FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2018/2019

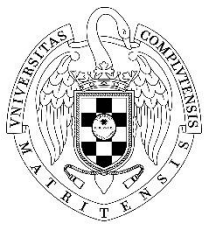
Trabajo de Fin de Máster

TITULO: Predicción del coste de un lesionado por accidente de tráfico en España utilizando técnicas de minería de datos

***Alumno*: Angélica María Bustos González**

***Tutor*: Juana María Alonso Revenga**

Septiembre de 2019



UNIVERSIDAD COMPLUTENSE
MADRID

Tabla de contenido

Índice de figuras

Índice de tablas

1. Introducción

Los accidentes de tráfico tienen un enorme impacto en la realidad social y económica de un país y se han convertido en un problema a nivel mundial, alcanzando en la mayoría de los países hasta 3% del producto interno bruto [1], en España ronda los 10.000 millones de euros anuales, el equivalente al 1% de su PIB según la OCDE [2]. Esto no solo repercute en pérdidas a nivel económico sino también a nivel humano, debido a que cada año los accidentes de tráfico causan la muerte de aproximadamente 1.25 millones de personas en todo el mundo y alrededor de 40 millones de personas resultan con lesiones según cifras de la OMS [2].

En España la situación no es ajena, el número de heridos ha presentado un leve crecimiento en los últimos años mientras que el número de fallecidos registraba un importante descenso hasta el año 2013, a partir de este momento se ha mantenido constante.

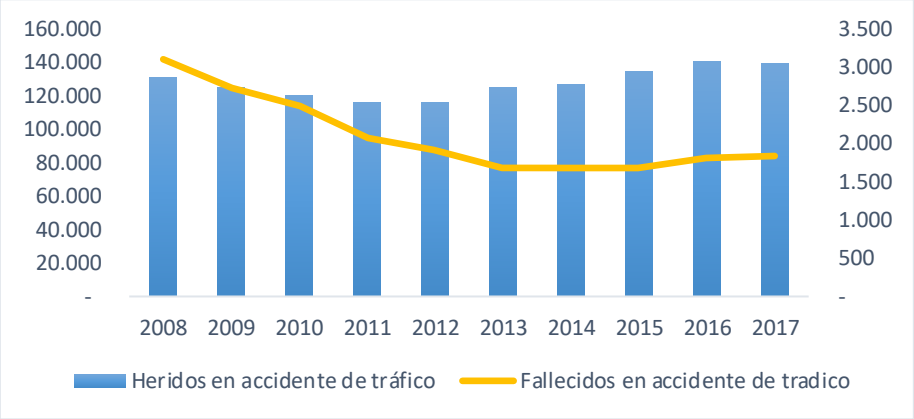


Figura 1. Lesionados por accidentes de tráfico en España (2008-2017) fuente: DGT

Para calcular la cuantía de las indemnizaciones derivadas de un daño corporal producido por un accidente de tráfico en España desde el año 2016 rige la ley 35/2015, la cual es una reforma del sistema de valoración de los daños y perjuicios más conocido como baremo que entró por primera vez en vigor en el año 1995. Tiene como finalidad realizar una reparación económica del daño ocasionado por un accidente de tráfico, es decir busca situar a la víctima en una posición lo más parecida posible a la que tendría de no haberse producido el accidente.

El principio de reparación integra daños y perjuicios de tres tipos: Indemnizaciones por lesiones temporales, por lesiones permanentes (secuelas) y por causa de muerte, donde se tiene en cuenta el perjuicio básico, particular y patrimonial. A criterio médico se valora el tiempo de curación, si hay daño funcional o estético, si han quedado impedimentos permanentes que impiden trabajar o desarrollar otras actividades sociales o lúdicas, que se traduce en pérdida de calidad de vida y afectaciones a nivel económico, etc. Esta cuantificación se expresa en puntos dependiendo de la gravedad e intensidad del daño,

también se tienen en cuenta otros factores como el número de actividades afectadas, la importancia, la edad del lesionado, etc.

1.1 Descripción del problema

Las empresas aseguradoras anualmente cubren grandes sumas de dinero en daños corporales, donde inicialmente realiza una previsión de lo que se considera va a ser la cuantía por cubrir de acuerdo con una valoración médica inicial. Sin embargo, en este proceso se congelan altas sumas de dinero por lo que es vital tener provisiones financieras lo más ajustadas a la realidad, lo que financieramente es crucial para el flujo de caja de la compañía.

El desarrollo de esta investigación se lleva a cabo en una empresa del sector de seguros en España. La variable objetivo corresponde a lo que la empresa aseguradora tiene que pagar por lesiones médicas tanto temporales como permanentes. Entre los factores a estudiar se tienen en cuenta no solo las características del lesionado como la edad, el sexo o situación laboral, si no también características del siniestro como el tipo de vehículo, la marca, el modelo, la provincia en la que se produjo el accidente además de variables particulares de la aseguradora como es el valorador médico.

Por lo tanto, el contenido de este trabajo se estructura de la siguiente manera:

En primer lugar, se obtienen los datos mediante la explotación de la base de datos, utilizando lenguaje de consulta estructurada SQL, donde se obtienen todas las variables que posiblemente influyen en el coste.

En segundo lugar, se realiza una exploración y depuración de la base datos, donde se identifica datos erróneos, atípicos, se realiza transformación de variables, etc.

Posteriormente se realizan diferentes modelos como regresión lineal, redes neuronales, random forest, gradient boosting y ensamblado para tres diferentes formas de predecir el coste, el primero es con la variable objetivo sin transformación, la segunda es un modelo de clasificación definiendo un punto de corte para posteriormente realizar dos modelos, uno cuando sea menor a ese punto de corte y otro en caso contrario y la tercera forma es con la variable objetivo-transformada. En la próxima sección se detallará la razón de probar con estas opciones.

Finalmente se realiza una comparación entre los modelos y se escoge el que mejor predice la variable objetivo mediante validación cruzada repetida.

1.2 Estado del arte

La minería de datos ha sido fundamental y ampliamente utilizada para abordar los temas asociados a los accidentes de tráfico. La mayoría de las investigaciones que se han llevado a cabo han sido enfocado a estudiar los factores claves relacionados en un accidente de tráfico, como lo son el tipo de vía, las condiciones meteorológicas, el tipo de vehículo, el perfil del conductor y factores relacionados con el lugar, el día de la semana, mes, etc.

Los modelos estadísticos tienen la finalidad de poder predecir la probabilidad de que ocurra un siniestro, así como los impactos que puede llegar a tener, donde finalmente se logren diseñar estrategias y políticas que ayuden a mitigar o disminuir el número de accidentes.

Entre las investigaciones se encuentra un estudio cuyo objetivo es estimar el número de personas lesionadas y fallecidas en accidentes de tránsito mediante la utilización de Redes neuronales artificiales, entre los principales resultados se destaca las causas más frecuentes de accidentes para diferentes perfiles de víctima. [3]. Otra investigación utiliza modelos tradicionales como la regresión lineal para predecir los accidentes de tráfico especialmente se centran en analizar perfiles jóvenes de conductores.[4].

Por su parte, un estudio que analiza los accidentes en el Reino unido y Australia cuyo objetivo es analizar la gravedad de las consecuencias de los accidentes, utiliza arboles de decisión como principal técnica, debido a que son fácilmente interpretables, pueden trabajar con grandes volúmenes de datos y permiten descubrir interacciones entre los datos [5] . En otro estudio, utilizan redes bayesianas para clasificar los accidentes de tráfico en función a la gravedad de la lesión, debido a que este algoritmo es capaz de predecir sin necesidad de suposiciones previas. Este estudio analizo 1536 accidentes en España, en el cual se concluyó que los principales factores que influyen en un accidente que involucran personas fallecidas son la edad del conductor, iluminación, tipo de accidente, etc. [6].

Por otro lado, se analiza un estudio relacionado con la clasificación del grado de severidad en el cual resultan las personas involucradas en accidentes de tráfico utilizando para ello máquinas de soporte vectorial (SVM) por sus siglas en inglés, que reconocen patrones basado en la metodología de aprendizaje supervisado, combinándolo con un algoritmo de optimización, Enjambres de Partículas (PSO) cuyo objetivo es estimar los mejores parámetros para la máquina clasificadora, los resultados de la investigación concluye que el mejor algoritmo que clasifica la gravedad de un accidente de tráfico es SVM con factor de inercia lineal con una tasa de acierto del 82%. [7].

Como se observa, en general las investigaciones en esta rama se han centrado en estudiar los factores, perfiles de víctimas, gravedad de los accidentes, etc. No obstante, lo más similar en cuanto a predicción del coste son los modelos que se han desarrollado en entidades prestadores de servicios de salud (EPS) que en el contexto español son los

centros de atención primarios (CAP). Estos modelos de predicción de costes en servicios de salud utilizan la simulación discreta, donde se evalúa diferentes escenarios como consulta médica general, urgencias, hospitalización y enfermedades catastróficas, entre otros. Los resultados obtenidos con el modelo permitieron determinar que la simulación para predecir el coste es efectivo, además realizaron una comparación entre el coste simulado vs coste promediado y se encontró que la predicción del costo total mediante el empleo de promedios puede llegar a tener altas variaciones en valor y porcentaje respecto a la predicción con simulación, dentro de intervalos estadísticamente aceptables [8].

Finalmente, una investigación realizada en la comunidad de Valencia evaluó un modelo de predicción del gasto farmacéutico en atención primaria de salud basado en variables demográficas, donde utiliza principalmente la regresión lineal para explicar la variabilidad del gasto farmacéutico ambulatorio, aunque el modelo obtenido no presento un resultado muy satisfactorio al tener un R cuadrado de 0.34. sí supone un avance en relación con modelos anteriores [9].

2. Objetivos y Metodología

2.1 Objetivos

El objetivo principal es predecir el coste de un lesionado como consecuencia de un accidente de tráfico en España, para conseguir el objetivo principal, se debe desarrollar los siguientes objetivos secundarios: Extracción de información del ERP, preparación y análisis exploratorio de las variables, realización de modelos de predicción y finalmente se realiza una evaluación y comparación de los modelos.

2.2 Metodología

Para alcanzar los objetivos establecidos, se propone utilizar la metodología desarrollada por SAS Institute, SEMMA[10], el cual es el acrónimo de las cinco fases: Sample, Explore, Modify, Model, Asses que si se traduce al castellano se refiere a muestrear, explorar, modificar, modelizar y evaluar. Sin embargo, es necesario aclarar que a menudo el orden no es exacto y en ocasiones una fase se puede repetir n veces.

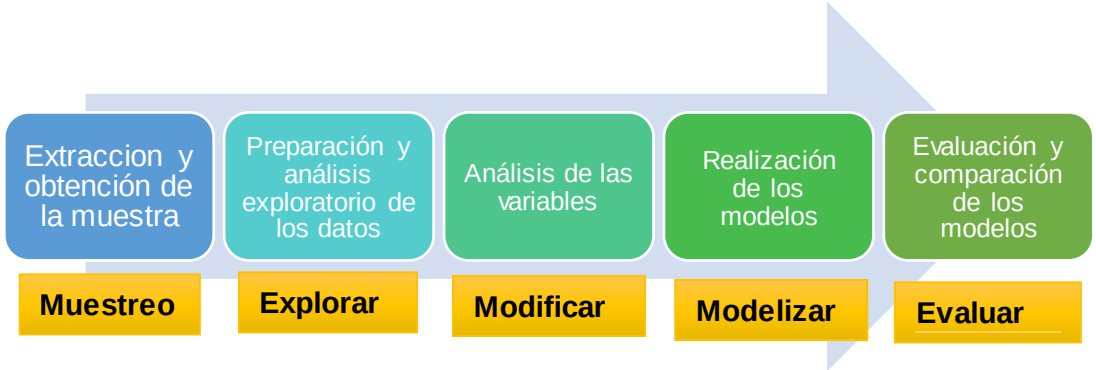


Figura 2 Metodología

Fase I: Extracción de información del ERP

Para la extracción de la información se realiza mediante el lenguaje de consulta estructurado SQL por sus siglas en inglés (Structured Query Language) aplicado a las bases de datos de la compañía, posteriormente se realiza una muestra la cual corresponde a la fase inicial de la metodología SEMMA.

Fase II: Exploración y depuración de la base de datos

Se realiza una exploración inicial de los datos, llevando a cabo una depuración de la base de datos con el fin de detectar errores, datos atípicos, se realiza una gestión de los datos ausentes, en este objetivo se pretende asegurar la calidad del dato.

Fase III: Análisis de las variables

Se procede a realizar un análisis de las variables, donde una opción es la discretización de las variables continuas para descubrir relaciones complejas (no lineales) entre las variables independientes y la variable objetivo. Además de evaluar la relación entre estas mediante el uso de estadísticos y tener una idea inicial de que variables son las más importantes para conseguir la predicción del coste. [11]

Fase IV: Modelización

Se propone realizar mediante tres opciones, las cuales se explican a continuación:

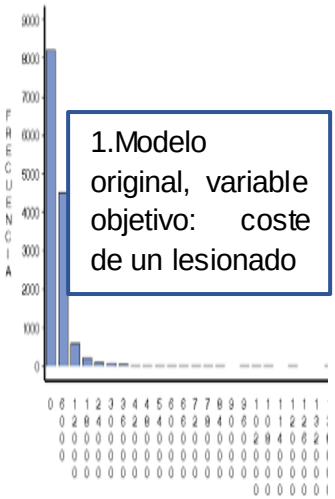


Figura 3.Modelo original sin transformación variable objetivo

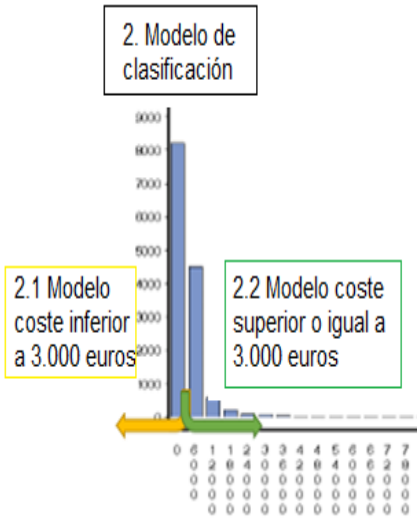


Figura 4. Modelo de clasificación

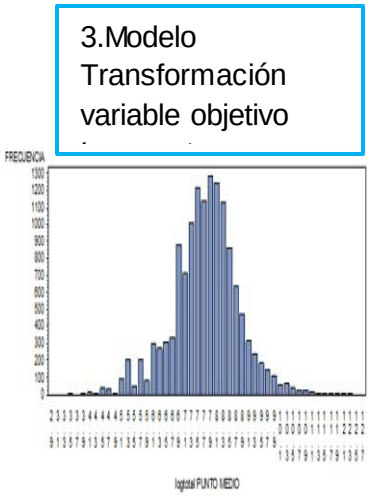


Figura 5. Modelo con variable objetivo-transformada

Primero se realiza el modelo original sin transformar la variable objetivo, sin embargo, como se observa en la figura 3 la variable objetivo presenta valores muy extremos por lo

que probablemente no se logre encontrar un modelo que se ajuste a la distribución de los datos, por lo tanto, una opción que se prueba es partir la muestra cómo se indica en la figura 4. Donde una vez clasificado se procede a realizar dos modelos, uno cuando es menor a los 3000 euros y el otro en caso contrario, se espera que al partir la muestra se logre encontrar modelos que se ajusten más a los datos. El punto de corte se define de acuerdo con que el 86% de la muestra ha tenido un coste inferior a los 6.000 euros por lo que resulta interesante fijar un punto intermedio para realizar la clasificación.

Por última opción se observa que al transformar la variable objetivo esta sigue una distribución normal por lo que es probable que se encuentre una mejor relación entre las variables independientes y la variable objetivo.

Los modelos estadísticos propuestos para predecir el coste de los lesionados son los presentados a continuación:

Regresión lineal

Los modelos de regresión lineal[12] son modelos matemáticos que permiten realizar predicciones de los valores que tomará la variable dependiente Y_t, con respecto a las variables independientes o explicativas X_t y un término de error aleatorio ε. Se puede representar con la siguiente ecuación:

$$Y_t = \beta_0 + \beta_1 X_1 + \ldots + \beta_i X_p + \varepsilon \qquad (1)$$

Donde:

- Y_t es la variable de respuesta o dependiente
- X_p son las variables independientes
- β_i son los parámetros respectivos de cada variable independiente y miden la influencia que las variables explicativas tienen, siendo β_0 el termino constante.
- ε es una variable aleatoria que recoge el error cometido en el modelo, generalmente sigue una distribución normal $(0,\sigma)$ independiente de cada observación y la varianza es constante para cualquier valor de X.

Generalmente los modelos de regresión proporcionan una descripción adecuada e interpretable de como las variables independientes o de entrada afectan a la variable de respuesta, uno de los problemas de estos modelos es que la relación entre las variables debe de ser lineal, sin embargo, se puede aplicar transformaciones a las variables de entrada para ampliar su alcance.

Las variables independientes cualitativas se incluyen en el modelo mediante el uso de variables dicotómicas ficticias llamadas *"dummy"*, si una variable categórica tiene *n* niveles, se incluye n-1 *dummies*.

Redes neuronales

La red neuronal artificial [13] se basa en la analogía que existe en el comportamiento de una red neuronal biológica que poseen bajas capacidades de procesamiento, sin embargo, toda su capacidad cognitiva se sustenta en la conectividad entre ellas. Una red neuronal artificial es un procesador elemental llamado neurona que posee la capacidad limitada de calcular, en general, una suma ponderada de sus entradas y luego le aplica una función de activación para obtener una señal que será transmitida a la próxima neurona. Estas neuronas artificiales se agrupan en capas o niveles y poseen un alto grado de conectividad entre ellas, conectividad que es ponderada por los pesos.

Si un conjunto de neuronas artificiales recibe simultáneamente el mismo tipo de información, se denomina capa. Una red se compone de nodos input o, de entrada, capa oculta y nodo de salida como se observa en la figura 6.

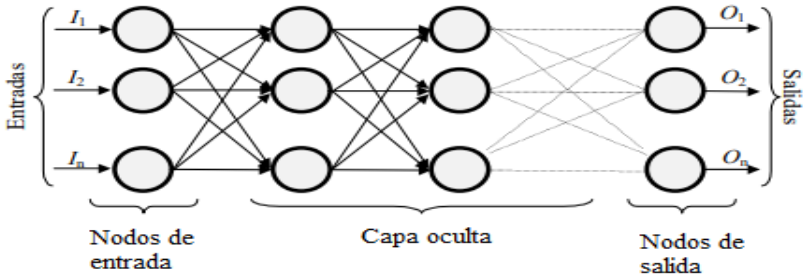


Figura 6. Estructura de una Red neuronal [14]

En general se utilizan las redes neuronales cuando es desconocida la función entre la variable dependiente y las variables independientes (no linealidad), cuando se tienen datos complejos como por ejemplo muchas variables categóricas. Adicional, se debe tener en cuenta que para tener una garantía de la red esta requiere de muchas observaciones. Se recomienda no utilizar la red cuando el objetivo del estudio sea explicar y no predecir, ya que la red es una caja negra y es difícil extraer información. Este no es el caso de la presente investigación ya que el objetivo es predecir mas no explicar.

Random Forest

Este algoritmo [15] es una combinación de factores predictivos de árboles, de modo que cada árbol depende de los valores de un vector aleatorio muestreado independientemente y con la misma distribución para todos los árboles, es decir consiste en incorporar dos fuentes de variabilidad, una es el remuestreo de observaciones y la otra es la aleatoriedad en las variables utilizadas para segmentar cada nodo del árbol. En general se utiliza este algoritmo cuando existen relaciones no lineales, cuando existen muchas variables categóricas, interacciones ocultas, etc.

El algoritmo de random forest [16] es el siguiente:

Dados los datos de tamaño N.

1. Repetir m veces a), b), c):
   a) Seleccionar N observaciones con reemplazamientos de los datos originales.
   b) Aplicar un árbol de la siguiente manera:
      En cada nodo, seleccionar p variables de las k originales y de las p elegidas, escoger la mejor variable para la partición del nodo.
   c) Obtener predicciones para todas las observaciones originales N.
2. Promediar las m predicciones obtenidas en el apartado 1.

En general los parámetros a controlar son:

- El tamaño de la muestra, si se va a utilizar Bootstrap (con reemplazamiento) o sin reemplazamiento.
- El número de iteraciones m y el número de variables p que sortear, si el número de variables a sortear es igual al número inicial de variables entonces el algoritmo random forest sería equivalente al bagging.
- Características de los arboles como el número de hojas final, número de divisiones máxima en cada nodo, el p valor para las divisiones de cada nodo.

Gradient Boosting

Es uno de los algoritmos más potentes en la actualidad, consiste en repetir la construcción de árboles de regresión o clasificación, donde se modifica levemente las predicciones iniciales, intentando ir minimizando los residuos en la dirección de decrecimiento [16] obteniendo así de manera gradiente modelos que convergen en un modelo final donde los errores son mínimos.

Ya que la base de este modelo es el algoritmo de random forest, lo que significa que utiliza las variables definidas en random forest y a su vez ganan nuevos parámetros, como las iteraciones que reflejan el número de etapas del modelo y el parámetro shrinkage que refleja el grado con el que se ajustará el modelo en cada una de las iteraciones.

Los principales parámetros que controlar son la constante de regularización shrinkage, el número de iteraciones y las características propias de los arboles como el número de hojas finales, número de observaciones mínimo en una rama del nodo, el p valor, etc.

Support vector machine

En esta investigación se utilizará SVM[16] por sus siglas en inglés, en el modelo de clasificación, este algoritmo consiste en la separación lineal de clases con métodos algebraicos buscando el hiperplano de separación, se basa en tres principales ideas. La

primera es el concepto de separador como máximo margen, donde no solo se trata de separar las clases por un hiperplano si no poder realizar como su nombre lo indica con la máxima distancia. La segunda idea considera que la separación perfecta no existe por lo que es necesario permitirse errores para evitar el sobreajuste. Y la tercera es que la separación entre clases en muchos casos no es lineal por lo que se debe de trabajar en una dimensión superior donde tenga más sentido la separación lineal.

Ensamblado

Consiste en la construcción de predicciones mediante la combinación de varios algoritmos. El ensamble se puede realizar con diferentes combinaciones, las técnicas básicas de combinado son bagging, boosting, stacking, entre otros.

En esta investigación se centra en la técnica de stacking que en general se puede calcular de tres formas, la primera es realizar el promedio de las predicciones de los algoritmos, este promedio también se puede realizar dándole un mayor peso a alguno de los modelos. La segunda forma es voto (para clasificación) que se predice el resultado como mayoría entre las predicciones y por último una combinación a partir de otro algoritmo (esto es estrictamente stacking). Por ejemplo, se introducen en una regresión o árbol y1, y2, y3 como variables independientes. En regresión equivaldría a un promediado de modelos con pesos diferentes.

Fase V: Evaluación y comparación de los modelos

Se realiza mediante validación cruzada repetida, donde se utiliza principalmente el estimador Error Cuadrático Medio (ASE) en datos test, el cual mide el promedio de los errores al cuadrado, es decir, la diferencia entre la observación real y la predicción. Su fórmula es la que se muestra en ecuación 2.

$$\text{ASE} = \frac{1}{n}\sum_{i=1}^{n}(\widehat{Y_i} - Y_i)\text{^}2 \qquad (2)$$

3. Depuración y exploración de variables

3.1 Origen de los datos

El estudio se lleva a cabo desde enero del 2016 hasta enero del 2019. En total se tiene 13.752 observaciones y 30 variables. El conjunto de datos es proporcionado por una empresa de seguros de España.

3.2 Variables continuas

Se tiene 15 variables continuas, que en su mayoría son variables que describen condiciones de tipo médica como son los puntos estéticos y puntos funcionales, además también se tiene la edad del lesionado, etc.

NOMBRE VARIABLE	DESCRIPCIÓN
días_básico días_grave días_mgrave días_modera	Corresponde a los días que considera el médico va a tener el lesionado en su proceso de recuperación, se divide entre días básico, moderado, grave y muy grave
importe_ilt	Corresponde al importe (€) que el médico considera que va a tener de incapacidad temporal
edad	Edad del lesionado
importe_sec	Corresponde al importe (€) que el médico considera que va a tener de secuelas
número diagnósticos	Indica cuantos diagnósticos recibe la persona en la primera valoración
días desde alta hasta valo	Corresponde a los días desde que se de alta a la víctima en el sistema hasta la valoración inicial médica
día alta	Día de alta de la víctima en el sistema
mes alta	Mes que se registra el alta de la victima
impt_ptos_est	Corresponde al importe (€) que el médico considera que va a tener por puntos estéticos
impt_ptos_fun	Corresponde al importe (€) que el médico considera que va a tener por puntos funcionales
ptos_est	Indica los puntos estéticos que va entre 0 a 50 puntos
ptos_fun	Indica puntos funcionales que esta entre 0 a 100 puntos
max_gravedad	Es un indicador que cuantifica la gravedad máxima de la lesión, es estipulado por el médico
min_gravedad	Es un indicador que cuantifica la gravedad mínima de la lesión, es estipulado por el médico

Tabla 1. Variables continuas

Se empieza con la exploración de las variables, el objetivo es conocer como es la distribución de las variables independientes mediante los histogramas de cada variable permitiendo detectar posibles errores, datos atípicos, curtosis, etc.

Histograma de las variables continuas

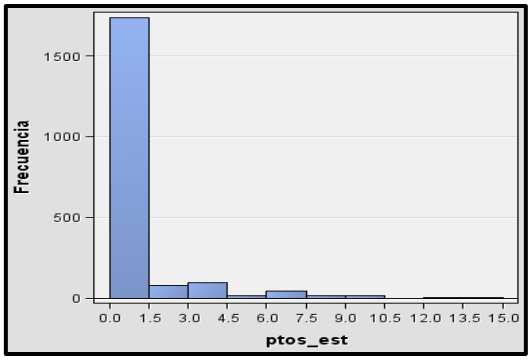


Figura 7. Histograma ptos_est

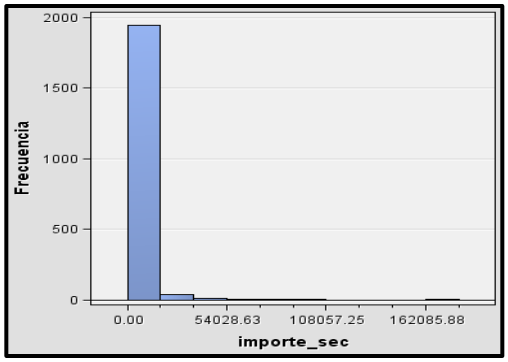


Figura 8. Histograma importe_sec

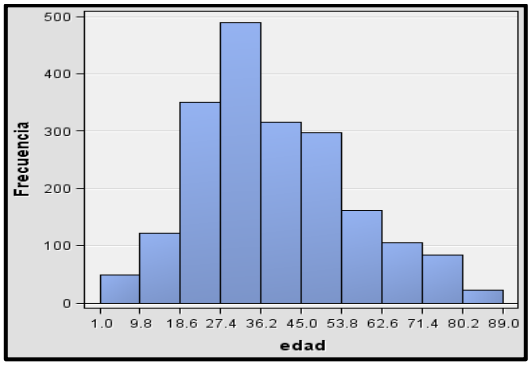


Figura 9. Histograma edad

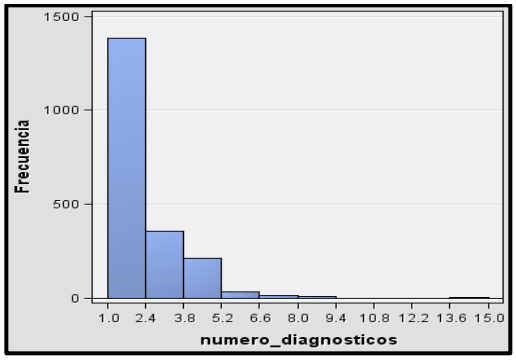


Figura 10. Histograma número_diagnósticos

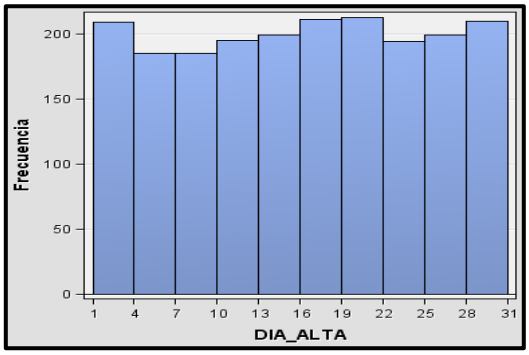


Figura 11. Histograma Día_alta

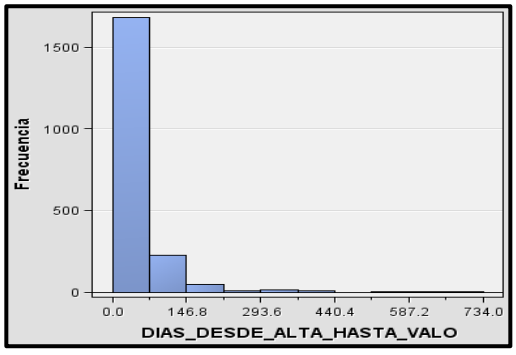


Figura 12. Histograma Días_desde_alta_valo

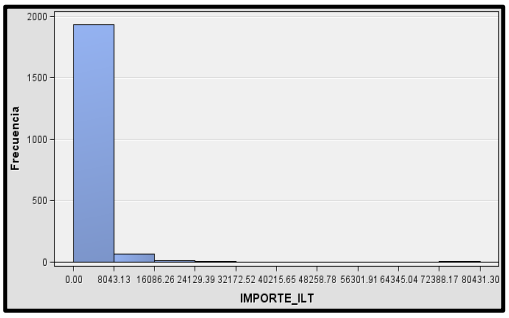


Figura 13. Histograma Importe ILT

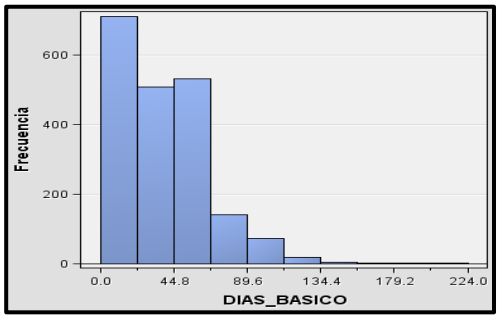


Figura 14. Histograma Días_Básico

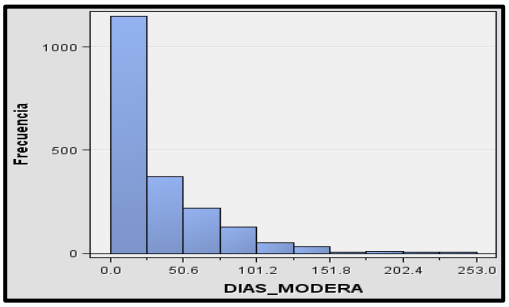


Figura 15. Histograma Días_Modera

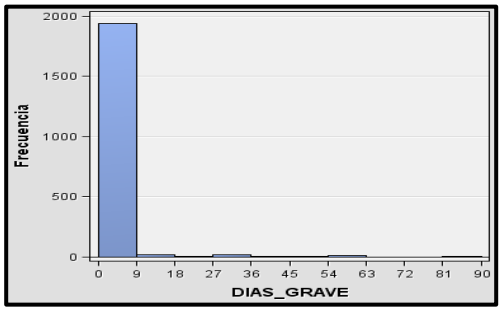


Figura 16. Histograma Días_Grave

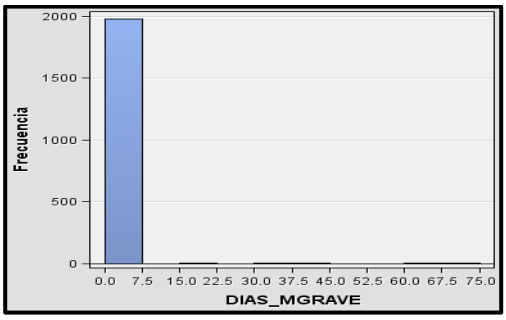


Figura 17. Histograma Días_MGrave

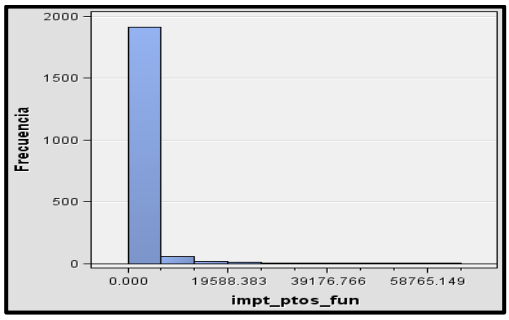


Figura 18. Histograma Impt_ptos_fun

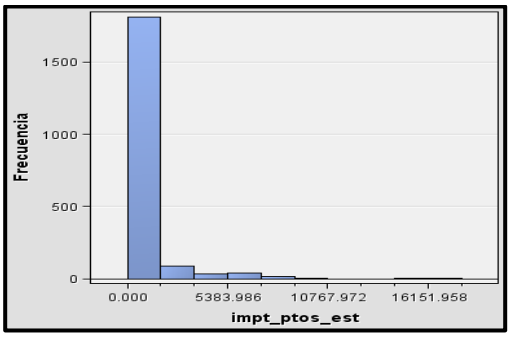


Figura 19. Histograma Impt_ptos_est

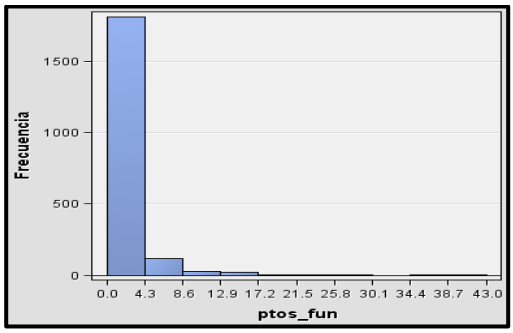


Figura 20. Histograma Ptos_fun

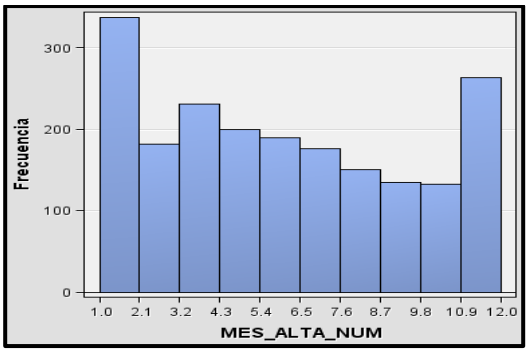


Figura 21. Histograma Mes_alta_num

En general las variables son asimétricas hacia la derecha con excepción de la edad, día alta y mes alta. La categoría más frecuente en la edad son los lesionados entre 27 y 36 años. La mayoría de las variables presenta una alta curtosis lo que podría indicar datos atípicos, como por ejemplo la variable importe de incapacidad laboral (importe ilt) se concentra en valores inferiores a 8.043 euros sin embargo se observa que tiene valores que superan los 70.000 euros. Por lo tanto, se procede a mostrar un resumen con los principales estadísticos de cada variable.

Variable	Mediana	Ausente	No ausente	Mínimo	Máximo	Media	Desviación estándar	Asimetría	Curtosis
DIAS_MGRAVE	0	109	13643	0	180	0.235945	4.143231	24.36285	705.2456
DIAS_GRAVE	0	109	13643	0	265	0.876933	7.124546	16.15525	384.7371
importe_sec	776.1	0	13752	0	1239562	2640.566	18086.77	49.19545	3031.049
impt_ptos_fun	0	0	13752	0	221297.1	1463.553	5314.737	22.23823	756.2796
impt_ptos_est	0	0	13752	0	47460.79	542.3842	1705.995	6.962062	94.31223
ptos_est	0	0	13752	0	31	0.625727	1.783835	4.576062	30.66378
ptos_fun	0	0	13752	0	100	1.570681	3.537485	8.443042	140.4048
Total	2428.37	0	13752	30	200000	3943.768	6579.593	9.282767	155.3836
DIAS_BASICO	30	109	13643	0	5530	36.2056	55.15067	72.59484	7218.426
DIAS_DESDE_ALTA_HA...	23	0	13752	0	959	44.5789	64.39413	4.623651	32.26898
DIAS_MODERA	15	109	13643	0	1120	31.58873	40.38713	3.536969	46.28372
IMPORTE_ILT	2351.7	0	13752	0	168688.3	2917.147	3230.573	18.64439	760.4007
DIA_ALTA	16	0	13752	1	31	15.85798	8.696725	-0.01446	-1.18199
edad	37	16	13736	0	99	39.02468	16.86214	0.515806	-0.06707

Tabla 2. Estadísticos de las variables continuas

3.3 Variables nominales

Se tiene cuatro variables binarias y once variables nominales, en general las variables hacen referencia tanto a condiciones médicas como por ejemplo el diagnóstico que recibe la persona en la valoración inicial y variables que describen el siniestro como la marca del vehículo, si el lesionado es un ciclista, conductor o acompañante, etc. En la tabla 3 se presenta una breve descripción de cada variable.

NOMBRE VARIABLE	DESCRIPCIÓN
cod_cond_victima	Si el lesionado es ciclista, conductor, ocupante, peatón, otros
cod_des_tipo_siniestro	Descripción del siniestro: Colisión con animal, gira o cambia de sentido, etc
cod_provincia	Lugar donde ocurre el siniestro
cod_sit_lab	Ocupación del lesionado
descri_tipo_vehiculo	Descripción tipo de vehículo asociado a la póliza: Turismo, moto, camión
diagnóstico 1	Corresponde al primer diagnóstico registrado en la valoración medica
diagnóstico 2	Corresponde al segundo diagnóstico registrado en la valoración medica
ref_marca	Marca del vehículo
ref_modelo	Modelo del vehículo
ind_colision	Indica si hubo colisión o no
ind_sexo_victima	Sexo de la victima
ind_sin_contrario	Indica si en el accidente sucedió con un contrario
ind_veh_reposo	Indica si el vehículo estaba en reposo al momento del accidente
médico	Razón social del médico que evalúa la lesión
vehículo	Corresponde si la víctima es un asegurado de la compañía o un contrario

Tabla 3. Variables nominales

Se presenta un resumen de los niveles y el número de ausentes que presenta cada variable, posteriormente se muestra unos ejemplos de la frecuencia que tienen las variables mediante una tabla de frecuencia en donde se identifica los niveles poco representados y se realiza una agrupación de aquellas categorías con la finalidad de que todos estén estadísticamente bien representados. Cabe resaltar que por confidencialidad la empresa proporciona la mayoría de los datos codificados, por lo tanto, se muestra tres ejemplos para ilustrar al lector.

Variable	Etiqueta	Tipo	Número de niveles	Ausente ▼
IND_SEXO_VICTIMA	IND_SEXO...	C	2	7625
VEHICULO	VEHICULO	C	2	2803
MAX_GRAVEDAD	MAX_GRAV...	N	6	42
MIN_GRAVEDAD	MIN_GRAV...	N	6	42
IND_SIN_CONTRARIO	IND_SIN_C...	C	2	4
COD_COND_VICTIMA	COD_CON...	C	5	0
COD_DES_TIPO_SINIESTRO	COD_DES...	C	23	0
COD_PROVINCIA	COD_PRO...	C	26	0
COD_SIT_LAB	COD_SIT_...	C	6	0
COD_VEHICULO_VICTIMA	COD_VEHI...	C	4	0
DESCRI_TIPO_VEHICULO	DESCRI_TI...	C	18	0
DIAGNOSTICO_1	DIAGNOSTI...	C	26	0
DIAGNOSTICO_2	DIAGNOSTI...	C	26	0
IND_COLISION	IND_COLIS...	C	2	0
IND_VEH_REPOSO	IND_VEH_...	C	2	0
MEDICO	MEDICO	C	26	0
REF_MARCA	REF_MARCA	C	26	0
REF_MODELO	REF_MOD...	C	26	0

Tabla 4. Número de ausentes variables nominales

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
REF_MARCA	SEAT		1295
REF_MARCA	CITROEN		1147
REF_MARCA	PEUGEOT		1120
REF_MARCA	VOLKSWAGEN		1119
REF_MARCA	FORD		1101
REF_MARCA	RENAULT		1092
REF_MARCA	OPEL		969
REF_MARCA	AUDI		603
REF_MARCA	BMW		558
REF_MARCA	MERCEDES		503
REF_MARCA	TOYOTA		480
REF_MARCA	NISSAN		387
REF_MARCA	HYUNDAI		355
REF_MARCA	HONDA		351
REF_MARCA	KIA		300
REF_MARCA	FIAT		290
REF_MARCA	SUZUKI		204
REF_MARCA	SKODA		162
REF_MARCA	MAZDA		148
REF_MARCA	YAMAHA		138
REF_MARCA	VOLVO		131
REF_MARCA	CHEVROLET-GM		111
REF_MARCA	DACIA		106
REF_MARCA	PIAGGIO-VESPA	**OTRO**	105
REF_MARCA	MITSUBISHI	**OTRO**	89
REF_MARCA	MINI	**OTRO**	85
REF_MARCA	ALFA ROMEO	**OTRO**	64

Tabla 5. Frecuencia de la variable marca

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
DESCRI_TIPO_VEHICULO	TURISMOS TURBO-DIESEL		4967
DESCRI_TIPO_VEHICULO	TURISMO		3453
DESCRI_TIPO_VEHICULO	MONOVOLUMEN		1576
DESCRI_TIPO_VEHICULO	TODO TERRENO TD		999
DESCRI_TIPO_VEHICULO	MOTOCICLETAS		746
DESCRI_TIPO_VEHICULO	TUR. FAMILIAR TURBO DIE		577
DESCRI_TIPO_VEHICULO	DERIVADO DE TURISMO		412
DESCRI_TIPO_VEHICULO	TODO TERRENO		332
DESCRI_TIPO_VEHICULO	CICLOMOTORES		156
DESCRI_TIPO_VEHICULO	CAMIONES LIGEROS Y FUR		137
DESCRI_TIPO_VEHICULO	FURGON HABILITABLE A PA		137
DESCRI_TIPO_VEHICULO	TUR. FAMILIAR		136
DESCRI_TIPO_VEHICULO	TURISMO DESCAPOTABLE	**OTROS**	98
DESCRI_TIPO_VEHICULO	PICK UP	**OTROS**	13
DESCRI_TIPO_VEHICULO	TODO TERRENO DESCAPOT	**OTROS**	6
DESCRI_TIPO_VEHICULO	TODO TERRENO TD DESC	**OTROS**	3
DESCRI_TIPO_VEHICULO	VEHICULOS ESPECIALES	**OTROS**	3
DESCRI_TIPO_VEHICULO	CAMIONES PESADOS	**OTROS**	1

Tabla 6. Frecuencia de la variable Descri_tipo_vehiculo

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
COD_DES_TIPO_SINIESTRO	DS07		2518
COD_DES_TIPO_SINIESTRO	DS06		2508
COD_DES_TIPO_SINIESTRO	DS03		2162
COD_DES_TIPO_SINIESTRO	DS05		893
COD_DES_TIPO_SINIESTRO	DS01		863
COD_DES_TIPO_SINIESTRO	DS11		850
COD_DES_TIPO_SINIESTRO	DS02		811
COD_DES_TIPO_SINIESTRO	DS08		688
COD_DES_TIPO_SINIESTRO	DS13		563
COD_DES_TIPO_SINIESTRO	DS09		465
COD_DES_TIPO_SINIESTRO	DS14		436
COD_DES_TIPO_SINIESTRO	DS12		319
COD_DES_TIPO_SINIESTRO	DS10		282
COD_DES_TIPO_SINIESTRO	DS04		152
COD_DES_TIPO_SINIESTRO	DS16		101
COD_DES_TIPO_SINIESTRO	DS18	**OTRO**	95
COD_DES_TIPO_SINIESTRO	DS17	**OTRO**	18
COD_DES_TIPO_SINIESTRO	DS15	**OTRO**	13
COD_DES_TIPO_SINIESTRO	DS19	**OTRO**	7
COD_DES_TIPO_SINIESTRO	DS22	**OTRO**	4
COD_DES_TIPO_SINIESTRO	DS20	**OTRO**	2
COD_DES_TIPO_SINIESTRO	DS21	**OTRO**	1
COD_DES_TIPO_SINIESTRO	DS24	**OTRO**	1

Tabla 7.Frecuencia tipo de siniestro

3.4 Búsqueda de datos atípicos

Se procede a la búsqueda de atípicos para las variables de intervalo, se analiza que método aplicar dependiendo de la asimetría de las variables, para las variables asimétricas se utiliza desviación absoluta, para las que son asimétricas y con mediana cero se aplica percentiles extremos, por último, para las variables simétricas se utiliza desviación típica, mediante el software SAS Miner.

Variable	Entrenamiento ▼
REP_importe_sec	769
REP_DIAS_DESDE_ALTA_...	579
REP_IMPORTE_ILT	258
REP_DIAS_MODERA	217
REP_DIAS_MGRAVE	67
REP_edad	16
DIA_ALTA	0
REP_DIAS_BASICO	0
REP_DIAS_GRAVE	0
REP_impt_ptos_est	0
REP_impt_ptos_fun	0
REP_ptos_est	0
REP_ptos_fun	0

Tabla 8. Número de atípicos por variable

Se observa que las variables que presentan atípicos son importe sec, días desde alta, importe ILT, días modera y mgrave y la edad. Se procede a utilizar el rango intercuartílico para las variables anteriormente mencionadas con el fin de construir los respectivos límites. Se establecen los límites y se dice colocar las observaciones fuera del rango como ausentes, para gestionarlos en el siguiente apartado.

3.5 Tratamiento de datos ausentes

En la tabla 9. se muestra el número de ausentes de cada variable continua, donde se observa que la variable que presenta un mayor número es importe sec con 769 esto equivale al 5% lo que no se considera representativo. También se crea una variable para contar el número de ausente por observación la cual arroja un máximo de 8 por lo que se considera que no es significativamente representativo.

Variable	Ausente ▲
DIA_ALTA	0
REP_impt_ptos_est	0
REP_impt_ptos_fun	0
REP_ptos_est	0
REP_ptos_fun	0
Total	0
numMissing	0
REP_edad	16
REP_DIAS_BASICO	109
REP_DIAS_GRAVE	109
REP_DIAS_MODERA	109
REP_REP_DIAS_MGRAVE	176
REP_REP_IMPORTE_ILT	258
REP_REP_DIAS_DESDE_ALTA_HASTA_VA	453
REP_REP_importe_sec	769

Tabla 9.Número de ausentes variables continuas

Por otra parte, las variables nominales, la que mayor número de ausentes presenta es el sexo con 7.625 equivalente al 55% de la muestra, lo que se considera representativo, seguido de vehículo, con 2.803 observaciones ausentes, equivalente al 20%. Debido de que la proporción de ausentes en el sexo es muy alta se decide eliminar la variable, por su parte para vehículo se decide imputar, sin embargo, se crea una variable que especifique si la observación ha sido imputada tomando el valor de 1 y 0 de lo contrario.

La imputación se realiza mediante la opción del SAS Miner, se elige el árbol como método de imputación para las variables de clase y para las variables de intervalo se utiliza distribución. Una vez realizada la imputación se verifica que no haya ausentes.

Variable	Etiqueta	Tipo	Número de niveles	Ausente
COD_SIT_LAB	COD_SIT...	C	6	0
IMP_REP_COD_VEHICULO_VICTIMA	Imputed: ...	C	3	0
IMP_REP_DIAGNOSTICO_1	Imputed: ...	C	26	0
IMP_REP_IND_SIN_CONTRARIO	Imputed: ...	C	2	0
IMP_REP_MAX_GRAVEDAD	Imputed: ...	N	2	0
IMP_REP_MIN_GRAVEDAD	Imputed: ...	N	2	0
IMP_REP_VEHICULO	Imputed: ...	C	2	0
IND_COLISION	IND_COL...	C	2	0
IND_VEH_REPOSO	IND_VEH...	C	2	0
M_Variable	Imputatio...	N	7	0
REP_COD_COND_VICTIMA	Replace...	C	4	0
REP_COD_DES_TIPO_SINIESTRO	Replace...	C	16	0
REP_COD_PROVINCIA	Replace...	C	26	0
REP_DESCRI_TIPO_VEHICULO	Replace...	C	13	0
REP_DIAGNOSTICO_2	Replace...	C	26	0
REP_MEDICO	Replace...	C	26	0
REP_REF_MARCA	Replace...	C	24	0
REP_REF_MODELO	Replace...	C	26	0

Tabla 10.Verificación del número de ausentes variables nominales

Variable	Etiqueta	Ausente	N
IMP_REP_REP_DIAS_BASICO	Imputed: R...	0	13752
IMP_REP_REP_DIAS_GRAVE	Imputed: R...	0	13752
IMP_REP_REP_DIAS_MGRAVE	Imputed: R...	0	13752
IMP_REP_REP_DIAS_MODERA	Imputed: R...	0	13752
IMP_REP_REP_IMPORTE_ILT	Imputed: R...	0	13752
IMP_REP_REP_edad	Imputed: R...	0	13752
IMP_REP_REP_importe_sec	Imputed: R...	0	13752
IMP_numero_diagnosticos	Imputed: nu...	0	13752
REP_DIAS_DESDE_ALTA_HAST...	Replaceme...	0	13752
REP_REP_DIA_ALTA	Replaceme...	0	13752
REP_REP_MES_ALTA_NUM	Replaceme...	0	13752
REP_REP_impt_ptos_est	Replaceme...	0	13752
REP_REP_impt_ptos_fun	Replaceme...	0	13752
REP_REP_ptos_est	Replaceme...	0	13752
REP_REP_ptos_fun	Replaceme...	0	13752
Total	Total	0	13752

Tabla 11.Verificación número de ausentes variables continuas

4. Modelo I: Modelo original sin transformación de la variable objetivo

En esta sección se lleva a cabo la primera opción, realizando inicialmente una selección de variables, donde se conoce cuáles son las variables más importantes y posteriormente se realiza todos los algoritmos. En este apartado se hará énfasis especialmente en cómo obtener un modelo de regresión lineal y redes neuronales tanto en el software SAS Miner como en el software estadístico R

4.1 Variable Objetivo sin transformación

Se nombra la variable objetivo como **Total**, la cual corresponde al valor de la incapacidad temporal más secuela. Es decir, representa lo que la empresa desembolsa al lesionado por los daños físicos tanto temporales como permanentes, en este caso se toma la primera valoración médica como momento cero y se relaciona con el importe económico final que pago la empresa (momento en que ya se cierra el proceso).

Se presenta la distribución de la variable, donde se observa que la mayor parte de los datos se concentra en un coste inferior a los 6.000 euros. Se tiene un mínimo de 30 y un máximo de 200.000 euros y una media de 3.943 euros.

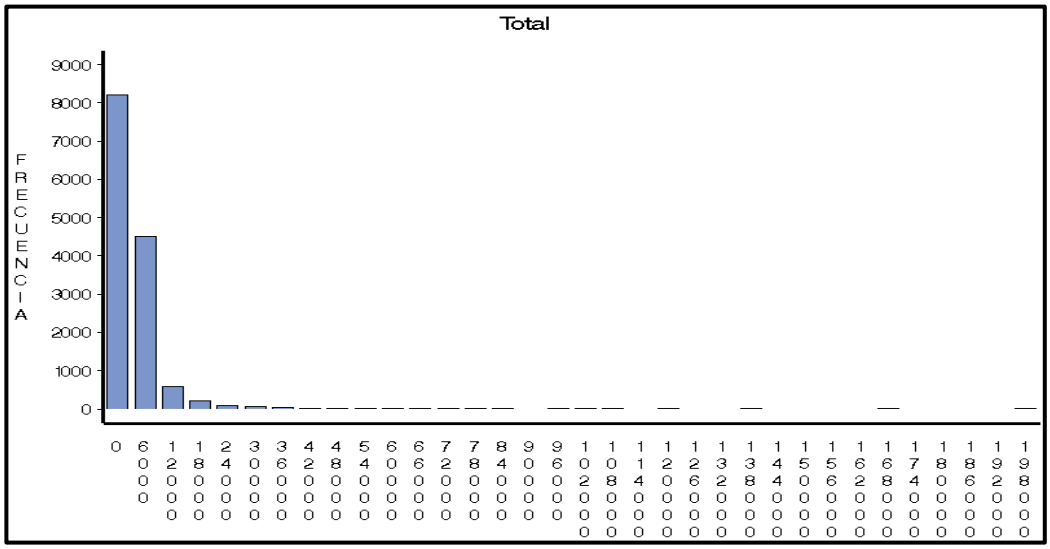


Figura 22. Variable Objetivo

4.2 Transformación de las variables independientes

Posteriormente se realiza una transformación de las variables de intervalo utilizando correlación máxima e indicando que se mantengan las variables originales.

NOMBRE CORTO	VARIABLE TRANSFORMADA
Z1	LOG_imp_rep_dias_basico
Z2	EXP_imp_rep_dias_grave
Z3	EXP_imp_rep_dias_mgrave
Z4	PWR_imp_rep_dias_modera
Z5	SQR_imp_numero_diagnósticos
Z6	LOG_rep_dias_desde_alta_hasta_va
Z7	LOG_rep_dia_alta
Z8	LOG_rep_mes_alta_num
Z9	PWR_rep_impt_ptos_est
Z10	EXP_rep_impt_ptos_fun
Z11	PWR_rep_ptos_est
Z12	PWR_rep_ptos_fun

Tabla 12.Transformaciones variables independientes Modelo I

Se decide renombrar las variables asignándoles un nombre corto con la finalidad de simplicidad.

NOMBRE VARIABLE	NOMBRE CORTO
días_básico	Y1
días_grave	Y2
días_mgrave	Y3
días_modera	Y4
importe_ilt	Y5
edad	Y6
importe_sec	Y7

Tabla 13. Recodificación de variables continuas

NOMBRE VARIABLE	NOMBRE CORTO
número diagnósticos	Y8
días desde alta hasta valo	Y9
día alta	Y10
mes alta	Y11
impt_ptos_est	Y12
impt_ptos_fun	Y13
ptos_est	Y14
ptos_fun	Y15

Tabla 14.Recodificación de variables continuas

4.3 Análisis de correlación

Se realiza un analisis de correlación entre las variables independientes en relación con la variable objetivo, se presenta el analisis por grupos.

Pearson Correlation Coefficients, N = 13752
Prob > |r| under H0: Rho=0

	Y1	Y2	Y3	Y4	Y5	Total
Y1 Y1	1.00000	-0.03369 <.0001	-0.01028 0.2282	-0.39231 <.0001	0.06301 <.0001	-0.02975 0.0005
Y2 Y2	-0.03369 <.0001	1.00000	0.19006 <.0001	0.36454 <.0001	0.28951 <.0001	0.54449 <.0001
Y3 Y3	-0.01028 0.2282	0.19006 <.0001	1.00000	0.07364 <.0001	0.06371 <.0001	0.15120 <.0001
Y4 Y4	-0.39231 <.0001	0.36454 <.0001	0.07364 <.0001	1.00000	0.75707 <.0001	0.49483 <.0001
Y5 Y5	0.06301 <.0001	0.28951 <.0001	0.06371 <.0001	0.75707 <.0001	1.00000	0.37107 <.0001
Total Total	-0.02975 0.0005	0.54449 <.0001	0.15120 <.0001	0.49483 <.0001	0.37107 <.0001	1.00000

Tabla 15. Análisis de correlación primer grupo

Se observa que entre las variables independientes que están más relacionadas son Y4 y Y5, que corresponden a días moderados e importe de incapacidad temporal (importe_ilt) con un coeficiente de 0.75 y un p valor inferior al 0.05. Con respecto a la variable objetivo la que tiene un mayor coeficiente de relación es Y2 (días graves) con un 0.54 y un p valor menor de 0.001, por lo tanto, se rechaza la hipótesis nula de independencia.

Pearson Correlation Coefficients, N = 13752
Prob > |r| under H0: Rho=0

	Y6	Y7	Y8	Y9	Y10	Total
Y6	1.00000	0.08418	0.03938	-0.01590	0.00537	0.12604
Y6		<.0001	<.0001	0.0623	0.5289	<.0001
Y7	0.08418	1.00000	0.21544	-0.07876	0.01481	0.19891
Y7	<.0001		<.0001	<.0001	0.0824	<.0001
Y8	0.03938	0.21544	1.00000	-0.13745	0.00063	0.17916
Y8	<.0001	<.0001		<.0001	0.9416	<.0001
Y9	-0.01590	-0.07876	-0.13745	1.00000	0.00513	-0.00487
Y9	0.0623	<.0001	<.0001		0.5478	0.5683
Y10	0.00537	0.01481	0.00063	0.00513	1.00000	0.00148
Y10	0.5289	0.0824	0.9416	0.5478		0.8618
Total	0.12604	0.19891	0.17916	-0.00487	0.00148	1.00000
Total	<.0001	<.0001	<.0001	0.5683	0.8618	

Tabla 16. Análisis de correlación segundo grupo

Se observa que tanto entre las variables independientes como en relación con la variable objetivo presenta un coeficiente de correlación pequeño.

Pearson Correlation Coefficients, N = 13752
Prob > |r| under H0: Rho=0

	Y11	Y12	Y13	Y14	Y15	Total
Y11	1.00000	0.03768	0.00840	0.03835	0.00831	0.00065
Y11		<.0001	0.3245	<.0001	0.3301	0.9389

Pearson Correlation Coefficients, N = 13752
Prob > |r| under H0: Rho=0

	Y11	Y12	Y13	Y14	Y15	Total
Y12	0.03768	1.00000	0.35809	0.99796	0.33501	0.37988
Y12	<.0001		<.0001	<.0001	<.0001	<.0001
Y13	0.00840	0.35809	1.00000	0.36298	0.99167	0.56972
Y13	0.3245	<.0001		<.0001	<.0001	<.0001
Y14	0.03835	0.99796	0.36298	1.00000	0.34126	0.38379
Y14	<.0001	<.0001	<.0001		<.0001	<.0001
Y15	0.00831	0.33501	0.99167	0.34126	1.00000	0.54696
Y15	0.3301	<.0001	<.0001	<.0001		<.0001

Tabla 17. Análisis de correlación tercer grupo

Se observa que entre las variables independientes hay una correlación entre Y12 y Y14, con un coeficiente de 0.99 que corresponde al importe de puntos estéticos (€) y a los puntos estéticos. También entre Y13 y Y15, que corresponde al importe de puntos funcionales (€) y puntos funcionales, respectivamente. Con respecto a la variable objetivo presenta una correlación con Y13 de 0.56 y Y15 con un 0.54 y un p valor inferior a 0.001 donde se rechaza la hipótesis nula de independencia.

Ahora se realiza el analisis de correlación con las variables independientes transformadas

Pearson Correlation Coefficients, N = 13752
Prob > |r| under H0: Rho=0

	Z1	Z2	Z3	Z4	Z5	Z6	Total
Z1	1.00000	-0.04296	-0.01309	-0.24526	-0.00648	0.12783	-0.04452
Z1		<.0001	0.1247	<.0001	0.4474	<.0001	<.0001

Pearson Correlation Coefficients, N = 13752
Prob > |r| under H0: Rho=0

	Z1	Z2	Z3	Z4	Z5	Z6	Total
Z2 Z2	-0.04296 <.0001	1.00000	0.18115 <.0001	0.42640 <.0001	0.15809 <.0001	-0.05430 <.0001	0.53854 <.0001
Z3 Z3	-0.01309 0.1247	0.18115 <.0001	1.00000	0.08623 <.0001	0.06648 <.0001	-0.00986 0.2477	0.14240 <.0001
Z4 Z4	-0.24526 <.0001	0.42640 <.0001	0.08623 <.0001	1.00000	0.08720 <.0001	0.03727 <.0001	0.52062 <.0001
Z5 Z5	-0.00648 0.4474	0.15809 <.0001	0.06648 <.0001	0.08720 <.0001	1.00000	-0.09336 <.0001	0.23175 <.0001
Z6 Z6	0.12783 <.0001	-0.05430 <.0001	-0.00986 0.2477	0.03727 <.0001	-0.09336 <.0001	1.00000	-0.00733 0.3898
Total Total	-0.04452 <.0001	0.53854 <.0001	0.14240 <.0001	0.52062 <.0001	0.23175 <.0001	-0.00733 0.3898	1.00000

Tabla 18. Análisis de correlación cuarto grupo

Se observa una correlación con la variable objetivo Total y Z2 (EXP_imp_rep_dias_grave) de 0.53 y con Z4 (PWR_imp_rep_dias_modera) de 0.52.

Pearson Correlation Coefficients, N = 13752
Prob > |r| under H0: Rho=0

	Z7	**Z8**	**Z9**	**Z10**	**Z11**	**Z12**	**Total**
Z7	1.00000	0.00571	0.01402	0.00090	0.01392	-0.00160	0.00276
Z7		0.5032	0.1001	0.9161	0.1025	0.8512	0.7463
Z8	0.00571	1.00000	0.01895	0.00824	0.02133	0.00454	0.00177
Z8	0.5032		0.0263	0.3338	0.0124	0.5945	0.8355
Z9	0.01402	0.01895	1.00000	0.38993	0.99290	0.40162	0.40443
Z9	0.1001	0.0263		<.0001	<.0001	<.0001	<.0001
Z10	0.00090	0.00824	0.38993	1.00000	0.40248	0.92751	0.59405
Z10	0.9161	0.3338	<.0001		<.0001	<.0001	<.0001
Z11	0.01392	0.02133	0.99290	0.40248	1.00000	0.41499	0.41331
Z11	0.1025	0.0124	<.0001	<.0001		<.0001	<.0001
Z12	-0.00160	0.00454	0.40162	0.92751	0.41499	1.00000	0.57984
Z12	0.8512	0.5945	<.0001	<.0001	<.0001		<.0001
Total	0.00276	0.00177	0.40443	0.59405	0.41331	0.57984	1.00000
Total	0.7463	0.8355	<.0001	<.0001	<.0001	<.0001	

Tabla 19. Análisis de correlación quinto grupo

Se observa que entre variables independientes las que tienen un coeficiente de correlación alto son Z10 y Z12, que corresponde (EXP_rep_impt_ptos_fun) y PWR_rep_ptos_fun respectivamente con un coeficiente de 0.92 y entre Z9 y Z11, es decir entre PWR_rep_impt_ptos_est y PWR_rep_ptos_est. Con respecto a la variable objetivo,

presenta correlación alta con Z10 (EXP_rep_impt_ptos_fun) de 0.59, Z11 (PWR_rep_ptos_est) de 0.41 y Z12 (PWR_rep_ptos_fun) de 0.57.

4.4 Modelización de la variable objetivo sin transformar

4.4.1 Regresión lineal SAS Miner

Se decide realizar varios caminos para encontrar el mejor modelo de regresión que si bien no es el óptimo, al probar muchas opciones se tiene una mayor probabilidad de encontrar un buen modelo. Por lo tanto, lo primero que se realiza es crear una variable aleatoria, con el fin de identificar que variables no son importantes para el modelo, esta variable se crea desde el nodo de transformación de variables, pero sin realizar ninguna transformación por el momento.

Se realiza un primer modelo el cual considera todas las variables, se utiliza una partición de 70 15 15. Se obtiene un RASE entrenamiento de 4465 y en prueba de 5287, por lo que no se considera un modelo estable, el número de parámetros es de 573. El R cuadrado es de 0.55. Es importante también identificar cuáles son las mejores variables y cuales se podrían eliminar (Type 3 Analysis of Effects).

En el análisis tipo 3 se identifica que las variables más importantes son imp_rep_dias_grave, rep_impt_ptos_fun, imp_rep_dias_modera. Las variables que se podrían eliminar son imp_rep_vehiculo, ind_veh_reposo, imp_rep_dias_desde_alta_hast, ind_colision, rep_cod_cond_victima, dia_alta, imp_rep_ind_sin_contrario, imp_rep_cod_vehiculo_victima.

Se utiliza cuatro modelos de selección de variables, normalmente se conocen los tres primeros modelos para seleccionar variables, no obstante, SAS da la opción de utilizar la regresión Lasso como método de selección ya que teóricamente a partir de cierto valor del parámetro de penalización (landa) el estimador Lasso produce estimaciones nulas para algunos coeficientes y no nulas para otros, con lo cual Lasso realiza una especie de selección de variables continua. [17].

- Selección hacia adelante (Forward): Este método introduce secuencialmente las variables. La primera variable que ingresa es aquella que tenga una mayor correlación con la variable dependiente, que cumpla con el criterio de entrada. El proceso finaliza cuando no haya variables que satisfagan el criterio.

- Selección hacia atrás (Backward): Este método parte del modelo con todas las variables, el cual va eliminando una a una, empieza por la que menos presente correlación con la variable dependiente y que cumpla con el criterio de eliminación. El procedimiento termina cuando ya no queden variables que satisfagan este criterio.

- Paso a paso (Stepwise): Este método es una combinación de los dos anteriores, comienza con introducir paulatinamente las variables, pero en cada etapa se plantea si todas las variables introducidas deben permanecer en el modelo mediante el no cumplimiento del criterio de salida.

Los criterios utilizados para seleccionar las variables que consideran el ajuste y el exceso de parámetros son BIC, AIC y validación. El BIC es el más estricto, es decir permite menos parámetros.

A continuación, se presenta los diferentes modelos y caminos a evaluar. El modelo Manual hace referencia a eliminar las variables con valores inferiores a la variable aleatoria como se presentó anteriormente. También para el Step se considera un nivel de significación 0.2, efectos Solo y un número máximo de pasos de 50.

Modelos propuestos
Todas Las Variables
Manual
Backward Error Validación
Forward Aic
Step Error Validación
Step Bic
Lasso Sbc
Lasso Aic

Tabla 20.Modelos para regresión lineal

Se evalúa los modelos mediante seis caminos para posteriormente escoger el mejor modelo por cada método y luego realizar un training test. Los caminos se componen por: solo imputación, imputación y selección de variables, transformación y clustering, árbol sin transformación de variables, transformación y selección de variables, transformación de variables y árbol, como se observa en la figura 23.

Se utiliza principalmente los estadísticos Averange Squared Error tanto en entrenamiento como en prueba, el Error cuadrático de la media de la raíz (TRASE) debido a que visualmente es más fácil identificar qué modelo proporciona el menor error, y por último el DFM, estos estadísticos se utilizan para determinar el mejor modelo de cada método.

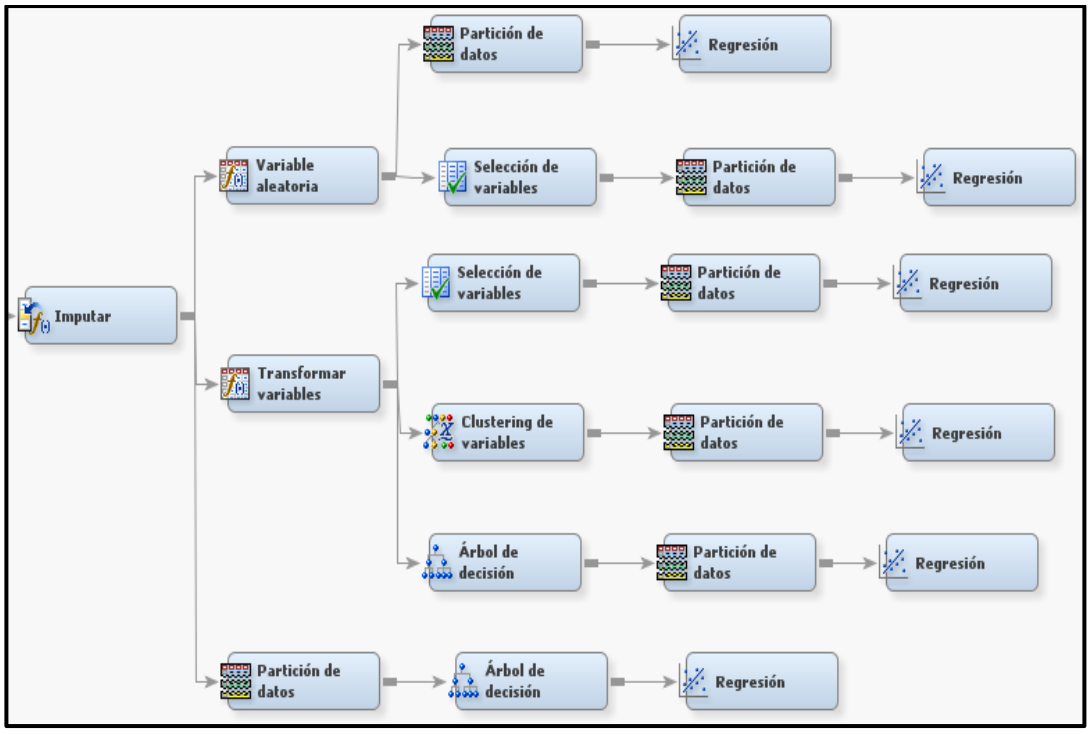


Figura 23. Caminos para obtener modelos de regresión lineal SAS Miner

Training-test regresión lineal SAS Miner

Se realiza training test con los mejores modelos obtenidos de cada método, se realizan 10 iteraciones.

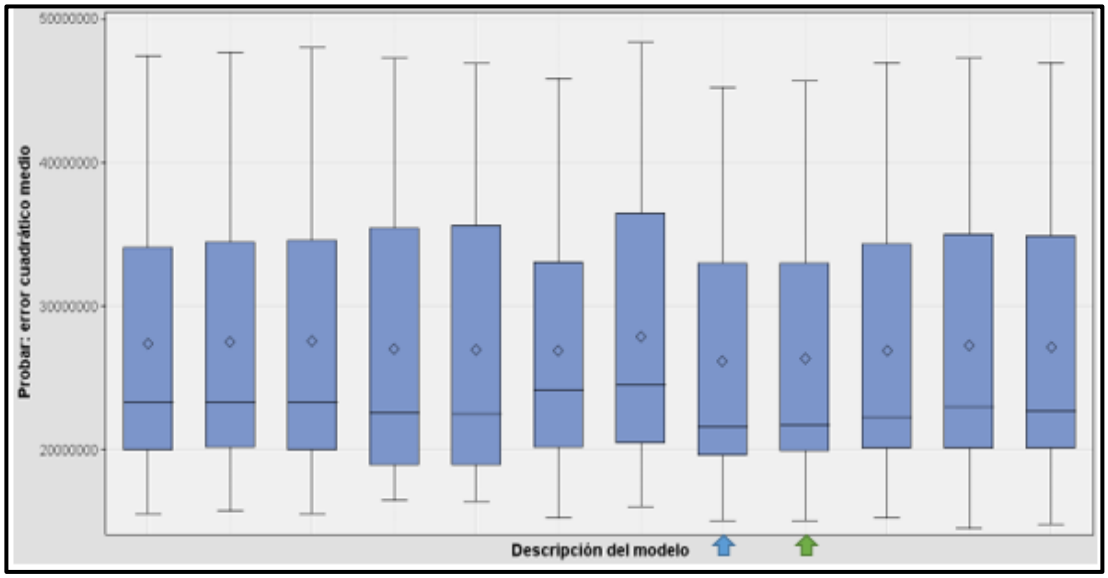


Figura 24. Training test regresión lineal SAS Miner

Se observa que los modelos en general están similares y presentan mucha variabilidad, no obstante, se elige el modelo de acuerdo con el número de parámetros y el que presenta el error más bajo. Se ha señalado en la figura 24. los dos modelos que podrían ser elegidos que corresponden a los modelos obtenidos mediante transformación, Forward AIC (señalado en azul) presenta 19 parámetros y Step BIC (señalado con verde) tiene 11 parámetros, por lo tanto, se decide elegir ese modelo.

4.4.2 Regresión lineal con R

Ahora bien, se utiliza R principalmente para realizar validación cruzada repetida. Se realiza un primer modelo el cual es el mejor modelo del Miner, también se realiza una selección de variables mediante Stepwise, Backward y Forward, utilizando el criterio de selección AIC y BIC, Posteriormente se realiza otro modelo utilizando Lasso como método de selección de variables, se realiza validación cruzada repetida para decidir cuál es el mejor modelo y si se logra encontrar un mejor modelo con respecto al del Miner.

```
modelos<-
sapply(list(modeloPreliminar,modeloBackAIC,modeloBackBIC,modeloForBIC,modeloForwardAIC,modeloStepAIC,modeloStepBIC),formula)
```

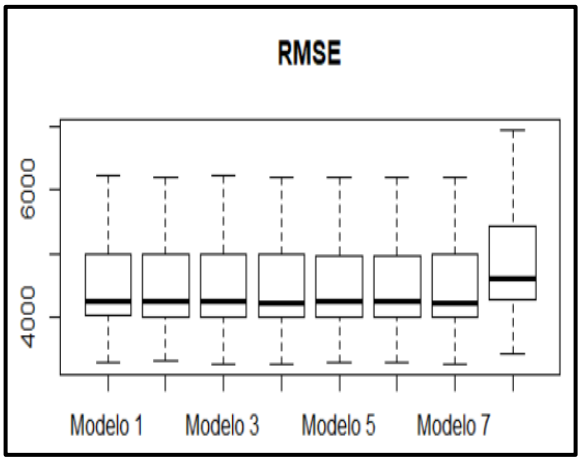


Figura 25. RMSE regresión lineal en R

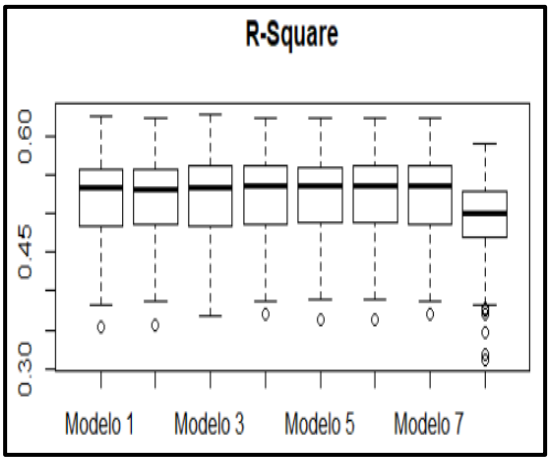


Figura 26. R-square regresión lineal en R

En la figura 25. se observa el error de los diferentes modelos en validación cruzada repetida donde visualmente presentan errores muy similares excepto lasso que tiene el error superior, de igual manera se comporta el R cuadrado, figura 26. Como visualmente es difícil conocer exactamente cuál es el mejor modelo, se generan los valores del R cuadrado en media y desviación.

```
media
    modelo  Rsquared
1 Modelo 1 0.5168032
2 Modelo 2 0.5182056
3 Modelo 3 0.5202380
4 Modelo 4 0.5216771
5 Modelo 5 0.5209190
6 Modelo 6 0.5214492
7 Modelo 7 0.5216771
8    LASSO 0.4854664
```

```
> aggregate(Rsquared~modelo,
    modelo   Rsquared
1 Modelo 1 0.05352088
2 Modelo 2 0.05160236
3 Modelo 3 0.05272319
4 Modelo 4 0.05199369
5 Modelo 5 0.05101517
6 Modelo 6 0.05100769
7 Modelo 7 0.05199369
8    LASSO 0.05931812
```

Se observa que los modelos que presenta un mayor R cuadrado en media son el modelo 4 y modelo 7, que corresponde a Forward con BIC y Step con BIC respectivamente. Se decide cual es el mejor modelo según el número de parámetros. Forward BIC tiene 16 parámetros y Step BIC presenta 18. Por lo tanto, se escoge el modelo Forward BIC.

Modelo Forward BIC:

z10	exp_rep_rep_impt_ptos_fun
y2	dias_grave
z4	pwr_imp_rep_rep_días_modera
z5	sqr_imp_numero_diagnósticos
z11	pwr_rep_rep_ptos_est
y4	días_modera
y7	importe_sec
y1	dias_básico
y5	importe_ilt
y8	numero_diagnósticos
g6	g_rep_médico
y3	dias_mgrave
g4	g_rep_diagnóstico_2
y6	edad
y15	ptos_fun
g2	g_imp_rep_diagnóstico_1

4.4.3 Redes Neuronales

Primero se construye un set con todas las variables, tanto nominales, intervalo, transformadas, dummies, agrupadas por selección de variables. Las dummies se crean a partir del nodo de transformación de variables en Miner, como se tienen 15 variables

nominales, la mayoría de ellas con más de 20 niveles como son la provincia, el diagnóstico 1, diagnóstico 2, médico, etc. El resultante es un fichero con 281 variables.

Selección de variables

De acuerdo con Carrasco[18] una de las cuestiones más importantes es realizar una correcta selección de variables, debido a que, si se incluye más variables de las que se requiere para explicar la variable dependiente, se puede caer en el sobreajuste. Por el contrario, si se eligen menos variables de las necesarias, las varianzas se reducen, pero los sesgos aumentan obteniéndose una mala descripción de los datos. Por otra parte, algunas variables predictoras pueden perjudicar la confiabilidad del modelo, especialmente si están correlacionadas con otras. Por lo tanto, el objetivo de los métodos de selección de variables es buscar un modelo que se ajuste bien a los datos y que a su vez sea posible obtener un equilibrio entre bondad de ajuste y sencillez.

Ahora bien, existe un problema en redes neuronales y es que se debe de realizar a priori una selección de variables. Por esto se realiza diferentes métodos para obtener las variables, los cuales son a través de mínimos cuadrados, incremento gradiente, árbol, selección de variables y diferentes modelos de regresión que se utilizan tanto para selección de variables como para comparar los resultados de las redes.

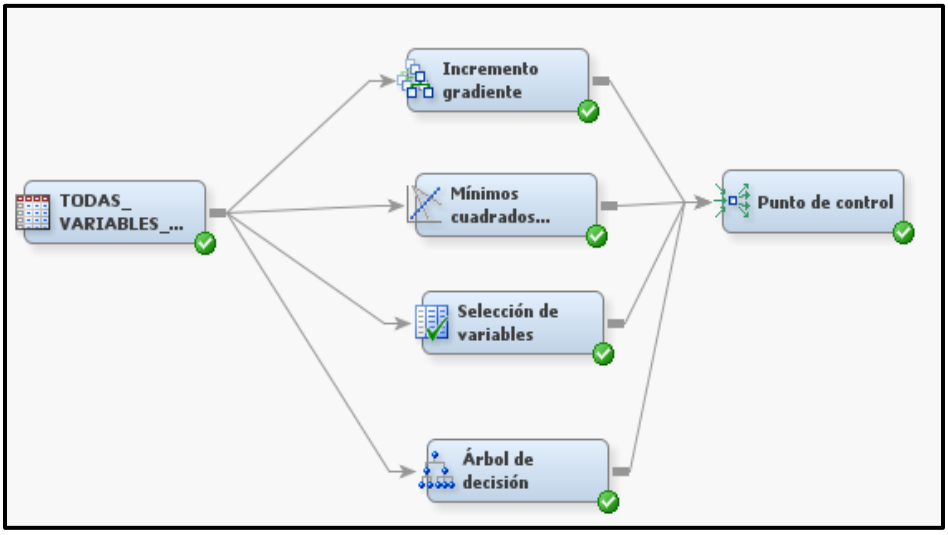


Figura 27. Selección de variables SAS Miner

Minimos cuadrados		Regresión Forward BIC	
Y13	impt_ptos_fun	Z10	EXP_REP_REP_impt_ptos_fun
Y15	ptos_fun	Y2	DIAS_GRAVE
Y2	DIAS_GRAVE	Z4	PWR_IMP_REP_REP_DIAS_MODERA
Y4	DIAS_MODERA	Z5	SQR_IMP_numero_diagnosticos
Y5	IMPORTE_ILT	Z11	PWR_REP_REP_ptos_est
Y8	numero_diagnosticos	Y4	DIAS_MODERA
Z10	EXP_REP_REP_impt_ptos_fun	Y7	importe_sec
Z12	PWR_REP_REP_ptos_fun	Y1	DIAS_BASICO
Z2	EXP_IMP_REP_REP_DIAS_GRAVE	Y5	IMPORTE_ILT
Z4	PWR_IMP_REP_REP_DIAS_MODERA	Y8	numero_diagnosticos
Z5	SQR_IMP_numero_diagnosticos	G6	G_REP_MEDICO
		Y3	DIAS_MGRAVE
		G4	G_REP_DIAGNOSTICO_2
		Y6	edad
		Y15	ptos_fun
		G2	G_IMP_REP_DIAGNOSTICO_1
Arbol		**Incremento gradiente**	
Z12	PWR_REP_REP_ptos_fun	Y13	impt_ptos_fun
Y2	DIAS_GRAVE	Y4	DIAS_MODERA
Y5	IMPORTE_ILT	x45	MAX_GRAVEDAD2-3-4
Z4	PWR_IMP_REP_REP_DIAS_MODERA	Y12	impt_ptos_est
x57	COD_DES_TIPO_SINIEST_DS05	A13	DIAGNOSTICO_1
x11	COD_SIT_LAB3	Y5	IMPORTE_ILT
Y12	impt_ptos_est	Y15	ptos_fun
x1	IND_COLISION_SI	x46	MAX_GRAVEDAD0-1
Y7	importe_sec	G1	G_M_Variable
Y4	DIAS_MODERA	Y7	importe_sec
x45	MAX_GRAVEDAD2-3-4	Z4	PWR_IMP_REP_REP_DIAS_MODERA
Z9	PWR_REP_REP_impt_ptos_est	Y17	M_Variable
x18	DIAGNOSTICO1_OTRO	A9	MEDICO
G6	G_REP_MEDICO	A8	DIAGNOSTICO_2
G4	G_REP_DIAGNOSTICO_2	Y1	DIAS_BASICO
Z11	PWR_REP_REP_ptos_est	Z12	PWR_REP_REP_ptos_fun
Y9	DIAS_DESDE_ALTA_HASTA_VALO	Y9	DIAS_DESDE_ALTA_HASTA_VALO

Tabla 21. Variables seleccionadas por cada método

Estudio de early stopping

Las redes neuronales pueden infra ajustar o sobre ajustar por eso una de las soluciones a este último problema es dividir los datos en training y validación y detener el proceso de estimación cuando el error en los datos de validación comience a aumentar.

Se muestra un ejemplo de early stopping figura 28. Donde se utiliza la función de activación Levmar y el número de nodos 10, se observa que la red neuronal no requiere de early stopping debido a que la línea de validación (línea azul) no crece por lo tanto se concluye que no es necesario.

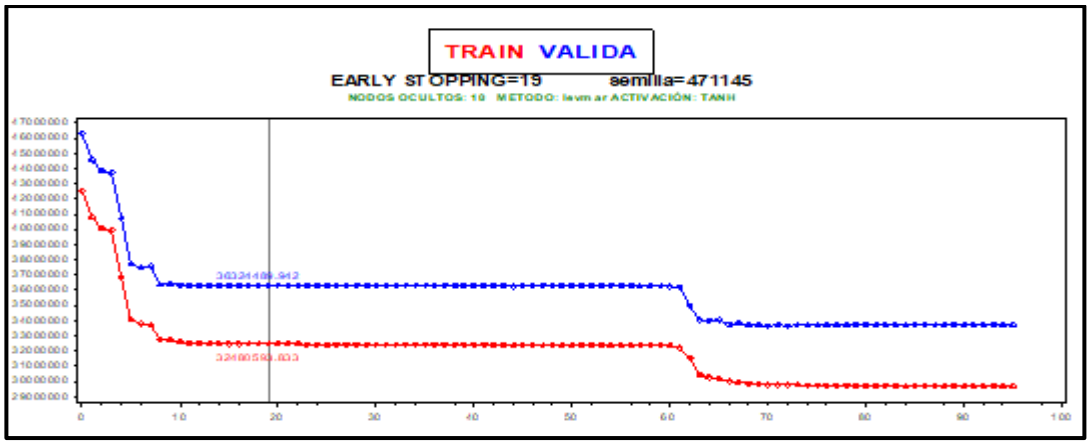


Figura 28. Early stopping Levmar 10 nodos

Se procede a realizar las redes con los diferentes modelos de selección de variables y configuración de early stopping, se prueba diferentes redes donde se varía el número de nodos, la función de activación, el algoritmo y la semilla. Cabe resaltar que primero se probó con unas redes iniciales y dependiendo del resultado se fue probando con las redes que presentaban mejor resultado. Se realiza un cálculo teórico del número de nodos a utilizar en las redes cuyo resultado es de 25 a 37 nodos. A continuación, se presenta todas las redes evaluadas y la media de los errores.

Modelo	Selección de variables	N.Variables	Nodos	Algoritmo	Activación	Early stopping	Semilla	Media Error
27	Arbol	17	25	Levmar	TANH	-	12365 12375	22,102,395
25	Arbol	17	25	Levmar	TANH	31	12345 12355	21,644,923
16	Arbol	17	15	Levmar	TANH	31	12345 12355	21,728,814
10	Arbol	17	20	Levmar	TANH	31	12345 12355	21,802,989
18	Mejor Regresión R	16	20	Levmar	LIN	21	12345 12355	21,844,970
19	Arbol	17	15	Levmar	TANH	31	12360 12370	21,923,770
22	Arbol	17	15	Levmar	TANH	-	12345 12355	22,302,486
21	Minimos cuadrados	11	20	Levmar	TANH	-	12345 12355	22,185,354
9	Minimos cuadrados	11	20	Levmar	TANH	42	12345 12355	22,327,990
13	Arbol	17	20	Levmar	TANH	-	12345 12355	22,365,922
20	Arbol	17	15	QUANEW	TANH	31	12345 12355	25,325,786
23	Arbol	17	15	Levmar	LOG	31	12345 12355	26,219,641
12	Regresión modelo 4	16	20	Levmar	TANH	27	12345 12355	28,259,765
17	Mejor Regresión R	16	20	Levmar	TANH	21	12345 12355	29,498,345
14	Regresión modelo 6	20	20	Levmar	TANH	21	12345 12355	29,436,851
15	Regresión modelo 5	19	20	Levmar	TANH	28	12345 12355	32,167,890
24	Arbol	17	15	NRRIDG	TANH	31	12345 12355	37,983,572
11	Arbol	17	20	BPROP -MOM:0.2-LEARN:0.1	TANH	7	12345 12355	3.00E+10
26	Arbol	17	25	BPROP -MOM:0.2-LEARN:0.1	TANH	-	12356 12366	3.10E+10

Tabla 22. Redes neuronales

En conclusión, se escogen dos redes y sus respectivas variaciones (cambio de semilla) para comparar con los modelos de regresión, se realiza mediante validación cruzada repetida con 10 iteraciones.

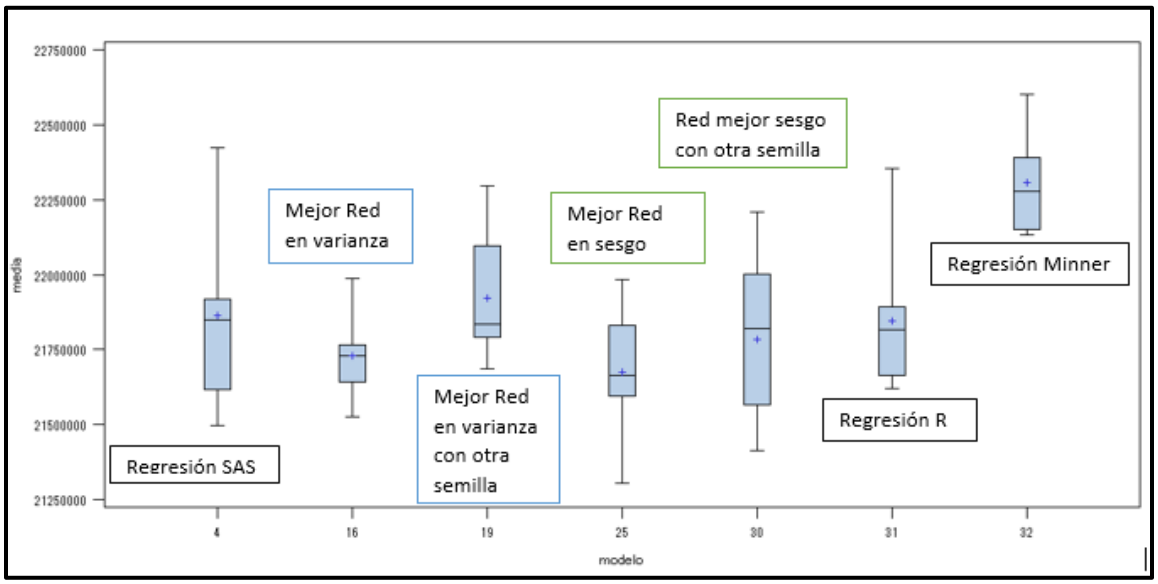


Figura 29. Comparación entre la red neuronal y regresión lineal

Se observa en la figura 29. Que el modelo 16 el cual es la mejor red en cuanto a varianza y sesgo no es estable, debido a que cuando se realiza el cambio de semilla se empeora, por lo tanto, se puede concluir que se prefiere la regresión obtenida en R por encima de la red, debido a que son modelos más sencillos y que se pueden explicar.

Posteriormente se realiza los modelos random forest y gradient boosting, (se explicarán estos modelos en la próxima sección) donde se realiza un tuneado en R para conocer la configuración de los parámetros a controlar en cada modelo.

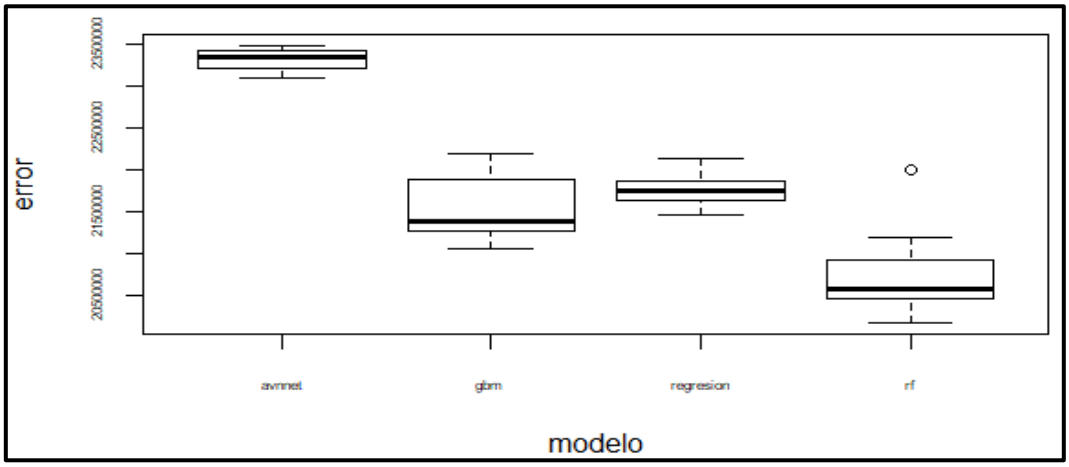


Figura 30. Validación cruzada modelo I

Se observa que el mejor modelo en cuanto a sesgo es el encontrado mediante random forest sin embargo presenta un dato atípico, no obstante, no supera el máximo umbral de los otros modelos.

Ensamblado

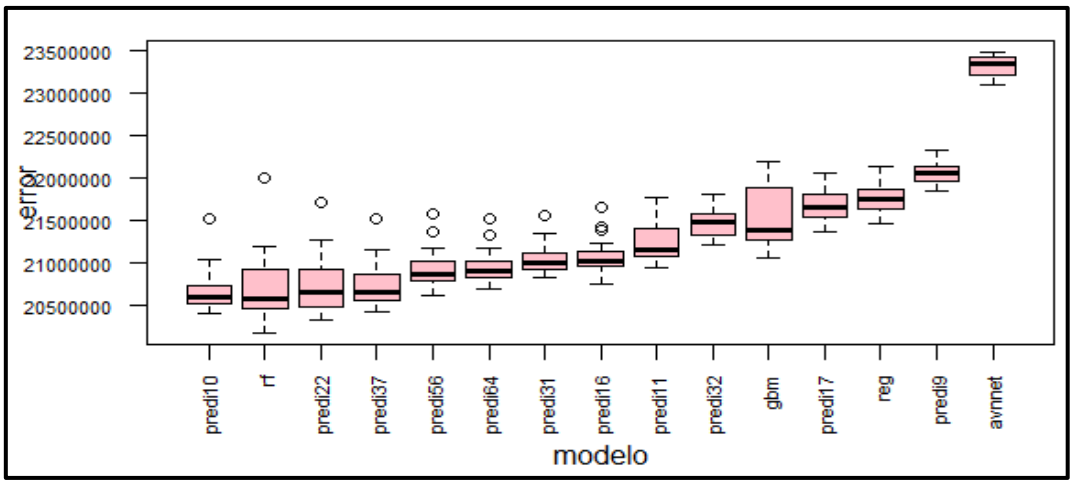


Figura 31. Diagrama de cajas ensamblado modelo variable objetivo continua

Se observa que al realizar ensamblado se obtiene un mejor modelo en comparación con los modelos tradicionales, predi10 es la combinación entre regresión y random forest. En la tabla 23. se muestra las variables seleccionadas y el modelo ganador.

Variables seleccionadas		Modelo ganador
Z10	impt_ptos_fun	**Ensamblado:** Regresión lineal y random Forest.
Y2	días_grave	
Z4	pwr_días_modera	**Configuración:**
Z5	sqr_imp_número_diagnóstico	
Z11	pwr_ptos_est	**Random Forest:**
Y4	días_modera	
Y7	importe_sec	ntree=200, sampsize=300, nodesize=10 y mtry=4
Y1	días_básico	
Y5	importe_ilt	
Y8	número_diagnóstico	
G6	G_rep_médico	
Y3	días_mgrave	
G4	G_rep_diagnóstico_2	
Y6	edad	
Y15	ptos_fun	
G2	G_rep_diagnóstico_1	

Tabla 23. Resultados variable objetivo continua

Validación

Se realiza una validación, con la librería caret para el modelo de random forest y regresión (ensamblado predi10). Calculando las predicciones con datos test y con un margen de error sugerido por la empresa del ±30%, se obtiene que de 2.748 observaciones se acierta 1.124 es decir un 41%.

Rango	Número aciertos	Muestra	%Aciertos
Menor o igual a 1.000 euros	29	512	6%
Entre 1.000 a 2.000 euros	212	648	33%
Entre 2.000 a 3.000 euros	324	487	67%
Entre 3.000 a 4.000 euros	228	358	64%
Entre 4.000 a 5.000 euros	115	204	56%
Mayor de 5.000 euros	216	539	40%
Total	**1124**	**2748**	**41%**

Tabla 24.Validación del modelo original

5. Modelo II: Modelo de clasificación

Hasta el momento no se ha logrado encontrar un buen modelo que prediga el coste de un lesionado, en la tabla 24 se evidencia que especialmente le cuesta al modelo predecir valores pequeños del coste, por lo que se propone limitar la muestra con un punto de corte de 3.000 euros, con el fin de si al reducir la muestra y tener unos valores más homogéneos se logre captar estos valores.

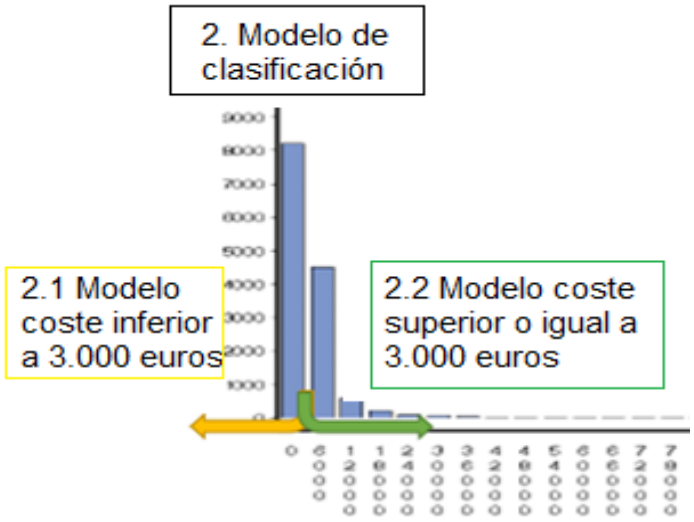


Figura 32. Representación del modelo de clasificación

La otra razón para fijar este punto de corte es que el 86% de la muestra ha tenido un coste inferior a los 6.000 euros por lo que resulta interesante fijar un punto intermedio para realizar la clasificación.

Por otra parte, esta sección se centra en explicar los modelos de random forest, gradient boosting y support vector machine. Para el modelo de clasificación se reduce la muestra a 10.752 observaciones para reducir el tiempo computacional.

5.1 Variable Objetivo binaria modelo de clasificación

La variable objetivo toma el valor de 1 cuando el coste de los lesionados supera los 3.000 euros de lo contrario 0. La proporción de 1 es de 5.212 observaciones equivalente al 48%, por su parte la proporción de 0 es de 5.540 equivalente al 52%.

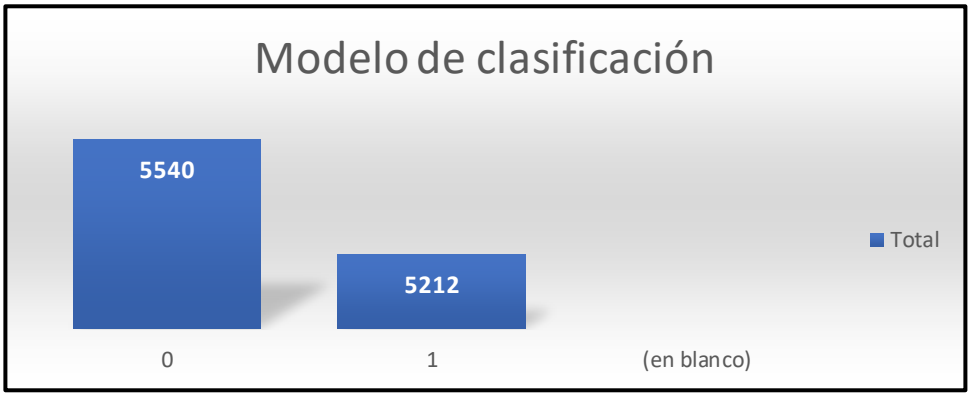


Figura 33. Variable objetivo modelo de clasificación

Se realiza una nueva selección de variables debido que al cambiar la naturaleza de la variable objetivo probablemente se encuentren variables que pueden explicar mejor la variable dependiente, igualmente se realizan los modelos de regresión logística y redes neuronales, tal como se explicó en el capítulo 4.

5.2 Modelización variable objetivo binaria: modelo de clasificación

5.2.1 Random Forest SAS base

Incorpora dos fuentes de variabilidad, el remuestreo de observaciones y el de variables. Es por esto que uno de los principales parámetros a controlar son el número de variables a muestrear en cada nodo, se elige probar con 3, 4 y 5. También se controla otros parámetros como las características propias de los árboles, la profundidad del árbol, donde se prueba con 5, 6 y 7. El número de divisiones máximas de cada nodo, se prueba con 2 y 4. El p valor de 0.05, 0.1, 0.2 y 0.3 y el número de observaciones mínimo en una rama-nodo de 15, 20 y 25.

Modelo	Maxtrees	Maxbranch	Tamhoja	Maxdepth	pvalor	variables	Media Error
19	100	4	15	6	0.05	5	0.4535
20	100	4	20	6	0.1	5	0.4534
37	200	2	20	6	0.1	4	0.4535
38	300	2	15	5	0.05	3	0.4530
39	400	2	25	7	0.2	3	0.4531
40	500	2	25	7	0.3	3	0.4533

Tabla 25. Random Forest modelo clasificación SAS base

5.2.2 Random Forest en R

Al igual que como se realizó en SAS base, se prueba diferentes combinaciones. En la tabla 26 se presenta las configuraciones y en la tabla 27 se muestra el promedio de la tasa de fallo y del área bajo la curva obtenidos.

mtry	ntree	nodesize	Model
12	200	10	Bagging
3	300	15	rf
2	400	20	rf2
4	500	20	rf3
3	1000	20	rf4

Tabla 26. Configuración RF modelo clasificación en R

Modelo	Promedio tasa error	Promedio AUC
Rf4	0.2099	0.8715
Rf	0.2106	0.8710
Rf3	0.2124	0.8714
Rf2	0.2114	0.8683
Bagging	0.2177	0.8654

Tabla 27. Resultados RF modelo clasificación R

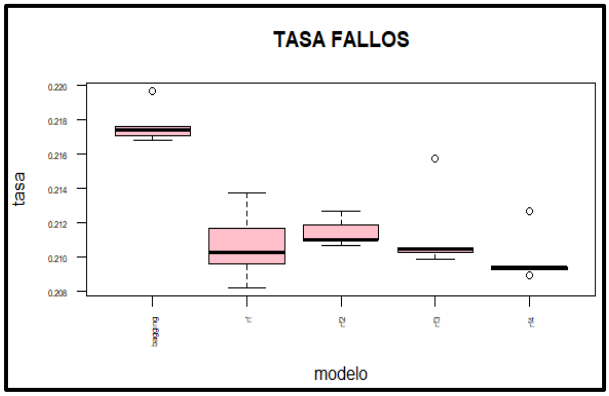


Figura 34. Diagrama de cajas de la tasa de fallos RF

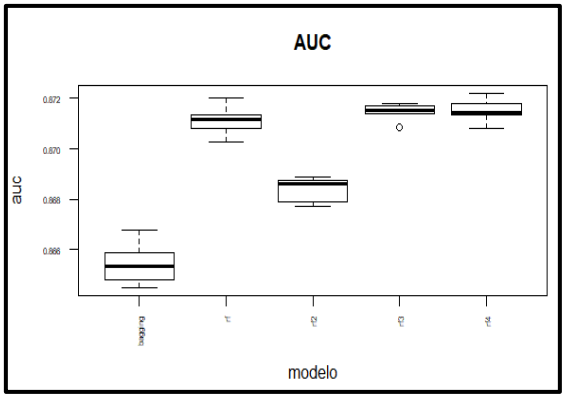


Figura 35. Diagrama de cajas de la AUC RF

Se observa que el número de mtry con el que se obtiene una menor tasa de error es de 3 (modelo rf y rf4), esto ya se había evidenciado en SAS, igualmente se prueba con ntree de 1000 y 300 dónde se observa que es mejor con 1000.

5.2.3 Gradient Boosting (gbm) SAS

Se realiza una primera prueba donde se cambia el parámetro shrink de 0.0001 a 0.2 dejando todo lo demás constante, se observa que cuanto más alto (0.2) se obtiene un mayor sesgo. Por su parte la mejor configuración se obtiene cuando es de 0.01. Posteriormente se prueba cambiando el número de iteraciones, donde con 400 se mejora el resultado (modelo28), luego se prueba cambiando la configuración propia del árbol donde se utiliza una división de 2, una profundidad de 4, 5 y 6 un mínimo número de observaciones de variables categóricas de 5, 10 y 15, un mínimo de observaciones de 15, 20, 30, 35 y 40 y por último un tamaño de hoja de 10, 15 y 20.

Modelo	Iteracciones	Shrink	Maxbranch	Maxdepth	Mincatsize	Minobs	leafsize	Sinicial	Sfinal	Error media
23	200	0.05	4	4	15	20	15	13345	13355	0.2174
24	200	0.1	4	4	15	20	15	13345	13355	0.2286
25	200	0.2	4	4	15	20	15	13345	13355	0.2427
26	200	0.001	4	4	15	20	15	13345	13355	0.2214
27	200	0.01	4	4	15	20	15	13345	13355	0.2118
28	400	0.01	4	4	15	20	15	13345	13355	0.2110
29	400	0.01	2	5	15	20	15	13345	13355	0.2074
30	400	0.01	2	5	15	30	15	13345	13355	0.2074
31	400	0.01	2	5	15	35	20	13345	13355	0.2073
32	400	0.01	2	5	5	40	20	13345	13355	0.2073
41	300	0.01	2	6	10	15	10	13345	13355	0.2081
70	400	0.01	2	5	5	35	20	13365	13375	0.2072
71	400	0.01	2	5	5	35	20	13385	13395	0.2076
72	400	0.01	2	5	15	35	20	13365	13375	0.2073

Tabla 28. Gradient Boosting modelo clasificación

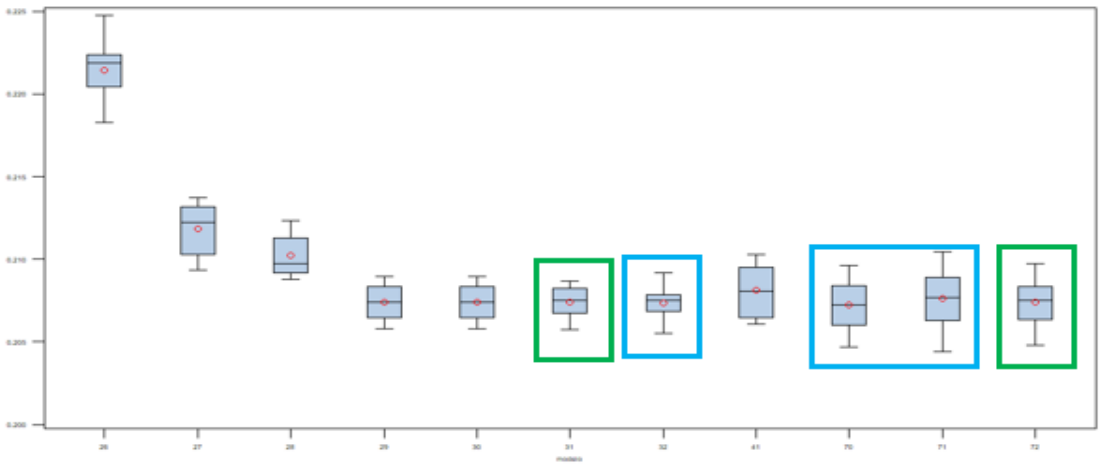


Figura 36. Diagrama de cajas Gradient Boosting

Se observa que el mejor modelo es el 32, sin embargo, cuando se cambia la semilla se observa que aumenta la variabilidad, por lo tanto, se prueba con el modelo 31 (subrayado en verde) al cambiar la semilla no aumenta considerablemente la variabilidad, entonces se elige el modelo 31 como el mejor modelo de gradient boosting.

5.2.4 Gradient Boosting (gbm) y Xgboost en R

Se realiza un tuneado, se presenta la configuración de los modelos con los que obtiene una mayor tasa de exactitud. También se representa gráficamente los parámetros (n.minobsnnode,shrinkage y n.trees) en los que se muestra las diferentes tasa de acierto.

shrinkage	n.minobsinnode	n.trees	Tasa acierto	Kappa
0.030	20	1000	0.7909226	0.5815966
0.030	10	1000	0.7907366	0.5812100
0.010	10	5000	0.7904576	0.5806488
0.050	5	1000	0.7904576	0.5806351
0.100	5	500	0.7904576	0.5806825
0.010	5	5000	0.7901786	0.5800833
0.030	5	1000	0.7901786	0.5801074
0.100	10	500	0.7900856	0.5799232
0.050	20	500	0.7899926	0.5797398
0.010	20	5000	0.7898065	0.5793480
0.050	5	500	0.7898065	0.5793527
0.050	10	1000	0.7898065	0.5793194
0.050	10	500	0.7897135	0.5791549
0.050	20	1000	0.7897135	0.5791548
0.100	20	500	0.7890625	0.5778734
0.030	10	500	0.7885045	0.5767392
0.030	5	500	0.7883185	0.5763620

Tabla 29. Tuneado Gradient Boosting

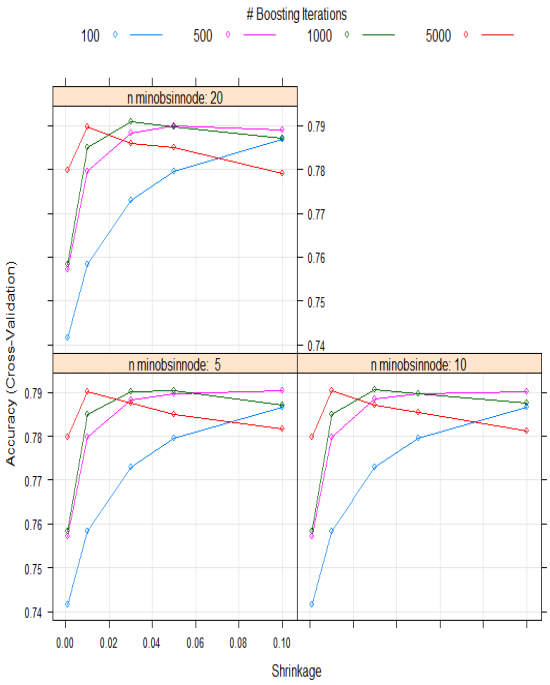


Figura 37.Representación gráfica principales parámetros Gradient Boosting

Se procede a realizar el modelo que sugiere el tuneado y también se prueba con diferentes opciones, en la siguiente tabla se presenta las configuraciones y la tasa de fallos y el AUC.

Shrinkage	ntree	nminobsinnode	Interac. Depth	Modelo	Tasa	AUC
0.01	400	35	2	gbm	0,21661088	0,86709502
0.01	800	20	2	gbm2	0,21210936	0,8740179
0.001	1000	25	2	gbm3	0,22546504	0,8506899
0.2	1000	20	2	gbm4	0,21400668	0,8694435
0.03	1000	20	2	gbm5	0,2079613	0,8774937

Tabla 30. Resultados Gradient Boosting

Se observa que el mejor modelo efectivamente se obtiene por medio de la configuración del tuneado, con una tasa de error de 0.20 y un AUC del 0.877. Gráficamente

se observa que es el que presenta menor sesgo y varianza a pesar de que tiene un dato atípico este no alcanza a superar el mínimo de los otros modelos.

Posteriormente se realiza el Xboost, el modelo que presenta una menor tasa de fallos es el xgbm2, el cual tiene un eta más pequeña, un nrounds más alto y una profundidad de árbol de 6.

Modelo	eta	nrounds	max_depth	gamma	Lambda	Tasa de fallos	AUC
xgbm	0.008	1000	2	0	0	0,21222098	0,87382434
xgbm1	0.08	100	6	0	0	0,21024926	0,87564464
xgbm2	0.001	5000	6	0	0	0,21021206	0,87650978

Tabla 31. Resultados Xgboost

5.2.5 Support Vector Machine (SVM) en R

El problema de optimización[16] puede plantearse en función de productos escalares, si se sustituye este producto por su Kernel, implícitamente se está aumentando la dimensión del espacio de variables utilizadas en el hiperplano de separación, sin la creación realmente de nuevas variables. Los Kernels más frecuentes son el polinomial, lineal y gaussiano.

Modelo	C	Nombre	Sigma	Tasa	AUC
Lineal	0.2	SVM		0,22038692	0,87066416
Polineal	0.01	SVMPOLY		0,21121652	0,85985544
Polineal	0.1	SVMPOLY2		0,21114212	0,85972416
Polineal	0.2	SVMPOLY4		0,21119172	0,86009462
Gaussiano	5	SVMRBF	0.01	0,21212796	0,86616286
Gaussiano	5	SVMRBF2	0.1	0,21789434	0,84142412
Gaussiano	1	SVMRBF3	0.1	0,21429812	0,84973154

Tabla 32. Resultados Support Vector Machine

5.2.6 Comparación de resultados SAS y R

Se observa que en general se obtienen resultados similares, regresión logistica en SAS tiene una tasa media de fallos de 0.2085 y en R de 0.2083. Redes en SAS de 0.2104 y en R 0.2105. Gradient boosting en SAS de 0.2073 y en R de 0.2079. Las grandes diferencias radican en random forest que en SAS obtuvo una tasa de fallo muy alta del 0.45 en cambio en R de 0.21, por lo que no se confia del resultado obtenido en SAS, no obstante en los dos programas este algoritmo individualmente es uno de los peores modelos.

SAS

Modelo	Tasa error medio
Regresión logística	0.2085
Redes neuronales	0.21043
Bagging	0.4543
Random Forest	0.4530
Gradient Boosting	0.2073
SVM lineal	0.3682

Tabla 33. Resultados de los modelos en SAS base

R

Modelo	Tasa error medio
Regresión logística	0.2083
Redes neuronales	0.2105
Random Forest	0.2141
Gradient Boosting	0.2079
SVM Lineal	0.2214
SVM Polinomial	0.2112
SVM Gaussiano	0.2124

Tabla 34. Resultados de los modelos en R

5.2.7 Ensamblado

Se realiza la comparación de los mejores ensamblados con los modelos básicos donde se observa que se logra reducir la tasa de error tanto en sesgo como varianza con los ensamblados.

El modelo predi 42 se componen por los algoritmos de regresión logística, gradient boosting y xgboost (los tres mejores) cuyo error es de 0.205, predi 11 por su parte se observa que tiene una caja más grande y es la unión de regresión logística y gradient boosting (gbm), por otro lado, predi64 se obtiene con 4 algoritmos entre ellos dos que individualmente no tienen un buen resultado los cuales son random forest y la red neuronal.

Por otra parte, se observa que uno de los algoritmos que tanto en SAS como R no se obtuvo un buen modelo es random forest.

TOP	Modelo	Tasa Fallo
1	predi42	0.2056455
2	predi11	0.2057850
3	predi64	0.2059338
4	predi32	0.2060361
5	predi31	0.2060547
6	predi33	0.2060826
7	predi65	0.2061291
8	predi66	0.2061291
9	predi56	0.2062779
10	gbm	0.2079148
11	logi	0.2083426
12	xgbm	0.2099237
13	avnnet	0.2105190
14	svmPoly	0.2112165
15	svmRadial	0.2124721
16	rf	0.2141183

Tabla 35. Ensamblado

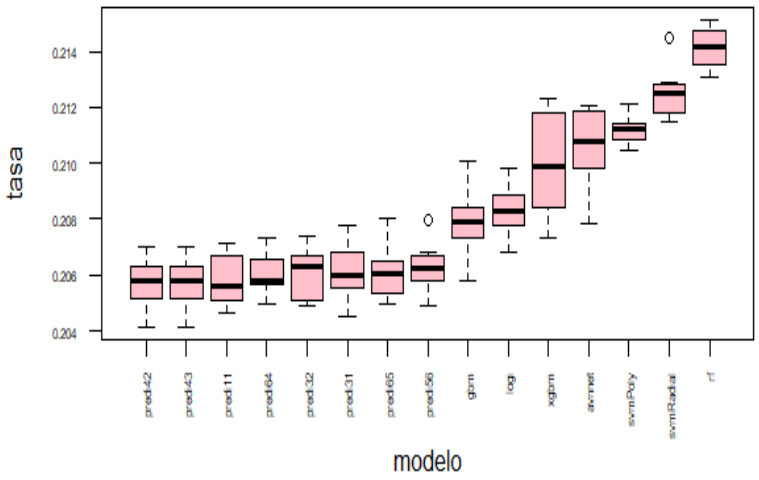


Figura 38. Diagrama de cajas Ensamblado

Por último, se representa gráficamente como clasifica un modelo tradicional contra un ensamblado, en este caso se toma el mejor ensamblado (predi42) y gradient boosting. Para interpretar el gráfico se nombra el cuadrante superior izquierdo como el cuadrante 1, posteriormente se nombra el cuadrante 2, 3 y 4 en sentido de las manecillas del reloj. Entonces del eje vertical hacia la derecha predi42 predice yes, por su parte del eje horizontal hacia arriba gbm predice yes. Es decir, un caso óptimo es que el cuadrante 2 sea todo de verde y el cuadrante 4 todo rojo y en el cuadrante 1 y 3 no hubiera puntos (son los cuadrantes de discrepancia). Se observa que hay levemente más puntos rojos en el primer cuadrante en comparación con el tercero, por lo que se concluye que gbm se equivoca ligeramente más que predi42.

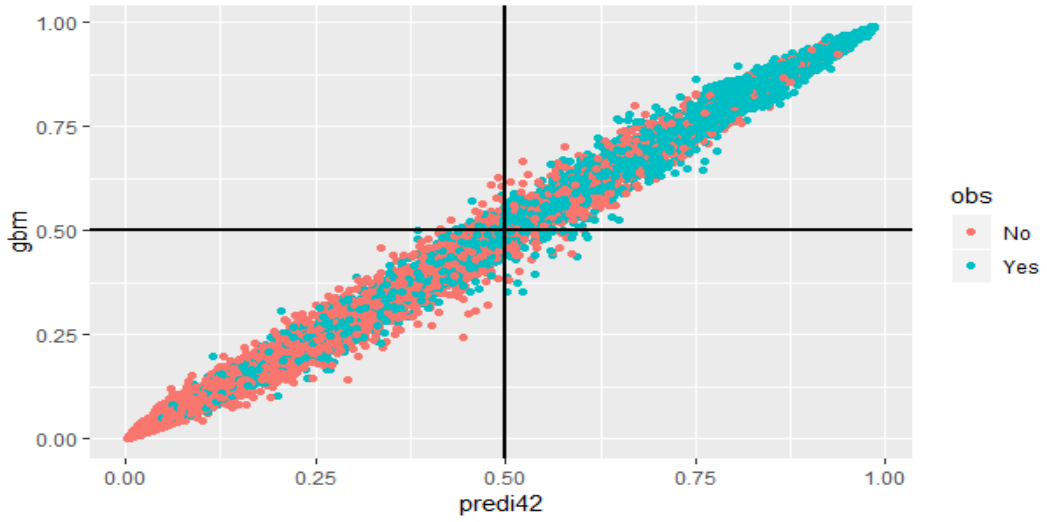


Figura 39. Representación de la clasificación ensamblado y un modelo tradicional

Se concluye que el modelo de clasificación encontrado es bueno debido a que la tasa de acierto es del 80%. En la tabla 36 se muestra las variables seleccionadas y el modelo ganador.

Variables seleccionadas		Modelo ganador
Y14	ptos_est	**Ensamblado:** Regresión logística, gradient boosting y xgboost unipredi$predi42<- (unipredi$logi+unipredi$gbm+unipredi$xgbm)/3 **Configuración:** **Gradient Boosting:** n.minobsinnode=20,shrinkage=0.03,n.trees=1000,interaction.depth=2
Y4	días_modera	
Y9	días_desde_hasta_valo	
GA6	Gcod_provincia	
Y1	días_básico	
Y8	número_diagnósticos	
GA13	Gdiagnóstico_1	
W_Y2	inv_días_grave	
W_Y15	LG10_ptos_fun	

T_Y5 OPT_importeilt T_Y6 OPT_edad	**Xgboost:** min_child_weight=10,eta=0.001,nrounds=5000,max_depth=6, gamma=0, colsample_bytree=1, subsample=1,alpha=0, lambda=0,lambda_bias=0

Tabla 36. Resultados modelo clasificación

5.3 Modelo de predicción del coste inferior a 3000 euros

Una vez que se clasifica el coste del siniestro, se procede a realizar un modelo con la variable objetivo continua, es decir poder predecir exactamente el coste que tiene el lesionado.

Se parte de la muestra depurada, donde se vuelve a realizar la transformación de variables independientes con el fin de encontrar la correlación máxima entre la variable objetivo y las variables explicativas y se realizan todos los modelos explicados anteriormente y se realiza validación cruzada repetida.

En la figura 40 se presenta el diagrama de cajas, donde se observa que el modelo que presenta menor sesgo es gradient Boosting (gbm), sin embargo, el de menor varianza es la red neuronal (avnnet).

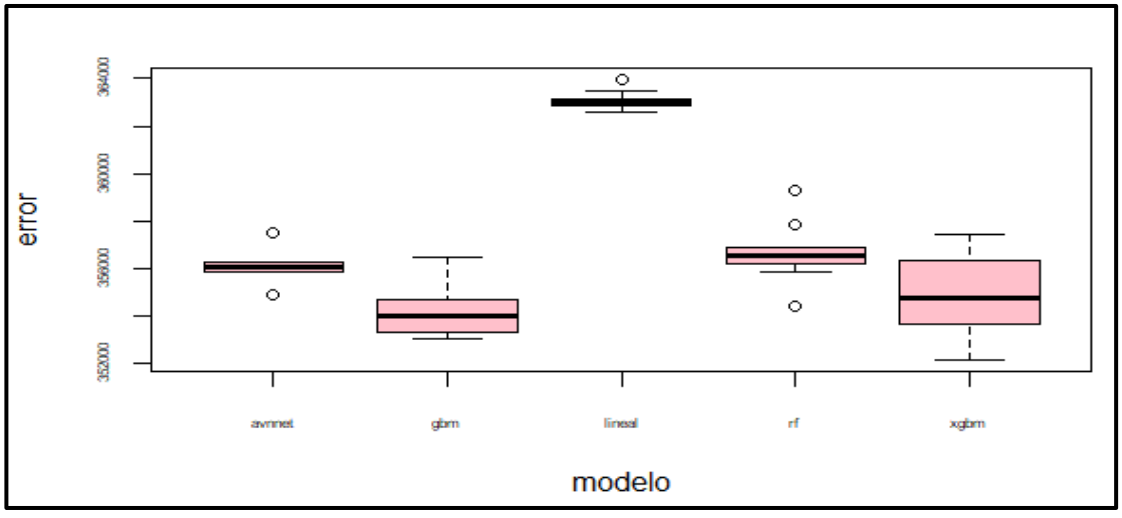


Figura 40. Diagrama de cajas modelo inferior 3000 euros

Ensamblado

Posteriormente se realiza el ensamblado, la mejor combinación se obtiene de los modelos regresión lineal, random forest, gradient boosting y red neuronal (predi 64). En la tabla 37 se muestra las variables seleccionadas y el modelo ganador.

Variables seleccionadas		Modelo ganador
W20	importe_ilt	**Ensamblado:** Regresión, random forest, gbm y red.unipredi$predi64<-(unipredi$reg+unipredi$rf+unipredi$gbm+unipredi$avnnet)/4
W9	días_básico	
W19	días_modera	**Configuración:**
Y7	importe_sec	
Y5	importe_ilt	**Random Forest:**
W21	edad	
W23	días_desde_hasta_valo	ntree=200, sampsize=300, nodesize=10, mtry=4
Y12	impt_ptos_est	
W22	importe_sec	**Gradient boosting:**
Y6	edad	
Y8	número_diagnósticos	shrinkage=c(0.010), n.minobsinnode=c(20), n.trees=c(5000), interaction.depth=c(2)
G6	G_Rep_médico	
G4	G_Rep_diagnóstico_2	**Red neuronal:**
G2	G_imp_rep_diagnóstico_1	size=c(20), decay=c(0.01)
A1	Cod_sit_lab	

Tabla 37. Resultados modelo del coste inferior 3000 euros

5.3.1 Validación del modelo inferior a 3000 euros

Una vez escogido el modelo ganador mediante validación cruzada repetida, se procede a realizar nuevamente la validación, para esto se parte la muestra en datos de entrenamiento y datos test (80,20) se construyen el modelo ganador utilizando la librería Caret de R, en este caso es un ensamblado de regresión lineal, random forest, gradient boosting y red neuronal. Se obtiene la predicción para cada observación, y con un margen de error, de más o menos el 30% (sugerido por la compañía) se obtienen los siguientes resultados:

Rangos	Aciertos	Total muestra	%Aciertos
Menor 1000 euros	80	498	16%
Mayor=1000 y menor que 2000	471	643	73%
Mayor=2000 y menor 3000	320	499	64%
Total	**871**	**1640**	**53%**

Tabla 38. Validación modelo inferior a 3.000 euros

5.4 Modelo de predicción del coste mayor o igual a 3000 euros

Se empieza realizando la transformación de variables independientes. Posteriormente se realiza la selección de variables, se calcula varios modelos de regresión, árbol y se estudia la importancia de las variables con random forest.

Se realizan diferentes modelos de cada algoritmo y se realiza validación cruzada repetida con los mejores modelos. En la figura 41 se presenta el diagrama de cajas de la validación cruzada. Se observa que random forest es el mejor modelo en cuanto a sesgo y varianza.

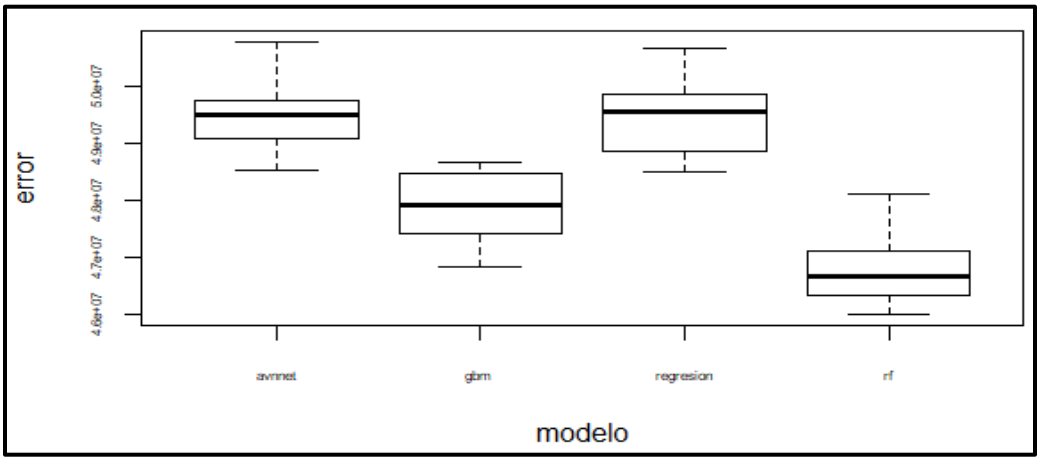


Figura 41. Boxplot validación cruzada modelo superior o igual 3000 euros

No obstante, se realiza un ensamble con el fin de intentar reducir el error al combinar diferentes modelos, en la figura 42 se muestra los resultados obtenidos, el cual predi 37 que es la combinación entre regresión lineal, random forest y gradient boosting (gbm), es el mejor modelo, aunque la diferencia no es muy significativa con respecto a ramdon forest, sin embargo, se observa que se logra disminuir levemente la varianza.

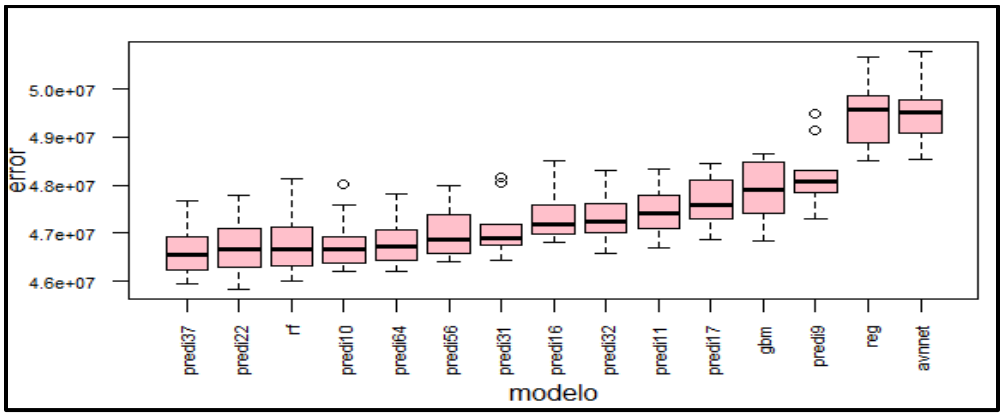


Figura 42. Boxplot validación cruzada ensamble y modelos tradicionales

Variables seleccionadas	
SQR_x45	max_gravedad2-3-4
PWR_Y13	impt_ptos_fun
EXP_Y2	días_grave
PWR_Y4	días_modera
Y3	días_mgrave
PWR_Y5	importe_ilt
PWR_Y8	número_diagnósticos
PWR_Y14	ptos_est
A5	cod_des_tipo_siniestro
A6	cod_provincia
EXP_Y3	días_mgrave
PWR_Y1	días_básico
PWR_Y11	mes_alta_num
PWR_Y6	edad
PWR_Y7	importe_sec
Y1	días_básico
Y13	impt_ptos_fun
Y4	días_modera
Y5	importe_ilt
Y6	edad

Modelo ganador

Ensambaldo: Regresión, random forest, gradient boosting

unipredi$predi37<-(unipredi$reg+unipredi$rf+unipredi$gbm)/3

Configuración:

Random Forest:

nodesize=10, ntree=600, mtry=4

Gradient Boosting:

n.minobsinnode=20,shrinkage=0.001,n.trees=5000,interaction.depth=2

Tabla 39. Resultado del modelo coste superior a 3000 euros

5.4.1 Validación del modelo superior o igual a 3000 euros

Nuevamente se realiza la validación del modelo mediante la librería caret de R, donde se parte la muestra en datos de entrenamiento y datos test, se construye el ensamble y se halla la predicción para cada observación, con un margen de error razonable para la empresa del ±30%, se acierta el 47%.

6. Modelo III: Transformación de la variable objetivo

6.1 Variable objetivo transformada

Por último se prueba si al transformar la variable objetivo, en este caso como el logaritmo del coste, se encuentren relaciones mas fuertes con las variables independientes. En la figura 43 se observa la distribución de la variable dependiente, cuya distribución se comporta como una distribución normal.

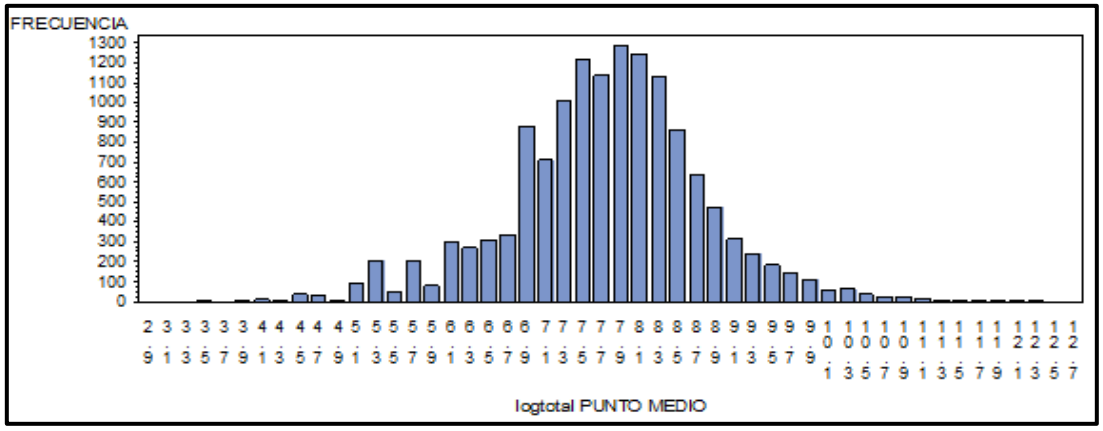


Figura 43.Logaritmo de coste variable objetivo transformada

6.2 Análisis de correlación

Se realiza el análisis de correlación de las variables independientes con el logaritmo de Total (variable objetivo transformada) para conocer si se encuentran una mayor correlación entre la variable transformada o la original. Por lo tanto, se presentan dos tablas, la primera es la correlación entre las variables independientes originales (sin transformar) y las variables dependientes transformada (LNTotal) y sin transformar (Total). En la segunda tabla por su parte, se muestra las correlaciones entre las variables independientes transformadas y la variable objetivo original y transformada.

	LN_TOTAL	Total
Y1	0.11267	-0.02975
Y1	<.0001	0.0005
Y2	0.35494	0.54449
Y2	<.0001	<.0001
Y3	0.08622	0.1512
Y3	<.0001	<.0001
Y4	0.60012	0.49483
Y4	<.0001	<.0001
Y5	0.63253	0.37107
Y5	<.0001	<.0001
Y6	0.18597	0.12604

	LN_TOTAL	Total
Z1	0.09828	-0.04452
Z1	<.0001	<.0001
Z2	0.34371	0.53854
Z2	<.0001	<.0001
Z3	0.0823	0.1424
Z3	<.0001	<.0001
Z4	0.4379	0.52062
Z4	<.0001	<.0001
Z5	0.19311	0.23175
Z5	<.0001	<.0001

	LN_TOTAL	Total
Y6	<.0001	<.0001
Y7	0.37626	0.19891
Y7	<.0001	<.0001
Y8	0.20708	0.17916
Y8	<.0001	<.0001
Y9	0.09163	-0.00487
Y9	<.0001	0.5683
Y10	0.00623	0.00148
Y10	0.4652	0.8618
Y11	0.00417	0.00065
Y11	0.6251	0.9389
Y12	0.32046	0.37988
Y12	<.0001	<.0001
Y13	0.55213	0.56972
Y13	<.0001	<.0001
Y14	0.32401	0.38379
Y14	<.0001	<.0001
Y15	0.56018	0.54696
Y15	<.0001	<.0001
LN_TOTAL	1	1

Tabla 40. Correlación var independ. y var depend. sin y con transf.

	LN_TOTAL	Total
Z6	0.09221	-0.00733
Z6	<.0001	0.3898
Z7	0.00669	0.00276
Z7	0.4325	0.7463
Z8	0.00344	0.00177
Z8	0.687	0.8355
Z9	0.31577	0.40443
Z9	<.0001	<.0001
Z10	0.52786	0.59405
Z10	<.0001	<.0001
Z11	0.32506	0.41331
Z11	<.0001	<.0001
Z12	0.42691	0.57984
Z12	<.0001	<.0001
LN_TOTAL	1	1

Tabla 41. Correlación var independ. Transf. y var depend. sin y con transf.

Se observa que al transformar la variable objetivo se logra una mayor correlación con las variables independientes originales por el contrario se tiene una correlación mayor entre las variables independientes transformadas con la variable objetivo original.

Entre las variables que se logra obtener una mayor correlación con la variable objetivo transformada es Y4 y Y5, días moderado e importe ilt con coeficientes de 0,6 y 0.63, respectivamente.

6.3 Modelización variable objetivo transformada

Se realiza una selección de variables y los diferentes modelos, se elige el modelo ganador mediante validación cruzada repetida. Como se observa en la figura 44, el mejor modelo es la Red neuronal de 15 nodos, decay 0.01.

Modelo	Promedio del error
Red (avnnet)	22.477.504
Gradient Boosting (gbm)	23.392.643
Regresión lineal	25.889.184
Random Forest	28.713.585

Tabla 42. Errores de los modelos con la variable objetivo transformada

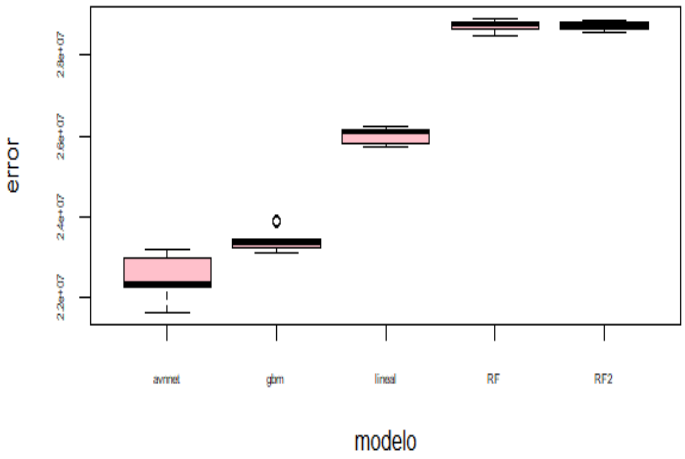


Figura 44.Boxplot modelo variable objetivo transformada

Finalmente se realiza el ensamblado, la mejor combinación se obtiene de los modelos red neuronal y gradient Boosting, como se observa en la figura 45.

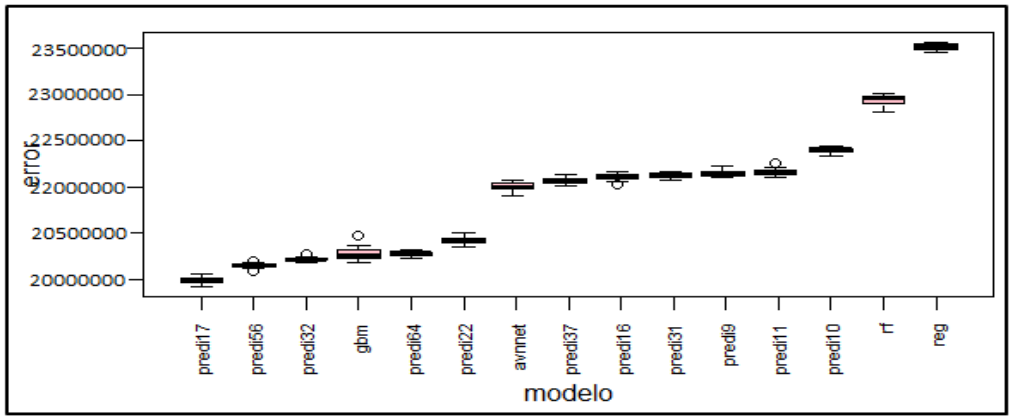


Figura 45. Boxplot ensamblado y modelos con la variable objetivo transf.

Variables seleccionadas	Modelo ganador
SQRT_impt_ptos_fun LOG_importe_ILT x45 max_gravedad2-3-4 LOG_días_básico LOG_edad Y13 impt_ptos_fun Y5 importe_ILT Y4 días_modera LOG_ptos_est	**Ensamblado:** Red neuronal y gradient boosting. unipredi$predi17<- (unipredi$avnnet+unipredi$gbm)/2 **Configuración:**

Y2 días_grave Y8 número_diagnósticos Y6 edad LOG_impt_ptos_est Y1 días_básico Y7 importe_sec Y3 días_mgrave A9 médico A13 diagnóstico_1 A8 diagnóstico_2 A6 cod_provincia	**Red neuronal:** 15 nodos y decay 0.01. **Gradient Boosting:** shrinkage=c(0.03), n.minobsinnode=c(20), n.trees=c(5000), interaction.depth=c(2)

Tabla 43. Resultado variable objetivo transformada

6.4 Validación modelo con la variable objetivo transformada

Se procede a realizar la validación, se construye la red neuronal de 15 nodos y decay 0.01 y el modelo gradient Boosting con un shrinkage=c(0.03), n.minobsinnode=c(20), n.trees=c(5000), interaction.depth=c(2). Se divide la muestra en datos entrenamiento y test y se predice para cada observación. Se deshace la transformación aplicando la inversa del logaritmo, es decir la exponencial tanto para la predicción como para la observación real. Con un intervalo del más o menos el 30% (sugerido por la compañía) se obtienen los siguientes resultados:

Rango	Número aciertos	Muestra	%Aciertos
Menor o igual a 1.000 euros	114	512	22%
Entre 1.000 a 2.000 euros	302	639	47%
Entre 2.000 a 3.000 euros	319	496	64%
Entre 3.000 a 4.000 euros	221	358	62%
Entre 4.000 a 5.000 euros	118	204	58%
Mayor de 5.000 euros	171	539	32%
Total	**1245**	**2748**	**45%**

Tabla 44. Resultados de la validación de la variable objetivo transformada

7. Comparación de los métodos y conclusiones

En esta sección se realiza una comparación entre la variable objetivo original y transformada y finalmente se presentan unas conclusiones.

7.1 Comparación entre el modelo original y la variable objetivo transformada

Se realiza una validación cruzada repetida con los dos mejores modelos de cada método, en la figura 46 se observa que predi 17 el cual corresponde al modelo ganador de

la variable objetivo transformada es levemente mejor que el modelo con la variable objetivo original.

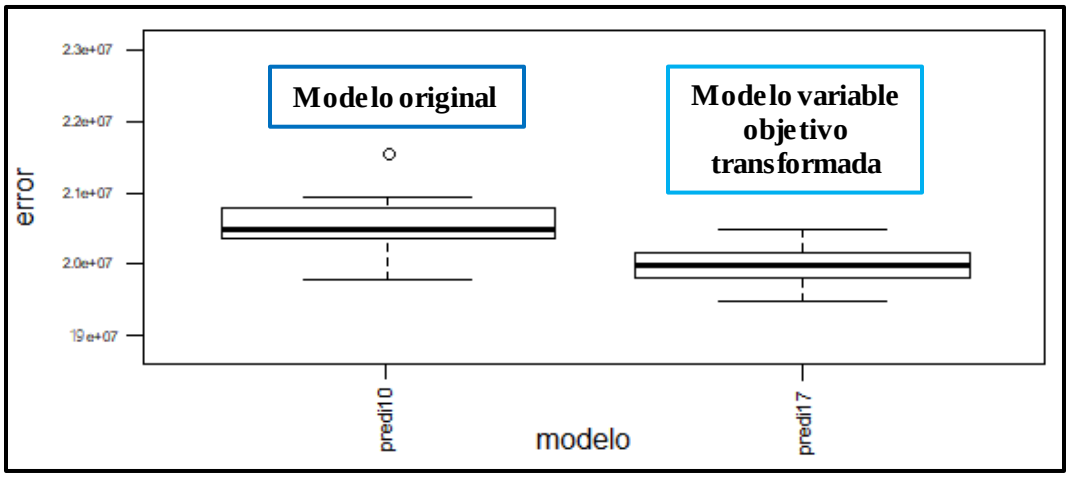


Figura 46. Diagrama de cajas de la variable objetivo original y transformada

Ahora bien, se realiza nuevamente la validación donde por rangos se compara la tasa de acierto, donde se observa que se logra aumentar la predicción para los valores más pequeños, sin embargo, se empeora para los valores altos, en general se acierta levemente más transformando la variable.

Rangos	Tasa de acierto	
	VO. Original	VO. Transformada
Menor o igual a 1.000 euros	6%	22%
Entre 1.000 a 2.000 euros	33%	47%
Entre 2.000 a 3.000 euros	67%	64%
Entre 3.000 a 4.000 euros	64%	62%
Entre 4.000 a 5.000 euros	56%	58%
Mayor de 5.000 euros	40%	32%
Total	**41%**	**45%**

Tabla 45. Comparación tasa de acierto entre modelo original y la transformación

7.2 Conclusiones

El objetivo principal de esta investigación es poder encontrar un modelo que prediga el coste que va a tener un lesionado con la primera valoración médica, independientemente de conocer en qué medida los factores influyen en la determinación del coste, es decir se trata de predecir mas no de explicar. Por tal motivo se realizó una extensa búsqueda, donde se plantea tres métodos diferentes, cada uno con sus respectivos algoritmos. Las conclusiones que se encontraron son las siguientes:

1. Al transformar la variable objetivo se logra reducir el error del modelo, especialmente en los valores pequeños del coste.

2. Se concluye que la mejor opción para predecir el coste es realizar primero el modelo de clasificación y posteriormente realizar los dos modelos para predecir exactamente el coste de un lesionado. Ya que al limitar la muestra se encuentra una mejor relación entre la variable objetivo y las independientes, sin embargo, se debe de tener en cuenta que el modelo de clasificación lleva consigo un error, que no es relativamente alto, pero lo tiene.

3. Se observa que con los variables independientes seleccionadas se obtiene un buen modelo de clasificación sin embargo al momento de predecir exactamente el valor del coste le cuesta más al algoritmo.

4. La combinación de diferentes algoritmos (ensamblado) permitió disminuir notablemente el error que al utilizar un modelo tradicional.

5. En cuanto a las variables más influyentes en el coste son los puntos funcionales y estéticos, el número de días que el médico determina que va a necesitar la lesión, la edad y el diagnóstico como tal.

8. Bibliografía

[1] «Más de 1,3 millones de españoles han sufrido lesiones en accidentes de tráfico durante la última década». [En línea]. Disponible en: https://www.abc.es/motor/reportajes/abci-mas-13-millones-espanoles-sufrido-lesiones-accidentes-trafico-durante-ultima-decada-201710161327_noticia.html. [Accedido: 22-abr-2019].

[2] «El coste de los lesionados por accidentes de tráfico asciende a 13.000 millones de euros anuales». [En línea]. Disponible en: https://confilegal.com/20171017-el-coste-de-los-lesionados-por-accidentes-de-trafico-asciende-a-13-000-millones-de-euros-anuales/. [Accedido: 22-abr-2019].

[3] «Análisis de accidentes de tránsito con inteligencia computacional | Congreso Chileno de Ingeniería de Transporte». [En línea]. Disponible en: https://revistas.uchile.cl/index.php/CIT/article/view/28446. [Accedido: 17-jul-2019].

[4] A. de Smet, *Transportation Accident Analysis and Prevention*. Nova Publishers, 2008.

[5] M. M. H. Rodriguez, «GRADO DE MÁSTER EN INGENIERÍA Y TECNOLOGÍA DE SISTEMAS SOFTWARE», p. 75.

[6] J. de Oña, R. O. Mujalli, y F. J. Calvo, «Analysis of traffic accident injury severity on Spanish rural highways using Bayesian networks», *Accid. Anal. Prev.*, vol. 43, n.º 1, pp. 402-411, ene. 2011.

[7] C. Montt, F. Castro, y N. Rodríguez, «Análisis de Accidentes de Tránsito con Máquinas de Soporte Vectorial LS-SVM», *Ing. Transp.*, vol. 15, n.º 2, nov. 2011.

[8] J. M. Vargas y J. A. Giraldo, «Modelo de Predicción de Costos en Servicios de Salud Soportado en Simulación Discreta», *Inf. Tecnológica*, vol. 25, n.º 4, pp. 175-184, 2014.

[9] S. Pérez y C. Mestre «Evaluación de un modelo de predicción del gasto farmacéutico en atención primaria de salud basado en variables demográficas | CIEGS», 2013.

[10] «SAS® Enterprise Miner™ 14.3: Reference Help», p. 1808.

[11] A.Calviño, «Apuntes de la asignatura Técnicas y metodología de la minería de datos (SEMMA).», Facultad de Estudios Estadísticos, Universidad Complutense de Madrid, 2018.

[12] J. G. Molina y M. F. Rodrigo, «El modelo de regresión lineal», p. 18.

[13] R. Salas, «Redes Neuronales Artificiales», p. 7.

[14] C. A. Ruiz, M. S. Basualdo, y D. J. Matich, «Redes Neuronales: Conceptos Básicos y Aplicaciones.», p. 55.

[15] L. Breiman, «Random Forests», *Mach. Learn.*, vol. 45, n.º 1, pp. 5-32, oct. 2001.

[16] J. Portela, «Apuntes de la asignatura Machine learning.», Facultad de Estudios Estadísticos, Universidad Complutense de Madrid, 2018.

[17] Alonso Revenga J.M, «Técnicas avanzadas de predicción» Facultad de Estudios Estadísticos, Universidad Complutense de Madrid, 2018.

[18] M. C. Carrasco, «Facultad de Matemáticas Departamento de Estadística e Investigación Trabajo Fin de Grado en Matemáticas», p. 52.

9. Anexos

Tablas de frecuencia:

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
COD_PROVINCIA	PR15		833
COD_PROVINCIA	PR07		798
COD_PROVINCIA	PR05		671
COD_PROVINCIA	PR26		357
COD_PROVINCIA	PR16		355
COD_PROVINCIA	PR22		316
COD_PROVINCIA	PR03		309
COD_PROVINCIA	PR11		271
COD_PROVINCIA	PR27		263
COD_PROVINCIA	PR28		256
COD_PROVINCIA	PR08		230
COD_PROVINCIA	PR37		228
COD_PROVINCIA	PR25		191
COD_PROVINCIA	PR42		166
COD_PROVINCIA	PR13		159
COD_PROVINCIA	PR21		137
COD_PROVINCIA	PR18		133
COD_PROVINCIA	PR23		122
COD_PROVINCIA	PR17		118
COD_PROVINCIA	PR19		112
COD_PROVINCIA	PR20		112
COD_PROVINCIA	PR31		111
COD_PROVINCIA	PR38		105
COD_PROVINCIA	PR24		103
COD_PROVINCIA	PR12		101
COD_PROVINCIA	PR40	**OTRA**	97
COD_PROVINCIA	PR35	**OTRA**	93

Tabla 46. Agrupación de provincias

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
COD_SIT_LAB	LA03		5714
COD_SIT_LAB	LA02		4238
COD_SIT_LAB	LA05		1557
COD_SIT_LAB	LA04		1222
COD_SIT_LAB	LA01		612
COD_SIT_LAB	LA06		409
COD_SIT_LAB	_UNKNOWN_	**_DEFAULT_**	.
COD_SIT_LAB	_UNKNOWN_	**_DEFAULT_**	.

Tabla 47. Frecuencia de la sit. laboral

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
DIAGNOSTICO_1	X20		3642
DIAGNOSTICO_1	X56		1697
DIAGNOSTICO_1	X169		1197
DIAGNOSTICO_1	X145		919
DIAGNOSTICO_1	X47		490
DIAGNOSTICO_1	X55		392
DIAGNOSTICO_1	X58		369
DIAGNOSTICO_1	X65		264
DIAGNOSTICO_1	X39		254
DIAGNOSTICO_1	X32		239
DIAGNOSTICO_1	X28		228
DIAGNOSTICO_1	X40		203
DIAGNOSTICO_1	X186		193
DIAGNOSTICO_1	X92		135
DIAGNOSTICO_1	X38		107
DIAGNOSTICO_1	X121		104
DIAGNOSTICO_1	X73		99
DIAGNOSTICO_1	X64		97
DIAGNOSTICO_1	X84		94
DIAGNOSTICO_1	X62		90
DIAGNOSTICO_1	X27		89
DIAGNOSTICO_1	X89		89
DIAGNOSTICO_1	X99		87
DIAGNOSTICO_1	X2		86
DIAGNOSTICO_1	X85		84
DIAGNOSTICO_1	X1		81
DIAGNOSTICO_1	X61	**OTRO**	79

Tabla 48. Frecuencia del diagnóstico1

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
DIAGNOSTICO_2	0		5507
DIAGNOSTICO_2	X145		786
DIAGNOSTICO_2	X56		744
DIAGNOSTICO_2	X47		676
DIAGNOSTICO_2	X169		643
DIAGNOSTICO_2	X39		584
DIAGNOSTICO_2	X32		569
DIAGNOSTICO_2	X28		352
DIAGNOSTICO_2	X40		228
DIAGNOSTICO_2	X27		217
DIAGNOSTICO_2	X24		204
DIAGNOSTICO_2	X186		181
DIAGNOSTICO_2	X35		169
DIAGNOSTICO_2	X38		145
DIAGNOSTICO_2	X55		144
DIAGNOSTICO_2	X2		128
DIAGNOSTICO_2	X33		124
DIAGNOSTICO_2	X65		106
DIAGNOSTICO_2	X58		92
DIAGNOSTICO_2	X25		87
DIAGNOSTICO_2	X1		80
DIAGNOSTICO_2	X121		80
DIAGNOSTICO_2	X36		77
DIAGNOSTICO_2	X164		66
DIAGNOSTICO_2	X112		63
DIAGNOSTICO_2	X61		63
DIAGNOSTICO_2	X64		62

Tabla 49. Frecuencia del diagnóstico 2

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
MEDICO	Y18		454
MEDICO	Y56		401
MEDICO	Y22		393
MEDICO	Y42		367
MEDICO	Y4		338
MEDICO	Y44		331
MEDICO	Y33		301
MEDICO	Y1		296
MEDICO	Y63		284
MEDICO	Y38		261
MEDICO	Y23		254
MEDICO	Y7		245
MEDICO	Y65		243
MEDICO	Y14		235
MEDICO	Y41		233
MEDICO	Y24		230
MEDICO	Y51		229
MEDICO	Y40		228
MEDICO	Y61		217
MEDICO	Y25		215
MEDICO	Y13		213
MEDICO	Y5		210
MEDICO	Y50		202
MEDICO	Y58		201
MEDICO	Y32		200
MEDICO	Y11		195
MEDICO	Y15		195

Tabla 50. Frecuencia de Médico

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
MIN_GRAVEDAD	1	**0-1**	12679
MIN_GRAVEDAD	2	**2-3-4**	819
MIN_GRAVEDAD	-1	**_MISSING_**	196
MIN_GRAVEDAD	.	**_MISSING_**	42
MIN_GRAVEDAD	0	**0-1**	12
MIN_GRAVEDAD	3	**2-3-4**	3
MIN_GRAVEDAD	4	**2-3-4**	1
MIN_GRAVEDAD	_UNKNOWN_	**_DEFAULT_**	.
MIN_GRAVEDAD	_UNKNOWN_	**_DEFAULT_**	.

Tabla 51. Frecuencia de Min_Gravedad

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
REF_MODELO	IBIZA		133
REF_MODELO	MEGANE		114
REF_MODELO	GOLF		111
REF_MODELO	LEON		85
REF_MODELO	FOCUS		79
REF_MODELO	ASTRA	**OTRO**	66
REF_MODELO	FIESTA	**OTRO**	59
REF_MODELO	XSARA	**OTRO**	58
REF_MODELO	CORSA	**OTRO**	53
REF_MODELO	POLO	**OTRO**	52
REF_MODELO	SERIE 3	**OTRO**	52
REF_MODELO	206	**OTRO**	50
REF_MODELO	PASSAT	**OTRO**	45

Tabla 52. Frecuencia del modelo

Variable	Valor formateado	Valor de reemplazo	Número de ocurrencias
IND_COLISION	S		10150
IND_COLISION	N		3602
IND_COLISION	_UNKNOWN_	**_DEFAULT_**	.
IND_COLISION	_UNKNOWN_	**_DEFAULT_**	.
IND_SEXO_VICTIMA		**_MISSING_**	7625
IND_SEXO_VICTIMA	H		3316
IND_SEXO_VICTIMA	M		2811
IND_SEXO_VICTIMA	_UNKNOWN_	**_DEFAULT_**	.
IND_SEXO_VICTIMA	_UNKNOWN_	**_DEFAULT_**	.
IND_SIN_CONTRARIO	S		1124
IND_SIN_CONTRARIO	N		250
IND_SIN_CONTRARIO		**_MISSING_**	4
IND_SIN_CONTRARIO	_UNKNOWN_	**_DEFAULT_**	.
IND_SIN_CONTRARIO	_UNKNOWN_	**_DEFAULT_**	.
IND_VEH_REPOSO	N		1367
IND_VEH_REPOSO	S		73
IND_VEH_REPOSO	_UNKNOWN_	**_DEFAULT_**	.
IND_VEH_REPOSO	_UNKNOWN_	**_DEFAULT_**	.
MAX_GRAVEDAD	1	**0-1**	1106
MAX_GRAVEDAD	2	**2-3-4**	2536
MAX_GRAVEDAD	3	**2-3-4**	60
MAX_GRAVEDAD	.	**_MISSING_**	42
MAX_GRAVEDAD	-1	**_MISSING_**	28
MAX_GRAVEDAD	0	**0-1**	11
MAX_GRAVEDAD	4	**2-3-4**	8
MAX_GRAVEDAD	_UNKNOWN_	**_DEFAULT_**	.
MAX_GRAVEDAD	UNKNOWN	**DEFAULT**	.

Tabla 53. Frecuencia del resto de variables

Código:

Modelo original

```
## Librerias
library(glmnet)
library(sas7bdat)
library(caret)
## Función R2
Rsq<-function(modelo,varObj,datos){
  testpredicted<-predict(modelo, datos)
  testReal<-datos[,varObj]
  sse <- sum((testpredicted - testReal) ^ 2)
  sst <- sum((testReal - mean(testReal)) ^ 2)
  1 - sse/sst
}

## Lectura datos

datos<-read.sas7bdat('C:/)

## Partición de datos
set.seed(2611)
partitionIndex <- createDataPartition(datos$Total, p=0.8, list=FALSE)
data_train <- datos[partitionIndex,]
data_test <- datos[-partitionIndex,]
modeloganador<-lm(Total~G2+ G4+ G6+ Y1+ Y15+ Y2+ Y3+
  Y4+ Y5+ Y6+ Y7+ Y8+ Z10+ Z11+ Z4+ Z5, data=data_train)

## Selección de variables sin INTERACCIONES

#aic

null<-lm(Total~1, data=data_train)

full<-lm(Total~., data=data_train)

modeloStepAIC<-step(null, scope=list(lower=null, upper=full), direction="both")
```

```
modeloBackAIC<-step(full, scope=list(lower=null, upper=full), direction="backward")

modeloForwardAIC<-step(null,scope=list(lower=null, upper=full), direction="forward")

#bic

modeloStepBIC<-step(null,scope=list(lower=null, upper=full),
direction="both",k=log(nrow(data_train)))

modeloBackBIC<-step(full, scope=list(lower=null, upper=full),
direction="backward",k=log(nrow(data_train)))

modeloForBIC<-step(null, scope=list(lower=null, upper=full),
direction="forward",k=log(nrow(data_train)))


## Regresión LASSO

y <- as.double(as.matrix(data_train[, 1])) # 7 es la columna de la Variable Objetivo
x<-model.matrix(Total~., data=data_train)[,-1]# Tranformar
set.seed(1712)
cv.lasso <- cv.glmnet(x,y)
plot(cv.lasso)
betas<-coef(cv.lasso, s=cv.lasso$lambda.1se)
row.names(betas)[which(betas!=0)]

## Validación cruzada repetida

total<-c()
modelos<-
sapply(list(modeloPreliminar,modeloBackAIC,modeloBackBIC,modeloForBIC,modeloStep
AIC,modeloStepBIC),formula)
for (i in 1:length(modelos)){
  set.seed(1712)
  vcr<-train(as.formula(modelos[[i]]), data = data_train, method = "lm", trControl =
trainControl(method="repeatedcv", number=5, repeats=20,
                               returnResamp="all")
  )
  total<-rbind(total,cbind(vcr$resample[,1:2],modelo=rep(paste("Modelo",i),
                                        nrow(vcr$resample))))
}

#falta el lasso

set.seed(1712)
vcr<-train(as.formula(modelos[[i]]), data = data_train,
        method = "glmnet",
```

```
            tuneGrid=expand.grid(.alpha=1,.lambda=cv.lasso$lambda.1se),
            trControl = trainControl(method="repeatedcv", number=5, repeats=20,
                                     returnResamp="all")
        )
        total<-rbind(total,cbind(vcr$resample[,1:2],modelo=rep('LASSO',
                                                  nrow(vcr$resample))))

        boxplot(RMSE~modelo,data=total,main="RMSE")
        boxplot(Rsquared~modelo,data=total,main="R-Square")
        aggregate(Rsquared~modelo, data = total, mean)
        aggregate(Rsquared~modelo, data = total, sd)
```

Ensambaldo Modelo original:

```
        archivo<-datos
        vardep="Total"
        listconti=c("Z10","Y2","Z4","Z5","Z11","Y4","Y7","Y1","Y5","Y8","G6","Y3","G4","Y6","
Y15","G2")
        listclass=c("")
        grupos=4
        sinicio=12345
        repe=20

        medias1<-cruzadalin(data=archivo,
         vardep=vardep,listconti=listconti,
         listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe)

        medias1bis<-as.data.frame(medias1[1])
        medias1bis$modelo<-"regresion"
        predi1<-as.data.frame(medias1[2])
        predi1$reg<-predi1$pred

        medias2<-cruzadaavnnet(data=archivo,
         vardep=vardep,listconti=listconti,
         listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe,
            size=c(10),decay=c(0.1),repeticiones=5,itera=200,trace=FALSE)

        medias2bis<-as.data.frame(medias2[1])
        medias2bis$modelo<-"avnnet"
        predi2<-as.data.frame(medias2[2])
        predi2$avnnet<-predi2$pred

        medias3<-cruzadarf(data=archivo,
          vardep=vardep,listconti=listconti,
```

```
 listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe,
mtry=4,ntree=200,nodesize=10,replace=TRUE)

medias3bis<-as.data.frame(medias3[1])
medias3bis$modelo<-"rf"
predi3<-as.data.frame(medias3[2])
predi3$rf<-predi3$pred
medias4<-cruzadagbm(data=archivo,
 vardep=vardep,listconti=listconti,
 listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe,
n.minobsinnode=10,shrinkage=0.1,n.trees=100,interaction.depth=2)

medias4bis<-as.data.frame(medias4[1])
medias4bis$modelo<-"gbm"
predi4<-as.data.frame(medias4[2])
predi4$gbm<-predi4$pred

## Librerias
library(caret)
## Partición de datos
set.seed(2611)
partitionIndex <- createDataPartition(datos$Total, p=0.8, list=FALSE)
data_train <- datos[partitionIndex,]
data_test <- datos[-partitionIndex,]

rfgrid<-expand.grid(mtry=c(4))

control<-trainControl(method = "none",savePredictions = "all")

rf<- train(Total~Y1+    Y2+    Y3+    Y4+    Y5+    Y6+    Y7+    Y8+    Y15+
 Z4+    Z5+    Z10+   Z11+   G2+    G4+    G6,
```

```
data=data_train,
        method="rf",trControl=control,tuneGrid=rfgrid,
        ntree=200,sampsize=300,nodesize=10,replace=TRUE,
        importance=TRUE)

t2<-as.data.frame(cbind(predict(rf,data_test),data_test))
t1<-as.data.frame(cbind(predict(modeloganador,data_test),data_test))
write.csv(t2,"resultadosrfOriginal.csv",row.names = FALSE, na="")
write.csv(t1,"resultadosregresionOriginal.csv",row.names = FALSE, na="")
```

Modelo de clasificación (segundo modelo):

```
library(sas7bdat)
datos<-read.sas7bdat("C:/")
datos$vo<-factor(datos$vo)

# EJEMPLOS RANDOMFOREST

# TUNEADO DE MTRY CON CARET

library(caret)

set.seed(12345)
rfgrid<-expand.grid(mtry=c(3,4,5,6,7,8,9,10,11))
control<-trainControl(method = "cv",number=4,savePredictions = "all",
                classProbs=TRUE)
rf<- train(factor(vo)~.,data=datos,
        method="rf",trControl=control,tuneGrid=rfgrid,
        linout = FALSE,ntree=1000,sampsize=200,nodesize=10,replace=TRUE,
        importance=TRUE)
rf
summary(saheartbis)
summary(datos)

# IMPORTANCIA  DE VARIABLES
```

```
final<-rf$finalModel
tabla<-as.data.frame(importance(final))
tabla<-tabla[order(-tabla$MeanDecreaseAccuracy),]
tabla

barplot(tabla$MeanDecreaseAccuracy,names.arg=rownames(tabla))



library(randomForest)
set.seed(12345)
rfbis<-randomForest(factor(vo)~.,
                    data=datos,
                    mtry=3,ntree=1000,sampsize=300,nodesize=10,replace=TRUE)
plot(rfbis$err.rate[,1])



for (muestra in seq(100,450,50))
{
  # controlamos la semilla pues bagging depende de ella
  set.seed(12345)
  rfbis<-randomForest(factor(vo)~.,
                      data=datos,
                      mtry=5,ntree=5000,sampsize=muestra,nodesize=10,replace=TRUE)

  plot(rfbis$err.rate[,1],main=muestra,ylim=c(0.25,0.5))

}

# Ahora se comprueba con validaciÃ³n cruzada con caret

rfgrid<-expand.grid(mtry=c(5))



rf<- train(factor(chd)~.,data=saheartbis,
           method="rf",trControl=control,tuneGrid=rfgrid,
           linout = FALSE,ntree=1000,sampsize=100,nodesize=10,replace=TRUE)
```

```
rf

# La funciÃ³n cruzadarfbin permite plantear random forest
load ("C:/ ")
source ("c:/cruzadas avnnet y log binaria.R")
source ("c:/cruzada arbolbin.R")
source ("c:/cruzada rf binaria.R")

medias1<-cruzadalogistica(data=datos,

 vardep="vo",listconti=c("Y1",  "Y4",  "Y8",  "Y9",  "Y14",  "Y17",  "GA6",  "GA13",  "W_Y2",
"W_Y15"), listclass=c("T_Y5","T_Y6"), grupos=4,sinicio=1234,repe=10)

medias1$modelo="LogÃ???stica"

medias2<-cruzadaavnnetbin(data=datos,

                          vardep="vo",listconti=c("Y1",  "Y4",  "Y8",  "Y9",  "Y14",  "Y17",  "GA6",
"GA13",  "W_Y2",  "W_Y15"),

                          listclass=c("T_Y5","T_Y6"),grupos=4,sinicio=1234,repe=10,

                          size=c(15),decay=c(0.1),repeticiones=5,itera=200)

medias2$modelo="avnnet"


medias3<-cruzadaarbolbin(data=datos,
                         vardep="vo",listconti=c("Y1",  "Y4",  "Y8",
                                                 "Y9",       "Y14",    "Y17",    "GA6",    "GA13",    "W_Y2",
"W_Y15"),
                         listclass=c("T_Y5","T_Y6"),grupos=4,sinicio=1234,repe=5,
                         cp=c(0),minbucket =5)


medias3$modelo="arbol"
#Este era medias 4
medias54<-cruzadarfbin(data=datos, vardep="vo",
                       listconti=c("Y1", "Y4", "Y8",
```

```
                "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
        listclass=c("T_Y5","T_Y6"),
        grupos=4,sinicio=1234,repe=5,nodesize=10,
        mtry=12,ntree=200,replace=TRUE)

medias54$modelo="bagging"

#Este era medias 5
medias55<-cruzadarfbin(data=datos, vardep="vo",
        listconti=c("Y1", "Y4", "Y8",
                "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
        listclass=c("T_Y5","T_Y6"),
        grupos=4,sinicio=1234,repe=5,nodesize=15,
        mtry=3,ntree=300,replace=TRUE)

medias55$modelo="rf"
medias51<-cruzadarfbin(data=datos, vardep="vo",
        listconti=c("Y1", "Y4", "Y8",
                "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
        listclass=c("T_Y5","T_Y6"),
        grupos=4,sinicio=1234,repe=5,nodesize=20,
        mtry=2,ntree=400,replace=TRUE)

medias51$modelo="rf2"
medias52<-cruzadarfbin(data=datos, vardep="vo",
         listconti=c("Y1", "Y4", "Y8",
                 "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
         listclass=c("T_Y5","T_Y6"),

         grupos=4,sinicio=1234,repe=5,nodesize=20,

         mtry=4,ntree=500,replace=TRUE)

medias52$modelo="rf3"
medias53<-cruzadarfbin(data=datos, vardep="vo",
         listconti=c("Y1", "Y4", "Y8",
```

```
                    "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
          listclass=c("T_Y5","T_Y6"),
          grupos=4,sinicio=1234,repe=5,nodesize=20,
          mtry=3,ntree=1000,replace=TRUE)

medias53$modelo="rf4"
medias56<-cruzadarfbin(data=datos, vardep="vo",
          listconti=c("Y1", "Y4", "Y8",
                    "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
          listclass=c("T_Y5","T_Y6"),
          grupos=4,sinicio=1245,repe=5,nodesize=20,
          mtry=3,ntree=1000,replace=TRUE)

medias56$modelo="rf4_2"


union90<-rbind(medias51,medias52,medias53,medias54,medias55,medias56)
par(cex.axis=0.5)
boxplot(data=union90,tasa~modelo,main="TASA FALLOS",col="pink")
boxplot(data=union90,auc~modelo,main="AUC")

#medias
medias60<-cruzadagbmbin(data=datos, vardep="vo",
listconti=c("Y1", "Y4", "Y8",
        "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
listclass=c("T_Y5","T_Y6"),
grupos=4,sinicio=1234,repe=5,
n.minobsinnode=35,shrinkage=0.01,n.trees=400,interaction.depth=2)


 medias60$modelo="gbm"
  medias61<-cruzadagbmbin(data=datos, vardep="vo",
          listconti=c("Y1", "Y4", "Y8",
```

```
                    "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                listclass=c("T_Y5","T_Y6"),
                grupos=4,sinicio=1234,repe=5,
                n.minobsinnode=20,shrinkage=0.01,n.trees=800,interaction.depth=2)

    medias61$modelo="gbm2"

    medias62<-cruzadagbmbin(data=datos, vardep="vo",
                listconti=c("Y1", "Y4", "Y8",
                    "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                listclass=c("T_Y5","T_Y6"),
                grupos=4,sinicio=1234,repe=5,
n.minobsinnode=25,shrinkage=0.001,n.trees=1000,interaction.depth=2)

    medias62$modelo="gbm3"

    medias63<-cruzadagbmbin(data=datos, vardep="vo",
                listconti=c("Y1", "Y4", "Y8",
                    "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                listclass=c("T_Y5","T_Y6"),
                grupos=4,sinicio=1234,repe=5,
                n.minobsinnode=20,shrinkage=0.2,n.trees=1000,interaction.depth=2)

    medias63$modelo="gbm4"

    medias64<-cruzadagbmbin(data=datos, vardep="vo",
                listconti=c("Y1", "Y4", "Y8",
                    "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                listclass=c("T_Y5","T_Y6"),
```

```
                    grupos=4,sinicio=1234,repe=5,
n.minobsinnode=20,shrinkage=0.03,n.trees=1000,interaction.depth=2)

 medias64$modelo="gbm5"

 union91<-rbind(medias60,medias61,medias62,medias63,medias64)
 par(cex.axis=0.5)
 boxplot(data=union91,tasa~modelo,main="TASA FALLOS",col="pink")
 boxplot(data=union91,auc~modelo,main="AUC")

medias20<-cruzadaSVMbin(data=datos, vardep="vo",
                listconti=c("Y1", "Y4", "Y8",
                        "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                listclass=c("T_Y5","T_Y6"),
                grupos=4,sinicio=1234,repe=5,
                C=0.2)
medias20$modelo="SVM"
medias21<-cruzadaSVMbinPoly(data=datos, vardep="vo",
                  listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                  listclass=c("T_Y5","T_Y6"),
                  grupos=4,sinicio=1234,repe=5,
                  C=0.01,degree=2,scale=2)

medias21$modelo="SVMPoly"

medias23<-cruzadaSVMbinPoly(data=datos, vardep="vo",
                   listconti=c("Y1", "Y4", "Y8",
                           "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                   listclass=c("T_Y5","T_Y6"),
```

```
                    grupos=4,sinicio=1234,repe=5,
                    C=0.1,degree=2,scale=2)

medias23$modelo="SVMPoly2"

medias24<-cruzadaSVMbinPoly(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1234,repe=5,
                    C=0.2,degree=2,scale=2)

medias24$modelo="SVMPoly4"


medias22<-cruzadaSVMbinRBF(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1234,repe=5,
                    C=5,sigma=0.01)

medias22$modelo="SVMRBF"

medias25<-cruzadaSVMbinRBF(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1234,repe=5,
                    C=5,sigma=0.1)

medias25$modelo="SVMRBF2"
```

```
medias26<-cruzadaSVMbinRBF(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1234,repe=5,
                    C=1,sigma=0.1)

medias26$modelo="SVMRBF3"

union92<-
rbind(medias20,medias21,medias22,medias23,medias24,medias25,medias26)

par(cex.axis=0.5)

boxplot(data=union92,tasa~modelo,main="TASA FALLOS",col="pink")

boxplot(data=union92,auc~modelo,main="AUC")


medias7<-cruzadaxgbmbin(data=datos, vardep="vo",
listconti=c("Y1", "Y4", "Y8",
          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
listclass=c("T_Y5","T_Y6"),
grupos=4,sinicio=1234,repe=5,
min_child_weight=10,eta=0.008,nrounds=1000,max_depth=2,
gamma=0,colsample_bytree=1,subsample=1)
medias7$modelo="xgbm"



union1<-
rbind(medias1,medias2,medias3,medias4,medias5,medias6,medias20,medias21,medias2
```

```
2,medias23,medias24,medias25,medias26,medias51,medias52,medias61,medias62,medi
as63)
    medias8<-cruzadaxgbmbin(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                            "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1234,repe=5,
                    min_child_weight=10,eta=0.08,nrounds=100,max_depth=6,
                    gamma=0,colsample_bytree=1,subsample=1,
                    alpha=0,lambda=0,lambda_bias=0)
    medias8$modelo="xgbm1"
    medias9<-cruzadaxgbmbin(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                            "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1234,repe=5,
                    min_child_weight=10,eta=0.001,nrounds=5000,max_depth=6,
                    gamma=0,colsample_bytree=1,subsample=1,
                    alpha=0,lambda=0,lambda_bias=0)

    medias9$modelo="xgbm2"
    union91<-rbind(medias60,medias61,medias62,medias63,medias64)
    par(cex.axis=0.5)
    boxplot(data=union91,tasa~modelo,main="TASA FALLOS",col="pink")
    boxplot(data=union91,auc~modelo,main="AUC")
    par(cex.axis=0.5)
    boxplot(data=union1,tasa~modelo,main="TASA FALLOS",col="pink")
    boxplot(data=union1,auc~modelo,main="AUC")



    union2<-rbind(medias5,medias20,medias52,medias61)
    boxplot(data=union2,tasa~modelo,main="TASA FALLOS",col="pink")
```

```
boxplot(data=union2,auc~modelo,main="AUC")


medias100<-cruzadarfbin(data=datos, vardep="vo",
                listconti=c("Y1", "Y4", "Y8",
                        "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                listclass=c("T_Y5","T_Y6"),
                grupos=4,sinicio=1245,repe=10,nodesize=15,
                mtry=3,ntree=300,replace=TRUE)

medias100$modelo="rf"


medias101<-cruzadaSVMbin(data=datos, vardep="vo",
                 listconti=c("Y1", "Y4", "Y8",
                         "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                 listclass=c("T_Y5","T_Y6"),
                 grupos=4,sinicio=1245,repe=10,
                 C=0.2)

medias101$modelo="SVM"


medias102<-cruzadarfbin(data=datos, vardep="vo",
                listconti=c("Y1", "Y4", "Y8",
                        "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                listclass=c("T_Y5","T_Y6"),
                grupos=4,sinicio=1245,repe=10,nodesize=20,
                mtry=4,ntree=500,replace=TRUE)

medias102$modelo="rf3"
```

```
medias103<-cruzadagbmbin(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1245,repe=10,
                    n.minobsinnode=20,shrinkage=0.01,n.trees=800,interaction.depth=2)

medias103$modelo="gbm2"

medias104<-cruzadagbmbin(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
                    grupos=4,sinicio=1245,repe=10,

n.minobsinnode=20,shrinkage=0.03,n.trees=1000,interaction.depth=2)

medias104$modelo="gbm5"

medias105<-cruzadarfbin(data=datos, vardep="vo",
                   listconti=c("Y1", "Y4", "Y8",
                         "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                   listclass=c("T_Y5","T_Y6"),
                   grupos=4,sinicio=1245,repe=10,nodesize=20,
                   mtry=3,ntree=1000,replace=TRUE)

medias105$modelo="rf4"

medias106<-cruzadaxgbmbin(data=datos, vardep="vo",
                    listconti=c("Y1", "Y4", "Y8",
                          "Y9",  "Y14", "Y17", "GA6", "GA13", "W_Y2", "W_Y15"),
                    listclass=c("T_Y5","T_Y6"),
```

```
                    grupos=4,sinicio=1245,repe=10,
                    min_child_weight=10,eta=0.001,nrounds=5000,max_depth=6,
                    gamma=0,colsample_bytree=1,subsample=1,
                    alpha=0,lambda=0,lambda_bias=0)


    medias106$modelo="xgbm2"

    union2<-rbind(medias100,medias101,medias102,medias103,
medias104,medias105,medias106)

    boxplot(data=union2,tasa~modelo,main="TASA FALLOS",col="pink")

    boxplot(data=union2,auc~modelo,main="AUC")
```

Modelo menor 3000 euros:

```
    library(sas7bdat)

    datos<-read.sas7bdat("C:/ ")

    medias1<-cruzadaavnnet(data=datos,

    vardep="Total",listconti=c("W20","W9","W19","Y7","Y5","W21","W23","Y12","W22","Y
6","Y8","G6","G4","G2"),

    listclass=c("A1"),grupos=4,sinicio=1234,repe=5,

    size=c(20),decay=c(0.01),repeticiones=10,itera=100)

    medias1$modelo="avnnet"


    medias2<-cruzadaavnnet(data=datos,
                    vardep="Total",listconti=c("W20",  "W9",  "W19","Y7",  "Y5",
  "W21","W23", "Y12","W22",  "Y6",  "Y8",  "G6",  "G4",  "G2"),
                    listclass=c(""),grupos=4,sinicio=1234,repe=5,
                    size=c(15),decay=c(0.01),repeticiones=5,itera=100)
    medias2$modelo="avnnet2"
```

```
medias1<-cruzadaavnnet(data=compressbien,
                       vardep="cstrength",listconti=c("age","water","cement","blast"),
                       listclass=c(""),grupos=4,sinicio=1234,repe=5,
                       size=c(15),decay=c(0.01),repeticiones=5,itera=100)


medias1$modelo="avnnet"


medias2<-cruzadalin(data=datos,
vardep="Total",listconti=c("W20","W9","W19","Y7","Y5","W21","W23","Y12","W22","Y
6","Y8","G6","G4","G2"),
listclass=c("A1"),grupos=4,sinicio=1234,repe=10)

medias2$modelo="lineal"


medias3<-cruzadaarbol(data=data,
vardep="cstrength",listconti=c("age","water","cement","blast"),
listclass=c(""),
 grupos=4,sinicio=1234,repe=5,cp=0,minbucket=5)

medias3$modelo="arbol"


medias4<-cruzadarf(data=datos,

vardep="Total",listconti=c("W20","W9","W19","Y7","Y5","W21","W23","Y12","W22","Y6","Y
8","G6","G4","G2"),
listclass=c("A1"),
  grupos=4,sinicio=1234,repe=10,
  nodesize=10,replace=TRUE,ntree=200,mtry=4)


medias4$modelo="rf"


medias5<-cruzadagbm(data=datos,
```

```
vardep="Total",listconti=c("W20","W9","W19","Y7","Y5","W21","W23","Y12","W22","Y6","Y
8","G6","G4","G2"),
                    listclass=c("A1"),
                    grupos=4,sinicio=1234,repe=10,
                    n.minobsinnode=20,shrinkage=0.010,n.trees=5000,interaction.depth=2)

    medias5$modelo="gbm"


    medias6<-cruzadaxgbm(data=datos,

vardep="Total",listconti=c("W20","W9","W19","Y7","Y5","W21","W23","Y12","W22","Y6","Y
8","G6","G4","G2"),
                    listclass=c("A1"),
                    grupos=4,sinicio=1234,repe=10,
                    min_child_weight=20,eta=0.010,nrounds=500,max_depth=6,
                    gamma=0,colsample_bytree=1,subsample=1)


    medias6$modelo="xgbm"
    union1<-rbind(medias1,medias2,medias4,medias5,medias6)


    par(cex.axis=0.5)
    boxplot(data=union1,error~modelo,col="pink")

    union1<-rbind(medias4,medias5)


    par(cex.axis=0.5)
    boxplot(data=union1,error~modelo,col="pink")


    Validación:

    #Validación:

    control<-trainControl(method = "none",savePredictions = "all")
```

```
#Red

nnetgrid <-  expand.grid(size=c(20),decay=c(0.01))

rednnet<-
train(Total~W20+W9+W19+Y7+Y5+W21+W23+Y12+W22+Y6+Y8+G6+G4+G2,data=data
_train,

                method="nnet",linout                                                    =
TRUE,maxit=100,trControl=control,tuneGrid=nnetgrid)



t1<-as.data.frame(cbind(predict(rednnet,data_test),data_test))

write.csv(t1,"resultadosredmenor3000A1.csv",row.names = FALSE, na="")

#RF
rfgrid<-expand.grid(mtry=c(4))
rf<-
train(Total~W20+W9+W19+Y7+Y5+W21+W23+Y12+W22+Y6+Y8+G6+G4+G2+A1,
        data=data_train,
        method="rf",trControl=control,tuneGrid=rfgrid,
        ntree=200,sampsize=300,nodesize=10,replace=TRUE,
        importance=TRUE)

t2<-as.data.frame(cbind(predict(rf,data_test),data_test))

write.csv(t2,"resultadosrfmenor3000A1.csv",row.names = FALSE, na="")

#GBM
gbmgrid<-expand.grid(shrinkage=c(0.010),
                n.minobsinnode=c(20),
                n.trees=c(5000),
                interaction.depth=c(2))

gbm<-
train(Total~W20+W9+W19+Y7+Y5+W21+W23+Y12+W22+Y6+Y8+G6+G4+G2,data=data
_train,
```

```
        method="gbm",trControl=control,tuneGrid=gbmgrid,

        distribution="gaussian", bag.fraction=1,verbose=FALSE)

    t3<-as.data.frame(cbind(predict(gbm,data_test),data_test))

    write.csv(t3,"resultadosgbmmenor3000.csv",row.names = FALSE, na="")

    #REGRESION

    regresion<-
lm(Total~W20+W9+W19+Y7+Y5+W21+W23+Y12+W22+Y6+Y8+G6+G4+G2,
data=data_train)

    t4<-as.data.frame(cbind(predict(regresion,data_test),data_test))

    write.csv(t4,"resultadosregmenor3000A1.csv",row.names = FALSE, na="")
```

Modelo mayor o igual a 3.000 euros:

```
    library(sas7bdat)

    datos<-read.sas7bdat("C:/ ")


    medias1<-cruzadaavnnet(data=datos,

    vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR
_Y5","PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y
1","Y13","Y4","Y5","Y6"),

    listclass=c(""),grupos=4,sinicio=1234,repe=10,

    size=c(15),decay=c(0.01),repeticiones=5,itera=100)

    medias1$modelo="avnnet"

    medias12<-cruzadaavnnet(data=datos,
vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),

                listclass=c(""),grupos=4,sinicio=1234,repe=10,

                size=c(20),decay=c(0.01),repeticiones=5,itera=100)


    medias12$modelo="avnnet2"
```

```
medias13<-cruzadaavnnet(data=datos,

vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),
                        listclass=c(""),grupos=4,sinicio=1234,repe=10,
                        size=c(15),decay=c(0.1),repeticiones=5,itera=100)


medias13$modelo="avnnet3"
medias2<-cruzadalin(data=datos,
vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR
_Y5","PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y
1","Y13","Y4","Y5","Y6"),
listclass=c("A5","A6"),grupos=4,sinicio=1234,repe=10)


medias2$modelo="lineal"


medias3<-cruzadaarbol(data=data,
vardep="cstrength",listconti=c("age","water","cement","blast"),
listclass=c(""),
 grupos=4,sinicio=1234,repe=5,cp=0,minbucket=5)

medias3$modelo="arbol"


medias4<-cruzadarf(data=data,
 vardep="cstrength",listconti=c("age","water","cement","blast"),
listclass=c(""),
  grupos=4,sinicio=1234,repe=5,
  nodesize=10,replace=TRUE,ntree=200,mtry=4)


medias4$modelo="bagging"
medias5<-cruzadarf(data=datos,

vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
```

```
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),
        listclass=c("A5","A6"),
          grupos=4,sinicio=1234,repe=10,
          nodesize=10,replace=TRUE,ntree=600,mtry=4)


        medias5$modelo="rf"
        medias51<-cruzadarf(data=datos,

vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),
                        listclass=c("A5","A6"),
                        grupos=4,sinicio=1234,repe=10,
                        nodesize=15,replace=TRUE,ntree=800,mtry=4)


        medias51$modelo="rf1"


        medias6<-cruzadagbm(data=datos,

vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),
        listclass=c("A5","A6"),
          grupos=4,sinicio=1234,repe=10,
        n.minobsinnode=20,shrinkage=0.001,n.trees=5000,interaction.depth=2)


        medias6$modelo="gbm"




        medias61<-cruzadagbm(data=datos,

vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
```

```
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),
                    listclass=c("A5","A6"),
                    grupos=4,sinicio=1234,repe=10,
                    n.minobsinnode=20,shrinkage=0.001,n.trees=1000,interaction.depth=2)

      medias61$modelo="gbm1"
      medias7<-cruzadaxgbm(data=datos,
vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),
                     listclass=c("A5","A6"),
                     grupos=4,sinicio=1234,repe=10,
                     min_child_weight=10,eta=0.03,nrounds=1000,max_depth=6,
                     gamma=0,colsample_bytree=1,subsample=1)

      medias7$modelo="xgbm"
      medias71<-cruzadaxgbm(data=datos,

vardep="Total",listconti=c("SQR_x45","PWR_Y13","EXP_Y2","PWR_Y4","Y3","PWR_Y5","
PWR_Y8","PWR_Y14","EXP_Y3","PWR_Y1","PWR_Y11","PWR_Y6","PWR_Y7","Y1","Y1
3","Y4","Y5","Y6"),
                     listclass=c("A5","A6"),
                     grupos=4,sinicio=1234,repe=10,
                     min_child_weight=10,eta=0.001,nrounds=1000,max_depth=6,
                     gamma=0,colsample_bytree=1,subsample=1)

      medias71$modelo="xgbm1"


      union1<-rbind(medias1,medias2,medias3,medias4,medias5,medias6)

      par(cex.axis=0.5)
      boxplot(data=union1,error~modelo,col="pink")
```

```
union1<-rbind(medias1,medias3,medias4,medias5,medias6)


par(cex.axis=0.5)
boxplot(data=union1,error~modelo,col="pink")
```

Modelo transformación variable objetivo:

```
library(sas7bdat)

datos<-read.sas7bdat("C:/ ")
medias1<-cruzadaavnnet(data=datos,
vardep="LN_TOTAL",listconti=c("SQRT_Y13",  "LOG_Y5",  "x45",
  "LOG_Y1",  "LOG_Y6",  "Y13",  "Y5",  "Y4",  "LOG_Y14",  "Y2",  "Y8",
  "Y6",  "LOG_Y12",  "Y1",  "Y7",  "Y3"),
listclass=c(""),grupos=4,sinicio=1234,repe=10,
size=c(15),decay=c(0.01),repeticiones=5,itera=100)


medias1$modelo="avnnet"

medias2<-cruzadalin(data=datos,
vardep="LN_TOTAL",listconti=c("SQRT_Y13",  "LOG_Y5",  "x45",
  "LOG_Y1",  "LOG_Y6",  "Y13",  "Y5",  "Y4",  "LOG_Y14",  "Y2",  "Y8",
  "Y6",  "LOG_Y12",  "Y1",  "Y7",  "Y3"),


listclass=c("A9","A13","A8","A6"),grupos=4,sinicio=1234,repe=10)

medias2$modelo="lineal"


medias3<-cruzadaarbol(data=data,
vardep="cstrength",listconti=c("age","water","cement","blast"),
listclass=c(""),
 grupos=4,sinicio=1234,repe=5,cp=0,minbucket=5)


medias3$modelo="arbol"
```

```
medias4<-cruzadarf(data=datos,
 vardep="LN_TOTAL",listconti=c("SQRT_Y13",  "LOG_Y5",  "x45",
  "LOG_Y1",  "LOG_Y6",  "Y13",  "Y5",  "Y4",  "LOG_Y14",  "Y2",  "Y8",
  "Y6",  "LOG_Y12",  "Y1",  "Y7",  "Y3"),
listclass=c("A9","A13","A8","A6"),
  grupos=4,sinicio=1234,repe=10,
  nodesize=10,replace=TRUE,ntree=200,mtry=4)

medias4$modelo="RF"
medias5<-cruzadarf(data=datos,
              vardep="LN_TOTAL",listconti=c("SQRT_Y13",  "LOG_Y5",  "x45",
  "LOG_Y1",  "LOG_Y6",  "Y13",  "Y5",  "Y4",  "LOG_Y14",  "Y2",  "Y8",
  "Y6",  "LOG_Y12",  "Y1",  "Y7",  "Y3"),
              listclass=c("A9","A13","A8","A6"),
              grupos=4,sinicio=1234,repe=10,
              nodesize=10,replace=TRUE,ntree=600,mtry=4)

medias5$modelo="RF2"
medias6<-cruzadagbm(data=datos,
 vardep="LN_TOTAL",listconti=c("SQRT_Y13",  "LOG_Y5",  "x45",
  "LOG_Y1",  "LOG_Y6",  "Y13",  "Y5",  "Y4",  "LOG_Y14",  "Y2",  "Y8",
  "Y6",  "LOG_Y12",  "Y1",  "Y7",  "Y3"),
listclass=c("A9","A13","A8","A6"),
  grupos=4,sinicio=1234,repe=10,
n.minobsinnode=20,shrinkage=0.03,n.trees=5000,interaction.depth=2)

medias6$modelo="gbm"



union1<-rbind(medias1,medias2,medias4,medias5,medias6)
union2<-rbind(medias1,medias2,medias4,medias5,medias6)
par(cex.axis=0.5)
```

```
boxplot(data=union1,error~modelo,col="pink")


union1<-rbind(medias1,medias3,medias4,medias5,medias6)


par(cex.axis=0.5)
boxplot(data=union1,error~modelo,col="pink")


#Validación:

## Partición de datos
set.seed(2611)
partitionIndex <- createDataPartition(datos$LN_TOTAL, p=0.8, list=FALSE)
data_train <- datos[partitionIndex,]
data_test <- datos[-partitionIndex,]

control<-trainControl(method = "none",savePredictions = "all")

nnetgrid <-  expand.grid(size=c(15),decay=c(0.01))

rednnet<-
train(LN_TOTAL~SQRT_Y13+LOG_Y5+x45+LOG_Y1+LOG_Y6+Y13+Y5+Y4+LOG_Y14
+Y2+Y8+Y6+LOG_Y12+Y1+Y7+Y3,data=data_train,

                method="nnet",linout                                                    =
TRUE,maxit=100,trControl=control,tuneGrid=nnetgrid)

t2<-as.data.frame(cbind(predict(rednnet,data_test),data_test))


write.csv(t2,"resultadosrednnetTRANSFLNTOTAL.csv",row.names = FALSE, na="")


gbmgrid<-expand.grid(shrinkage=c(0.03),
                     n.minobsinnode=c(20),
                     n.trees=c(5000),
                     interaction.depth=c(2))
```

```
gbm<-
train(LN_TOTAL~SQRT_Y13+LOG_Y5+x45+LOG_Y1+LOG_Y6+Y13+Y5+Y4+LOG_Y14
+Y2+Y8+Y6+LOG_Y12+Y1+Y7+Y3+A9+A13+A8+A6,data=data_train,
                method="gbm",trControl=control,tuneGrid=gbmgrid,
                distribution="gaussian", bag.fraction=1,verbose=FALSE)
t3<-as.data.frame(cbind(predict(gbm,data_test),data_test))


write.csv(t3,"resultadosgbmTRANSFLNTOTAL.csv
```