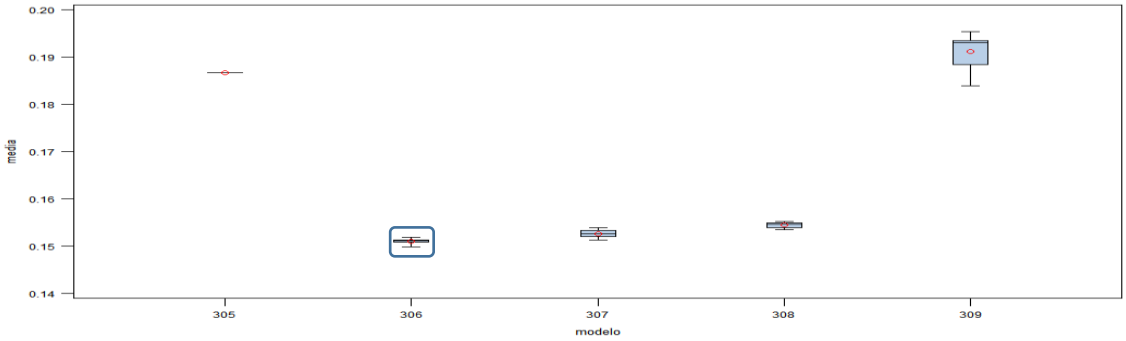


Gradient Boosting Machine

Grupo 2:

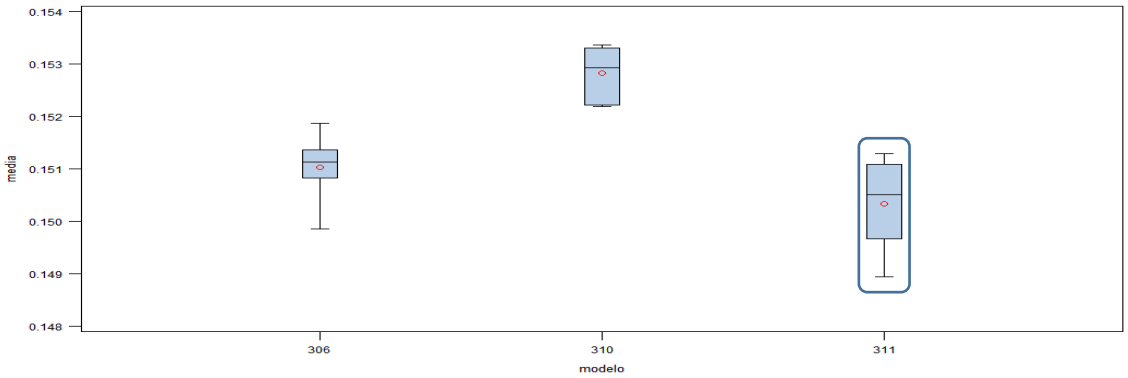
MODELOS:
305:leafsize=15,**iteraciones=400**,**shrink=0.001**,maxbranch=4,maxdepth=5,mincatsize=15,minob=20
306:leafsize=15,**iteraciones=300**,**shrink=0.01**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
307:leafsize=15,**iteraciones=200**,**shrink=0.05**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
308:leafsize=15,**iteraciones=100**,**shrink=0.1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
309:leafsize=15,**iteraciones=50**,**shrink=1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

Tasa de fallos:



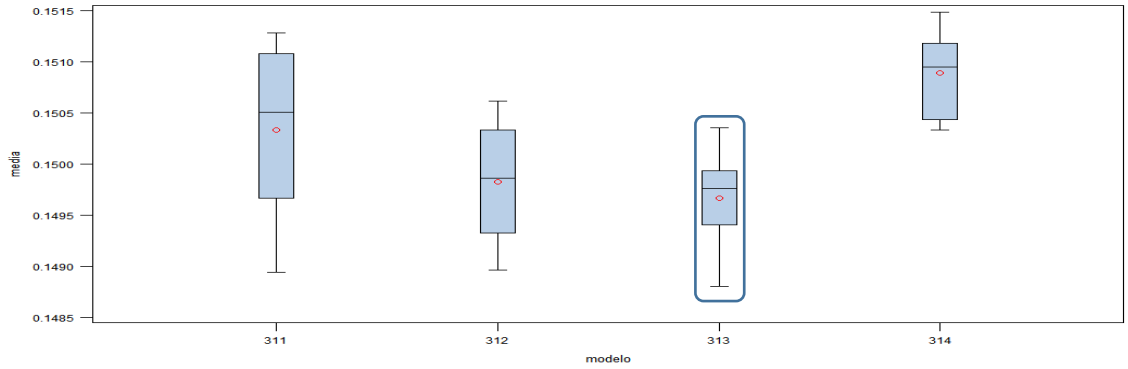
MODELOS:
310:leafsize=15,**iteraciones=200**,shrink=0.01,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
311:leafsize=15,**iteraciones=400**,shrink=0.01,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

Tasa de fallos:



MODELOS:
312:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=3**,mincatsize=15,minobs=20
313:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=4**,mincatsize=15,minobs=20
314:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=6**,mincatsize=15,minobs=20

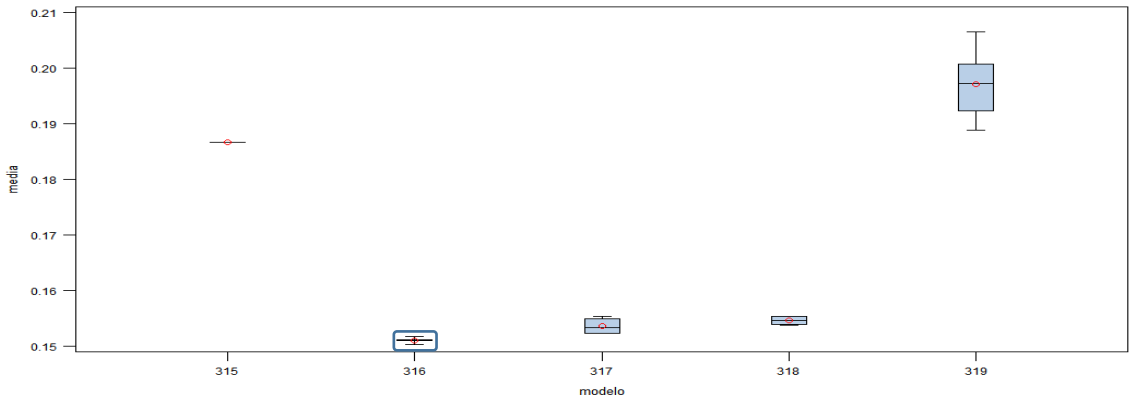
Tasa de fallos:



Grupo 3:

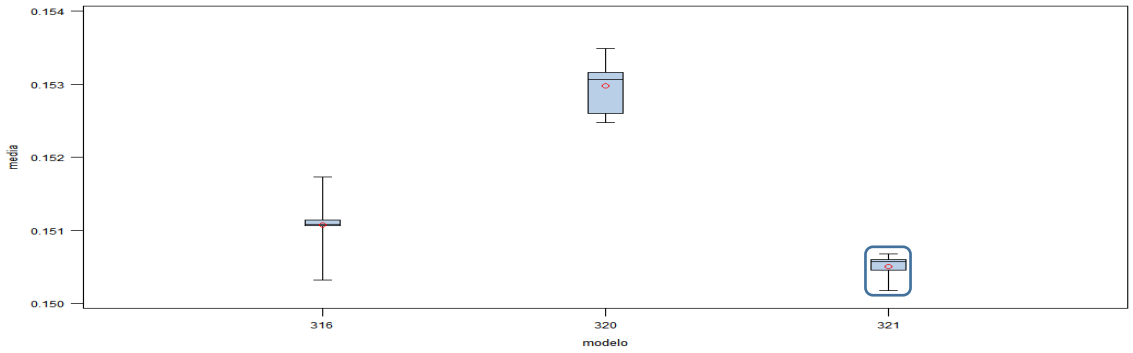
MODELOS:
315:leafsize=15,**iteraciones=400,shrink=0.001**,maxbranch=4,maxdepth=5,mincatsize=15,minob=20
316:leafsize=15,**iteraciones=300,shrink=0.01**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
317:leafsize=15,**iteraciones=200,shrink=0.05**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
318:leafsize=15,**iteraciones=100,shrink=0.1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
319:leafsize=15,**iteraciones=50,shrink=1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

Tasa de fallos:



MODELOS:
320:leafsize=15,**iteraciones=200**,shrink=0.01,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
321:leafsize=15,**iteraciones=400**,shrink=0.01,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

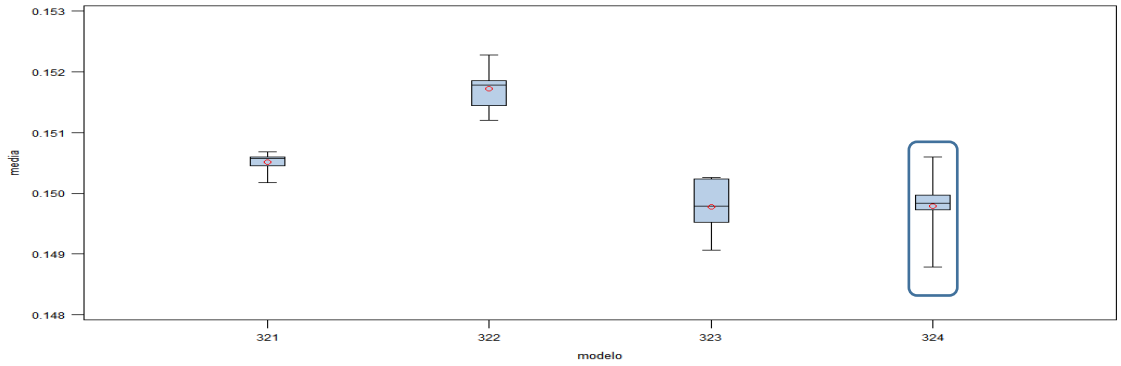
Tasa de fallos:



MODELOS:
322:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=3**,mincatsize=15,minobs=20
323:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=4**,mincatsize=15,minobs=20

324:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=6**,mincatsize=15,minobs=20

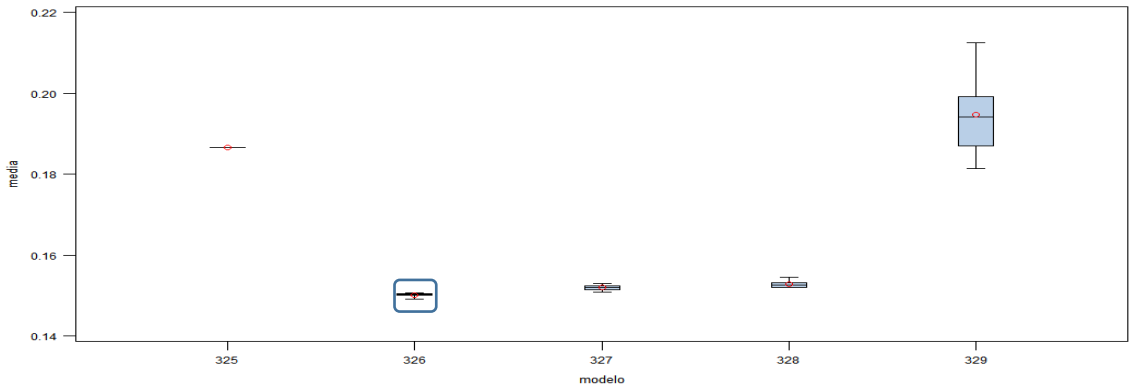
Tasa de fallos:



Grupo 4:

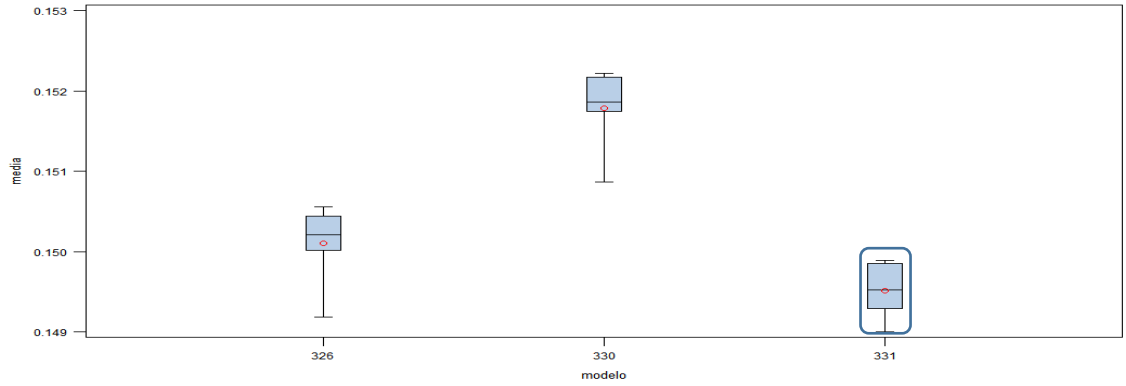
MODELOS:
325:leafsize=15,**iteraciones=400,shrink=0.001**,maxbranch=4,maxdepth=5,mincatsize=15,minob=20
326:leafsize=15,**iteraciones=300,shrink=0.01**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
327:leafsize=15,**iteraciones=200,shrink=0.05**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
328:leafsize=15,**iteraciones=100,shrink=0.1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
329:leafsize=15,**iteraciones=50,shrink=1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

Tasa de fallos:



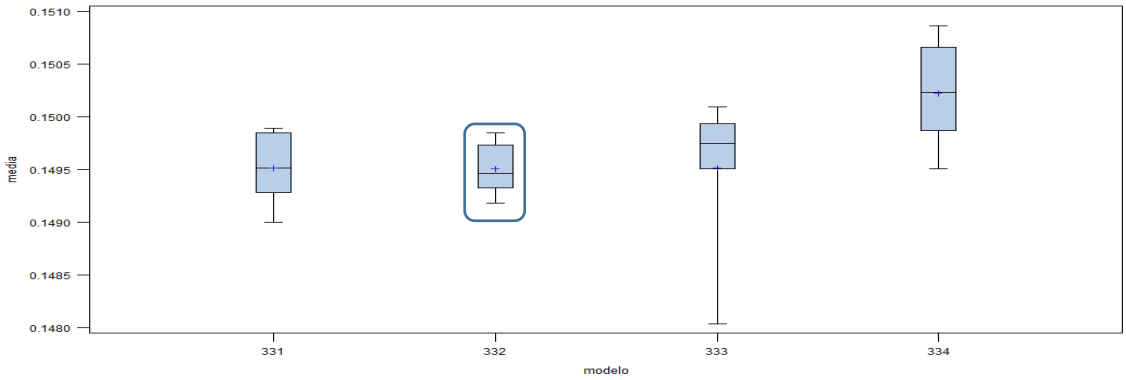
MODELOS:
330:leafsize=15,**iteraciones=200**,shrink=0.01,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
331:leafsize=15,**iteraciones=400**,shrink=0.01,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

Tasa de fallos:



MODELOS:
332:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=3**,mincatsize=15,minobs=20
333:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=4**,mincatsize=15,minobs=20
334:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=6**,mincatsize=15,minobs=20

Tasa de fallos:

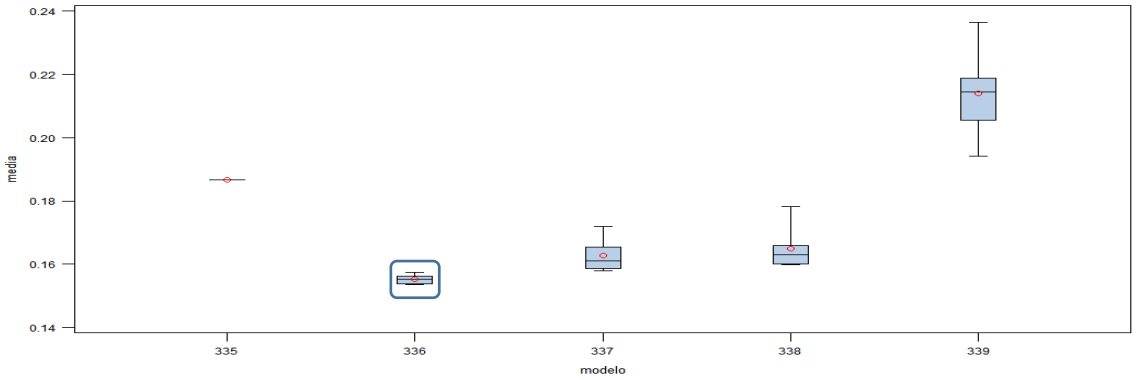


Grupo 5:

MODELOS:

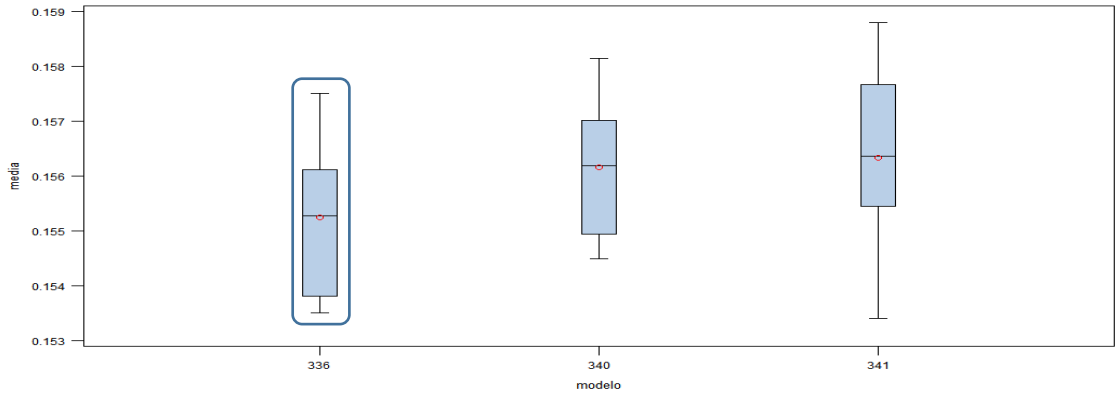
335:leafsize=7,**iteraciones=400,shrink=0.001**,maxbranch=4,maxdepth=5,mincatsize=15,minob=20
336:leafsize=7,**iteraciones=300,shrink=0.01**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
337:leafsize=7,**iteraciones=200,shrink=0.05**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
338:leafsize=7,**iteraciones=100,shrink=0.1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
339:leafsize=7,**iteraciones=50,shrink=1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

Tasa de fallos:



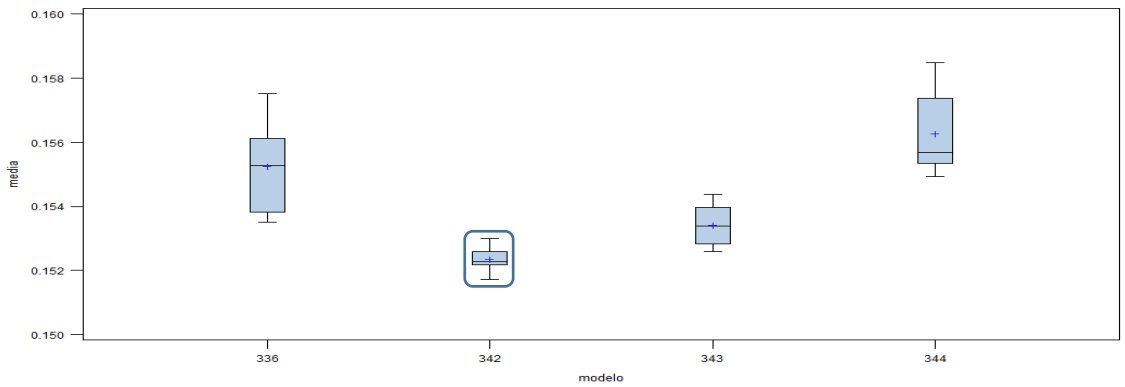
MODELOS:
340:leafsize=7,**iteraciones=200**,shrink=0.01,maxbranch=4,maxdepth=3,mincatsize=15,minobs=20
341:leafsize=7,**iteraciones=200**,shrink=0.01,maxbranch=4,maxdepth=4,mincatsize=15,minobs=20

Tasa de fallos:



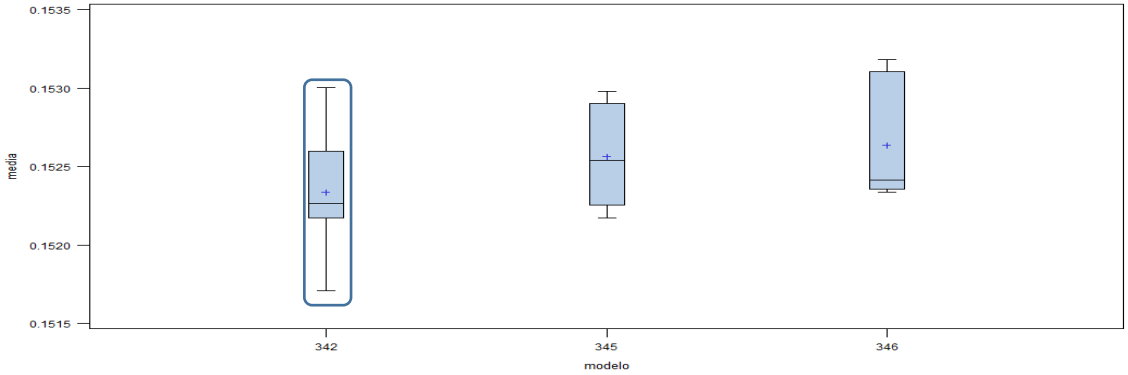
MODELOS:
342:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=3**,mincatsize=15,minobs=20
343:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=4**,mincatsize=15,minobs=20
344:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=6**,mincatsize=15,minobs=20

Tasa de fallos:



Grupos 6 y 7:

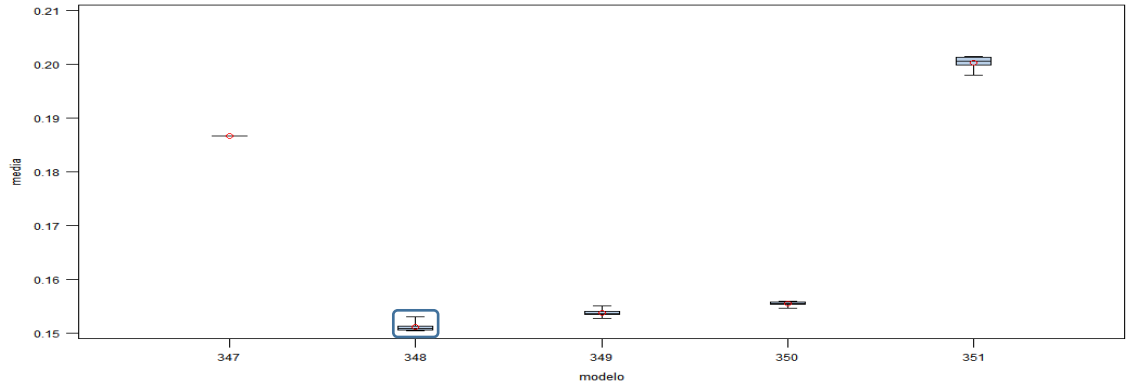
Tasa de fallos:



Grupo 8:

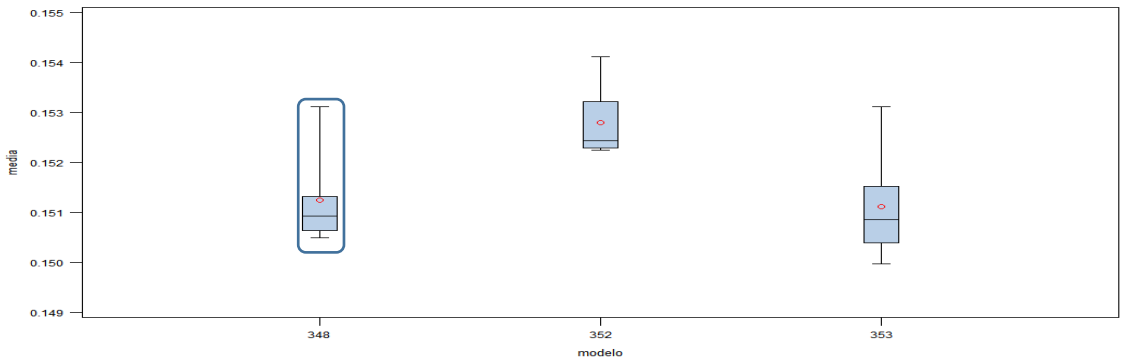
MODELOS:
347:leafsize=7,**iteraciones=400,shrink=0.001**,maxbranch=4,maxdepth=5,mincatsize=15,minob=20
348:leafsize=7,**iteraciones=300,shrink=0.01**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
349:leafsize=7,**iteraciones=200,shrink=0.05**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
350:leafsize=7,**iteraciones=100,shrink=0.1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20
351:leafsize=7,**iteraciones=50,shrink=1**,maxbranch=4,maxdepth=5,mincatsize=15,minobs=20

Tasa de fallos:



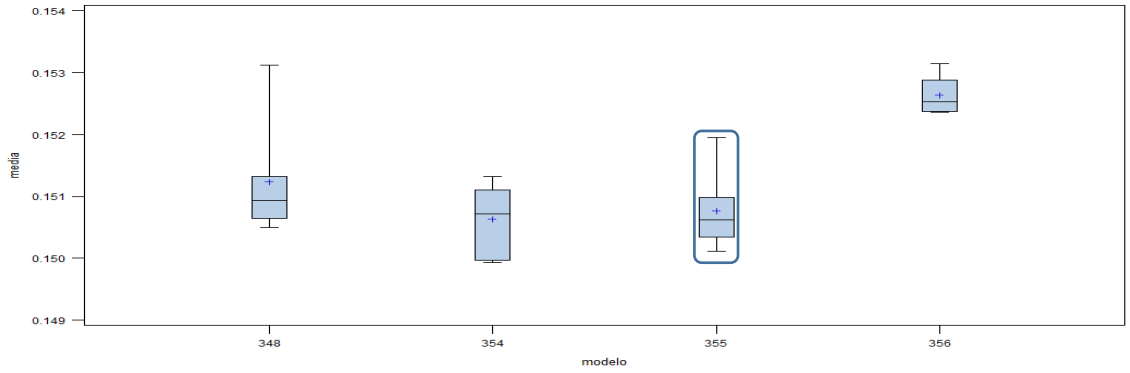
MODELOS:
352:leafsize=7,**iteraciones=200**,shrink=0.01,maxbranch=4,maxdepth=3,mincatsize=15,minobs=20
353:leafsize=7,**iteraciones=200**,shrink=0.01,maxbranch=4,maxdepth=4,mincatsize=15,minobs=20

Tasa de fallos:



MODELOS:
354:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=3**,mincatsize=15,minobs=20
355:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=4**,mincatsize=15,minobs=20
356:leafsize=15,iteraciones=200,shrink=0.01,maxbranch=4,**maxdepth=6**,mincatsize=15,minobs=20

Tasa de fallos:

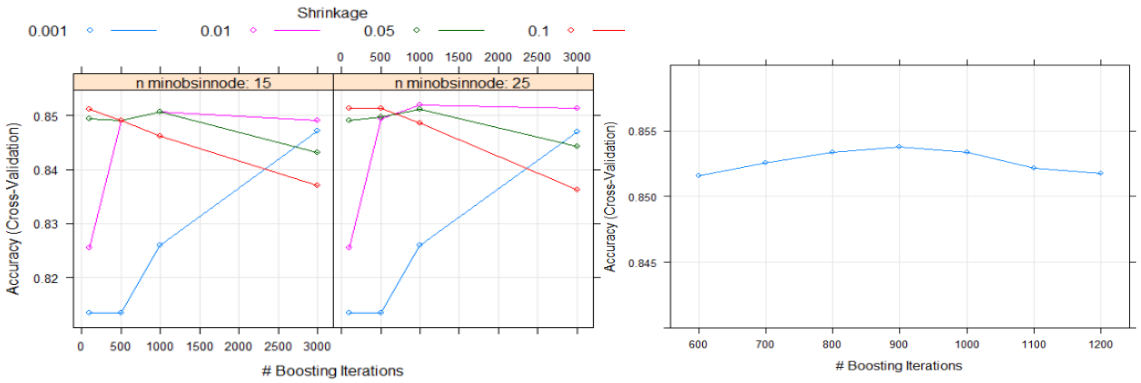


Anexo VI. Estudio de los híper parámetros de los distintos grupos de variables en RStudio

Gradient Boosting Machine

Grupo 2:

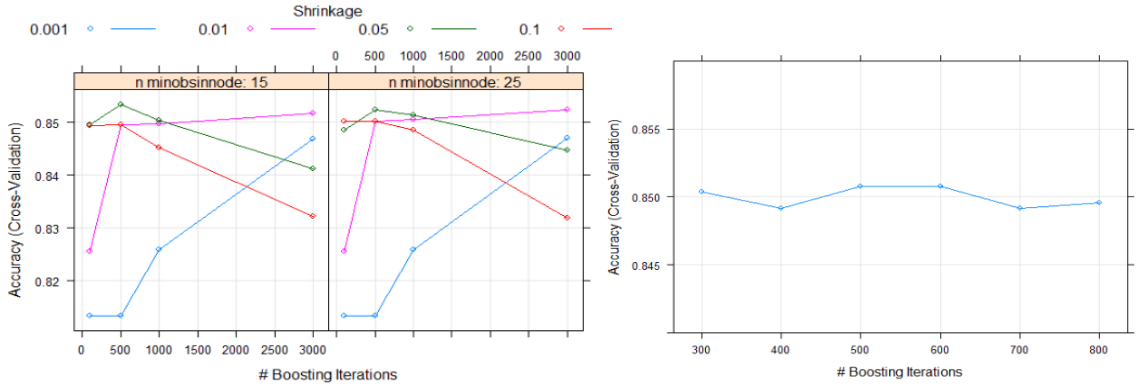
```
The final values used for the model were n.trees = 1000, interaction.depth = 2, shrinkage =
 0.01 and n.minobsinnode = 25.
```



La primera recomendación de R Studio es una constante de regularización de 0,01 y un tamaño mínimo de nodos de 25. Afinando la búsqueda de árboles vemos como el número ideal de iteraciones es de 900.

Grupo 3:

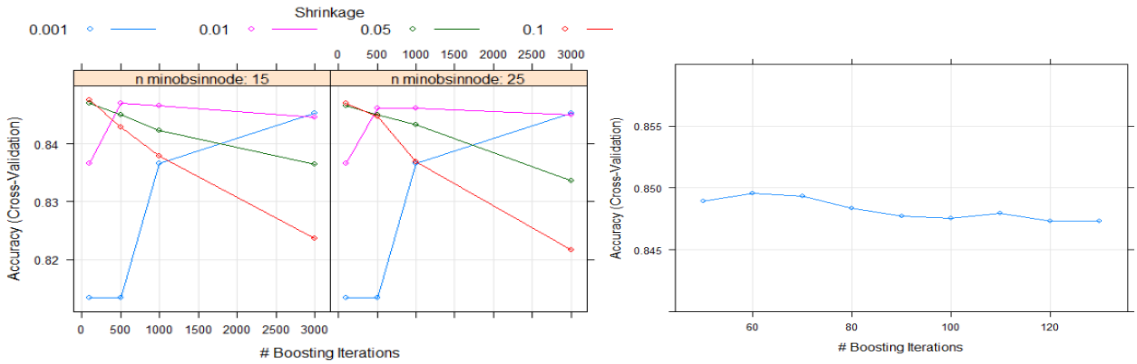
```
The final values used for the model were n.trees = 500, interaction.depth = 2, shrinkage = 0.05
 and n.minobsinnode = 15.
```



Para este caso, configuramos los parámetros que nos recomienda el programa.

Grupo 4:

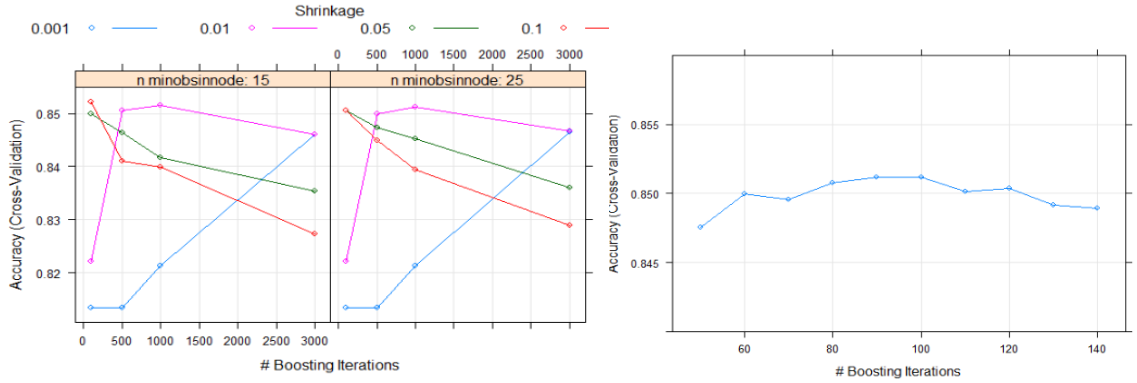
```
The final values used for the model were n.trees = 100, interaction.depth = 2, shrinkage = 0.1
 and n.minobsinnode = 15.
```



Para este caso, puliendo el resultado, se aprecia que 60 son las iteraciones que han de llevarse a cabo.

Grupos 5, 6 y 7:

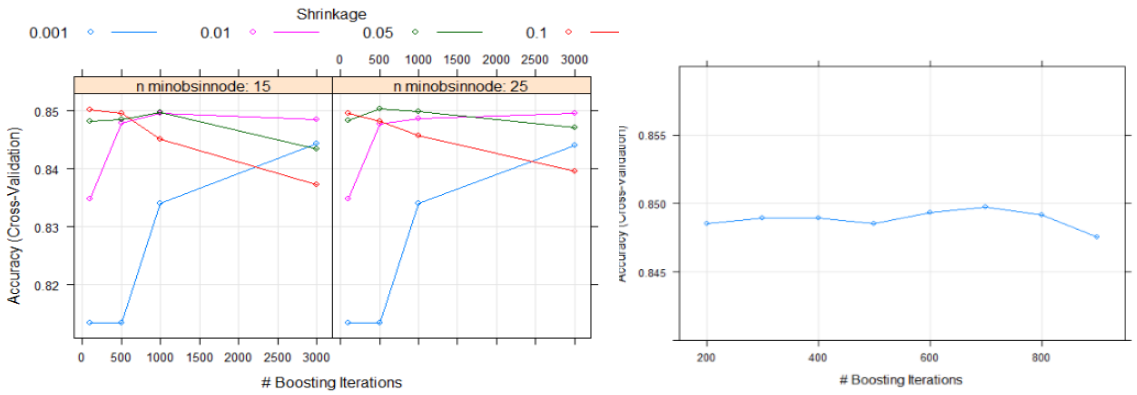
```
The final values used for the model were n.trees = 100, interaction.depth = 2, shrinkage = 0.1
 and n.minobsinnode = 15.
```



Usaremos, para las variables de los grupos 5, 6 y 7 un *shrinkage* de 0,1, un tamaño mínimo de nodos de 15 y 90 árboles.

Grupo 8:

```
The final values used for the model were n.trees = 500, interaction.depth = 2, shrinkage = 0.05
 and n.minobsinnode = 25.
```

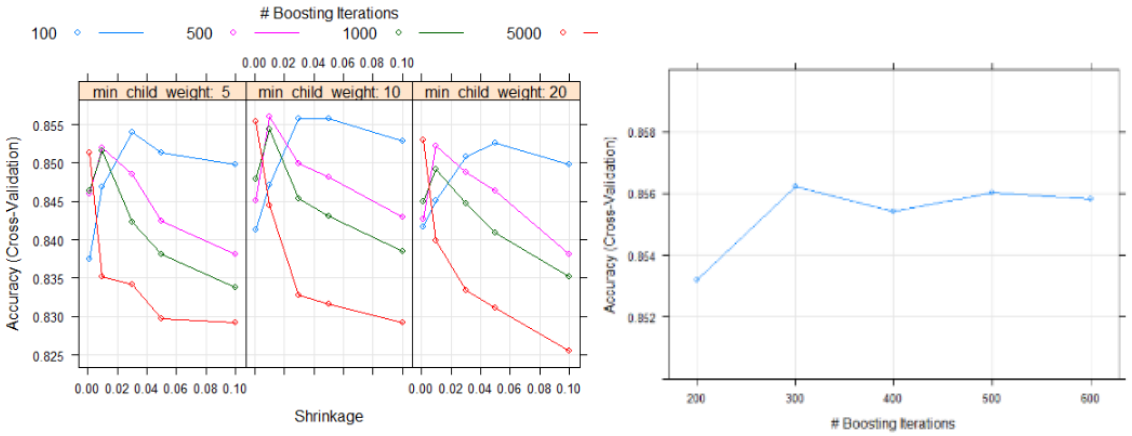


En este caso, utilizaremos una tasa de regularización de 0,05, un tamaño mínimo de nodos de 15 y 700 árboles.

Extreme Gradient Boosting Machine

Grupo 2:

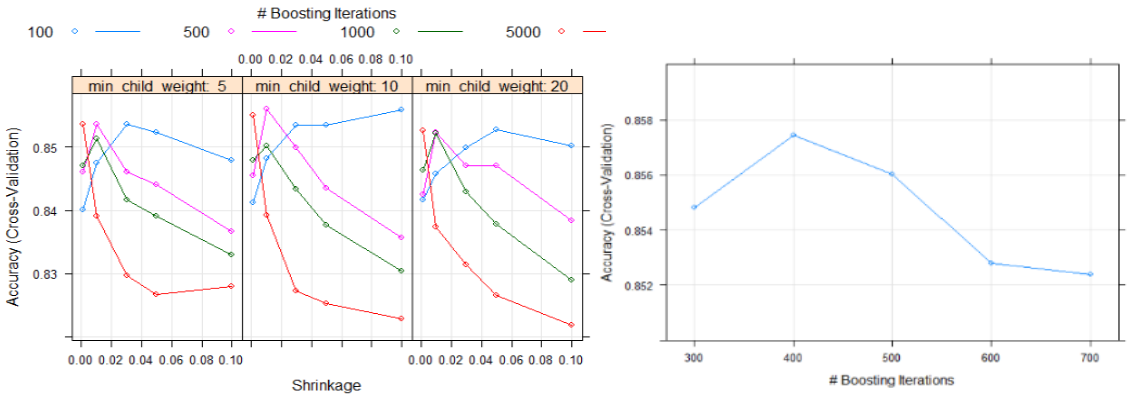
```
The final values used for the model were nrounds = 500, max_depth = 6, eta = 0.01, gamma =
 0, colsample_bytree = 1, min_child_weight = 10 and subsample = 1.
```



Para las variables del grupo 2 se usan los valores que saca R, aunque el número de árboles se reduce a 300 debido a que en el estudio de *early stopping* genera mejores resultados.

Grupo 3:

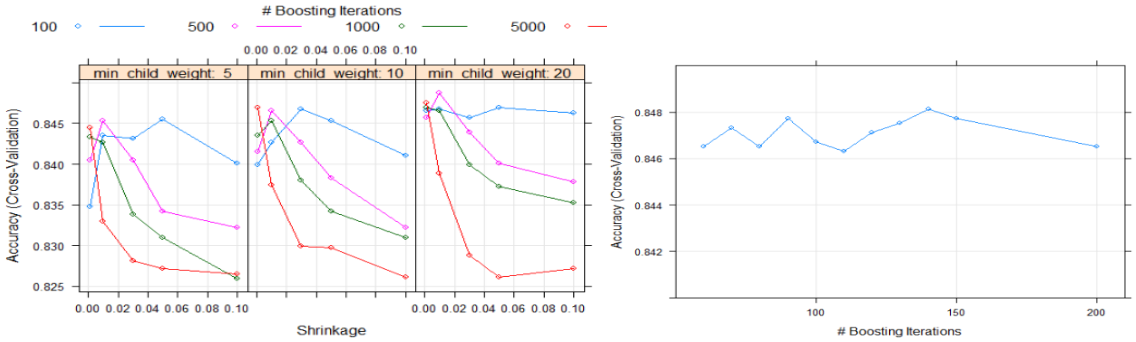
```
The final values used for the model were nrounds = 500, max_depth = 6, eta = 0.01, gamma =
 0, colsample_bytree = 1, min_child_weight = 10 and subsample = 1.
```



Para este caso, afinando el resultado, se ve que el número de iteraciones ideal es de 400.

Grupo 4:

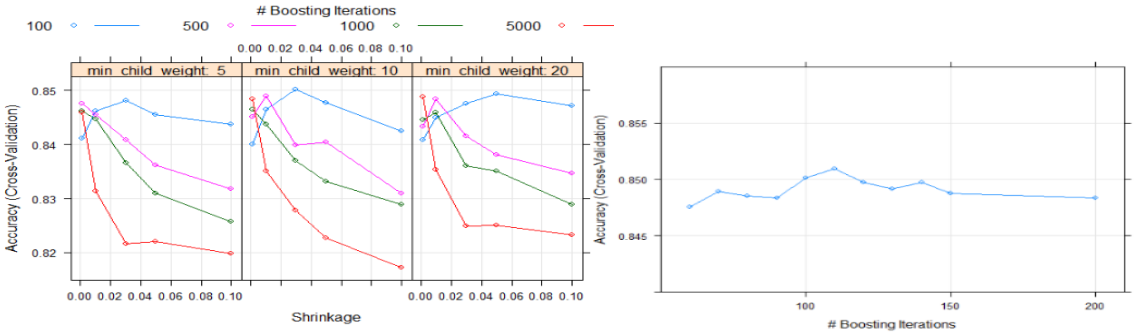
```
The final values used for the model were nrounds = 500, max_depth = 6, eta = 0.01, gamma =
 0, colsample_bytree = 1, min_child_weight = 20 and subsample = 1.
```



El número de iteraciones idóneo para este grupo es 140.

Grupos 5, 6 y 7:

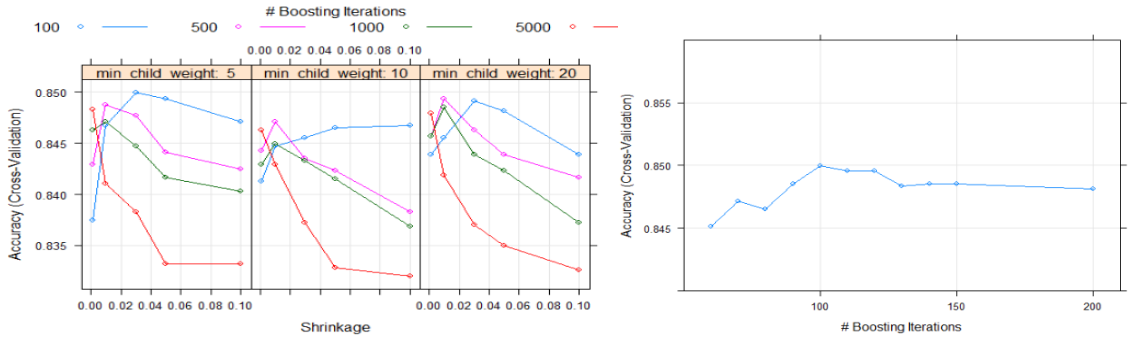
```
The final values used for the model were nrounds = 100, max_depth = 6, eta = 0.03, gamma =
 0, colsample_bytree = 1, min_child_weight = 10 and subsample = 1.
```



Los grupos de este sub-apartado optimizan la tasa de exactitud con 110 vueltas.

Grupo 8:

```
The final values used for the model were nrounds = 100, max_depth = 6, eta = 0.03, gamma =
 0, colsample_bytree = 1, min_child_weight = 5 and subsample = 1.
```

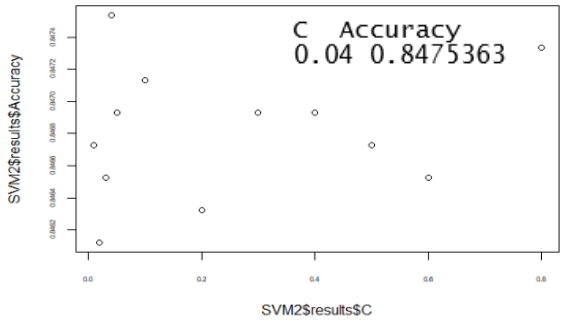


Para el grupo 8, se utilizan los híper parámetros que recomienda R Studio.

Support Vector Machine

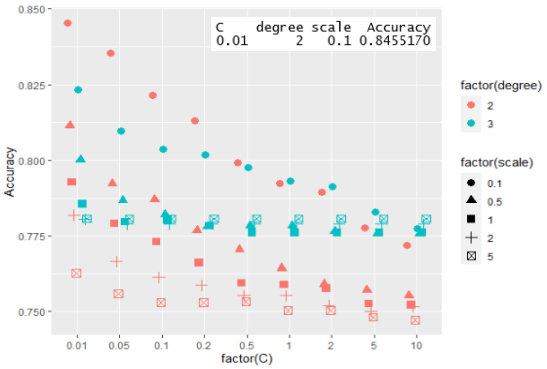
Grupo 2:

La constante 0,04 garantiza la mejor exactitud posible según los parametro analizados en el kernel lineal:

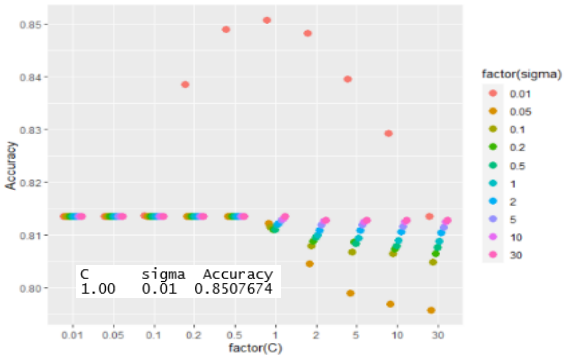


Analizamos ahora los otros dos kernels y utilizamos la salida de R para la configuración de ambos modelos:

SVM polinomial

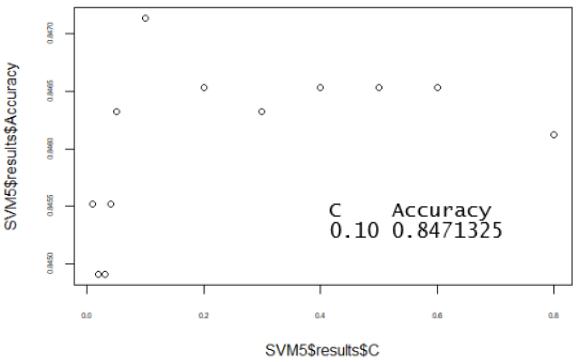


SVM *Radial Basis Function*

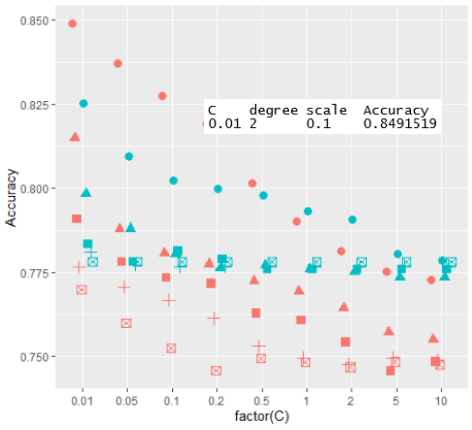


Grupo 3:

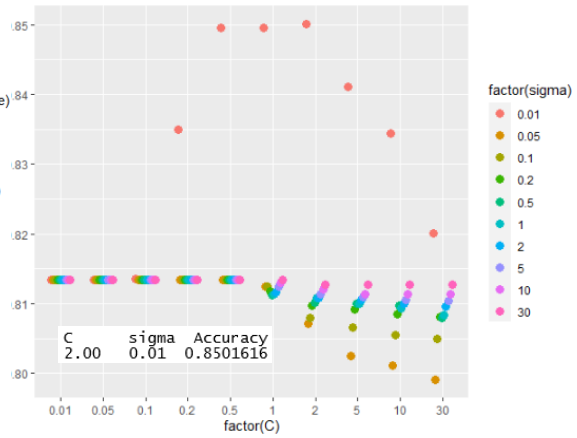
Para este grupo de variables, en SVM lineal configuramos un factor 0 de 0,1.



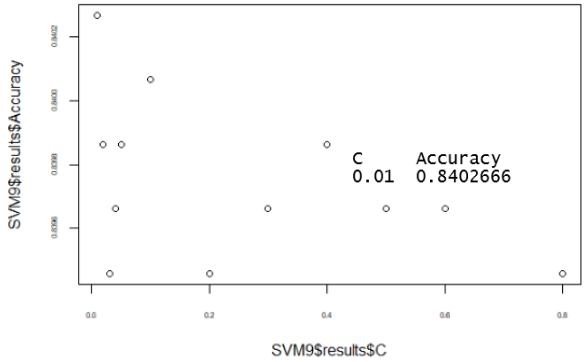
SVM polinomial



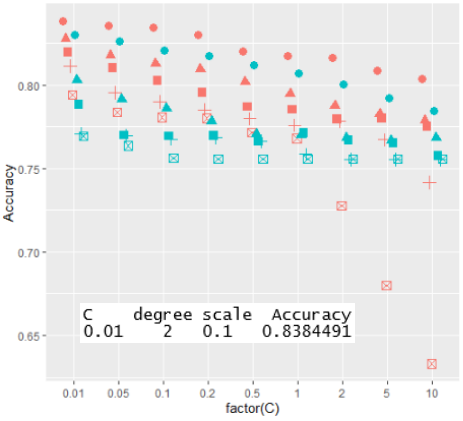
SVM *Radial Basis Function*



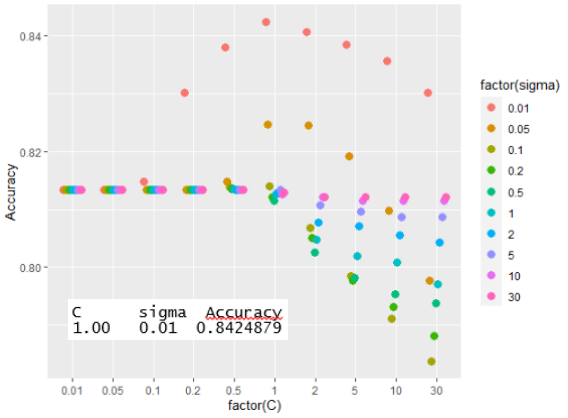
Grupo 4:



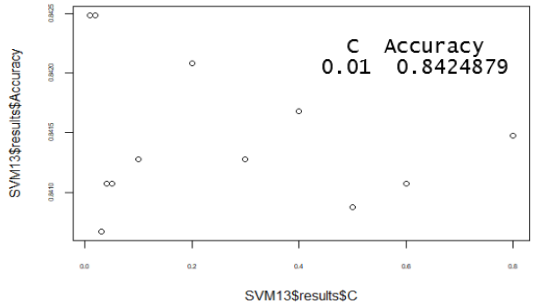
SVM polinomial



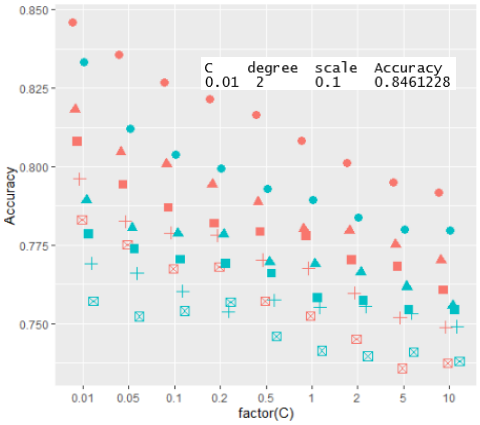
SVM *Radial Basis Function*



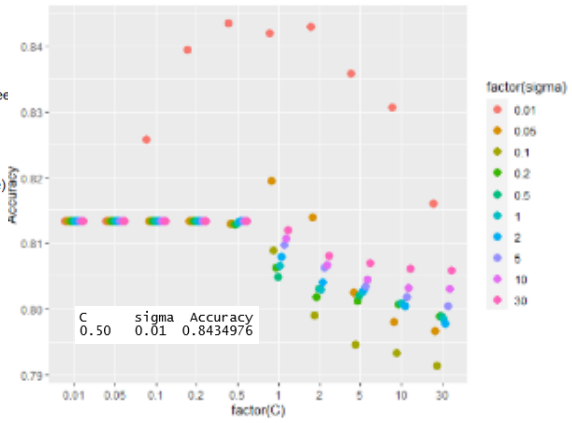
Grupos 5, 6 y 7:



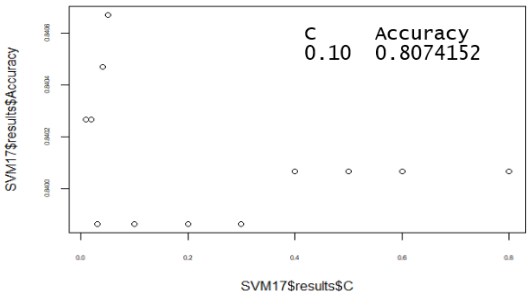
SVM polinomial



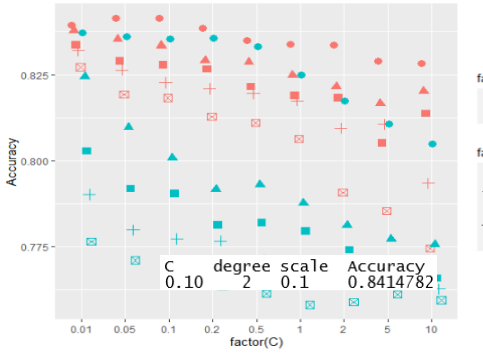
SVM *Radial Basis Function*



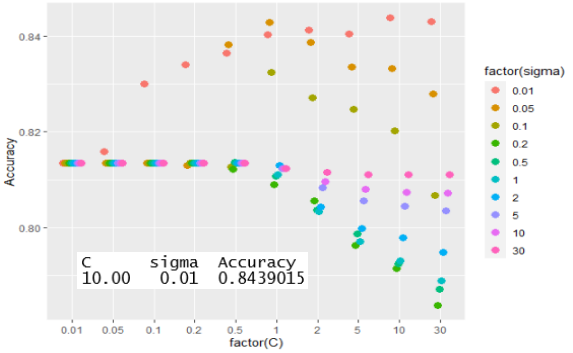
Grupo 8:



SVM polinomial



SVM *Radial Basis Function*



Anexo VII. Código

https://github.com/mariamoama/TFM-Maria-Otero-Alonso.git