**FACULTAD DE ESTUDIOS ESTADÍSTICOS**

**MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS**

**Curso 2021/2022**

**Trabajo de Fin de Máster**

***TITULO: Predicción del gasto sanitario en centros de atención médico-asistencial de la Comunidad de Madrid***

***Alumno*: Carlos Martínez Fárez**

***Tutor*: David Lora Pablos**

Junio de 2022



**Abstract**

Demand increasing in the medical-assistance centers of the main cities has generated a loss of quality on the services provided and a difficulty in supplying these centers with adequate resources to confront the growing needs of the population.

Nowadays, wides volumes of data are being generated to collect the demographic characteristics of patients and medical-care characteristics of health professionals of these centers. Therefore, in this data mining project, several machine learning techniques focused on supervised analysis will be applied, regarding to analyze multiple factors that may affect health spending, so a predictive model with the least possible bias and variance can be developed.

**Keywords:**

medical-assistance centers, health spending, data mining, machine learning*,* supervised analysis, bias, variance, cross-validation.

**Resumen**

El aumento de la demanda en los centros médicos-asistenciales de las principales ciudades, ha generado una pérdida en la calidad de los servicios suministrados y una dificultad para abastecer a estos centros con los recursos adecuados para hacer frente las necesidades crecientes de la población.

En la actualidad, se generan altos volúmenes de datos que recogen las características demográficas de los pacientes y médico-asistenciales de los profesionales sanitarios y de estos centros. Por lo tanto, en este proyecto de minería de datos se aplicarán diversas técnicas de *machine learning* enfocadas en el análisis supervisado, con el objetivo de analizar los múltiples factores que afectan al gasto sanitario, y en función de ellos, elaborar un modelo predictivo con el meno sesgo y varianza posible.

**Palabras clave:**

centros médico-asistenciales, gasto sanitario, minería de datos, *machine learning,* aprendizaje supervisado, sesgo, validación cruzada.

**Índice**

[1.- Introducción 7](#_Toc103776868)

[1.1.- Motivación 7](#_Toc103776869)

[1.2.- Centros de atención primaria 8](#_Toc103776870)

[1.3.- Inteligencia de negocios 9](#_Toc103776871)

[1.4.- Inteligencia artificial 10](#_Toc103776872)

[1.4.1.- Machine Learning 10](#_Toc103776873)

[1.5.- Minería de datos 11](#_Toc103776874)

[1.5.1.- Metodologías para la minería de datos 12](#_Toc103776875)

[2.- Hipótesis y objetivos 14](#_Toc103776876)

[3.- Material y métodos 15](#_Toc103776877)

[3.1-Muestra 15](#_Toc103776878)

[3.1.1.- Descripción de las variables 16](#_Toc103776879)

[3.2-Técnicas empleadas 18](#_Toc103776880)

[3.2.1.- Regresión lineal 18](#_Toc103776881)

[3.2.2- Redes Neuronales 20](#_Toc103776882)

[3.2.3 Árboles de regresión 22](#_Toc103776883)

[3.2.4.- Técnicas de ensamblado basadas en árboles 23](#_Toc103776884)

[3.2.5.- Support Vector Machine 26](#_Toc103776885)

[3.2.6.- Validación de modelos 28](#_Toc103776886)

[4.- Resultados 30](#_Toc103776887)

[4.1.- Análisis exploratorio 30](#_Toc103776888)

[4.1.1.- Importación y asignación de los roles de las variables. 31](#_Toc103776889)

[4.1.2.- Análisis descriptivo y gráfico del conjunto de datos. 31](#_Toc103776890)

[4.2.- Modificación 35](#_Toc103776891)

[4.2.1.- Corrección de errores 35](#_Toc103776892)

[4.2.2.- Tratamiento de valores atípicos 36](#_Toc103776893)

[4.2.3.- Imputación de valores ausentes 38](#_Toc103776894)

[4.2.4.- Transformación de variables 38](#_Toc103776895)

[4.3 Selección de variables 39](#_Toc103776896)

[4.4.- Modelización 41](#_Toc103776897)

[4.4.1.- Regresión lineal 42](#_Toc103776898)

[4.4.2.- Árboles de regresión 44](#_Toc103776899)

[4.4.3.- Redes neuronales 48](#_Toc103776900)

[4.4.4.- Bagging 52](#_Toc103776901)

[4.4.5.- Random Forest 54](#_Toc103776902)

[4.4.6.- Gradient Boosting 56](#_Toc103776903)

[4.4.7.- *Extreme Gradient Boosting* 60](#_Toc103776904)

[4.4.8.- Support Vector Machine 62](#_Toc103776905)

[4.5.- Modelos de ensamblado 66](#_Toc103776906)

[4.6.- Comparación de resultados 67](#_Toc103776907)

[5.- Conclusiones 72](#_Toc103776908)

[Anexos 73](#_Toc103776909)

[Bibliografía 74](#_Toc103776910)

**Índice de figuras**

[Figura 1.1: Inteligencia de negocios. Fuente: Juan Pablo Sánchez (2022) 13](#_Toc103859796)

[Figura 1.2: Machine Learning. Fuente: Javier Portela (2022) 15](#_Toc103859797)

[Figura 1.3: Metodología SEMMA. Fuente: (Calviño Martínez, 2021) 16](#_Toc103859798)

[Figura 3.1: Esquema relacional base de datos de centro Médico-Asistenciales de la Comunidad de Madrid 19](#_Toc103859799)

[Figura 3.2: Modelo de regresión lineal univariante. Fuente: Javier Álvarez (2021) 23](#_Toc103859800)

[Figura 3.3: Construcción de una red neuronal. Fuente: Javier Portela (2022) 25](#_Toc103859801)

[Figura 3.4: Esquema de árbol de decisión. Fuente: Javier Álvarez (2021) 26](#_Toc103859802)

[Figura 3.5: Bagging. Fuente: Javier Castro (2022) 28](#_Toc103859803)

[Figura 3.6: SVM maximal margin. Fuente: Juan del Rosal (2016) 31](#_Toc103859804)

[Figura 3.7: Kernel. Fuente: Daniel Gómez (2022) 32](#_Toc103859805)

[Figura 3.8: Validación cruzada. Fuente: Javier Portela (2022) 33](#_Toc103859806)

[Figura 4.1: Importación y asignación de roles 35](#_Toc103859807)

[Figura 4.2: Exploración de variables 36](#_Toc103859808)

[Figura 4.3: Exploración de variables intervalo 37](#_Toc103859809)

[Figura 4.4: Exploración de variables clase 37](#_Toc103859810)

[Figura 4.5: Dependencia lineal 38](#_Toc103859811)

[Figura 4.6: Nodo multigráfico 39](#_Toc103859812)

[Figura 4.7: Nodo reemplazo de variables de clase 40](#_Toc103859813)

[Figura 4.8: Nodo reemplazo de variables intervalo 40](#_Toc103859814)

[Figura 4.9: Detección de valores atípicos 41](#_Toc103859815)

[Figura 4.10: Valores atípicos detectados 41](#_Toc103859816)

[Figura 4.11: Resultado depuración 42](#_Toc103859817)

[Figura 4.12: Valor de las variables input respecto al output modificado 43](#_Toc103859818)

[Figura 4.13: Validación cruzada y boxplot. Regresión lineal 46](#_Toc103859819)

[Figura 4.14: Hipótesis paramétricas. Regresión lineal 47](#_Toc103859820)

[Figura 4.15: Minbucket óptimo. Árboles de regresión 50](#_Toc103859821)

[Figura 4.16: Árboles de regresión. Validación cruzada repetida 50](#_Toc103859822)

[Figura 4.17: Representación simplificada del árbol de regresión 51](#_Toc103859823)

[Figura 4.18: Importancia de las variables. Árbol de regresión 51](#_Toc103859824)

[Figura 4.19: Distribución las principales variables input 52](#_Toc103859825)

[Figura 4.20: Optimización de parámetros por early stopping. Redes neuronales 54](#_Toc103859826)

[Figura 4.21: Redes neuronales. Validación cruzada repetida 55](#_Toc103859827)

[Figura 4.22: Número de árboles óptimo (nodesize =85). Bagging 57](#_Toc103859828)

[Figura 4.23: Optimización del tamaño del nodo final. Bagging 57](#_Toc103859829)

[Figura 4.24: Bagging variando el sampsize. Validación cruzada repetida 58](#_Toc103859830)

[Figura 4.25: Remuestreo de variables. Random Forest 59](#_Toc103859831)

[Figura 4.26: Random Forest. Validación cruzada repetida 59](#_Toc103859832)

[Figura 4.27: Importancia de las variables. Random Forest 60](#_Toc103859833)

[Figura 4.28: Parámetros óptimos. Gradient Boosting 61](#_Toc103859834)

[Figura 4.29: Número de iteraciones (early-stopping). Gradient Boosting 62](#_Toc103859835)

[Figura 4.30: Gradient Boosting. Validación cruzada repetida 63](#_Toc103859836)

[Figura 4.31: Importancia de las variables. Gradient Boosting 63](#_Toc103859837)

[Figura 4.32: Parámetros óptimos. XGboost 64](#_Toc103859838)

[Figura 4.33: Número de iteraciones (early-stopping). XGboost 65](#_Toc103859839)

[Figura 4.34: XGboost. Validación cruzada repetida 65](#_Toc103859840)

[Figura 4.35: Importancia de las variables. XGboost 66](#_Toc103859841)

[Figura 4.36: Nivel de penalización óptimo. SVM lineal 67](#_Toc103859842)

[Figura 4.37: Nivel de penalización óptimo. SVM lineal 68](#_Toc103859843)

[Figura 4.38: Parametización óptima. SVM RBF 69](#_Toc103859844)

[Figura 4.39: Comparación de modelos SVM. Validación cruzada repetida 69](#_Toc103859845)

[Figura 4.40: Comparación modelos de ensamblado 71](#_Toc103859846)

[Figura 4.41: Comparación modelos finales. Validación cruzada repetida 72](#_Toc103859847)

[Figura 4.42: Comparación ampliada de los modelos finales. Validación cruzada repetida 73](#_Toc103859848)

[Figura 4.43. Coeficientes modelo de regresión lineal óptimo 74](#_Toc103859849)

[Figura 4.44: Nodos finales árbol de regresión simplificado 75](#_Toc103859850)

**Índice de tablas**

[Tabla 1.1: Funciones de la atención primaria de salud contempladas en la ley española de Cohesión y Calidad del Sistema Nacional de Salud (Ley 16/2003) 8](#_Toc103775279)

[Tabla 3.1: Descripción de las variables municipio 16](#_Toc103775280)

[Tabla 3.2: Descripción de las variables de la tabla médico 17](#_Toc103775281)

[Tabla 3.3: Descripción de las variables de la tabla paciente 18](#_Toc103775282)

[Tabla 3.4: Funciones de activación. Fuente: Javier portela (2022) 21](#_Toc103775283)

[Tabla 4.1: Selección de variables SAS Enterprise Miner 40](#_Toc103775284)

[Tabla 4.2: Selección de variables R-Studio 40](#_Toc103775285)

[Tabla 4.3: Selección de variables 41](#_Toc103775286)

[Tabla 4.4: Posibles combinaciones de modelos de ensamblado 66](#_Toc103775287)

[Tabla 4.5: Características de los mejores modelos obtenidos 68](#_Toc103775288)

# 1.- Introducción

## 1.1.- Motivación

## 

El elevado número de consultas de atención primaria (AP) realizadas en los centros médicos asistenciales de las principales ciudades, junto con la incapacidad de estos centros de generar una oferta efectiva que se adecúe a este exceso de demanda, genera una importante pérdida de calidad asistencial y un incremento del coste sanitario público.

El uso excesivo de las consultas de AP puede ser definido como aquellas actividades que consumen gran parte del tiempo en actividades curativas en detrimento de las preventivas, aumentando la carga de trabajo de los profesionales y dañando la relación médico-paciente. Para medirlo de manera adecuada se deben tener en cuenta múltiples factores relacionados con las características específicas del paciente y del beneficio que obtienen del sistema de salud, junto con toda la información profesional del médico que lo atiende.

Según Andersen R. la necesidad de atención médica está en relación con el nivel de salud percibido y con el deseado. Esta necesidad viene condicionada por una larga lista de factores predisponentes, entre los que se pueden encontrar características demográficas, culturales, sociales, y se puede ver facilitado por circunstancias relacionadas con el status económico, los recursos comunitarios, características del médico y de las instituciones sanitarias.

Cuantificar todos los factores que pueden influir en la relación médico-paciente requiere de un volumen enorme de datos y un correcto tratamiento de los mismos. Por ello, a través de la minería de datos, es posible descubrir patrones y/o relaciones ocultas y complejas de un conjunto de datos. Esta disciplina resulta de gran utilidad en el tema a tratar debido a que ayuda a estos centros médicos a descubrir pautas que puedan mejorar su calidad asistencial y ayudan a estimar los recursos necesarios que garanticen el correcto abastecimiento de un determinado municipio.

Gran parte de la literatura relacionada con este tema busca predecir este gasto sanitario utilizando únicamente información del paciente. En este sentido, el enfoque más utilizado son los Adjusted Clinical Groups (ACG) que son un sistema de agrupación de diagnósticos que clasifica a las personas según las enfermedades que presentan durante un período de tiempo. Fue desarrollado por Starfield et al con el objetivo de medir el grado de enfermedad de la población basándose en un índice de comorbilidad. Este sistema constituye uno de los principales métodos de ajuste del riesgo para concertar la financiación o la eficiencia en el uso de los servicios por parte de la administración.

Sin embargo, pese a que se dispone de algunas evidencias que destacan su comportamiento teórico, en la práctica, su capacidad para tratar de responder a las múltiples necesidades de gestión resulta incierta. Por esta razón, en el presente trabajo se ha decido desglosar el sistema de agrupación ACG en los múltiples factores relacionados con el paciente que lo componen, y se ha decido incorporar información relacionada con el médico y el entorno en el que se sitúa un determinado centro de atención primaria, con el objetivo de aportar un nuevo enfoque en la predicción del gasto sanitario.

En este capítulo se han discutido la importancia de este tipo de servicios junto con algunos aspectos fundamentales acerca de las técnicas de la minería de datos que serán tratadas y aplicadas en el presente trabajo.

## 1.2.- Centros de atención primaria

Los centros de atención primaria son aquellos lugares que ofrecen asistencia sanitaria y social, servicios de promoción de la salud, de atención preventiva curativa y rehabilitadora.

De acuerdo con la Organización Mundial de la Salud (OMS) en su informe sobre la salud en el mundo (2008), se incide en que la atención primaria resulta más necesaria que nunca puesto que supone un fuerte modo de afrontar las inequidades e ineficiencias de la atención de salud. Este informe precisa que estos centros requieren de una serie de reformas debido a los importantes cambios socio-demográficos y epidemiológicos en las poblaciones actuales, donde la tendencia a la dependencia y utilización creciente se estos servicios (medicalización de la vida cotidiana) y el crecimiento del gasto sanitario que dificulta el abastecimiento de los recursos necesarios son importantes razones para profundizar en este tipo de reformas.

En la siguiente tabla, de acuerdo con la ley 16/2003 de Cohesión y Calidad del Sistema Nacional de Salud, se contemplan las funciones atribuidas a estos centros.

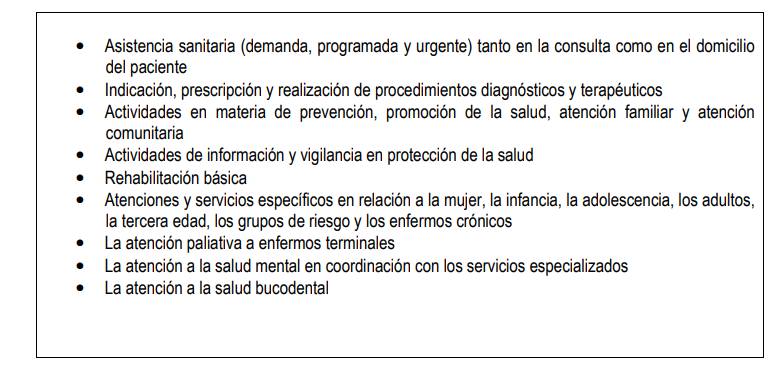


Tabla 1.1: Funciones de la atención primaria de salud contempladas en la ley española de Cohesión y Calidad del Sistema Nacional de Salud (Ley 16/2003)

Estas funciones son desarrolladas por un conjunto de profesionales sanitarios y no sanitarios que tienen su ámbito de actuación en su ubicación preferente en el centro de salud y comparten metodología de trabajo para mejorar el nivel de salud de la población asignada.

Con el objetivo de contribuir a las reformas necesarias requeridas por la OMS, en este trabajo utilizando las técnicas de minería de datos e inteligencia en los negocios, se tratará de estimar los costes de estos centros, en términos de gasto en farmacia, para facilitar la asignación eficiente de financiación y contribuir al desarrollo de las funciones esenciales en los terrenos asistencial, preventivo y de promoción de la salud.

## 1.3.- Inteligencia de negocios

La inteligencia de negocios es un conjunto de herramientas destinadas a la obtención, procesamiento y uso de información relevante que permiten a diferentes tipos de organizaciones, tomar decisiones más convenientes basadas en información precisa y adecuada.

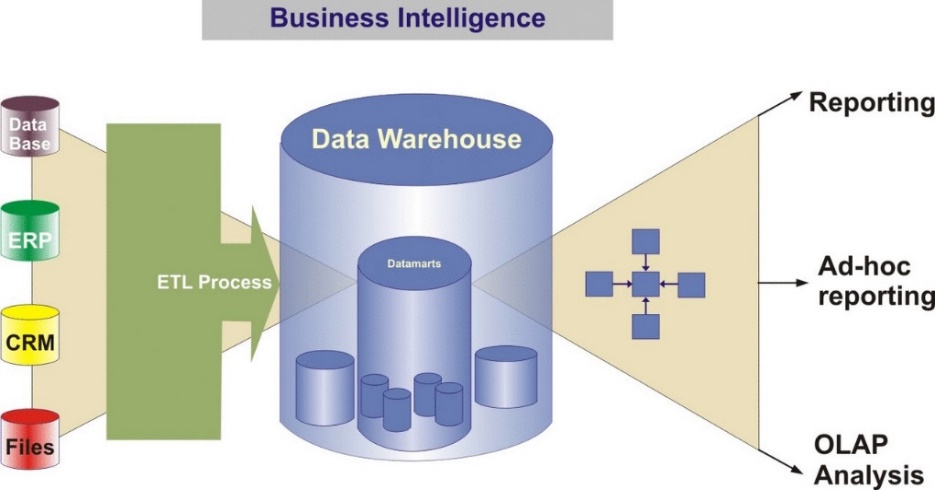


Figura 1.1: Inteligencia de negocios. Fuente: Juan Pablo Sánchez (2022)

Dentro de la inteligencia de negocios es importante señalar dos elementos fundamentales:

* **Inteligencia Competitiva:** Según Shrivastava y Grant (1985) la inteligencia competitiva es el sistema de aprendizaje sobre las capacidades y comportamientos de los competidores actuales y potenciales de la empresa así como de las circunstancias de la industria con objeto de ayudar a los responsables de la toma de decisiones estratégicas.
* **Vigilancia Tecnológica:** Proceso organizado, selectivo y permanente capaz de captar información del exterior y de la propia organización sobre ciencia y tecnología, seleccionarla, analizarla, difundirla y comunicarla, para convertirla en conocimiento para tomar decisiones con menor riesgo y poder anticiparse a los cambios (Norma UNE 166006:2011),

El adecuado uso de estas herramientas, para el caso concreto que nos ocupa en este proyecto, nos han permitido manejar de manera eficaz y eficiente grandes cantidades de datos proporcionadas por los centro médico asistenciales, de manera que permitan estimar de manera anticipada los gastos necesarios para abastecer sus necesidades.

## 1.4.- Inteligencia artificial

La inteligencia artificial (IA) es una rama de la informática formada por un conjunto de algoritmos lógicos que ayudan en la toma de decisiones para lograr un objetivo específico a partir de una serie de normas generales.

Gracias al desarrollo tecnológico de la computación, esta rama se ha ido incorporando a múltiples campos con una alta aplicabilidad. En el ámbito que nos ocupa, la aplicación de la IA en la medicina ha permitido mejorar la atención al paciente al acelerar los procesos y lograr una mayor precisión diagnóstica.

En este sentido, existen números proyectos dedicados a la exploración de la IA en todas las facetas sanitarias: asistencial (prevención de enfermedades, diagnóstico, tratamiento y seguimiento del paciente), investigación (entrenamiento de habilidades cuya ejecución tiene importante trascendencia social) y gestión de los recursos.

Concretamente, el ámbito de la gestión de recursos materiales y humanos es uno de los mayores beneficiados de la IA. En este trabajo, se utilizarán algunos de los modelos más desarrollados en la actualidad englobados dentro de la rama de *Machine Learining,* para examinar grandes cantidades de datos procedentes de registros históricos con el objetivo de prever los recursos necesarios de una situación concreta, de manera que se optimice el rendimiento, se impulse la productividad y se realice una mejor asignación de los recursos disponibles.

### 1.4.1.- Machine Learning

Según O’Reilly (2019) *Machine Learning* es la ciencia y el arte de programar ordenadores para que puedan aprender de los datos. Esta rama de la Inteligencia Artificial se fundamenta en la elaboración de técnicas que permitan a las maquinas aprender de manera automática a partir de una serie de algoritmos capaces de generalizar comportamientos y reconocer patrones a partir de cierta información.

Dependiendo del tipo datos a abordar, se pueden clasificar las herramientas de esta disciplina en:

* Aprendizaje supervisado: el algoritmo se programa para extraer patrones de comportamiento a partir de los datos recibidos, y en base dicha información aprendida, realizar la evaluación de nuevos *inputs.* De esta forma el algoritmo aprende de sus errores y se aproxima lo máximo posible a la realidad. Algunos de los algoritmos más conocidos por su efectividad son:
* Regresión lineal
* Regresión logística
* Redes neuronales
* Árboles de decisión, Random forest y gradient boosting
* Support Vector Machines (SVM)
* Aprendizaje no supervisado: el algoritmo realiza el procesamiento únicamente en base a las entradas. Es decir, tiene la capacidad de configurarse a medida que procesa los datos que va recibiendo. Este aprendizaje se destina fundamentalmente al clustering, segmentación y reducción de dimensiones y a la búsqueda de estructuras.

Algunos de los algoritmos más conocidos por su efectividad son:

* Algoritmos de clustering (k-means, jerárquicos entre otros)
* Análisis factorial.
* Análisis de correspondencias

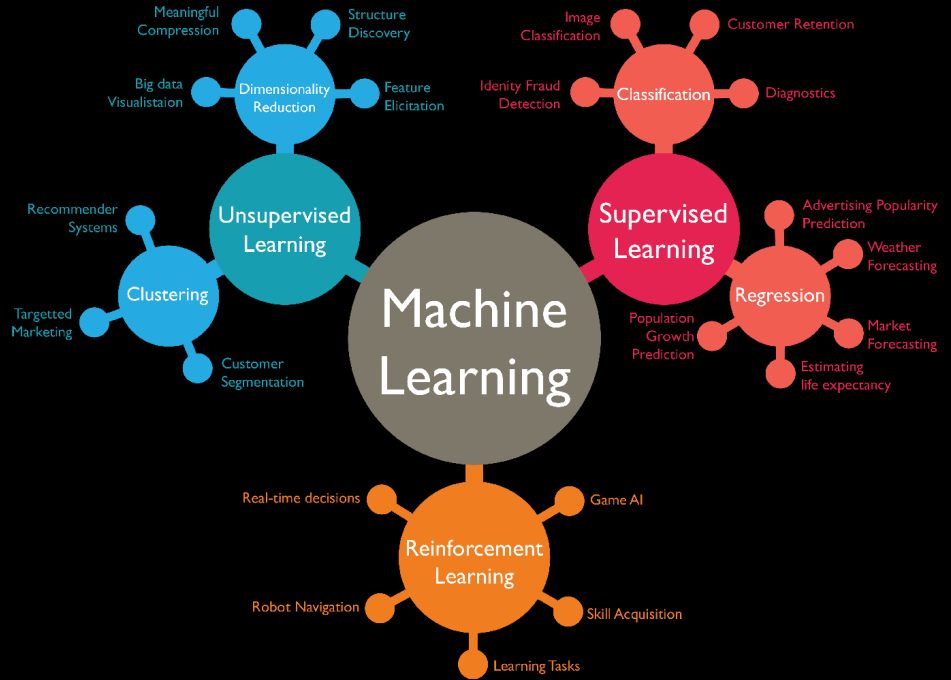


Figura 1.2: Machine Learning. Fuente: Javier Portela (2022)

## 1.5.- Minería de datos

Según IBM, “la minería de datos es una forma innovadora de obtener información comercial valiosa mediante el análisis de los datos contenidos en la base de datos de la empresa. Revela información comercial exhaustiva utilizando técnicas avanzadas de análisis y creación de modelos.”

Entre sus características, cabe destacar que la minería de datos trabaja con datos heterogéneos, abundantes y de calidad discutible. Es decir, utiliza un número elevado de observaciones y/o variables generalmente con gran cantidad de datos atípicos y/o faltantes. Este análisis se enfoca principalmente en el poder predictivo de sus modelos, más que en su capacidad de predicción.

En cuanto a las variables (Calviño Martínez, 2021) pueden ser clasificadas en dos tipos:

* Según su función:
* Identificativas: permiten únicamente identificar las observaciones.
* Input o de entrada: son las variables predictoras o independientes. Aportan información sobre la variable objetivo
* Objetivo: es la variable que se pretende predecir.
* Rechazadas: son variables eliminadas antes de la fase de modelización.
* Según su tipología:
* Continuas, cuantitativas o de intervalo: toman cualquier valor en un intervalo (que puede estar limitado o no).
* Nominales, cualitativas o categóricas: toman un número finito de valores que representan una categoría determinada.
* Binarias o dicotómicas: variables nominales que toman sólo dos valores.
* Fecha/Hora: son variables que representan una fecha y/o hora.

### 1.5.1.- Metodologías para la minería de datos

Dentro de la minería de datos existe una amplia gama de metodologías para la planificación y ejecución de un determinado proyecto. Algunas de ellas pueden ser el descubrimiento de conocimiento en bases de datos (KDD, del inglés *Knowledge Discovery in Databases*) que consiste en extraer patrones en forma de reglas o funciones para que el usuario lo analice, otras tienen un carácter más específico como CRISP-DM (del inglés *Cross-Industry Standard Process for Data Mining*) utilizado para la compresión de un determinado negocio, *Catalyst* que está ganando cada vez más popularidad al adaptarse a gran cantidad de escenarios entre otras. Sin embargo, este trabajo se ha realizado de acuerdo a la metodología SEMMA gracias al *software* SAS Enterprise Minerproporcionado por el instituto SAS.

Esta metodología se encarga de la exploración y modelado aplicado a cantidades significativas de datos almacenados que permitan el descubrimiento de patrones ocultos y/o relaciones complejas o desconocidas entre los datos a fin de establecer conclusiones útiles para el entorno de negocio.

La metodología SEMMA está compuesta por 5 fases diferenciadas:

1. *Sample* (muestrear): si el número de observaciones es muy elevado será necesario tomar una muestra lo suficientemente grande que contenga la máxima información posible y lo suficientemente pequeña para poder ser procesada.
2. *Explore* (explorar): detección y exploración los datos para detectar relaciones, anomalias y tendencias.
3. *Modify* (modificar): transformación de las variables con el objetivo de facilitar la modelización.
4. *Model* (modelizar): elaboración de modelos que ayuden a predecir el comportamiento de la variable objetivo.
5. *Asses*(evaluar): comprobación y comparación de la calidad de los modelos realizados

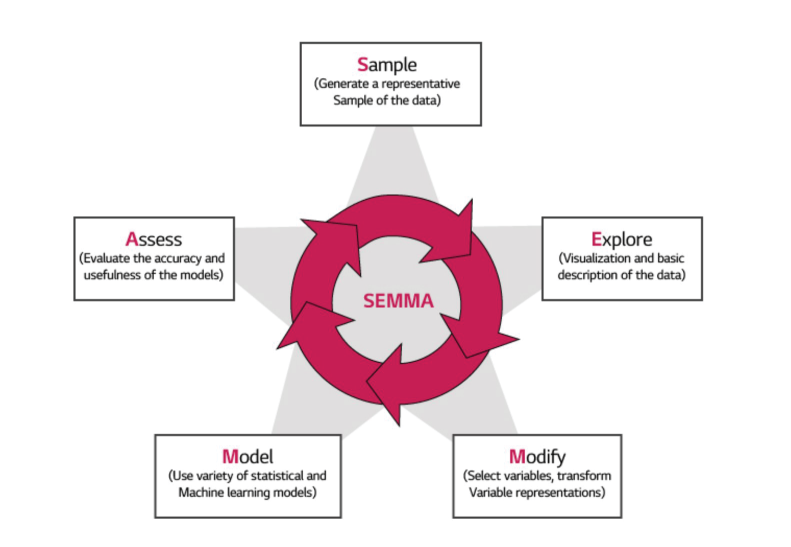


Figura 1.3: Metodología SEMMA. Fuente: (Calviño Martínez, 2021)

Es necesario destacar una peculiaridad de este tipo de metodología, ya que estas fases no se realizan siempre siguiente el mismo orden. Como se puede ver en la figura 1.3, el procedimiento es de carácter cíclico donde se encuentran continuamente nuevas relaciones entre las variables que llevan a volver ajustar los modelos realizados.

# 2.- Hipótesis y objetivos

El gasto en productos farmacéuticos puede ser identificado a través de las características demográficas de los pacientes y médico-asistenciales de los profesionales sanitarios y de los centros de la Comunidad de Madrid.

Con el presente trabajo se espera que haciendo uso de las técnicas de la minería de datos aplicadas a los centros médico-asistenciales de la Comunidad de Madrid, se obtenga un modelo predictivo los más preciso y robusto posible para predecir el gasto en farmacia, con el objetivo de establecer un plan de gasto en función del perfil médico-paciente, que permita adecuar los recursos de las administraciones públicas al exceso de demanda producido en estos centros. Esto devendría en una asignación más eficiente de los recursos del centro que permitiría reducir la sobrecarga de trabajo de los profesionales del sector y a mejorar la calidad asistencial proporcionada al paciente.

A través de técnicas que se comentarán posteriormente, se plantean una serie de objetivos secundarios relacionados con la identificación y descripción de los factores que influyen en mayor medida en el gasto sanitario generado por cada paciente, de manera que se asigne a cada departamento médico y municipio los recursos necesarios adaptados a sus características, con el objetivo de generar un mejor servicio sanitario.

Esta mejora en la calidad del servicio proporcionado, a través de una correcta estimación de los recursos necesarios para garantizar el abastecimiento de material sanitario del centro médico, se traduciría nuevamente en una reducción de la presión asistencial por parte el paciente y por ende en una disminución de los costes sanitarios públicos.

# 3.- Material y métodos

En este apartado se ha descrito el conjunto de datos sobre el que se realizará el análisis y las diferentes técnicas y modelos que se han utilizado a lo largo del proyecto. Se analizará tanto el funcionamiento como los motivos que justifican su aplicación en nuestro tema de estudio.

## 3.1-Muestra

En primer lugarse va a explicar la forma en que está estructurada la base de datos, así como el proceso de extracción de la misma, con el objetivo anteriormente mencionado de realizar un análisis predictivo del gasto sanitario (concretamente el gasto en farmacias) en centros de atención médico-asistenciales de la Comunidad de Madrid.

Los datos utilizados para la realización de este trabajo han sido extraídos a partir de un estudio sobre la utilización y variabilidad de los recursos en atención primaria de diseño transversal en una muestra representativa de la población de la Unidad Técnica-11 del Área de Salud Pública

En la siguiente figura se muestra un esquema relacional acerca de cómo se encuentra estructurada la base de datos, donde cada caja representada por un color recoge la entidad de cada componente a estudiar del centro médico-asistencial. Dentro de cada entidad se recogen un total de 18 variables agrupadas en función de las características del municipio en el que se encuentra el centro, de las características del médico y finalmente información relacionada con el paciente.

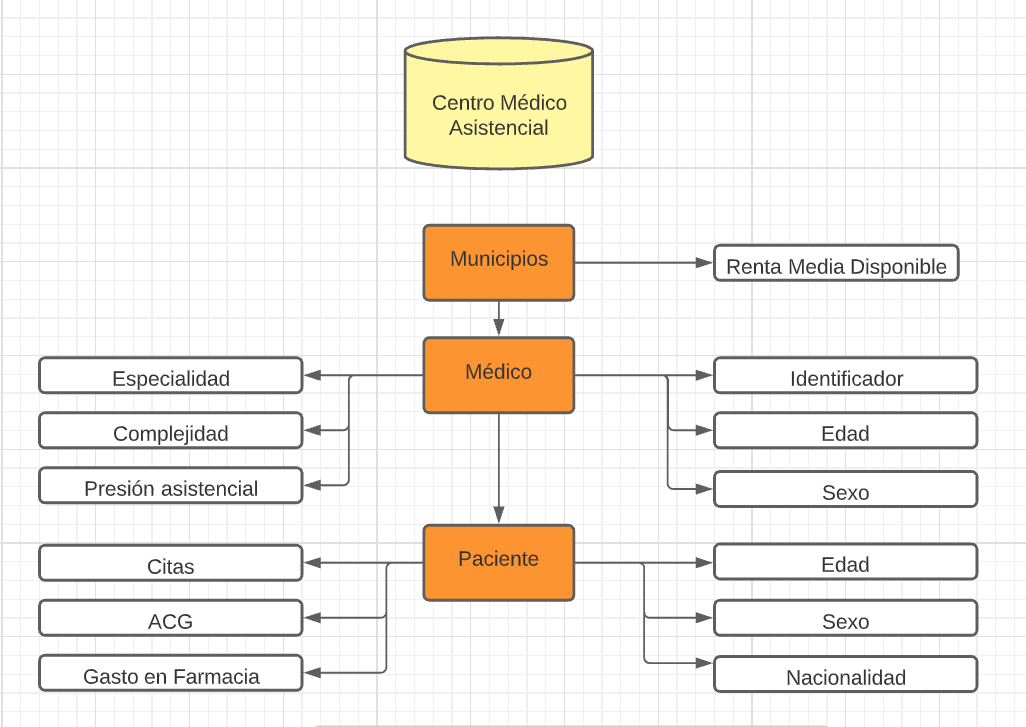


Figura 3.1: Esquema relacional base de datos de centro Médico-Asistenciales de la Comunidad de Madrid

### 3.1.1.- Descripción de las variables

Siguiendo la estructura representada en la figura anterior, se ha realizado un análisis descriptivo de las variables pertenecientes a las diferentes entidades. De esta forma, en primer lugar se ha obtenido la variable RMD (renta media disponible) perteneciente al conjunto de variables de *Municipio.* Para este conjunto, se ha recogido la renta de los 12 municipios que atiende la Unidad Técnica-11 del Área de Salud pública de la Comunidad de Madrid durante los años 1997-1999:

* Aranjuez
* Chinchon
* Ciempozuelos
* Colmenar de Oreja
* Arganzuela
* Carabanchel
* Usera
* Villaverde
* Titulcia
* Valdelaguna
* Valdemoro
* Villaconejos

Este conjunto se ha tratado como una variable numérica de tipo intervalo, de manera que únicamente se han extraido conclusiones acerca de si la renta media en el municipio afecta al gasto farmacéutico (variable objetivo) de cada paciente.

|  |  |
| --- | --- |
| Variables | Descripción |
| RMD | Representa el valor de la renta media disponible para los 12 municipios mencionados. Este indicador mide la capacidad económica de las familias o personas |

Tabla 3.1: Descripción de las variables municipio

En segundo lugar, se agrupan las variables pertenecientes a la tabla de *Médico*. Este conjunto representa tanto información personal como profesional de los médicos pertenecientes a los centros médico-asistenciales.

|  |  |
| --- | --- |
| Variables | Descripción |
| id | Código identificativo correspondiente al médico |
| edad\_med | Variable numérica discreta que representa la edad del médico |
| sexo\_med | Variable binaria que indica el sexo del médico   * Mujer (0) * Hombre (1) |
| especialidad | Variable binaria que indica si el médico dispone de especialidad o no   * Plaza en propiedad (0) * Sin plaza en propiedad (1) |
| i\_complejidad | Variable numérica que representa la complejidad del tratamiento |
| p\_asistencial | Representa en forma de porcentaje la carga de trabajo que tiene el médico |

Tabla 3.2: Descripción de las variables de la tabla médico

Finalmente, el último conjunto de variables se corresponde con la entidad de *Paciente.* Dentro de cada municipio se encuentran un conjunto de médicos que se encargan de atender a todos los pacientes que acuden al centro médico. Para ello, en la siguiente tabla se recoge por un lado información personal del paciente, y por otra parte, se recoge el número de visitas y el gasto farmacéutico desembolsado por cada uno de ellos.

|  |  |
| --- | --- |
| Variables | Descripción |
| TotCitas | Representa el número total de citas en todas las posibles consultas que ha realizado cada paciente |
| Totmed | Recoge las citas de cada paciente en la consulta de medicina |
| TotEnf | Recoge el total de citas de cada paciente en la consulta de enfermería |
| TotLab | Recoge el número de citas de cada paciente en el laboratorio |
| acg\_code | Esta variable representa el sistema de clasificación de pacientes “*Adjusted Clinical Group*” (ACG). Este índice captura la naturaleza multidimensional de la salud individual, a través de un índice que tiene en cuenta la edad, el sexo y la multimorbilidad presente durante un periodo de tiempo. |
| EDAD\_pac | Variable numérica discreta que representa la edad del paciente |
| sexo\_pac | Variable binaria que indica el sexo del paciente   * Mujer (0) * Hombre (1) |
| extranjero | Variable binaria que indica si el paciente es extranjero o no   * No extranjero (0) * Extranjero (1) |
| pais\_naci | Variable numérica discreta que representa el país de nacimiento de cada individuo. |
| registro | Esta variable recoge los años que los pacientes han acudido al centro médico (1997-1999) |

Tabla 3.3: Descripción de las variables de la tabla paciente

Con todo esto se han obtenido un total de 60243 observaciones. Dada la gran cantidad de registros, ha sido necesario extraer una muestra lo suficientemente grande que contenga toda la información necesaria y lo suficientemente pequeña para poder ser procesada, en este sentido se ha seleccionado una muestra aleatoria del 12% de los datos.

## 3.2-Técnicas empleadas

### 3.2.1.- Regresión lineal

La regresión lineal fue un modelo descrito por primera vez por el matemático Legendré en 1805, pero fue introducido formalmente por Galton en 1886. Este modelo permite predecir el comportamiento de la variable dependiente u *output*  a partir de una (regresión lineal simple) o varias (regresión multivariante) variables independientes o *inputs.*

Este método de predicción es útil cuando se cumplen las siguientes hipótesis sobre nuestra base de datos:

* **Hipótesis paramétrica**:



* **Linealidad:**
* **Homocedasticidad de los residuos**:



* **Normalidad de los residuos**:

****

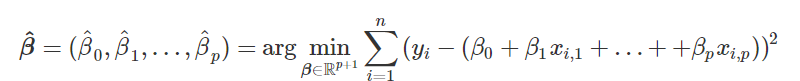
* **Residuos incorrelados:**

Esto se puede interpretar a través de la siguiente expresión:

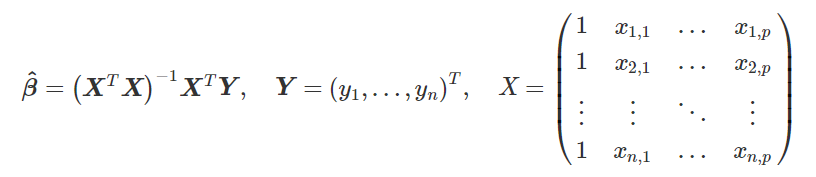


Siendo el coeficiente β0 la ordenada en el origen (valor esperado de la respuesta cuando el resto de parámetros X1 = … = Xp = 0), mientras que βj (j ≥ 1) representa el incremento medio esperado de la respuesta por cada incremento unitario de Xj.

Estos coeficientes se pueden ajustar por mínimos cuadrados de manera que minimicen la suma de residuos/errores al cuadrado



Obteniendo la solución general matricial:



La estimación de estos parámetros por mínimos cuadrados tratará de obtener el mínimo error estimado a través de la siguiente expresión

En la figura 3.2 se observa un ejemplo de regresión lineal en la que se recoge la distancia vertical medida desde cada observación hasta la línea estimada. Esto se denomina residuo y viene expresado en la anterior expresión con el término εj.

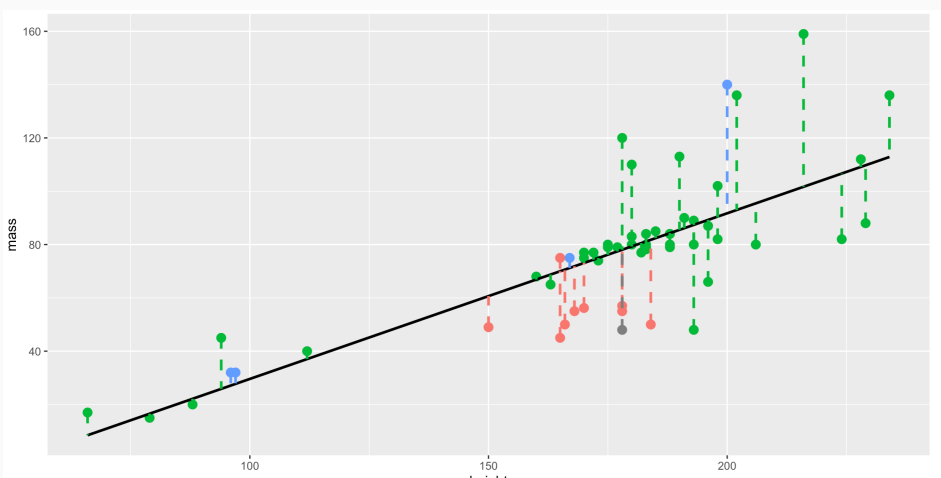


Figura 3.2: Modelo de regresión lineal univariante. Fuente: Javier Álvarez (2021)

El modelo de regresión lineal es uno de los más utilizados en la actualidad debido a su utilidad para predecir el comportamiento de variables continuas. Este tipo de variables toman valores a lo largo de un intervalo continuo de valores, que a diferencia de las variables discretas, nunca puede ser medido con exactitud (error de estimación o residuos comentados en la Figura 3.2). Por lo tanto, debido a que nuestra variable objetivo, gasto en farmacia, es de tipo continua este modelo será de gran utilidad para el análisis.

Por último, es necesario destacar una utilidad adicional del modelo de regresión lineal, debido a que no sólo tiene carácter predictivo, sino que permite una alta interpretabilidad de los coeficientes estimados, así como un proceso de selección de las variables más relevantes del conjunto. Entre sus funciones más utilizadas cabe señalar los diferentes métodos de selección de variables automáticas:

* Método Forward (selección hacia delante)
* Se selecciona la mejor variable del modelo para a continuación seleccionar otra variable que en combinación a la anterior ya introducida, proporcione el mejor ajuste.
* El proceso de introducción de variables se repite hasta que no queda ninguna fuera del modelo cuyo p-valor sea inferior al nivel de entrada.
* Es el modelo que requiere menos recursos computacionales
* Método Backward (eliminación hacia atrás)
* Se comienza con el modelo completo y en cada fase se van eliminando una variable.
* Las variables eliminadas serán aquellas que tengan el estadístico de t (en valor absoluto) más pequeño entre las variables incluidas aún en el modelo.
* Método Stepwise (selección paso a paso)
* Este método resulta de la combinación de los dos anteriores
* Comienza con un modelo de regresión simple y en cada paso se puede añadir una variable, y a su vez, se coteja si alguna de las variables ya incluidas en el modelo debe ser eliminada. Esta selección se realiza de acuerdo a la función de F-Snedecor (F) cuando F-in ≤ F-out.
* Este proceso se repite hasta que ninguna de las variables, que no han entrado aún, tienen suficiente relevancia como para ser introducidas en el modelo.

### 3.2.2- Redes Neuronales

El modelo de red neuronal artificial está inspirado en el funcionamiento de las células del sistema nervioso cuya función principal es recibir, procesar y transmitir información a partir de reacciones químicas y eléctricas. En el caso del modelo artificial, se imita este funcionamiento a través de la construcción de una relación entre variables *input* (nodos *input*), y las variables de salida (nodos output), introduciendo en un paso intermedio una serie de variables artificiales (nodos de la capa oculta).

En la figura 3.3 se puede ver el desarrollo de una red neuronal.

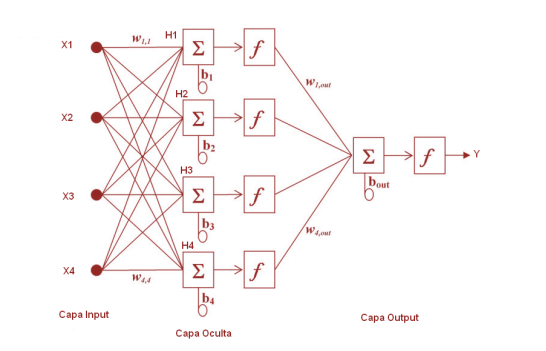
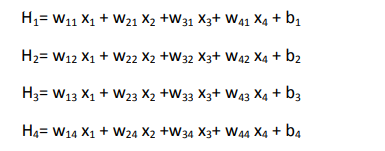


Figura 3.3: Construcción de una red neuronal. Fuente: Javier Portela (2022)

El funcionamiento de una red neuronal artificial consiste en que dado un conjunto de variables pertenecientes a una capa input (X1 X2 … Xn) que se encuentran conectadas en una capa oculta (H1,H2 … Hn) mediante un determinada función de combinación representada por Σ, donde los pesos wij hacen referencia al papel de parámetros a estimar. Se obtienen los valores de los siguientes nodos ocultos.

En este trabajo se utilizará una de las funciones de combinación más habituales, la lineal para minimizar el error estimado medio.



Una vez aplicada la función de combinación será necesario aplicar a cada nodo de la capa oculta una función de activación representada en la Figura 3.4. No obstante, estas funciones de activación son especialmente útiles cuando la variable objetivo es de tipo categórica, por lo que para nuestro caso de variable objetivo de tipo continua se ha considerado no necesario aplicar ninguna función de activación.

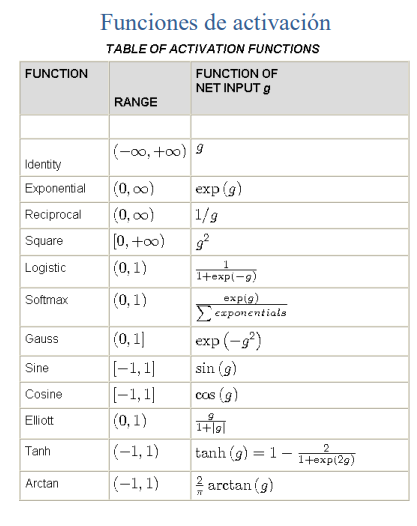


Tabla 3.4: Funciones de activación. Fuente: Javier portela (2022)

Este modelo es de gran utilidad para el presente trabajo debido a que las relaciones entre las variables *input* con la respuesta puede ser desconocida o no lineal, y es más apropiado un modelo con no linealidad implícita (red neuronal) que uno puramente lineal que se ajuste peor a nuestros datos. Es decir, la red es suficientemente flexible para ajustarse a esta relación.

### 3.2.3 Árboles de regresión

Los árboles de decisión son métodos de estimación no paramétricos propuestos por Breiman et al en 1984, que tratan de hallar los puntos de corte entre las variables independientes con el objetivo elaborar diferentes grupos de comportamiento homogéneo respecto a la variable respuesta. Entre sus utilidades es importante destacar por un lado su capacidad de análisis cluster, entendiendo esta acción como el agrupamiento de objetos que presentan ciertas similitudes para proporcionar información del conjunto agregado de datos, y por otro lado es importante destacar este modelo como un método predictivo del comportamiento de la variable objetivo.

Los árboles permiten tratar todas las variables de nuestro conjunto de datos ya que presentan gran tolerancia al ruido introducido por las variables no relevantes. Además es especialmente útil para modelar conjuntos de datos heterogéneos (variables continuas y discretas). Para el caso concreto que nos ocupa en este trabajo, el análisis se fundamentará acerca del árbol de regresión debido a las características de nuestra variable objetivo.

Los componentes que conforman los árboles de decisión son los siguientes:

* Nodos: segmentos que contienen subconjuntos de la muestra
* Nodo raíz: segmento original que contiene la totalidad de datos y sobre el que se realizan las subdivisiones
* Nodo padre/hijo: nodos predecesor/sucesor de otro nodo
* Hojas: nodo sin hijos que finalizan el camino de una rama
* Rama: ruta que toman los diferentes nodos a partir del nodo raíz y que acaba en una hoja

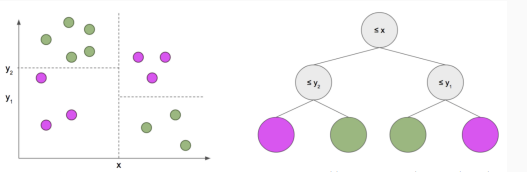
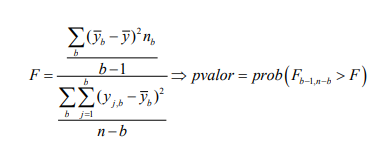


Figura 3.4: Esquema de árbol de decisión. Fuente: Javier Álvarez (2021)

Atendiendo a la figura 3.4 se puede observar un ejemplo de árbol que utiliza dos variables explicativas que separan los datos en varios subgrupos, obteniendo en cada división un valor de predicción. Para nuestro caso concreto, tanto la variable objetivo como las variables continuas *input* serán tratadas como nodos categóricos, obteniendo un punto de corte óptimo para cada división a través de los siguientes métodos iterativos.

**Estadístico F de Snedecor**

La división establecida para elaborar los subgrupos se realiza en función de aquella que hace variar más la media de la variable dependiente en los dos grupos.



Es decir, aquella que presenta el valor de estadístico de la F de Snedecor más alto.

**Criterio de varianza**

Calcula la variabilidad de la variable dependiente en cada grupo y se suma.



A continuación se elige la división que construye grupos más homogéneos internamente y diferentes entre sí.

### 3.2.4.- Técnicas de ensamblado basadas en árboles

Los árboles de clasificación y especialmente de regresión generalmente proporcionan poca fiabilidad en la predicción y mala generalización, debido a que cada hoja es un parámetro lo que provoca modelos finales sobreajustados e inestables para la predicción. Estas desventajas no han podido ser subsanadas a través de mejoras en las funciones de error o algoritmos de construcción, pero sí combinando el resultado de muchos árboles con diferentes criterios de parada, dando lugar a los modelos de ensamblado.

Los métodos de ensamblado son aquellos que pretenden predecir el comportamiento de una variable *output* a partir de la combinación de predicciones de varios modelos. En este apartado se van a analizar los principales métodos de ensamblado basados en los árboles de clasificación. El foco del análisis se ha centrado en los métodos *Bagging*, *Random Forest* y *Gradient Boosting* que son los métodos más utilizados en la actualidad y presentan una gran relevancia para nuestro tema de estudio. No obstante, es posible combinar cualquiera de los modelos expuestos en este proyecto con el objetivo de obtener los promedios de las predicciones de cada uno de ellos, y obtener de esta forma un modelo más robusto y frecuentemente menor sesgo.

#### 3.2.4.1.- Bagging

*Bootstrap aggregation* (o *Bagging*) es una técnica (L.Breiman, 1996) utilizada para generar múltiples versiones de un predictor y utilizarlos para obtener un predictor agregado. Es decir, el agrupamiento de los diferentes modelos de árboles de clasificación utilizando submuestras aleatorias con reemplazo en los datos, permite posteriormente combinar los resultados y mejorar la precisión del modelo. Esta agrupación sigue los siguientes pasos:

1. Se seleccionan n observaciones independientes e idénticamente distribuidas y se dividen en diferentes subconjuntos de datos admitiendo todo tipo de variaciones: con o sin reemplazamiento, estratificación entre otros.
2. Se aplican árboles de decisión con diferente complejidad para todas las observaciones de cada subconjunto.
3. Se promedian las m predicciones obtenidas en el apartado 1) y 2) obteniendo un único modelo predictivo.

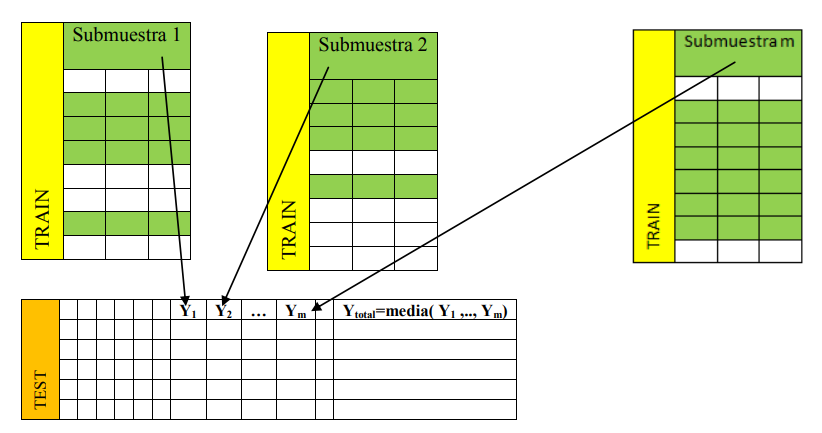


Figura 3.5: Bagging. Fuente: Javier Castro (2022)

El uso de diferentes submuestras ayuda a reducir la dependencia en la estructura de los datos completos para construir el modelo, y en consecuencia, permite reducir la varianza y generalmente el sesgo del modelo final.

En general, este método es especialmente útil cuando existen muchas variables con relación débil pero estable con la variable dependiente, cuando las relaciones entre los *inputs* y la variable respuesta no es lineal y cuando existen interacciones ocultas con muchas variables categóricas.

No obstante, el *Bagging* presenta algunas desventajas en el sentido de que se usan siempre las mismas variables para construir los diferentes árboles de clasificación. Esto implica que si alguna de las variables es dominante a la hora de construir los árboles, se generarán modelos iguales o parecidos, dando lugar a un resultado muy similar al árbol de decisión inicial.

#### 3.2.4.2.- Random Forest

La reducción de la varianza y del sesgo que se consigue mediante el modelo de ensamblado *Bagging,* dependía de la elevada correlación entre los clasificadores base creados. En este sentido, el algoritmo *Random Forest* (L.Breiman, 2001) tiene como objetivo mejorar la reducción de la varianza y sesgo a través de árboles de decisión menos correlados entre ellos, mediante la incorporación de aleatoriedad en las variables utilizadas para segmentar cada nodo del árbol. De forma análoga al caso anterior, se describirán los pasos seguidos en esta técnica.

1. Se seleccionan n observaciones independientes e idénticamente distribuidas y se dividen en diferentes subconjuntos de datos admitiendo todo tipo de variaciones: con o sin reemplazamiento, estratificación entre otros.
2. En cada nodo, se seleccionan *p* variables de las *k* originales de cada subgrupo y se escoge la mejor variable para la partición de cada nodo.
3. Se aplican árboles de decisión con diferente complejidad para las observaciones de cada nodo.
4. Se promedian las predicciones obtenidas por los árboles del apartado anterior, dando lugar a un único modelo predictivo.

Como se ha comentado en el apartado del *Bagging*, un problema de este método de ensamblado, resultaba en la existencia de variables predictoras muy dominantes que daban lugar a árboles muy parecidos. El *Random Forest* soluciona este inconveniente y ayuda a reducir la varianza añadiendo la aleatoriedad en este proceso de selección de variables.

Este algoritmo supone un avance adicional en el problema de selección de variables, evitando decidirse por un set de variables rígido, y aprovechando las ventajas del *Bagging*. Incorporar estas dos fuentes de variabilidad (remuestreo de observaciones y variables) permite ganar capacidad de generalización, y ayuda a reducir el sobreajuste a la vez que se adapta a las diferentes interacciones, no linealidad, problemas de extrapolación entre otros.

#### 3.2.4.3.- Gradient Boosting

El algoritmo *Gradient Boosting* ha supuesto uno de los avances más importantes en el área de *machine Learning* de los últimos años. Esta técnica, al igual que el *Bagging y Random Forest*, representan una familia de algoritmos que tiene como objetivo crear un clasificador con un error de entrenamiento reducido partiendo de un conjunto de clasificadores base.

El algoritmo introducido por primera vez por Friedman (2001) se basa en actualizar las predicciones en la dirección de decrecimiento dada por el negativo gradiente de la función de error *L(yi,f(xi)),* una función de predicción de yi basada en los valores de xi. Esta técnica sigue los siguientes pasos para el caso de regresión (variable objetivo continua):

1. Se da como valor predictivo de la variable *y*, para cada observación, la media de los valores de la variable *y*. A partir de este punto, en cada iteración del algoritmo la predicción de *y*, para cada observación se actualizará de manera individual . .
2. Para cada iteración de m se repiten los siguientes pasos:
3. Se calcula el residuo actual
4. Se construye un árbol de regresión para predecir los residuos, tomando ri(m)como variable respuesta, y el conjunto de las variables *input* como independientes. Se obtienen las predicciones de los residuos para los datos *train* y *test*.
5. Se actualiza la predicción de *y* para cada observación en la dirección de decrecimiento.
6. El proceso se detiene cuando se llega el número de iteraciones final deseado.

Esta técnica actúa mejor sobre bases de datos complejas y presenta ventajas en relación a que permanece invariante frente a las transformaciones monótonas, permite tratar los valores ausentes y las variables categóricas, requiere pocos parámetros a monitorizar (lo que ayuda a controlar el sobreajuste), presenta una gran eficiencia predictiva y es un modelo robusto respecto a la colinealidad y a las variables irrelevantes.

No obstante, este tipo de modelos se ven en gran medida superados por los modelos más simples (regresión) cuando los datos son relativamente sencillos.

#### 3.2.4.3.- Extreme Gradient Boosting

El algoritmo *Extreme Gradient Boosting* es una particularización de la variante más actual del Gradient Boosting, surge con destacado éxito en la plataforma de *Kaggle*, lugar habitual para las competiciones de *machine Learning*. Algunos ejemplos que destacan su aplicación incluyen (T.Chen, C.Guestrin, 2016): predicción de ventas de una compañía, predicción del comportamiento del consumidor, detección de movimiento, *text mining*, categorización de productos y por supuesto para el tema que nos ocupa en este proyecto, la predicción del gasto sanitario.

Esta técnica de ensamblado de árboles ha supuesto una de las alternativas más eficientes en términos computacionales y de implementación más escalable que el *Gradient Boosting,* ya que incorpora parámetros como λ, α, γ- (junto con otros parámetros base del Gradient Boosting) e incorpora una función de penalización basada en el número de hojas y la puntuación de predicción en cada hoja para controlar el sobreajuste y lograr de esta manera un mejor rendimiento (J.Brownlee, 2016).

### 3.2.5.- Support Vector Machine

La técnica *support vector machine* (SVM) tiene su origen en los trabajos sobre la teoría del aprendizaje estadístico y fue introducido port Vapnik y sus colaboradores Boser et al. (1992); Cortes & Vapnik (1995). Esta técnica constituye uno de los métodos más avanzados para la resolución de problemas de clasificación y regresión, aunque en la actualidad están siendo utilizadas con destacable éxito en múltiples campos relacionados con la visión artificial, reconocimiento de caracteres, categorización de texto, análisis de series temporales entre otras ramas.

Dentro del área de la predicción, el objetivo fundamental del SVM consiste en plantear un problema de separación lineal de clases con métodos algebraicos, también llamados hiperplanos, ya sea en el espacio original de los datos *input*, si son linealmente separables, o en un espacio transformado si los datos no son linealmente separables dentro del espacio original. Este método se basa en las siguientes tres ideas fundamentales:

* *Maximal Margin Classifier* (Vapnik y Chervonenkis, 1963): La idea de seleccionar este hiperplano radica en realizar una separación que equidista de los datos más cercanos de cada clase (función lineal) para conseguir lo que se denomina como un *margen máximo* a cada lado del hiperplano, considerando únicamente los datos de entrenamiento que caen justo en la frontera de dichos márgenes. Los puntos que se encuentren en la frontera de dichos márgenes se conocen como *vectores soporte*.

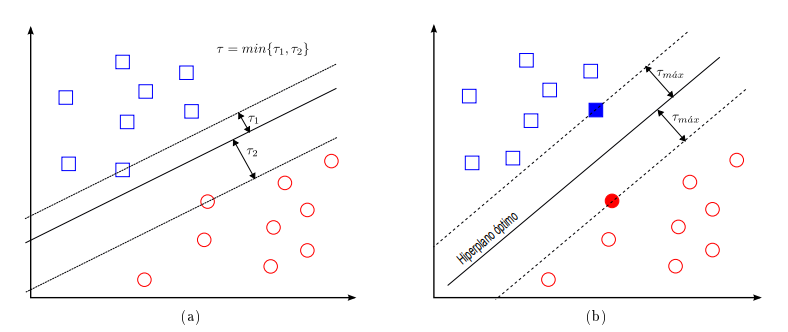
**

Figura 3.6: SVM maximal margin. Fuente: Juan del Rosal (2016)

* *Soft Margin SVM (*Cortes y Vapnik, 1995): La separación perfecta a la hora de determinar los puntos que se sitúan en cada hiperplano no suele ser perfecta, por lo que es necesario permitir observaciones mal clasificadas por los separadores para no incurrir en sobreajuste. La distancia a estos puntos respecto de los márgenes se determinan a través de añadir al problema de optimización una constante de error o residuo y otra de penalización *C* (coste*),* que se encuentra relacionada inversamente con la anchura del margen máximo. Dependiendo de los valores de *C* se estará penalizando mucho o poco la mala clasificación de dichos puntos.
* *Kernel (*Boser, Guyon, Vapnik, 1992): La separación entre clases en muchos casos no suele ser lineal. Por ello, trabajar con un espacio de dimensión superior, permite al algoritmo encontrar nuevas separaciones lineales y buen tamaño de margen (ver Figura 3.7).

Sin embargo, este incremento de dimensión complica el proceso y hace impracticables los cálculos. Por esta razón, se introduce el término “truco Kernel” (“the Kernel trick”) que enuncia que cualquier algoritmo que dependa solo de los productos escalares permite trabajar computacionalmente en una dimensión controlada a través de la función Kernel.

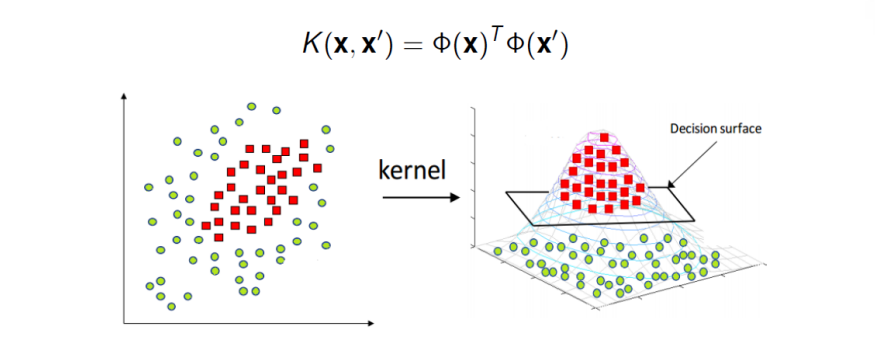


Figura 3.7: Kernel. Fuente: Daniel Gómez (2022)

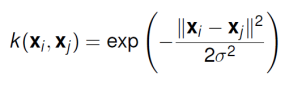
En el presente trabajo se estudiarán las funciones *Kernel* más habituales:

* **Kernel Lineal:** Emplea un *Kernel* lineal que es equivalente al Soft margin SVM expuesto anteriormente. Se determina a través de la siguiente función .

* **Kernel Polinómico:** Permite extenderse a separaciones no lineales

 **.** Siendo *c (*coste)la constante que representa la penalización y  *p (*grado del polinomio*)* la dimensión.

* ***Kernel radial basis function RBF:*** Es una función más general que en el modelo anterior. Dispone únicamente del parámetro σ que permite controlar el comportamiento del Kernel, de manera que valores altos de gamma implican menor sesgo y mayor sobreajuste. Se determina a través de la siguiente función.

******

### 3.2.6.- Validación de modelos

Una vez se ha modelizado el comportamiento de nuestro conjunto de datos, es necesario aplicar un conjunto de técnicas que nos permitan identificar cómo de bien se comportan nuestros modelos al aplicarlos sobre conjuntos de datos de los que no han aprendido en primer lugar. En este sentido, para este trabajo se aplicará el método de validación cruzada para determinar la combinación de parámetros óptima de cada modelo, y a partir de esta configuración óptima se evaluará el rendimiento de dicho modelo de manera más precisa a partir de validación cruzada repetida, uno de los sistemas más precisos a la hora de comprobar la eficacia de los respectivos modelos.

Mediante este enfoque de validación cruzada simple (ver Figura 3.8) se dividirán nuestros datos en *k* subconjuntos de igual tamaño en función del número de observaciones que se dispongan, en cada iteración se aplicará el modelo en cuestión sobre los subconjuntos k-1, dejando un grupo aparte (*test*) utilizado para validar y comprobar la eficacia de nuestro modelo, obteniendo una predicción del error para esa partición. Para eliminar el efecto de la forma en la que los datos han sido divididos, este proceso se repetirá *n* veces (*n-folds)* y se realizará el promedio o la suma de los errores de cada iteración.

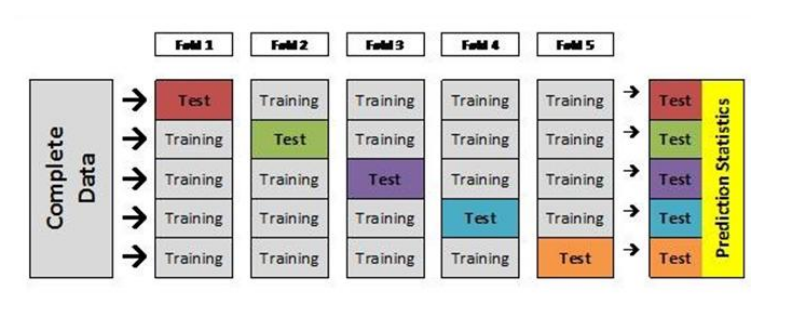


Figura 3.8: Validación cruzada. Fuente: Javier Portela (2022)

Este modelo resulta más preciso que otros métodos como el Training-Validación-Test ya que utiliza la totalidad de nuestros datos para realizar la predicción. No obstante, este método de validación presenta algunas desventajas en el sentido de que el resultado final depende en gran medida del número de grupos *k*, por lo que el error de predicción puede estar subestimado.

Una alternativa para solucionar este inconveniente es la validación cruzada repetida. Esta técnica es idéntica a la anterior, pero con ciertos matices, para evaluar en mayor medida la variabilidad del modelo, se repite el proceso de validación cruzada *q* veces variando en cada una de ellas la semilla utilizada. Esto ayuda a evitar la aleatoriedad en la selección de grupos y otorga robustez a la predicción realizada.

# 4.- Resultados

En este capítulo se realizará un análisis de la base de datos y se realizarán los respectivos modelos atendiendo a las fases de la metodología SEMMA expuestas anteriormente:

* *Explore* (exploración): Durante esta fase se explorarán los datos con el objetivo de detectar posibles errores, relaciones entre las variables y posibles tendencias.
* *Modify* (modificar): Se realizarán modificaciones sobre los datos, ya sea creando seleccionando o transformando variables con el objetivo de facilitar la modelización.
* *Model* (modelizar): elaboración de modelos que ayuden a predecir el comportamiento de la variable objetivo
* *Asses* (evaluar): comprobación y comparación de la calidad de los modelos realizados

Para el correcto desarrollo de esta metodología, en primer lugar, se analizará la estructura de los datos sobre los que se va a obtener información. Se seleccionarán aquellas variables más relevantes para el estudio, se depurarán los datos en base a la información observada y finalmente se llevará a cabo el respectivo modelaje.

## 4.1.- Análisis exploratorio

Una vez se ha extraído la muestra del 12% de nuestros datos, se comenzará a realizar el proceso de análisis y depuración de la base de datos. Una de fases más importantes de la metodología SEMMA para una correcta modelización es la fase del análisis exploratorio, la cual permite conocer la naturaleza de los datos, detectar posibles errores e identificar relaciones entre las variables y posibles tendencias, que ayudará a realizar las modificaciones necesarias para que los modelos aprendan bien de nuestros datos y saquen conclusiones precisas.

La depuración tras el periodo de exploración supone uno de los pasos más importantes ya que es la fase donde modificamos los datos para que se cumplan las especificaciones esperables de la base de datos. Es decir, en esta fase se tratará de evitar la presencia de valores anómalos y ausentes, de manera que se asegure que los datos cumplan las características necesarias que hagan posible su modelización a través de las diferentes técnicas planteadas, con el objetivo de obtener los mejores resultados posibles.

Para el correcto desarrollo del este análisis exploratorio se utilizará el software *SAS Enterprise Miner*, que permitirá de una forma muy intuitiva, comprender y visualizar la estructura de los datos.

### 4.1.1.- Importación y asignación de los roles de las variables.

Se ha importado la base de datos en el *SAS Enterprise Miner* a través del nodo de importar archivo. En la Figura 4.1, se muestran todas las variables *imput* de nuestra base de datos clasificadas según su rol y su nivel. Disponemos de un total de 18 variables conformadas por:

* 1 variable id
* 3 variables nominales
* 1 variable objetivo (Gasto en farmacia)
* 11 variables numéricas de tipo intervalo
* 3 variables binarias

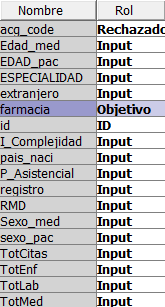


Figura 4.1: Importación y asignación de roles

La variable *acg\_code* no formará parte del análisis y modelización puesto que es un indicador que para su construcción engloba gran parte de las variables utilizadas dentro de la entidad paciente. Por lo tanto, para tener una visión más clara de los efectos individuales de cada variable sobre el gasto farmacéutico y evitar la introducción de ruido en la modelización se declarará la variable como rechazada.

### 4.1.2.- Análisis descriptivo y gráfico del conjunto de datos.

Una vez se han importado todas las variables que se van a utilizar en este proyecto, desde el propio nodo donde se ha aplicado la muestra se puede realizar una exploración de la distribución de todas las variables tanto numéricas como categóricas.

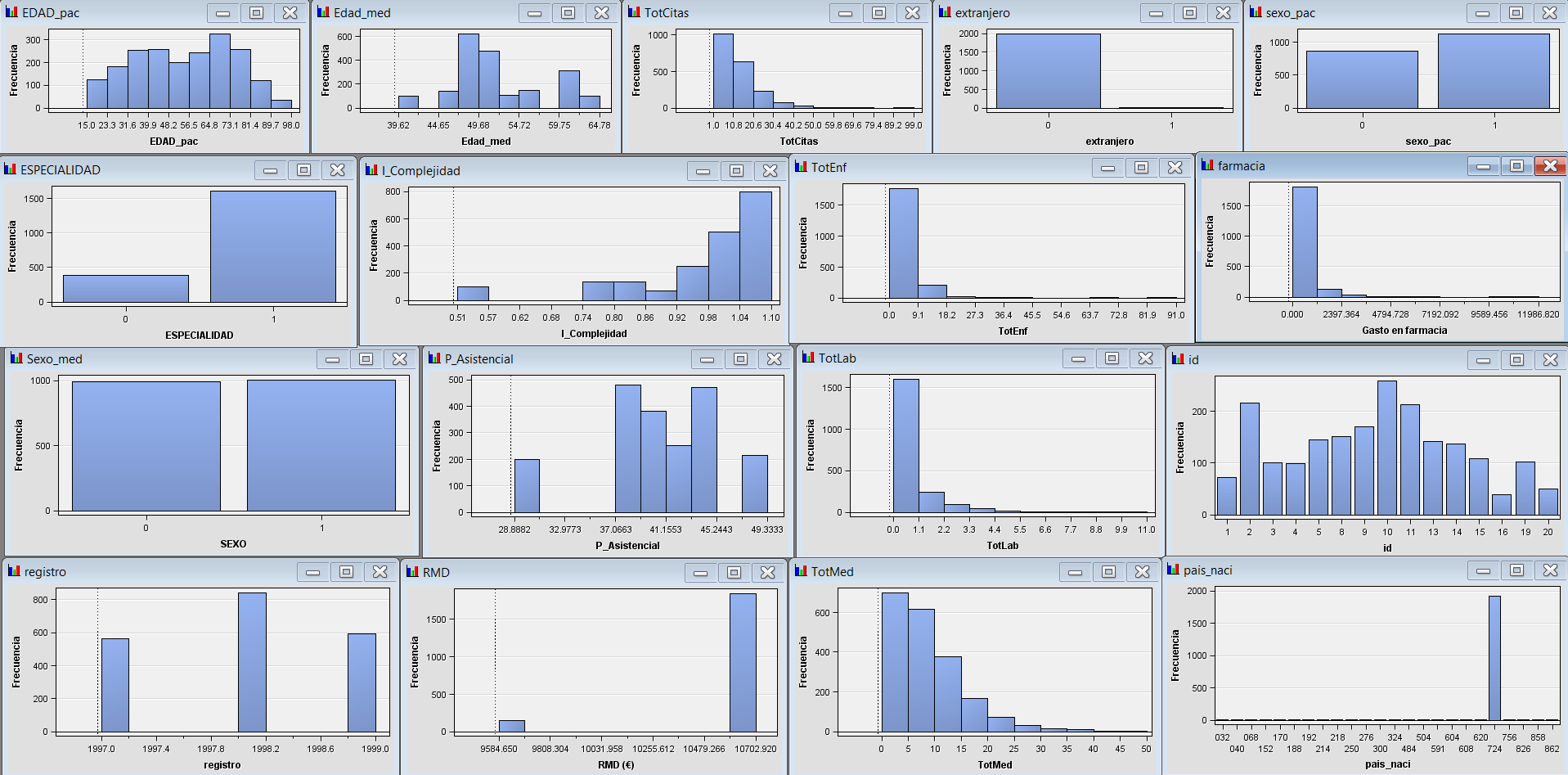


Figura 4.2: Exploración de variables

Atendiendo a los resultados podemos ver que la variable binaria *extranjero* está claramente desbalanceada, siendo la mayor parte de los pacientes españoles. Esto se traslada a la variable de *país\_naci* que dispone de 26 categorías con muy pocas observaciones en cada una de ellas (menos el 1%). Debido a que esta variable puede inducir ruido en el modelo en el sentido que aporta prácticamente la misma información que extranjero, se decide no usarla en el modelo.

Respecto al número de consultas por paciente, se puede ver que la mayor parte de ellas se realizan en las consultas de enfermería y medicina y presentan indicios de existencia de outliers.

En relación al conjunto de variables del médico cabe mencionar que la mayor parte de médicos no disponen de especialidad, es decir no disponen de una plaza en propiedad. La variable id parece arrojar información acerca de la carga de trabajo de cada médico, en el sentido que se agrupan un número de pacientes para cada id del médico. El médico 10 parece ser aquel que ha atendido a un mayor número de pacientes.

Por último, la variable objetivo de gasto en farmacia parece presentar *outliers* en el sentido que la mayor parte del gasto se sitúa entre 0-500€, pero hay registros que superan los 11.000.

Continuando con el análisis estadístico, a través del nodo de *Data Mining Database (DMDB)* obtenemos información estadística tanto de las variables de tipo intervalo como de clase. Esta base de datos dispone de 7229 observaciones y un total de 16 variables. De ellas, 11 son de tipo intervalo (incluyendo la variable objetivo), 1 nominal y 4 binarias.

Respecto a las variables de tipo intervalo (ver Figura 4.3) no disponemos de valores ausentes. Las variables relacionadas con el número de citas *TotCitas*, *TotEnf*, *TotMed* (recuadro rojo) presentan valores mínimos y máximos entre 0-165 con valores medios alejados de sus respectivos máximos. La variable objetivo (recuadro azul) presenta valores mínimos y máximos comprendidos entre 0 y 18140.03 con un valor medio de 362.88. Esto podría indicar la existencia de posibles valores atípicos. En suma a lo anteriormente mencionado en el recuadro verde, todas estas variables presentan una Curtosis positiva y mayor que 0, esto implica que los datos presentan valores atípicos más extremos que una distribución normal.

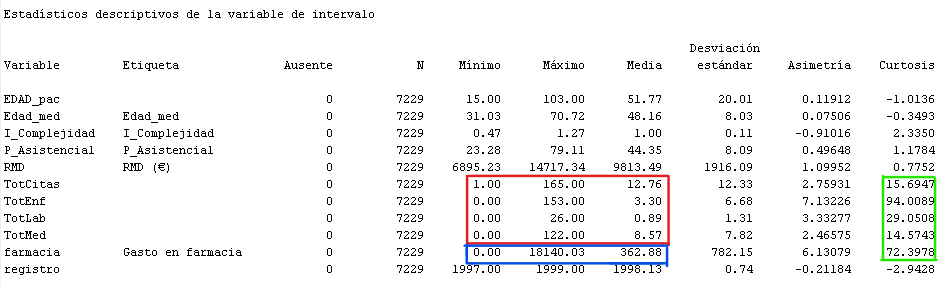
**

Figura 4.3: Exploración de variables intervalo

En cuanto a las variables de clase (ver Figura 4.4) no disponemos de valores ausentes, y podemos comprobar que *país\_naci* dispone de 26 niveles con la mayoría de categorías huérfanas.

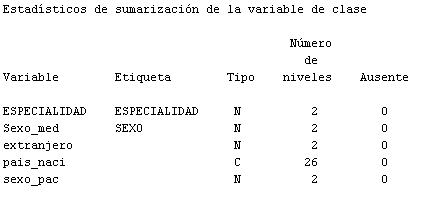
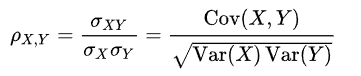


Figura 4.4: Exploración de variables clase

El nodo de explorador de estadísticos de *SAS Enterprise Miner* es otra herramienta estadística que permite examinar el comportamiento de las variables de nuestra base de datos.

En concreto este nodo entre otras funciones, aporta una salida bastante informativa acerca de la dependencia lineal de las variables cuantitativas con nuestra variable objetivo, esto es el coeficiente de correlación de Pearson. Este coeficiente se define de la siguiente forma:



Donde

* σ xy es la covarianza de (X,Y)
* σ x es la desviación estándar de la variable x
* σ y es la desviación estándar de la variable y

Este coeficiente se mostrará junto con la matriz de correlaciones (realizada en R-Studio) para analizar la dependencia lineal de las variables numéricas continuas (excluyendo las variables relacionadas con el id y el año de registro).

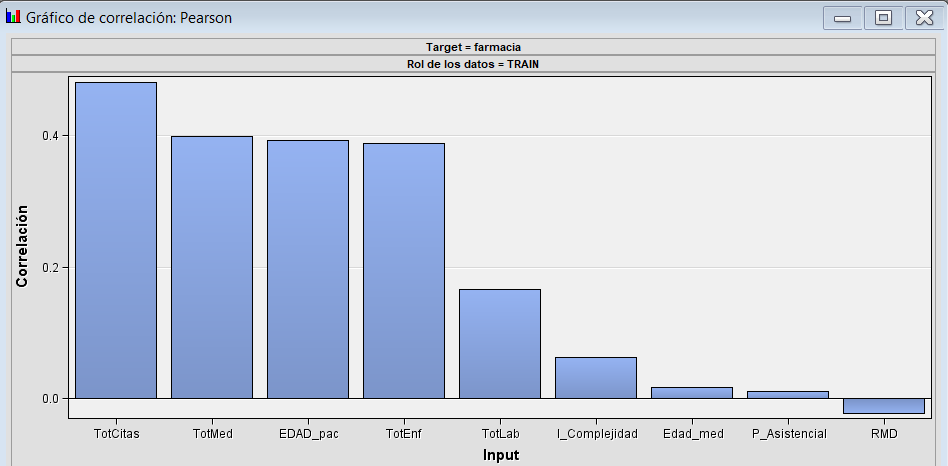
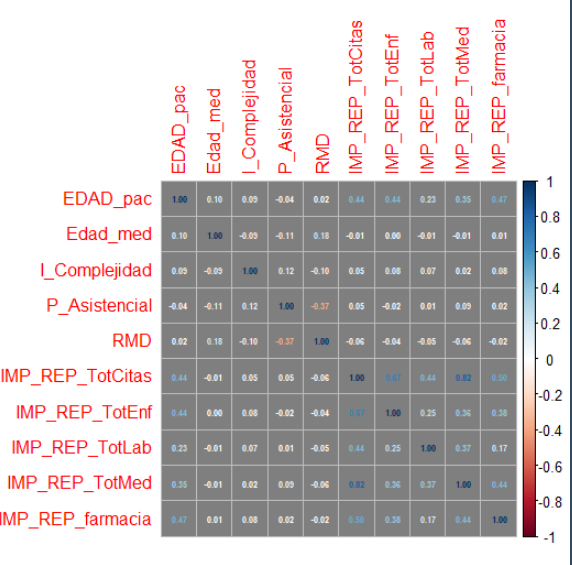
 

Figura 4.5: Dependencia lineal

En la Figura 4.5 se puede ver que tanto en la matriz de correlaciones (gráfico de la derecha) y en el gráfico de correlación de Pearson (gráfico de la izquierda) las variables que presentan una mayor dependencia lineal con la variable objetivo farmacia son el número de citas y la edad del paciente. Esta correlación positiva indica que cuando se incrementa el número de citas en cualquier departamento o la edad principalmente, el gasto en farmacia se incrementa. Es importante destacar que hay una correlación relativamente alta entre la edad y el número de citas, esto deberemos tenerlo en cuenta en futuros análisis de multicolinealidad.

Para concluir la parte exploratoria, el nodo de multigráfico de *SAS Enterprise Miner* nos permite observar de múltiples formas la distribución de nuestras variables. En este caso, a través de un gráfico de dispersión se va a analizar la supuesta dependencia lineal expresada el gráfico anterior (ver Figura 4.6), de la variable genérica TotCitas y la variable EDAD\_pac respecto a la variable objetivo farmacia.

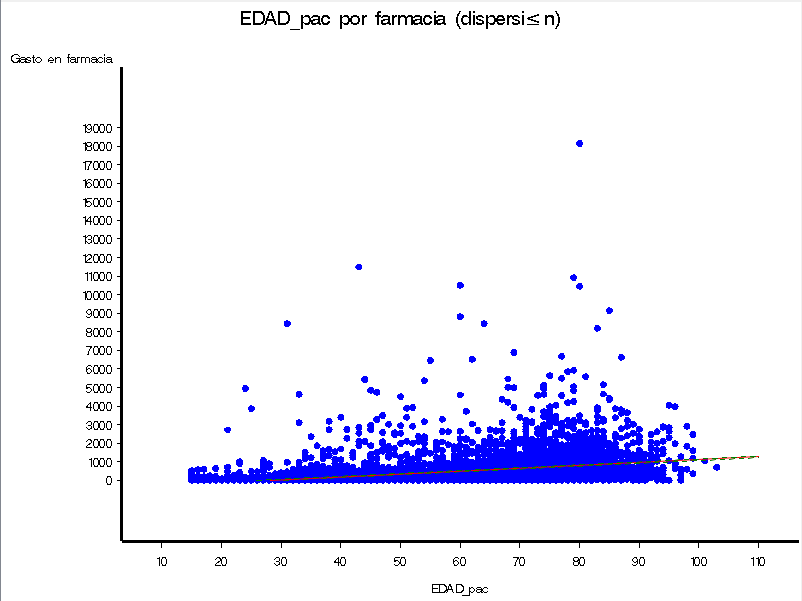
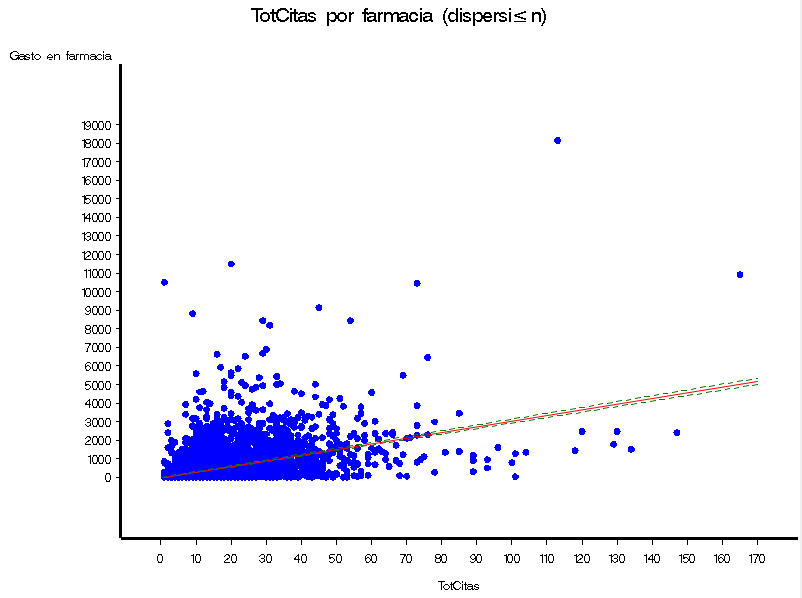
 

Figura 4.6: Nodo multigráfico

En el gráfico de la izquierda se puede comprobar el claro comportamiento lineal de la edad del paciente, también se observa la existencia de diversos valores atípicos que deberán ser tratados. Respecto al gráfico de la derecha, no se observa una linealidad tan clara como con la edad, habiendo una alta concentración de puntos en un tramo concreto, esto puede sugerir la necesidad de realizar alguna transformación sobre el gasto en farmacia. Se observan datos atípicos.

## 4.2.- Modificación

Una vez analizado el comportamiento de las variables de nuestra base de datos, toma lugar la siguiente fase de la metodología SEMMA, la modificación de los posibles errores detectados en la fase de exploración.

### 4.2.1.- Corrección de errores

La corrección de errores se realizará a través del nodo de Reemplazo de *SAS Enterprise Miner*. Este nodo es de gran utilidad puesto que permite por un lado recategorizar y agrupar aquellas variables de tipo clase que puedan inducir ruido en nuestro modelo, y por otro lado, permite acotar el rango de valores (antes del tratamiento de valores atípicos) de las variables numéricas con el objetivo de que muestren información lógica. Esto es, evitar la existencia de edades negativas o superiores a 120 años, gastos negativos entre otros casos.

Respecto a las variables de tipo clase (ver Figura 4.7), se obtiene en la salida del nodo únicamente las variables binarias, ya que se ha decidido no utilizar la variable del país de nacimiento. Por defecto, cada variable tomará el valor 0-1 y \_DEFAULT\_ en caso de haber algún error que no se corresponda con estos valores.

**

Figura 4.7: Nodo reemplazo de variables de clase

En relación a las variables de tipo intervalo (ver Figura 4.8) se van acotar que las variables *EDAD\_pac, EDAD\_med, RMD, TotCitas, TotEnf, TotLab, TotMed* y *farmacia* tomen únicamente valores positivos, de manera que en caso de haber valores ilógicos se transformarán en valores ausentes. Para ello, se establecerá como editor de reemplazo la opción de límites especificados por el usuario y en el subapartado de puntuación, valores de sustitución como ausentes.

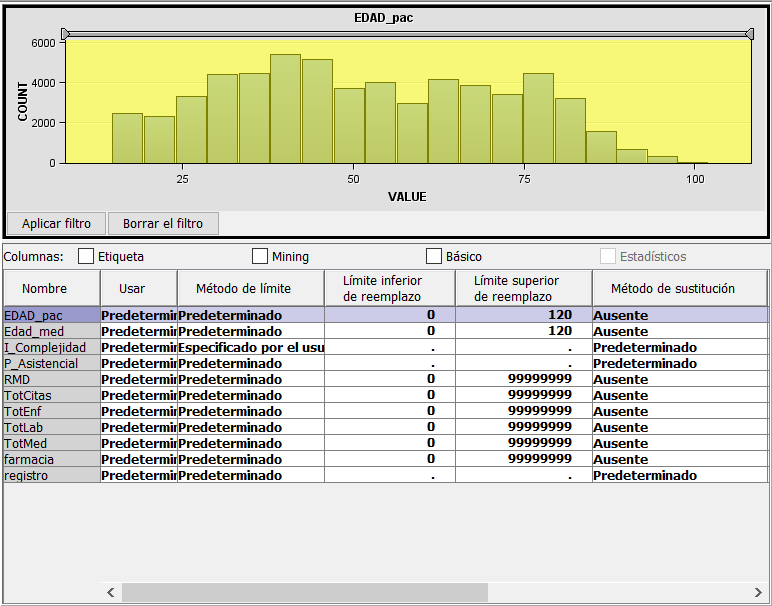


Figura 4.8: Nodo reemplazo de variables intervalo

### 4.2.2.- Tratamiento de valores atípicos

Un dato atípico (D.M Hawkins, 1980) es una observación que se desvía mucho de otras observaciones y despierta sospechas de ser generada por un mecanismo diferente. Para obtener resultados adecuados y predicciones precisas es importante realizar un proceso de detección de valores atípicos, ya que en caso contrario estos valores podrían inducir ruido en el modelo y podrían tener una gran influencia en los resultados. A continuación, se describen algunas de las técnicas utilizadas para la detección de este tipo de valores para variables continuas:

* Valores alejados de la media muestral: serán atípicos todos aquellos datos que estén alejados k ∗ σ veces de la media, siendo σ la desviación típica y k un parámetro comprendido normalmente entre 1.5 y 6. Este método asume que la distribución sigue una normal, por lo tanto sólo tiene sentido su uso para distribuciones aproximadamente simétricas.
* Valores alejados de la mediana muestral o MAD (*Median Absolute Deviation*): serán atípicos los datos que estén alejados k ∗ MAD veces de la mediana, siendo el valor MAD la mediana de las distancias absolutas a la mediana. El valor de k generalmente deberá estar comprendido entre 6 y 15. Al igual que el caso anterior, este método resulta más adecuado para distribuciones aproximadamente simétricas, pero solo cuando la mediana es distinta de 0.
* Métodos basados en percentiles extremos: serán atípicos los valores comprendidos entre el 0.5-1% de los valores más grandes y el 0.5-1% de los valores más pequeños. Este método se aplicará únicamente sobre la variable objetivo puesto que queremos mantener la gran variabilidad que dispone, perdiendo la mínima cantidad de observaciones.

En la Figura 4.9 se pueden observar las variables sobre las que se van a tratar los valores atípicos y la técnica utilizada en cada caso; valores alejados de la media muestral con un valor de k = 3 en color rojo, valores alejados de la MAD con un valor de k = 9 en color azul y percentil extremo en color verde.

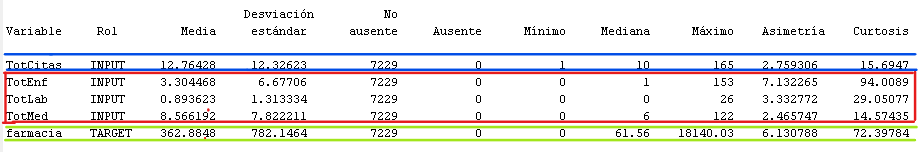


Figura 4.9: Detección de valores atípicos

Para realizar la depuración se ha vuelto a utilizar el nodo de Reemplazo, pero esta vez especificando los métodos de detección de valores atípicos para cada uno de los casos. Es necesario tener en cuenta el número de *outliers* detectados respecto al total de observaciones. En este sentido, se clasificarán como valores extremos siempre que representen menos del 5% del total de observaciones.



Figura 4.10: Valores atípicos detectados

En la figura 4.10 se recogen todos los valores extremos detectados. Debido a que suponen menos del 5% del total de las observaciones, podemos catalogarlos como atípicos y por lo tanto serán tratados como ausentes.

### 4.2.3.- Imputación de valores ausentes

Una vez se ha detectado los valores extremos, se ha procedido a realizar la imputación de los valores ausentes. Este proceso consiste en sustituir aquellos valores ausentes por valores reales, es necesario tener en cuenta la cantidad de datos que se imputan para no perder la variabilidad de la muestra. En nuestro caso, la proporción de ausentes sobre el total es ínfima por lo que se va a proceder a imputar sobre la media conservando la variabilidad.

Este proceso se realizará a través del nodo de imputar de *SAS Enterprise Miner.*

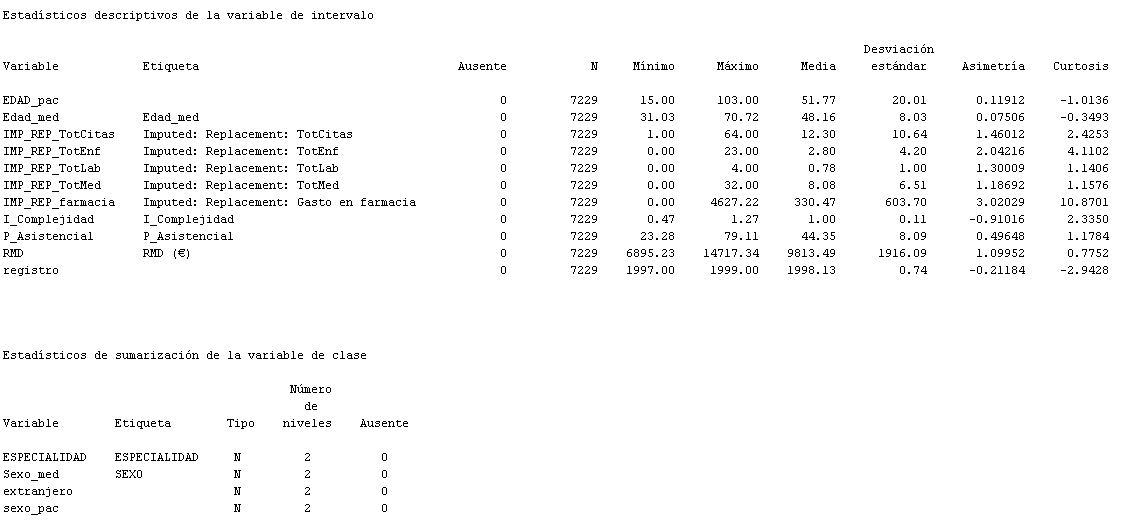


Figura 4.11: Resultado depuración

Una vez concluido este proceso en la figura 4.11 se obtienen todas las variables que pueden aportar información sobre el gasto en farmacia sin datos atípicos ni valores ausentes.

### 4.2.4.- Transformación de variables

En este apartado se ha transformado la variable objetivo *farmacia* con el objetivo de intentar maximizar la relación existente con el resto de variables *input*. Es decir, utilizando algún tipo de transformación ya sea en forma de logaritmo, exponencial entre otras, permite analizar mejor el comportamiento de la variable objetivo, de manera que durante los modelos de predicción posteriores se puedan establecer mayor relaciones entre las variables. Este proceso se ha realizado a través del nodo de *Transformar variables* del *SAS Enterprise Miner.*

Al ser nuestra variable objetivo de tipo continua, para captar en mayor medida el comportamiento del gasto en farmacia se ha realizado una transformación logarítmica sobre la variable dependiente. Esta transformación ha permitido captar en mayor medida el comportamiento de los principales predictores, reduciendo la alta dispersión inicial de los datos (duplicando prácticamente el valor del R2 inicial) y mitigando el impacto de posibles *outliers.*

A continuación, utilizando el nodo anterior, se ha creado una nueva variable llamada *random* que dispone de valores aleatorios para analizar la importancia de las variables después de la transformación logarítmica. Esto nos ha permitido identificar cómo de bien o mal predicen el resto de variables sobre nuestro *output* modificado.

En este sentido, utilizando nuevamente el nodo de explorador de estadísticos, si la variable *random* dispone de un valor superior o similar a otra variable *input* podemos deducir que dicha variable no está aportando más información que un componente aleatorio.

En la figura 4.12 observamos que nuestro componente aleatorio ocupa la posición 11 con un valor sobre la variable objetivo de 0.018. Por lo tanto, todas aquellas variables situadas a la derecha de este componente aleatorio (o muy cerca de él) parecen no resultar útiles para el análisis. Por otro lado, se ha observado que el valor de los principales predictores (edad y número de citas) se ha visto incrementado por la transformación logarítmica.

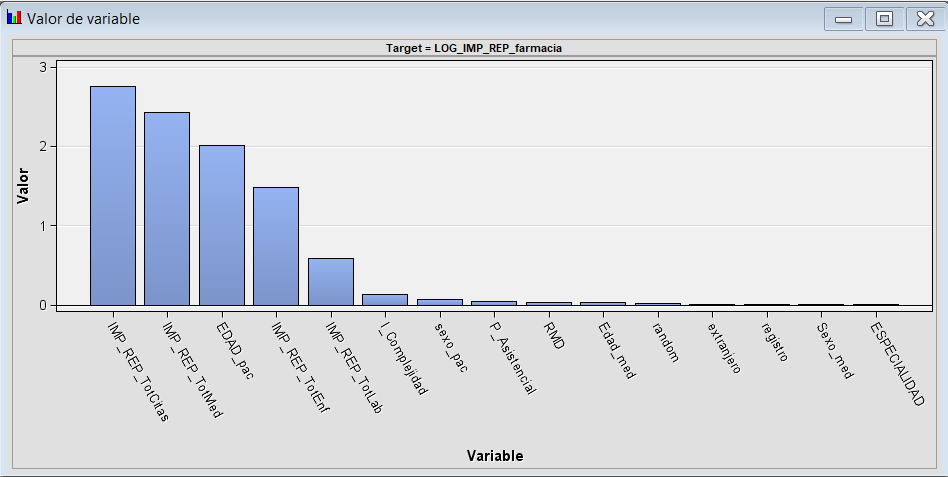


Figura 4.12: Valor de las variables input respecto al output modificado

## 4.3 Selección de variables

Antes de comenzar con la modelización, es necesario realizar el último paso de la depuración, que consiste en determinar que variables pueden resultar útiles para el análisis. Este proceso es de gran importancia ya que permite reducir el nivel de sobreajuste de nuestros modelos y ayuda a optimizar los tiempos computacionales. Es decir, la selección de variables otorga robustez a nuestros análisis a la hora de incorporar nuevos datos sobre los que no ha aprendido el modelo inicial.

Una de las peculiaridades de este *dataset*, consiste en que las principales variables que parecen explicar la variable respuesta (ver Figura 4.12) son la edad y el número de citas en los diferentes tipos de consultas. No obstante, estas variables son combinación lineal respecto al número total de citas, por lo que no pueden utilizarse todas de manera conjunta para la modelización. En este sentido para eliminar posibles problemas de multicolinealidad, se ha divido nuestro conjunto en un *set-1* de variables inicial eliminando el número total de citas, y otro *set-2* eliminando las variables relacionadas con el número de citas en las diferentes consultas.

En este proyecto se ha decidido trabajar con el *set-1* de variables, ya que permite explicar cómo afecta el gasto sanitario dependiendo del tipo de consulta que se lleva a cabo, y desde un punto administrativo, ayuda a conocer qué departamentos requieren un presupuesto superior para llevar a cabo sus actividades dentro de un centro médico asistencial

A partir de este set de variables, se aplican los siguientes métodos de selección a través de las herramientas de *SAS Enterprise Miner y R-Studio*:

|  |  |
| --- | --- |
|  | Selección de variables *SAS Enterprise Miner* |
| Selección de variables | En este método se ha utilizado el nodo de Selección de variables cuyo R2 sea superior a 0.005 |
| MCO | En este método se ha utilizado el nodo de mínimos cuadrados ordinarios con 200 iteraciones. |
| Lineal | En este método se han seleccionado las variables a través del nodo de regresión lineal a partir de los métodos forward, backward y step by step. (Se agrupan lo resultados ya que en las tres salidas se han obtenido las mismas variables) |
| GB | En este método se ha utilizado el nodo de incremento gradiente. Se han seleccionado aquellas variables utilizando un árbol de clasificación con 50 iteraciones y semilla 12345. |

Tabla 4.1: Selección de variables SAS Enterprise Miner

|  |  |
| --- | --- |
|  | Selección de variables *R-Studio* |
| AIC | Se ha realizado un modelo de regresión lineal utilizando el AIC como criterio de selección de variables. |
| BIC | Realiza un modelo de regresión lineal utilizando el BIC como criterio de selección de variables. Para ello, se especifica indica k=log(n) (en nuestro caso disponemos de 7229 observaciones k = 9) |
| Boruta | Hace copias de las variables inputs donde los valores fila están permutados (no en su lugar), y compara la importancia de la variable input original con la de su copia permutada. El proceso es secuencial y usa random forest. |

Tabla 4.2: Selección de variables R-Studio

Una vez realizados los siete métodos de selección de variables, en la siguiente tabla se recogen aquellas variables que cada algoritmo considera como importantes para explicar la variable respuesta. Para cada método, aquellas variables que se consideren importantes recibirán el valor de (1), (-) en caso contrario.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Obs | Nombre\_Var | SV | MCO | Lineal | GB | Boruta | AIC | BIC |
| 1 | Edad\_pac | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | Tot\_med | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | Tot\_enf | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | Tot\_Lab | 1 | 1 | 1 | 1 | 1 | 1 | - |
| 5 | I\_complejidad | 1 | - | 1 | 1 | 1 | 1 | - |
| 6 | Sexo\_pac | - | - | 1 | 1 | 1 | 1 | - |
| 7 | P\_Asistencial | - | - | 1 | 1 | 1 | 1 | - |
| 8 | Edad\_med | - | - | - | 1 | 1 | 1 | - |
| 9 | RMD | - | - | - | 1 | 1 | 1 | - |
| 10 | Sexo\_Med | - | - | - | - | 1 | 1 | - |
| 11 | Especialidad |  |  |  |  | 1 | - | - |
| 12 | Registro | - | - | - | - | - | - | - |
| 13 | Extranjero | - | - | - | - | - | - | - |

Tabla 4.3: Selección de variables

Atendiendo a los resultados obtenidos, el conjunto de *inputs*  que se consideran más importantes y que están englobados en al menos 6/7 algoritmos presentan el color verde. Las variables presentes en al menos 5/7 métodos disponen del color azul, 4/7 color amarillo, 3/7 naranja, entre 1/7 y 2/7 color rojo y finalmente aquellas que ningún método considera relevante se representan en gris.

Se ha decidido establecer como criterio uso, aquellas variables que hayan sido seleccionadas por al menos 4/7 métodos (color amarillo). Es importante señalar que esta selección de variables se utilizará únicamente para los modelos de regresión lineal, redes neuronales y SVM. Mientras que para los árboles y los métodos basados en árboles se utilizarán todas las variables, ya que estos modelos disponen de su propio criterio de entrada.

## 4.4.- Modelización

En este apartado se desarrollará la fase modelización correspondiente a la metodología *SEMMA* expuesta en puntos anteriores. Esta fase tiene el objetivo de elaborar modelos que puedan ayudar a predecir el comportamiento de la variable objetivo (gasto en farmacia). Para ello, se emplearán las técnicas expuestas anteriormente, comenzando con los modelos clásicos de predicción, y posteriormentese introducirán de manera progresiva modelos más complejos de *Machine Learning*.

Como medida de referencia para determinar el menor sesgo del modelo se utilizará el error cuadrático medio (RMSE), el cual mide el error medio de un valor predicho respecto a un valor observado o conocido.

El criterio de validación utilizado será la validación cruzada simple para determinar la configuración de parámetros óptima de cada conjunto de modelos, con el objetivo de poder comprobar una alta variedad de modelos controlando el tiempo computacional. A partir de esta configuración óptima, se seleccionarán aquellos modelos previsiblemente mejores y se aplicará validación cruzada repetida sobre ellos.

### 4.4.1.- Regresión lineal

A partir de los *sets* obtenidos por cada método de selección de variables del capítulo anterior, se realizará para cada uno de ellos un modelo de regresión lineal y se estudiará el sesgo y la varianza para cada uno de ellos. Este proceso se ha llevado a cabo utilizando la técnica de remuestreo de validación cruzada repetida y *boxplot*, ya que supone una de las opciones más precisas para estudiar el comportamiento de los diferentes modelos.

Para ello, se ha dividido nuestro conjunto de datos en 10 grupos (9 grupos *train* y 1 *test* variable) con 20 repeticiones obteniendo como resultados un error y una serie de estadísticos predictivos para cada semilla. Se ha representado la distribución de los errores (RMSE) de cada modelo a través del siguiente diagrama de cajas.

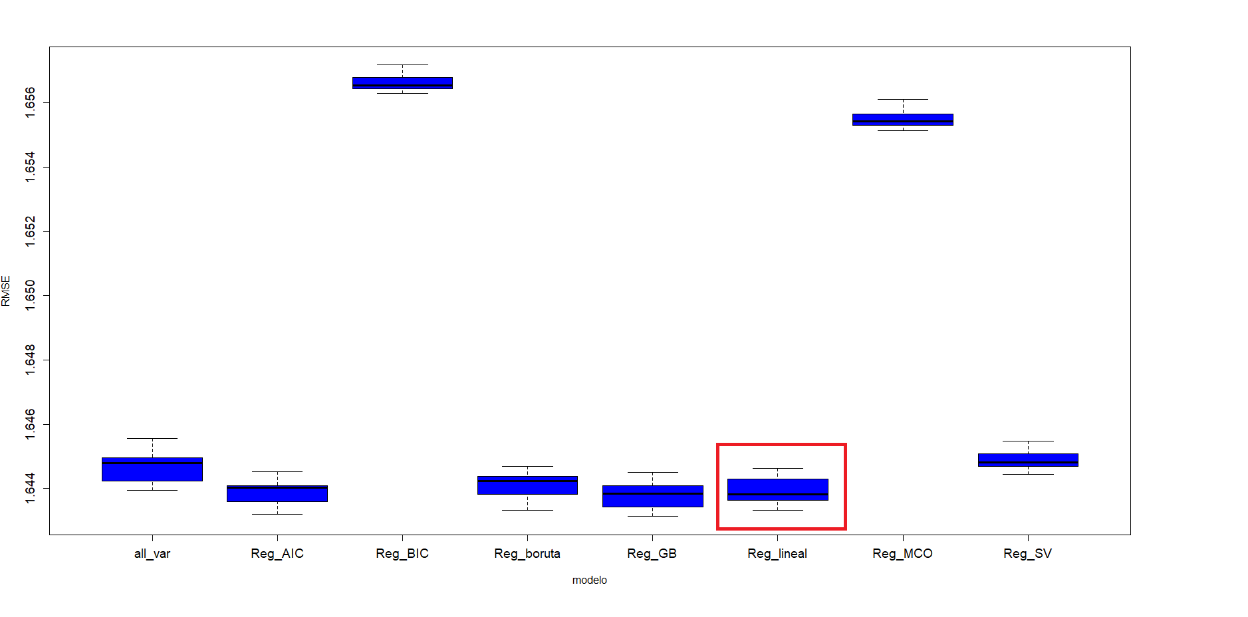


Figura 4.13: Validación cruzada y boxplot. Regresión lineal

En la Figura 4.13 se observa que los modelos MCO y BIC que únicamente incluyen las variables relacionadas con el paciente y que presentan un mayor peso sobre el gasto en farmacia, disponen de un sesgo muy superior frente al resto que incorpora *inputs* relacionados con el médico y el municipio. No obstante, ya que incorporan un menor número de variables independientes, la altura de la caja (varianza) es ligeramente menor al resto de conjuntos.

Por otro lado, para el resto de modelos la varianza y el sesgo es relativamente similar, concretamente observamos que las regresiones lineales cuyos *sets* de variables se corresponden con el método AIC, GB y por el consenso de *los métodos forward, bacward y step by step* (que es equivalente al *set* de variables amarrillo cuyo consenso se ha obtenido a partir de 4/7 algoritmos del punto anterior) obtienen una varianza muy similar a sus competidores pero un sesgo más reducido. Sin embargo, es importante destacar que el modelo construido a partir del consenso de los 3 métodos lineales (Reg\_lienal) presenta una varianza relativamente inferior y un sesgo mediano equivalente al GB e inferior al AIC, pero incorpora menos variables respecto a cada uno de ellos, por lo que lo hace preferible.

Por lo tanto a fin de establecer comparaciones futuras respecto a otros modelos más complejos que utilicen el mismo proceso de selección de variables (cuyo conjunto se obtiene por el consenso de 4/7 métodos de selección), se ha establecido como el mejor modelo de regresión el conjunto “Reg\_lineal”.

Para que este modelo óptimo sea útil, tenga capacidad de predictiva y sea interpretable en puntos posteriores, es tan importante que el sesgo y la varianza sean los más reducidos posibles, como que la estimación cumpla las hipótesis paramétricas fundamentales enunciadas en puntos anteriores (ver Apartado 4.1). En este sentido, se comprobará el cumplimiento de las siguientes hipótesis de manera gráfica.

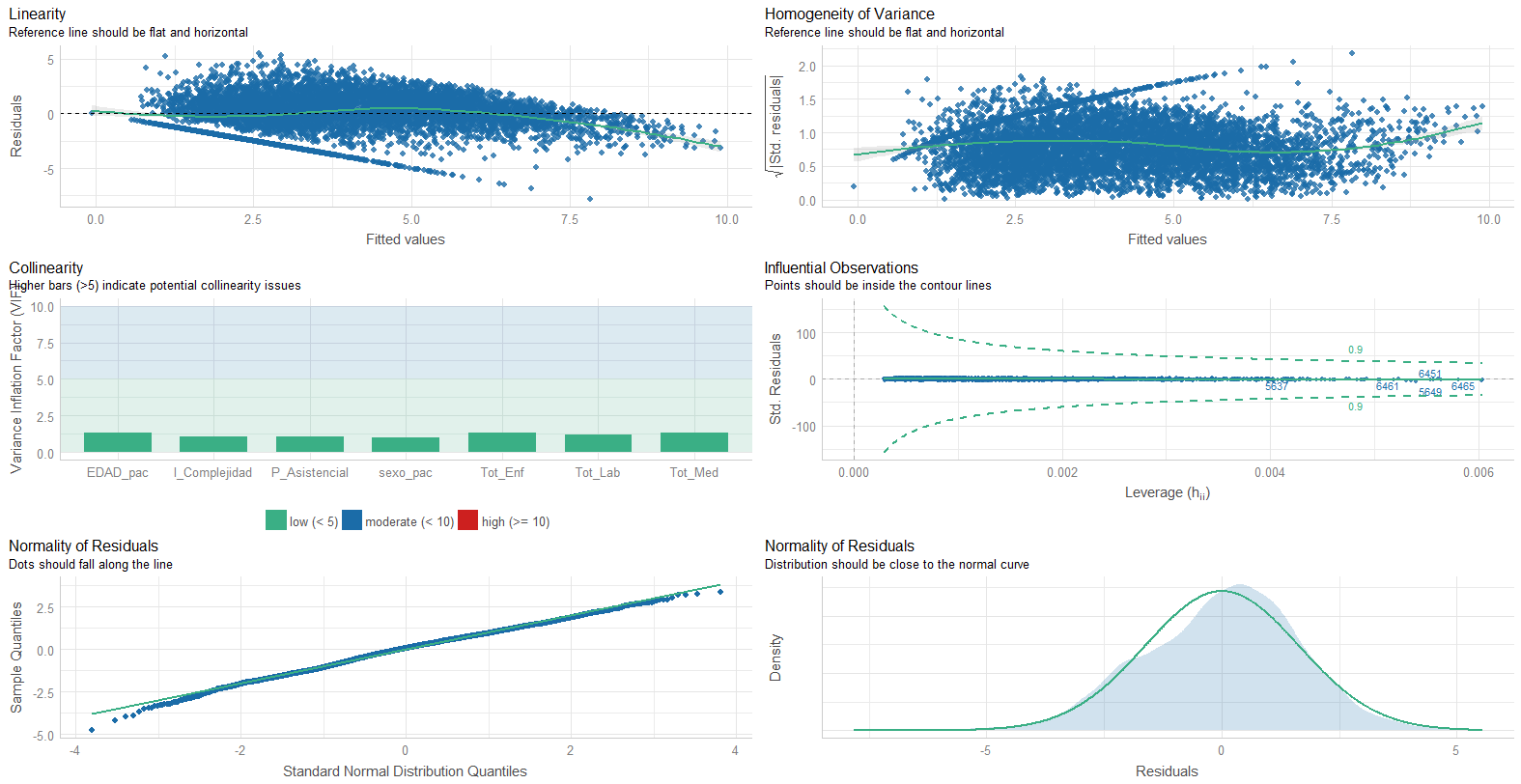


Figura 4.14: Hipótesis paramétricas. Regresión lineal

1. **Linearidad:** el gráfico titulado *linearity* muestra las predicciones del modelo (eje x) frente a los residuos (eje Y) para calibrar la linearidad entre *input* y *output*. Bajo esta premisa, se trata de observar que de forma significativa no exista una tendencia no lineal en el comportamiento de los residuos. Es decir, la línea verde debe ser plana y horizontal respecto a la línea punteada trazada sobre los datos. En nuestro caso podemos comprobar que esto se cumple en el primer tramo de las predicciones, pero se curva para valores más elevados. No obstante, consideramos que el comportamiento agregado de los residuos se asemeja en una medida aceptable a una línea recta.
2. **Homocedasticidad:** el gráfico titulado *homogeneity of variance* muestra la raíz cuadrada del valor absoluto de los residuos estandarizados frente a las predicciones. Para que se cumpla esta hipótesis de Homocedasticidad, la línea verde de referencia debe ser plana y horizontal. En nuestro caso comprobamos un efecto similar al caso anterior, donde la línea de referencia se mantiene plana y horizontal salvo en los últimos tramos de la estimación, que presenta una ligera curvatura.
3. **Normalidad en lo residuos:** los gráficos *normality of residuals* aportan evidencias acerca de la hipótesis de normalidad de los residuos. El primero de ellos se corresponde el *QQ-plot*, y grafica los percentiles teóricos de una normal frente a los percentiles empíricos de los residuos estandarizados. Bajo este principio, los puntos de la gráfica deben encontrarse distribuidos a lo largo de la diagonal verde de referencia. En nuestro caso observamos que los residuos son prácticamente normales. Esto se puede ver gráficamente en el segundo gráfico de normalidad *normality of residuals* donde la distribución es muy similar a la curva normal, salvo por la presencia de la cola izquierda ligeramente más larga.
4. **Colinealidad:** el gráfico titulado *colinearity* muestra la existencia de posible colinealidad en nuestros datos. La eliminación de la variable relacionada con el número total de citas que era combinación lineal respecto a las citas en las diferentes consultas, ha permitido eliminar este posible problema.
5. **Valores influyentes:** el gráfico titulado *influential observations* muestra el impacto atípico que puedan tener algunas observaciones, ya sea debido a un valor atípico de la variable respuesta que pueda distorsionar la varianza residual, o bien debido a algún valor atípico de alguna variable independiente. En el gráfico podemos ver que todos los puntos se encuentran distribuidos dentro de la banda establecida, lo que nos muestra que la depuración de *outliers* y la tranformación logarítmica realizada en apartados previos sobre la variable objetivo han funcionado correctamente para asegurar la robustez del modelo.

### 4.4.2.- Árboles de regresión

Los árboles de regresión tratan de hallar puntos de corte en las variables independientes que lleven a grupos de individuos con comportamiento homogéneo (análisis clúster) con el objetivo de evaluar el comportamiento de la variable respuesta (análisis predictivo). Por ello, se introducirán todas las variables de nuestro conjunto de datos depurado, de manera que sea el propio árbol aquel que vaya decidiendo qué variables tienen relevancia suficiente para entrar en cada división.

**Parámetros a optimizar**

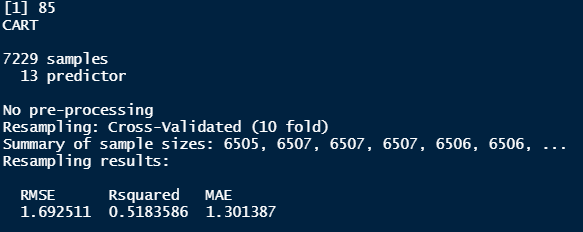
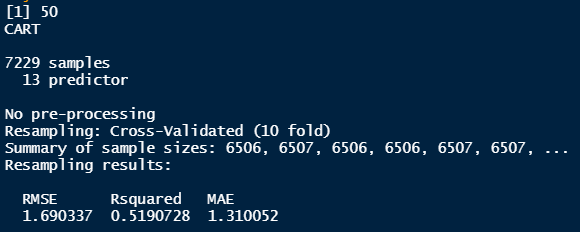
* **Criterio de punto de corte óptimo**: la modelización de los árboles se llevará a cabo haciendo uso de la librería *rpart* de *R-Studio*. En esta librería se especificará para variables continuas el criterio de varianza. Es decir, se calculará la variabilidad de la variable dependiente en cada grupo y se realizará el sumatorio de los valores obtenidos, eligiendo de esta forma la división que construye grupos más homogéneos internamente y diferentes entre sí.
* **Tamaño del nodo final (*minbucket)***: una vez definido el punto de corte sobre los que se realizarán las divisiones en cada nodo, es necesario determinar el criterio de parada adecuado para seleccionar la variable independiente o nodo que va a crear en la última división, ya que en caso contrario, el modelo final puede estar sobreajustado si los datos son complejos, o bien puede ser demasiado simple e incapaz de captar el comportamiento de la variable objetivo de manera precisa.

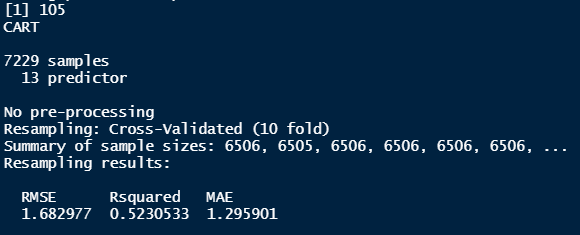
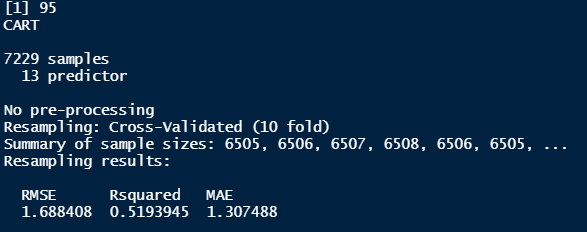
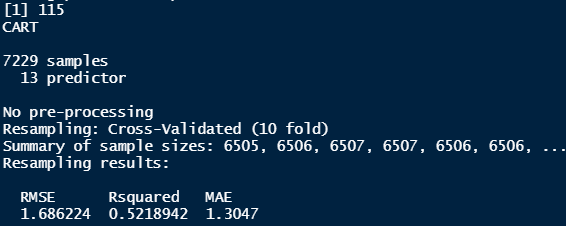
En este sentido, será necesario determinar el número de observaciones mínimo que debe tener un nodo para poder ser dividido. Cuantas menos observaciones tengamos en la hoja final, la complejidad del árbol se incrementará y la predicción podrá variar mucho al incorporar datos de los que no ha aprendido el modelo inicial (alta varianza). No obstante, si el número de observaciones es muy alto los árboles serán más sencillos y pueden tener poca potencia predictiva (alto sesgo), pero serán más robustos ante la incorporación de nuevos datos.

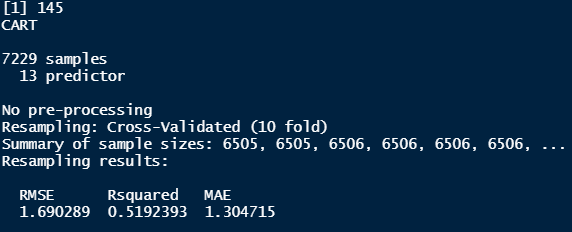
* **Importancia de variables (*masurrogate*)**: Por último, debido a que los árboles generalmente no utilizan todas las variables, sino que van incorporando de forma progresiva aquellas que tienen un mayor impacto sobre la variable respuesta. Para este caso, dado que no tenemos valores ausentes, fijaremos un parámetro extra para que solo nos presente la importancia de las variables que efectivamente participen en el modelo.

En función de estos parámetros se ha especificado un bucle en *R-Studio* que realice 26 árboles de regresión variando el valor del *minbucket* según el criterio de corte óptimo la varianza, por validación cruzada simple*.* Dado que disponemos de 7229 observaciones, se ha establecido el mínimo de observaciones en el nodo final en 50 observaciones, e irá incrementándose de manera progresiva en 5 hasta llegar a un máximo de 180.

De manera ascendente se han seleccionado los siguientes árboles cuyo *minbucket* minimiza el error de predicción (RMSE<1.695).







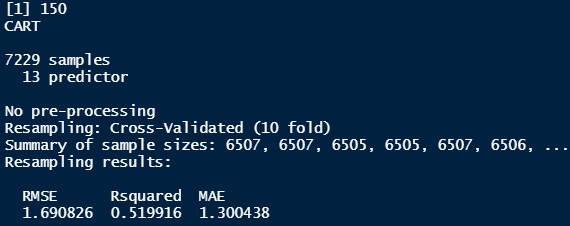


Figura 4.15: Minbucket óptimo. Árboles de regresión

A partir de las configuraciones óptimas obtenidas en la Figura 4.15,se construirán varios conjuntos de árboles en función del valor del *minbucket* utilizando validación cruzada repetida y representando la distribución de los errores obtenidos en un diagrama de cajas con el objetivo de identificar el sesgo y varianza de los distintos conjuntos.

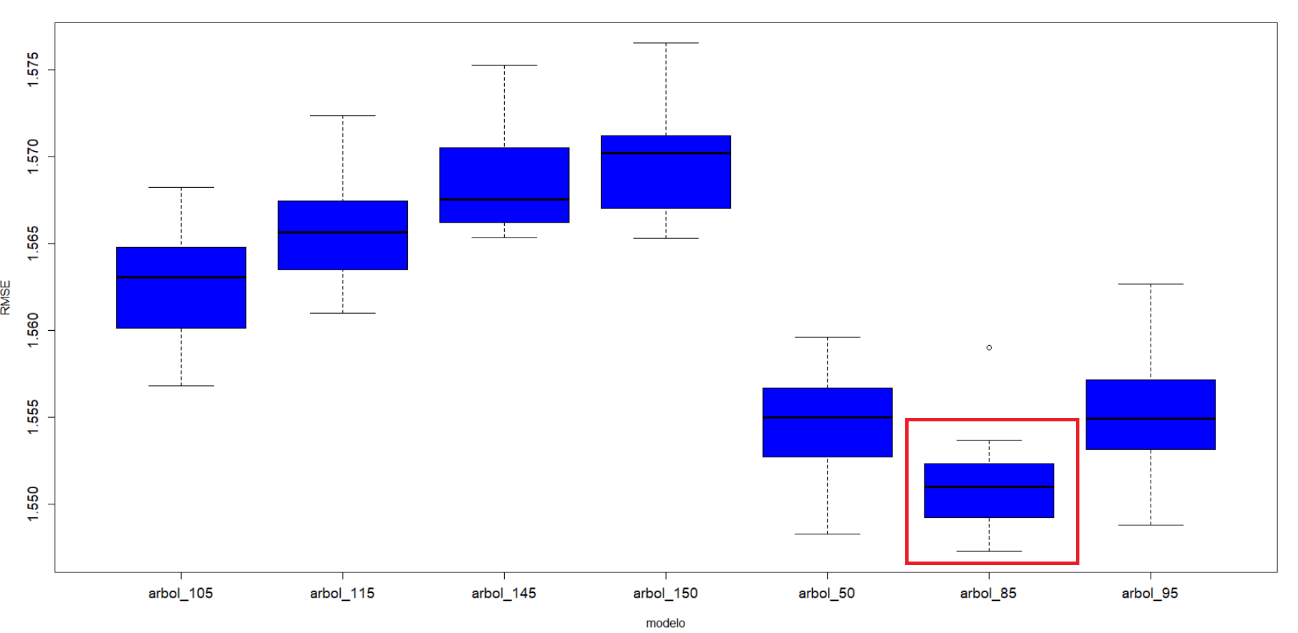


Figura 4.16: Árboles de regresión. Validación cruzada repetida

En la Figura 4.16 se observa que los árboles con *minbucket* más bajo funcionan mejor para nuestro conjunto de datos. Concretamente el árbol con *minbucket* = 85 presenta el valor más reducido tanto de varianza como de sesgo (altura de la caja).

Una de las ventajas de los árboles es su alta capacidad de interpretabilidad, ya que nos permite identificar qué variables tienen un mayor peso a la hora de explicar la variable objetivo y en qué cuantía la explican. Por lo tanto, a partir de este árbol obtenemos la representación gráfica de los diferentes nodos y la importancia de las variables que se han ido incorporando en cada división.

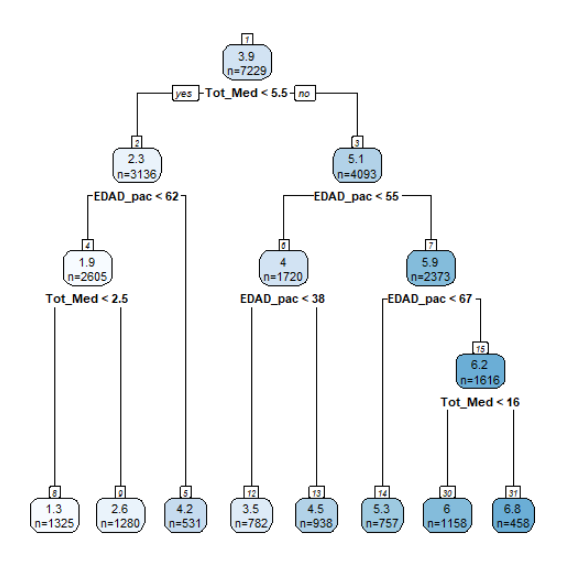


Figura 4.17: Representación simplificada del árbol de regresión

En la Figura 4.17se obtiene la representación gráfica de un árbol simplificado fijando un valor de *minbuket* superior, con el objetivo de poder realizar interpretaciones útiles en los apartados posteriores. Es importante destacar que el valor del gasto sanitario está expresado en escala logarítmica, por lo que se deberá aplicar la exponencial sobre estos valores para obtener conclusiones útiles.

Por último, identificamos qué variables han sido las más relevantes a la hora de establecer los diferentes criterios de división.

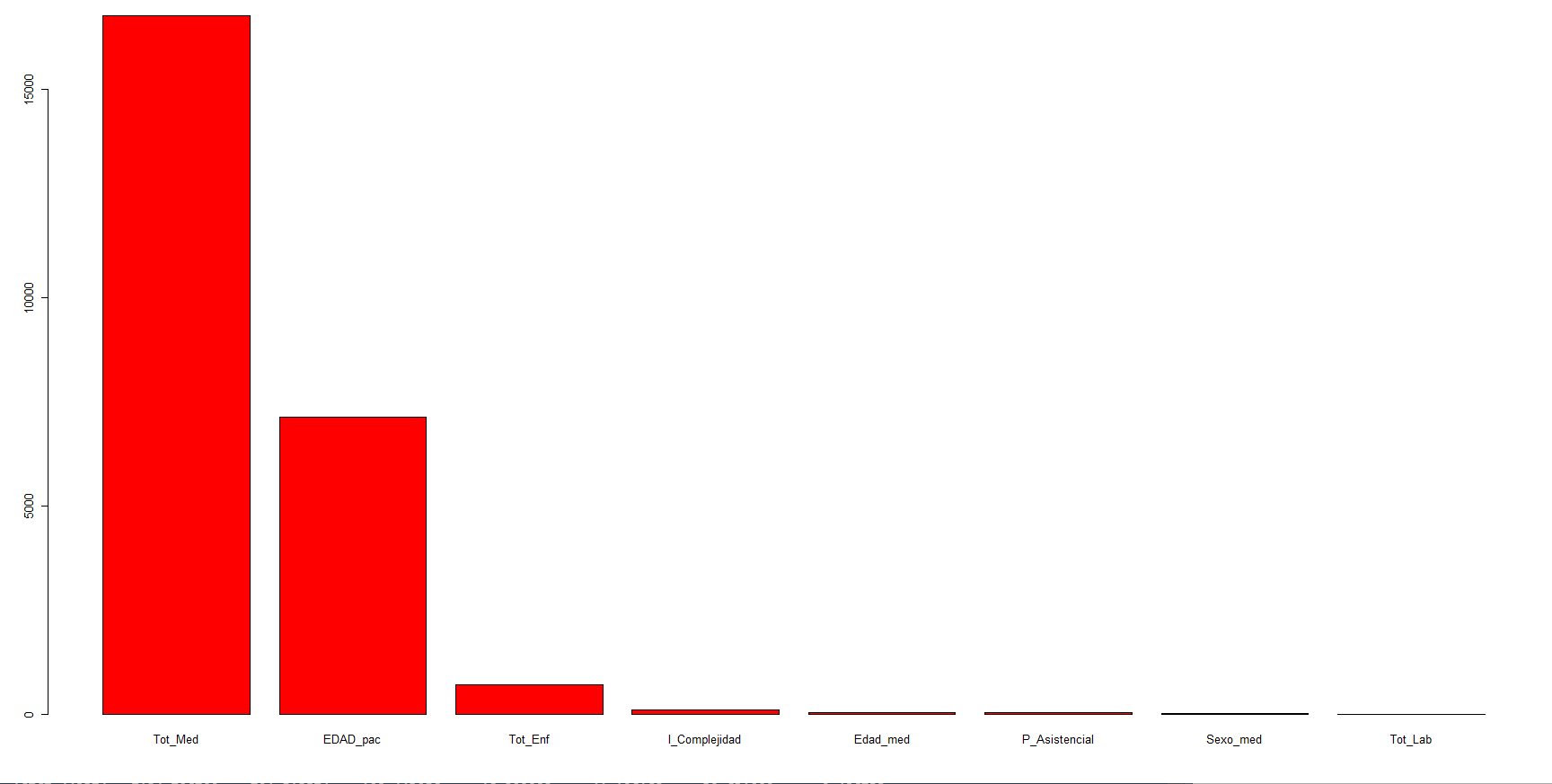


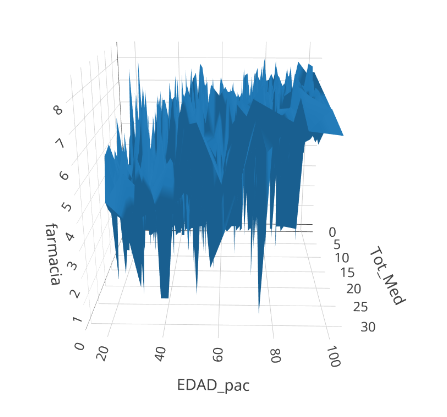
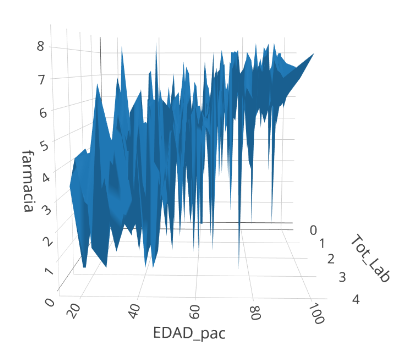
Figura 4.18: Importancia de las variables. Árbol de regresión

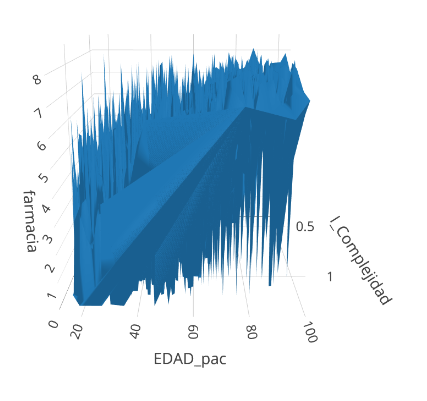
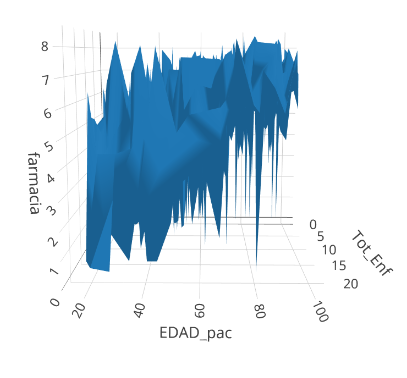
Como se puede observar en el gráfico, de manera similar a los métodos de selección de variables, los predictores que presentan una mayor ganancia en términos del criterio de corte óptimo son la edad y el número de citas en las respectivas consultas (a excepción de las citas en la consulta de laboratorio que ha adquirido una menor relevancia). No obstante, los árboles reportan mayor importancia a las variables relacionadas con el médico, concretamente llama la atención el peso relativamente superior de la edad y el sexo del médico a la hora de determinar el gasto sanitario.

### 4.4.3.- Redes neuronales

Las redes neuronales representan un tipo de modelización especialmente útil cuando se desconoce las relaciones de los inputs de nuestra base de datos para explicar la variable respuesta. Esto ocurre debido a que las relaciones entre las variables del modelo pueden no ser lineales, o bien existen patrones desconocidos en la distribución de los datos.

Por esta razón, antes de comenzar a definir los parámetros que conformarán la red de nuestro modelo, es necesario identificar qué relación existe entre las principales variables que explican el gasto en farmacia. Para el desarrollo de redes neuronales se utilizará el *set* de variables obtenido por el consenso de 4/7 métodos de selección de variables (*set amarillo)* especificado en el punto 4.3.





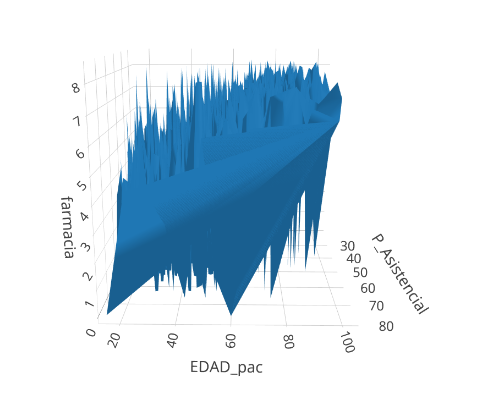


Figura 4.19: Distribución las principales variables input

En la figura 4.19 se observa que la distribución de las principales variables input es bastante asimétrica y distinta a un plano, lo que justifica utilizar el modelo de red neuronal para captar este efecto no lineal.

Antes de comenzar con la modelización es necesario realizar un tratamiento previo de nuestros datos. En primer lugar se ha agrupado el conjunto de *inputs* en una lista y se han creado dos vectores con los valores de las medias y de las desviaciones típicas de las variables continuas. A continuación para escalar las variables continuas, le restamos a cada una de ellas la media y lo dividimos por su desviación típica. Por último volvemos a agrupar los dos conjuntos de variables. Esta estandarización permite que nuestras variables continuas tengan media 0 y desviación típica 1 y pueden tomar valores negativos comprendidos en la recta real entre (-3,3), con el objetivo de controlar en cierta medida el comportamiento de las redes elaboradas.

Una vez finalizada la preparación de los datos previa a la construcción de redes, se definen los parámetros que conformarán estos modelos.

**Parámetros a optimizar**

* **Nodos de la capa oculta:** el primer parámetro a optimizar se corresponde con número de nodos en la capa oculta. Este valor se determinará a través de la siguiente expresión:

*h(k + 1) + h + 1 = obs / p*

Donde h es el número de nodos ocultos de la red, k es el número de nodos *input* y p es el número de parámetros. El valor del parámetro h viene muy relacionado con la configuración de nuestros datos, ya que si la combinación de los predictores es muy compleja se requerirán más nodos para reducir el sesgo, mientras que si los datos son relativamente más simples, demasiados nodos pueden provocar sobreajuste.

En este sentido, ya que disponemos de 7229 observaciones, utilizando el set de variables compuesto por 7 predictores, se ha decido que cada *input* disponga de al menos 30 observaciones por parámetro, obteniéndose la siguiente expresión:

h(7 + 1) + h + 1 = 7229 / 30 h = 26.662

Por lo tanto, se elaborarán redes neuronales con un número de nodos de la capa oculta (h) comprendido entre 5 y 26 nodos.

* ***Learning rate:*** el valor del *Learning rate* determina en qué medida van a cambiar los pesos *wij* en cada iteración. Respecto al nivel del learning rate se establecerá una combinación entre valores bajos (0.0001) y algo más elevados (0.1) con el objetivo de que los pesos se adapten de manera adecuada a la red y que converjan en la medida de lo posible a un mínimo global.
* **Número de iteraciones:** un número elevado de iteraciones puede afectar en algunos casos muy negativamente a la redajustando demasiado bien los pesos y provocando una mala generalización al aplicar el modelo sobre un conjunto de datos sobre el que no ha aprendido en primer lugar. Por lo tanto, el número de iteraciones que realizará la red neuronal vendrá determinada por la técnica *early stopping.* En este sentido se fijará el número mínimo de iteraciones hasta que el error sobre datos *test* se “estanque”, y comience a sobrejustar.
* **Algoritmo de optimización:**El algoritmo de optimización utilizado en el paquete se basará en el Algoritmo básico Gradient Descent. El objetivo es hallar los pesos wij que minimicen el error (RMSE).

A partir de estos parámetros se ha especificado un bucle en *R-Studio* en el que se comprobarán por validación cruzada simple 5 nodos en la capa oculta (5,10,15,20,26) con 4 tipos de *learning rate* (0.1,0.01,0.001,0.0001) con las siguientes iteraciones (10,50,100,250,500,1000,2000,3000) dando lugar a 160 redes neuronales.

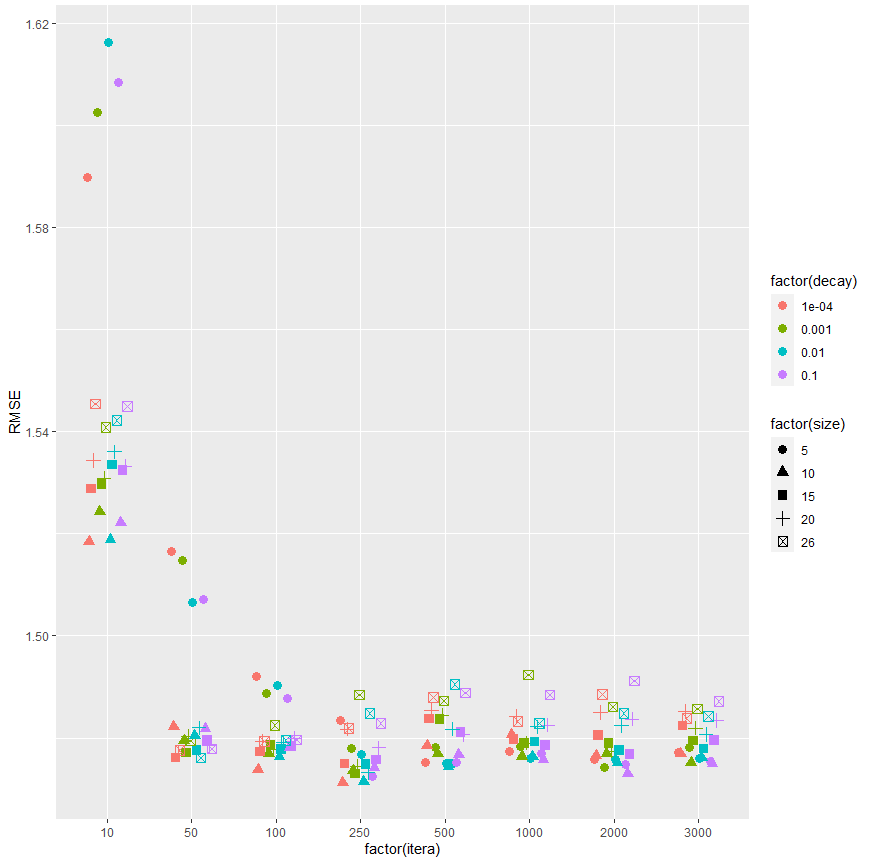


Figura 4.20: Optimización de parámetros por early stopping. Redes neuronales

Atendiendo a los resultados de la figura 4.20 se puede observar que a medida que se incrementa el número de nodos el error estimado disminuye hasta los 15 nodos, donde sobre los datos *test,* el RMSE se estanca *y las* redes más profundas empiezan a sobreajustar. Esto puede ser un indicativo de que nuestros datos no son excesivamente complejos, lo que hace que se requieran un menor número de nodos.

Respecto al número de iteraciones, observamos que el error disminuye de forma constante hasta las 250 iteraciones, punto a partir del cual el error en la gran mayoría de los escenarios se vuelve a incrementar, mostrando cierta tendencia al sobreajuste. Por lo tanto, consideraremos hasta un máximo de 500 iteraciones como valor óptimo para tratar de generar un equilibrio entre el tiempo computacional y el número adecuado de iteraciones para que los pesos converjan.

Por último en relación con el nivel del *learning rate*, parece alcanzarse un mínimo cercano al global cuando el este parámetro es igual a 0.0001. No obstante, dado que las diferencias no son claras se probarán por validación cruzada los diferentes valores con el objetivo de que los pesos se adapten de manera adecuada a la red y que converjan en la medida de lo posible a un mínimo global.

La inicialización aleatoria de los pesos puede tener gran influencia en los resultados, cambiando mucho los parámetros finales de la red, si ésta no está bien calibrada. En este sentido, se utilizará otra semilla por validación cruzada simple acotando el número de iteraciones en (250), utilizando nodos entre (0-15) y los mismos valores del *learning rate*, pero esta vez se obtiene el modelo óptimo (donde se acerca a conseguir el mínimo global) de forma analítica.



Los resultados obtenidos son coherentes con la Figura 6.8, por lo que la red parece estar bien especificada. Las redes que presenten un RMSE < 1.48 serán representadas por validación cruzada repetida a través de un *boxplot*, con el objetivo de evaluar el sesgo y varianza.

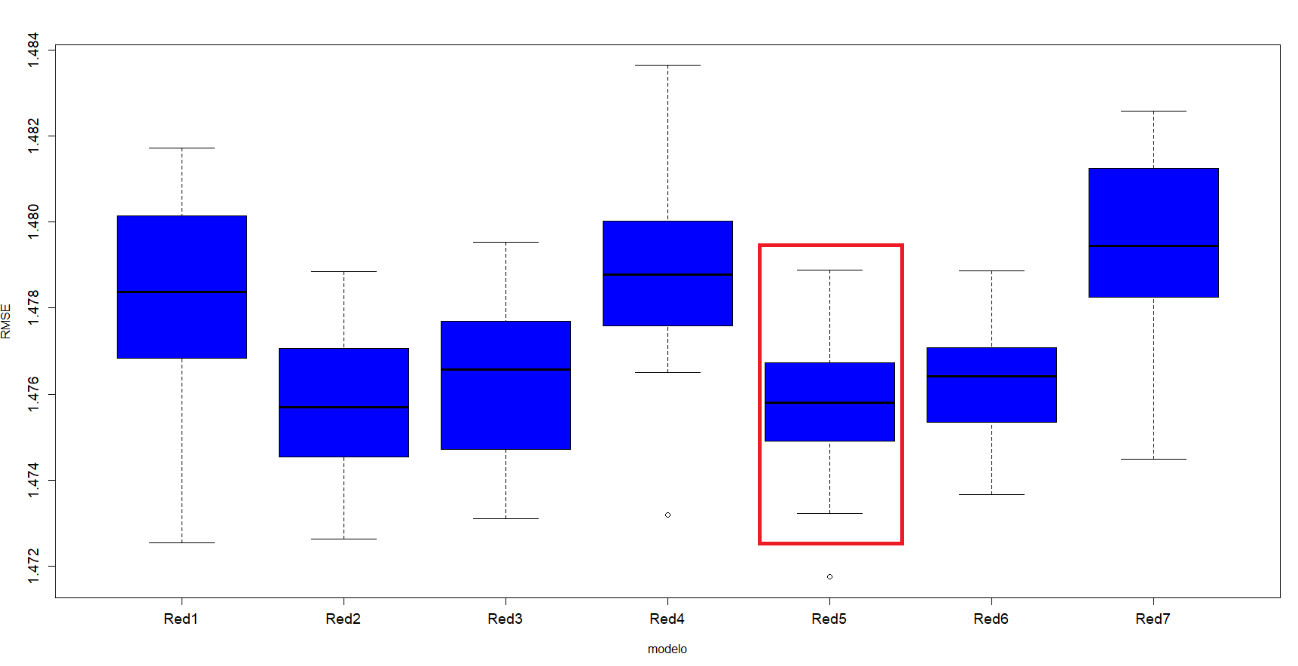


Figura 4.21: Redes neuronales. Validación cruzada repetida

En figura 4.21 se ha obtenido la distribución del error (RMSE) y la variabilidad de todos los set de variables para redes neuronales. Se puede observar que el sesgo en todos los casos es parecido, salvo para el caso cuando el número de iteraciones = 500 (red7), que se produce un ligero sobreajuste que aumenta el sesgo y la varianza. La red5 compuesta por 10 nodos en la capa oculta, 250 iteraciones y *learning rate = 0.0001* se ha seleccionado como mejor modelo de red ya que presenta un sesgo parecido a la red2 pero con varianza más reducida.

### 4.4.4.- Bagging

El método bagging consiste en dividir nuestro conjunto de datos en diferentes submuestras aleatorias con reemplazamiento de los datos sobre las que se aplicarán diferentes modelos de árboles, para luego combinar los resultados de las predicciones.

**Parámetros a optimizar**

* ***Mtry***: tomará valores comprendidos dependiendo del número de variables independientes que se introducen durante la modelización.
* ***Ntree***: muestra el número de árboles que vamos a aplicar sobre los subconjuntos de datos. En principio se fijará un máximo de 1000 árboles con el objetivo de identificar el punto a partir del cual el error se reduce hasta permanecer constante (en caso de que el error siguiera decreciendo significativamente, se incrementaría este valor).
* ***Sampsize***: mide el tamaño de cada muestra Bagging. Es importante señalar que el tamaño máximo de cada muestra de Bagging al realizar validación cruzada con n-fold=10, debe ser igual o inferior al tamaño de la muestra. Es decir, dado que se utilizarán 9/10\*7229 = 6505 observaciones para entrenar nuestro modelo, el sampsize no deberá ser superior a este valor.
* ***Nodesize***: es un parámetro que mide el tamaño máximo de nodos finales (complejidad de los árboles). Dado que este parámetro es equivalente al minbucket del modelo de árboles anterior, se mantienen las consideraciones al respecto y se fijan valores próximos al obtenido en el caso de árboles. Recordemos que el valor de este parámetro debía de ajustarse en función de un objetivo doble, de manera que se obtuviese el error más reducido posible sin llegar a sobreajustar, y que tuviera un tamaño suficiente que garantizase la representatividad futura de nuestros datos. Dado que el modelo Bagging es una combinación de árboles de decisión, se ha considerado lógico que dicha combinación se realice a partir del minbucket del árbol óptimo del apartado anterior.
* ***Replace=TRUE***: las muestras realizadas se llevan a cabo con reemplazamiento

En primer lugar, se ha especificado un bucle en *R-Studio* por validación cruzada simple, para controlar el número de árboles óptimo que vamos a realizar en nuestro modelo Bagging. En este sentido, se han ejecutado 1000 árboles sobre el conjunto de datos depurado compuesto por 13 predictores (*mtry* = 13), se ha empleado la configuración del árbol óptimo (*nodesize* = 85) obtenido en el apartado 4.4.2, y en última instancia se ha tratado como constante tamaño de la muestra de bagging. Se obtienen los siguientes resultados:

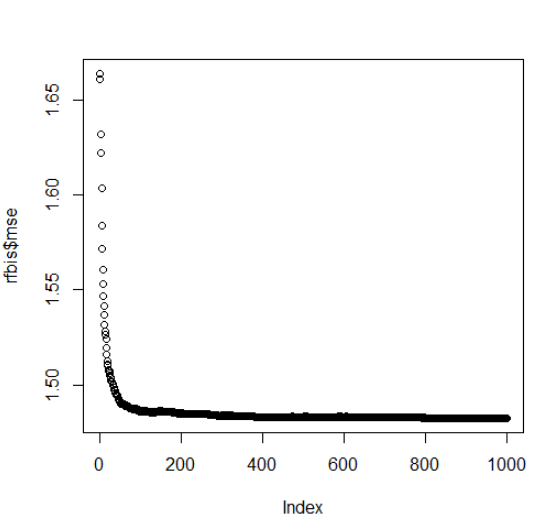


Figura 4.22: Número de árboles óptimo (nodesize =85). Bagging

En la Figura 4.22 se observa que el error se reduce progresivamente hasta los 100-200 árboles de regresión, hasta que se estanca y se mantiene estable independientemente del valor de *ntree*. Para ver más claramente el punto donde se obtiene un error más reducido, se realiza este mismo bucle con tamaños de los nodos finales superiores e inferiores al obtenido por el árbol de regresión óptimo.

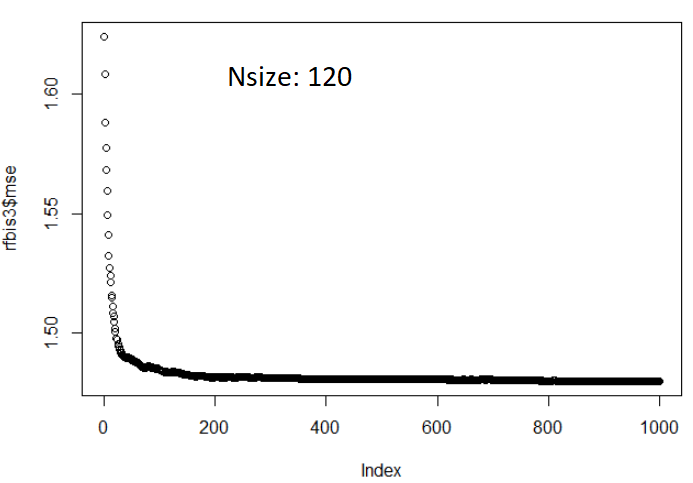
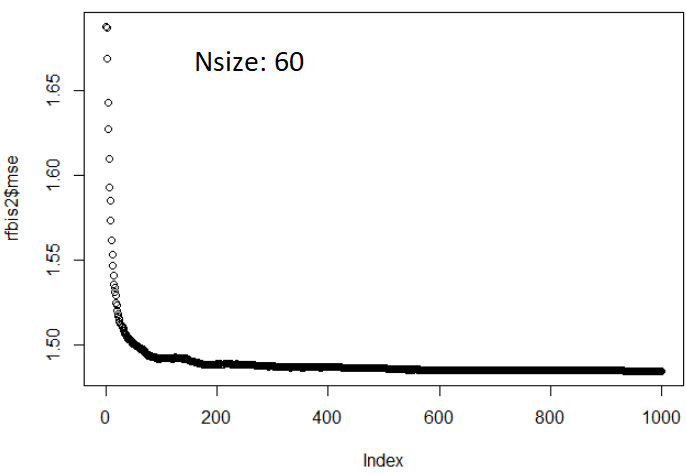


Figura 4.23: Optimización del tamaño del nodo final. Bagging

En ambos casos se puede observar que el punto donde el error es más reducido antes de estabilizarse, se sitúa alrededor de 130 árboles. En relación al tamaño del nodo final, se observa que sobre los datos de entrenamiento, tamaños superiores tienden a reducir ligeramente el valor del RMSE, hasta llegar un punto donde se estaca (alrededor de nodesize = 85). Por lo tanto, se ha decidido que el tamaño del nodo final = 85 y ntrees= 130, representa un tamaño adecuado para realizar futuras comparaciones con el modelo de árboles de regresión. Finalmente se han fijado estos parámetros, para realizar la comprobación final a través de validación cruzada repetida variando el valor de *sampsize,* hasta encontrar el Bagging que ofrezca tanto un sesgo como una varianza menor.

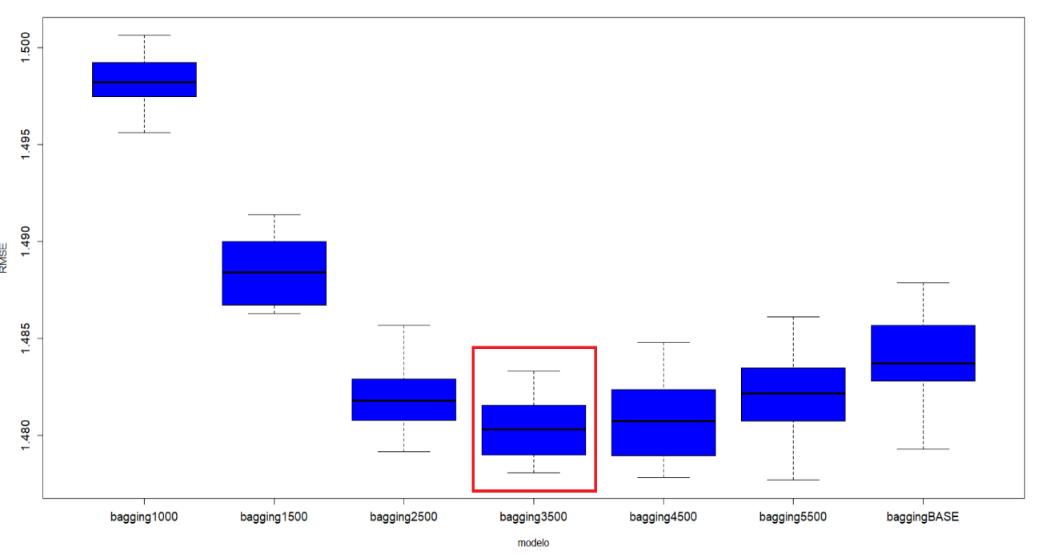


Figura 4.24: Bagging variando el sampsize. Validación cruzada repetida

En la Figura 4.24 se observa que a medida que se incrementa el tamaño de cada muestra de Bagging el error disminuye. No obstante, se puede ver claramente que a partir del Bagging con *sampsize = 3500,* se comienza a producir un ligero sobreajuste sobre los datos *test* incrementándose tanto el sesgo como la varianza (altura de la caja).

Por lo tanto, los parámetros del modelo Bagging óptimo son los siguientes: *Mtry*=13, *Ntree*=130, *Sampsize*=3500, *Nodesize*=85 con reemplazamiento.

### 4.4.5.- Random Forest

Como se ha comentado previamente, el Random Forest es una especificación del modelo Bagging pero incorporando remuestreo de variables. Por lo tanto, se incorporarán los parámetros óptimos relacionados con el número de árboles, tamaño de los nodos finales, tamaño de la muestra de Bagging con reemplazamiento obtenidos en el apartado 4.4.4.

**Parámetros a optimizar**

* ***Mtry*:** muestra el número de variables independientes que vamos a introducir durante la modelización. Dado que uno de los principales objetivos del Random Forest consiste en realizar remuestreo de variables, los predictores de nuestro conjunto depurado se irán introduciendo de manera aleatoria, obteniendo de esta forma la importancia de las diferentes combinaciones de variables.

De manera análoga al modelo *Bagging,* se han probado diferentes valores del parámetro *sampsize,* tamaño del nodo final y número de árboles. No se ha obtenido ninguna mejora significativa, por lo que con el objetivo de realizar comparaciones respecto al modelo anterior, se ha decidido utilizar la configuración óptima del Bagging.

A continuación se ha realizado en *R-Studio* un bucle por validación cruzada simple que contempla la configuración del modelo óptimo de Bagging e incorpora las diferentes combinaciones de variables en un rango comprendido entre (0-13), sobre las que se construirán los diferentes árboles. Se obtienen los siguientes resultados:

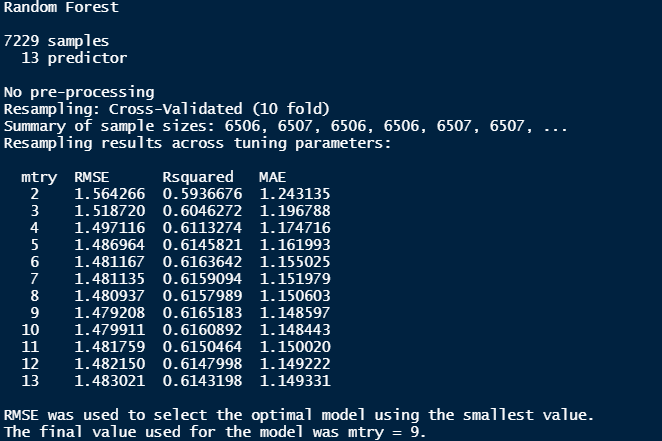


Figura 4.25: Remuestreo de variables. Random Forest

Los conjuntos de variables correspondientes con un valor del *mtry* comprendido entre (6-12) ofrecen el menor valor de RMSE (por debajo del modelo óptimo de *Bagging*), siendo el valor de mtry=9 el valor aparentemente óptimo por validación cruzada simple. Para el caso aparentemente óptimo de mtry=9, esto significa que en cada árbol creado sortearíamos en cada nodo 9 variables de las 13 variables *input* candidatas que tiene el archivo depurado, y de esas 9 se elegiría la mejor. Este proceso se repetiría hasta llegar al nodo final (cuyo tamaño máximo serán 85 observaciones).

Una vez se han obtenido obtenido los valores óptimos del parámetro *mtry* que reducen el error respecto al *Bagging*, se realizarán 7 modelos por validación cruzada repetida con *nfolds*=10 y 20 repeticiones (semilla1234). Con el objetivo de obtener gráficamente a través de un *boxplot* el modelo con menor sesgo y varianza posible.

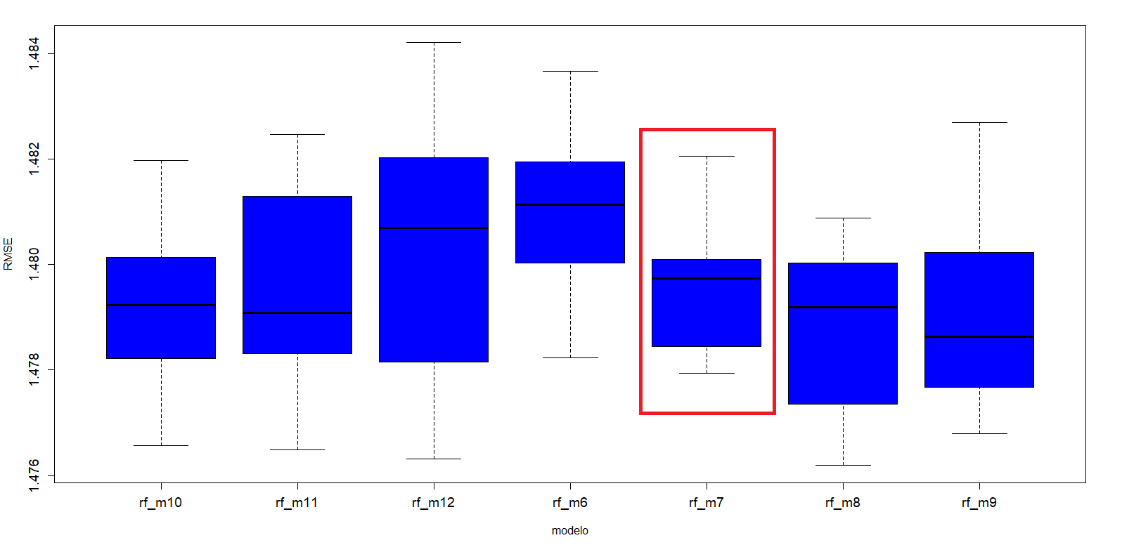


Figura 4.26: Random Forest. Validación cruzada repetida

Como se puede ver en el gráfico, todos los modelos han mejorado ligeramente el sesgo y varianza respecto Bagging óptimo. No obstante, el modelo *rf\_m7* presenta un sesgo ligeramente superior al *rf\_m9 (*modelo óptimo obtenido por validación cruzada simple), pero dado que para el remuestreo de variables se han utilizado un menor número de *inputs*, la varianza (altura de la caja) es significativamente menor. Por lo tanto, el mejor modelo seleccionado ha sido el *Random Forest* con mtry=7. Esto implica que que en cada árbol creado, sortearíamos en cada nodo 7 variables de las 13 variables *input* candidatas que tiene el archivo, y de esas 7 se elegiría la mejor.

Por último, se ha extraído la puntuación de cada variable utilizada para la construcción del modelo de Random Forest, en función de las veces que se ha repetido en cada división hasta llegar al nodo final, y se han agrupado en un *dataframe* para construir el siguiente gráfico de barras.

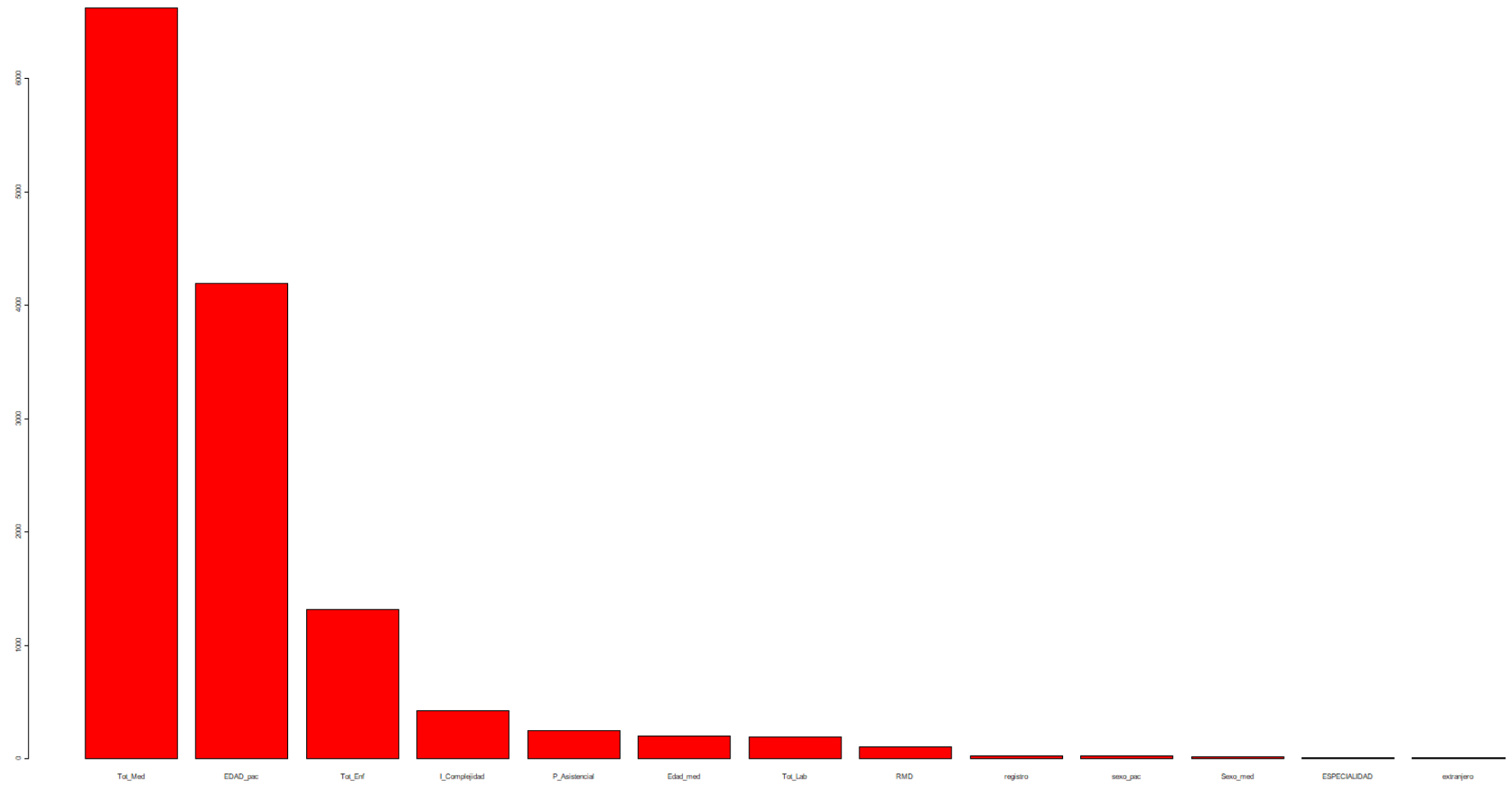


Figura 4.27: Importancia de las variables. Random Forest

Atendiendo al siguiente gráfico de barras podemos observar como las variables que reciben mayor importancia (edad y número de citas en la consulta del médico) coinciden con aquellas que presentaban una mayor ganancia en función del criterio del punto de corte óptimo para el caso de árboles de regresión individuales, algo lógico dado que para la construcción del Random Forest se han realizado cientos de árboles de regresión. Sin embargo, a pesar de que el comportamiento de las variables con mayor relevancia es muy similar a los árboles, para este caso algunas variables relacionadas con el conjunto del médico o del municipio, parecen adquirir mayor relevancia.

### 4.4.6.- Gradient Boosting

Como se ha comentado en apartados anteriores, la idea del *Gradient Boosting* consiste en ajustar de manera secuencial varios modelos de árboles. En cada ajuste se utiliza información del modelo anterior para aprender de sus errores y mejorar el resultado de cada iteración.

**Parámetros a optimizar:**

* **La constante de regularización v (*shrinkage*):** el valor de este parámetro determina cómo de rápido convergerá el modelo de *Gradient Boosting* al punto de corte óptimo. De manera que cuanto más alto sea este valor, más rápido convergerá al punto de corte óptimo, y más alto será el sesgo del modelo. Por otro lado, si se ajusta un valor demasiado bajo, serán necesarias muchas iteraciones para que este converja.
* **Número de iteraciones (*ntrees*):** este valor se corresponde con el número de árboles de regresión que se realizarán para para que el modelo pueda aprender de sus errores y mejorar el resultado de cada iteración. Es importante controlar el número de árboles realizados, ya que existe un punto donde la mejora del modelo sobre el conjunto de test se estanca y comienza a sobreajustar.
* **Tamaño del nodo final (*n.minosbsnode*):** este parámetro mide el tamaño máximo de los nodos finales (complejidad de los árboles), y se corresponde con el minbucket en el caso de los árboles, por lo que se ajustará en un rango entre (50-200) teniendo en cuenta los resultados del modelo de árbol de regresión óptimo inicial.
* ***Interaction.depth*** = 2: se corresponde con el número de divisiones máxima década nodo. Por defecto se dejará en 2, árboles binarios.

De manera análoga al resto de modelos, a través de la librería *caret* se ha especificado un bucle por validación cruzada simple con 10 grupos que recoja los parámetros especificados anteriormente. En primer lugar introducen los valores del shrinkage comprendidos entre (0.0001-0.2) con el objetivo de observar los diferentes momentos donde el punto de corte converge. Seguidamente se introducen los diferentes tamaños del nodo final (50-200). Por último se introducen las iteraciones y se especifican las divisiones como binarias. Se obtienen los siguientes resultados:

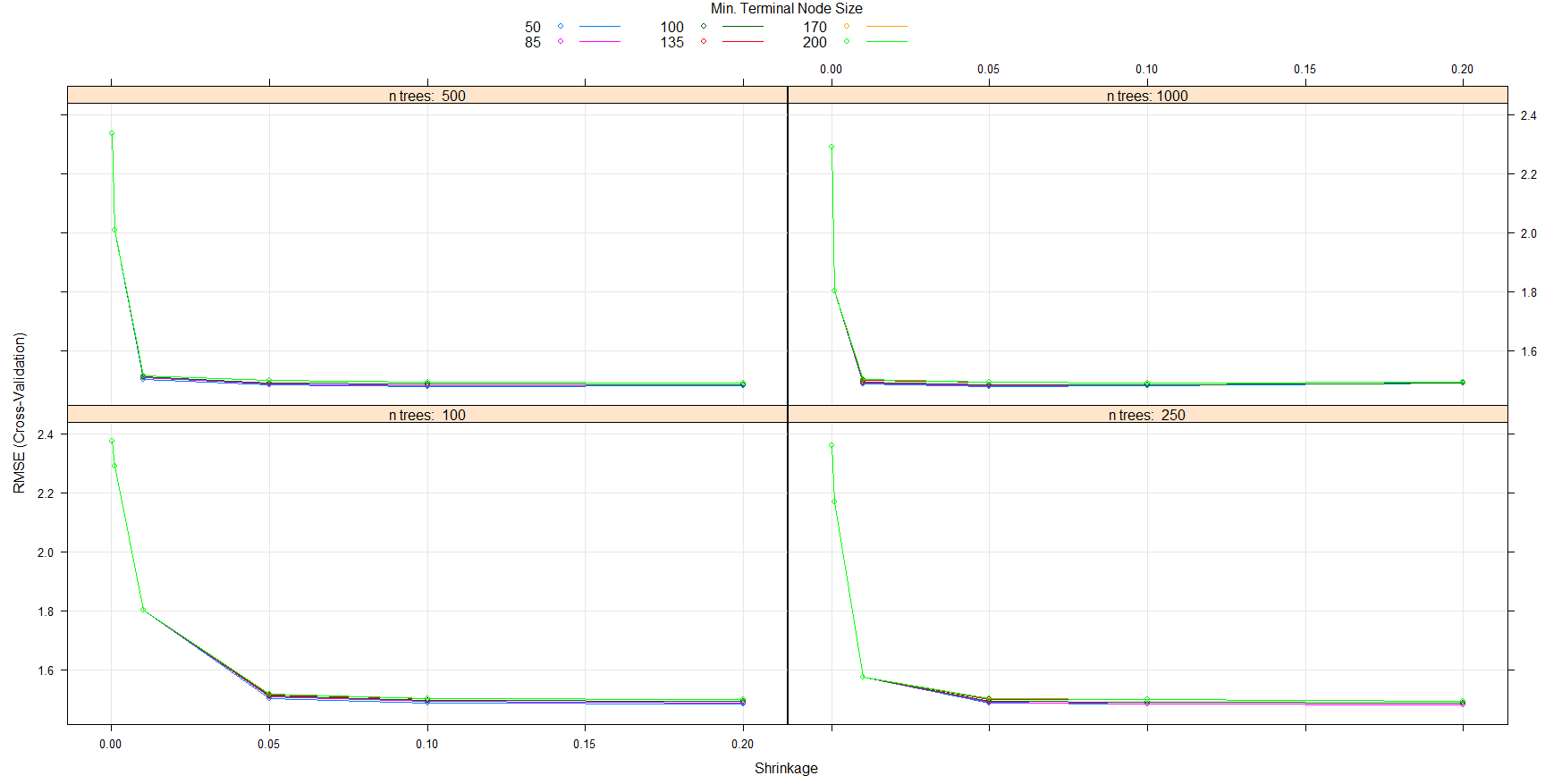


Figura 4.28: Parámetros óptimos. Gradient Boosting

En la Figura 4.28 se puede observar que la diferencia del error (RMSE) entre el número de observaciones en el nodo final es muy reducida salvo para *n.minosbsnode*=200. A su vez, se observa que para obtener el mínimo error cuando se realizan muchas iteraciones (*ntrees*=1000) el valor de la constante de regularización es muy baja (0.001). Por otro lado, cuando el número de iteraciones es muy bajo, se llega a un punto de corte óptimo cuando el *shrinkage* es relativamente más alto (0.1), siendo el modelo aparentemente óptimo con semilla (1234) aquel con *n.trees* = 500, *interaction.depth* = 2, *shrinkage* = 0.1 and *n.minobsinnode* = 50.

Las pruebas realizadas anteriormente, se han realizado sobre un rango de árboles muy distinto entre sí, por lo que es posible que no podamos obtener conclusiones demasiado precisas sobre el sobreajuste relacionado con el número de árboles. En este sentido, utilizando la técnica de *early-stopping*  se ha decidido tomar los parámetros del *Gradient Boosting* óptimo del apartado anterior como constantes y someterlo a un nuevo rango de valores de *ntrees* (100,150,200,250,300,350,400,450,500,550,600,650,800, 1000,1500,2000) más preciso.

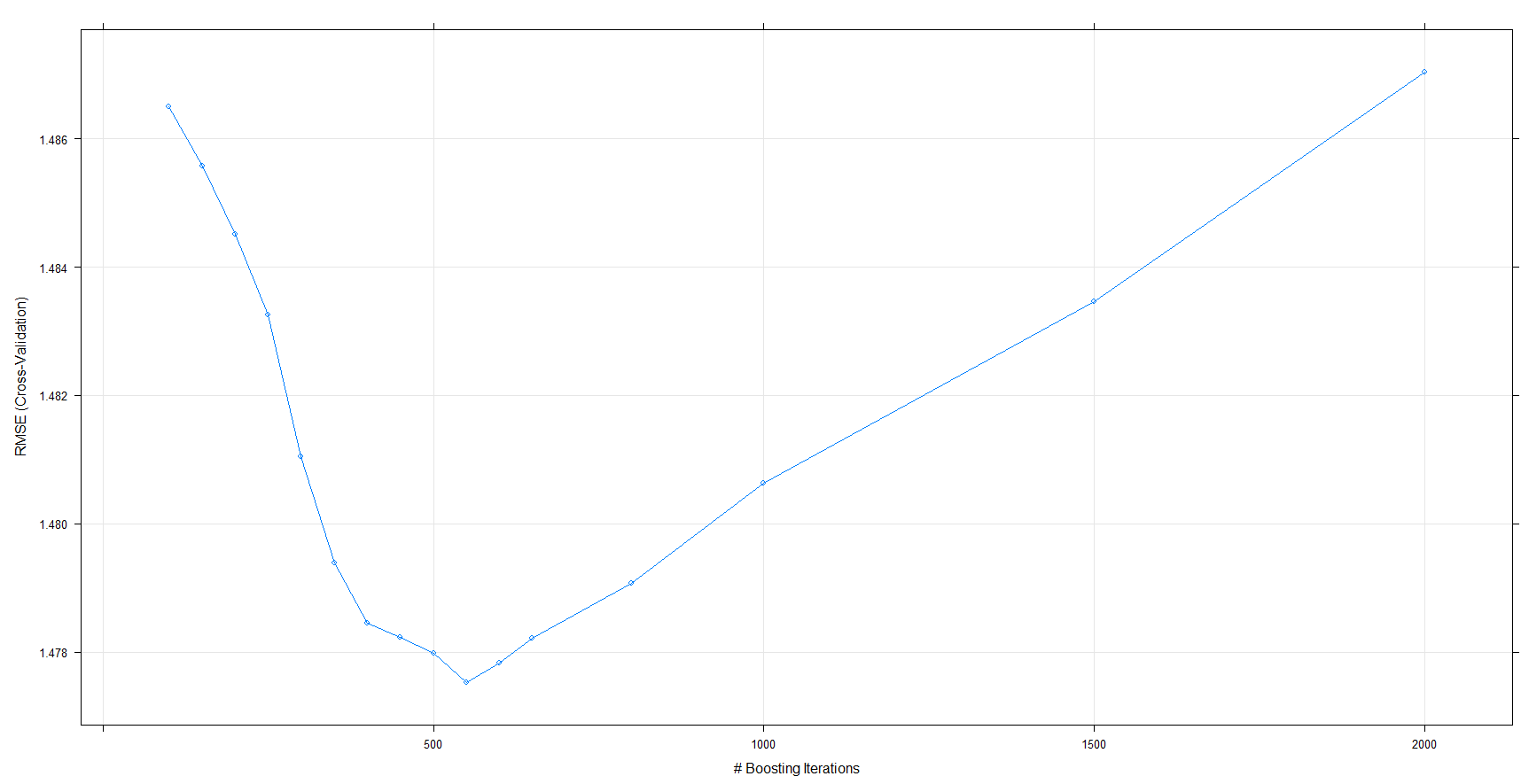


Figura 4.29: Número de iteraciones (early-stopping). Gradient Boosting

En la Figura 4.29 se puede ver claramente como el error (RMSE) disminuye de forma progresiva a medida que aumentan las iteraciones, hasta llegar un punto en el que se estanca (550 iteraciones). A partir de este punto, el error se vuelve a incrementar debido al sobreajuste producido.

Finalmente se han realizado 6 modelos de validación cruzada repetida con nfolds=10 y 20 repeticiones (semilla1234) a partir de los modelos de los anteriores bucles cuyo RMSE haya sido inferior a 1.480. Se han agrupado distribución de los errores con el objetivo de obtener gráficamente a través de un *boxplot* el modelo con menor sesgo y varianza posible. Se han obtenido los siguientes resultados.

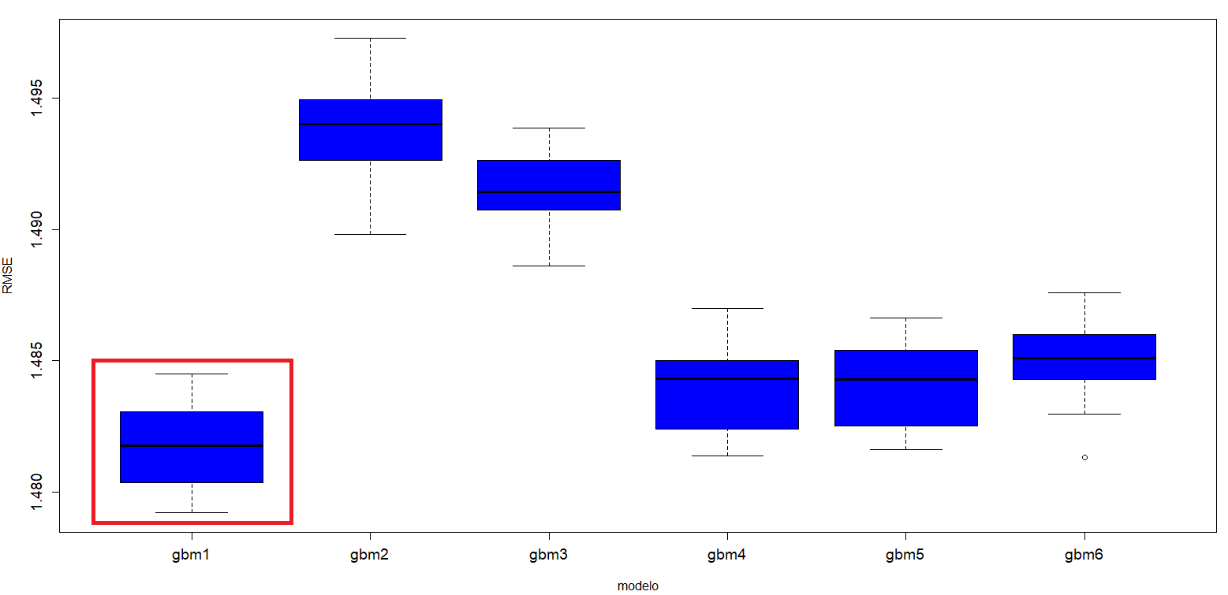


Figura 4.30: Gradient Boosting. Validación cruzada repetida

En la Figura 4.30 se ha obtenido para todos los modelos una varianza bastante similar y un sesgo ligeramente superior al observado por validación cruzada simple. Lo que nos muestra que en este caso, la validación simple podría estar subestimando el error de predicción. Por otro lado, se observa que el modelo “GBM1” compuesto *n.trees* = 1000, *interaction.depth* = 2, *shrinkage* = 0.05 and *n.minobsinnode* = 50 presenta el sesgo más reducido, no obstante ha requerido de un elevado número de iteraciones y una constante de regularización muy baja para converger al punto de corte óptimo

Finalmente de forma análoga al resto de modelos basados en árboles, se obtiene la importancia de las variables que han construido dicho modelo óptimo.

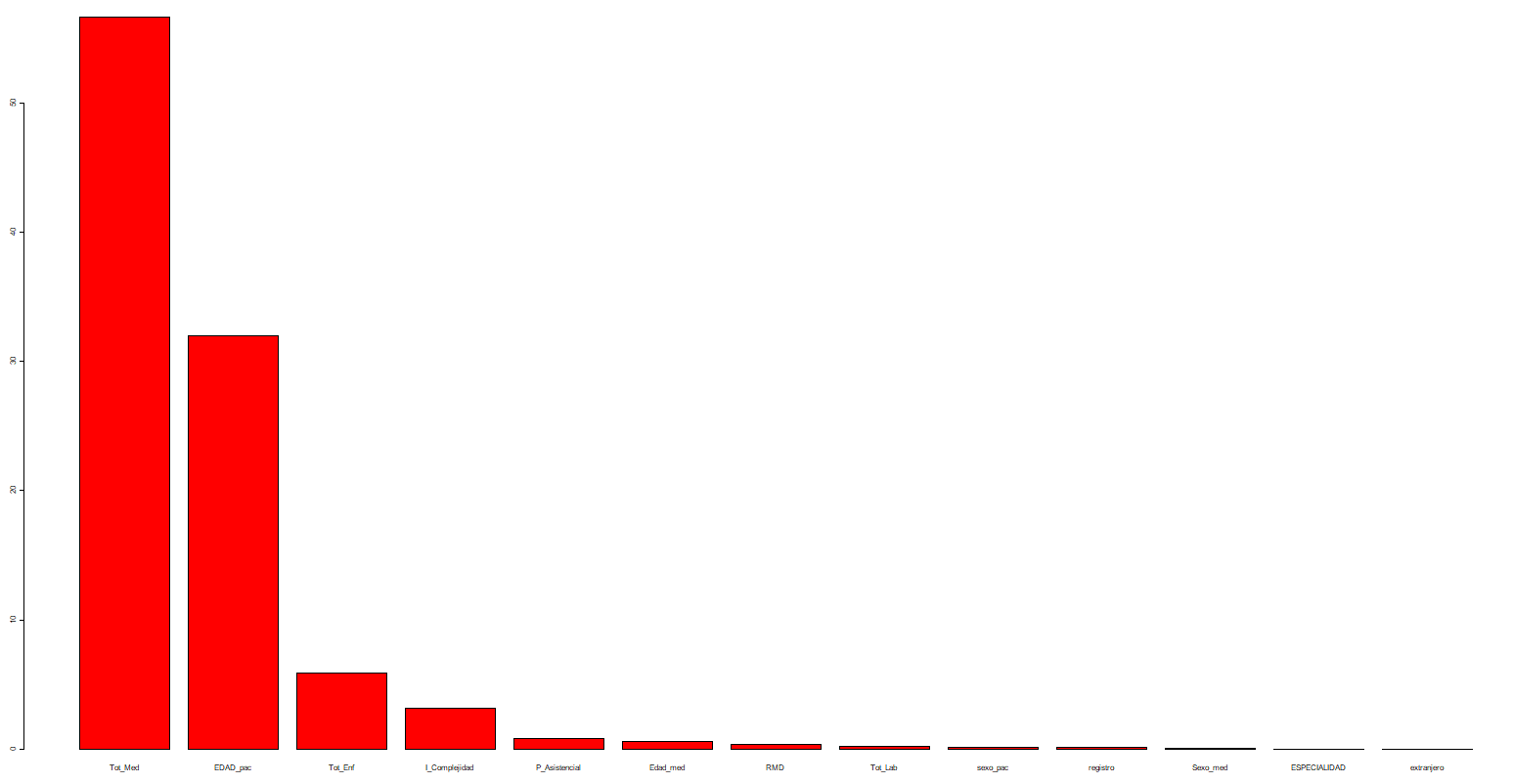


Figura 4.31: Importancia de las variables. Gradient Boosting

Como es lógico, la importancia de las variables de los modelos basados en árboles son muy similares, siendo las variables que explican cerca del 80% del modelo aquellas relacionadas la edad y el número de citas del paciente en las consultas del médico y enfermería. Las variables relacionadas con el médico y el municipio siguen manteniendo un bajo porcentaje explicativo en el modelo (por debajo del 10%).

### 4.4.7.- *Extreme Gradient Boosting*

Como se ha comentado en anteriores apartados la idea de este algoritmo consiste en una modificación del *Gradient Boosting* en el momento de construir un árbol con una función de penalización basada en el número de hojas y en la puntuación de cada hoja.

**Parámetros a optimizar:**

* **Número de iteraciones (*ntrees*):** este valor se corresponde con el número de árboles de regresión que se realizarán para para que el modelo pueda aprender de sus errores y mejorar el resultado de cada iteración.
* **Eta (*shrinkage*):** el valor de este parámetro determina cómo de rápido converge el modelo al punto de corte óptimo.
* **Tamaño del nodo final (min\_child\_weight):** este parámetro mide el tamaño máximo de los nodos finales (complejidad de los árboles)

A partir de estos parámetros se ha especificado un bucle por validación cruzada simple con 10 grupos, a través de la librería *caret.* En primer lugar introducen los valores de eta comprendidos entre (0.001-0.1) con el objetivo de observar la rapidez donde el punto de corte converge. Seguidamente se introducen los diferentes tamaños del nodo final (50-200). Por último se introducen las iteraciones (este parámetro se optimizará más adelante) y se fijan una serie de parámetros para realizar sorteo de variables (de forma análoga al *Random forest)* y para la constante de regulación. Se obtienen los siguientes resultados:

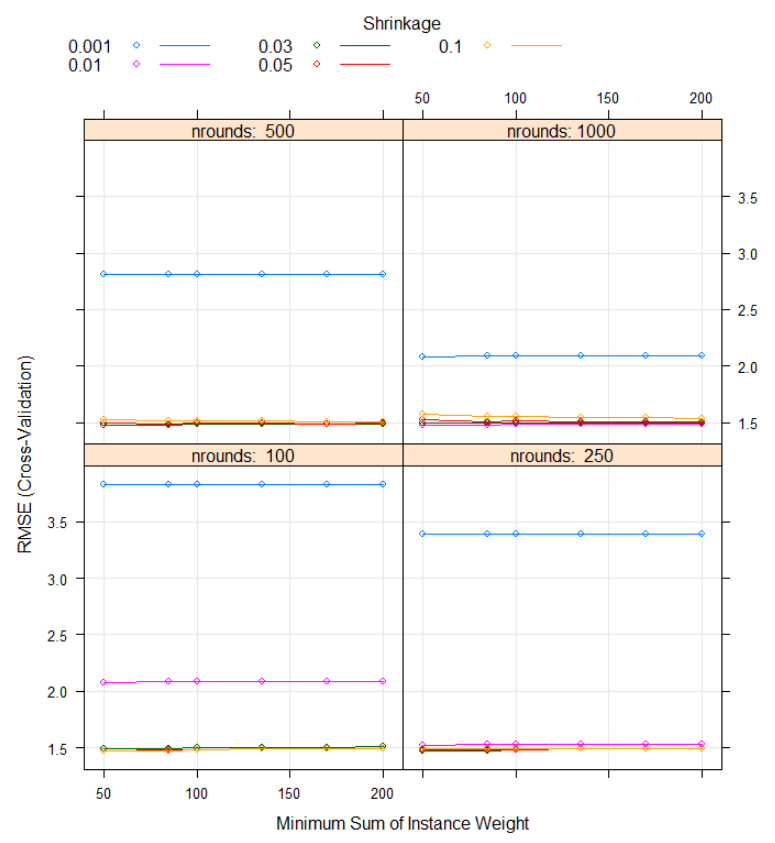


Figura 4.32: Parámetros óptimos. XGboost

En la Figura 4.32 se observa que el error converge a un punto de corte aparentemente óptimo cuando el valor de eta es igual o superior a 0.01 y el número de iteraciones es superior a 250. Siendo el mejor modelo observado aquel con eta = 0.01, tamaño del nodo final = 50 y número de iteraciones = 500.

A partir de esta configuración óptima se ha decidido realizar un análisis por *early-stopping* para evaluar en mayor medida el punto donde incrementar el número de iteraciones empieza a producir sobreajuste sin reducir el sesgo de manera significativa.

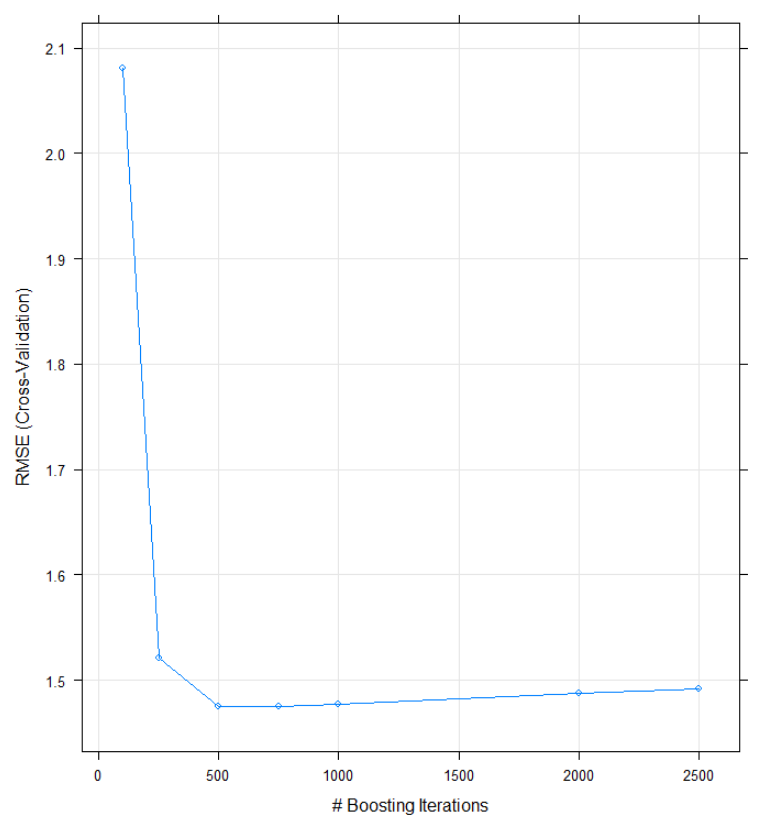


Figura 4.33: Número de iteraciones (early-stopping). XGboost

En la Figura 4.33 se puede ver que el error (RMSE) disminuye de forma progresiva a medida que aumentan las iteraciones, hasta llegar un punto en el que se estanca. Cuando el número de iteraciones = 500 parece ser un tamaño adecuado, a partir del cual el error se incrementa debido al sobreajuste producido.

A continuación se ha utilizado validación cruzada repetida con nfolds=10 y 20 repeticiones (semilla1234), para comprobar aquellos modelos que han reportado un menor sesgo (RMSE <1.480) en el anterior bucle junto con otros modelos variando ligeramente el tamaño del nodo final para evaluar la varianza producida.

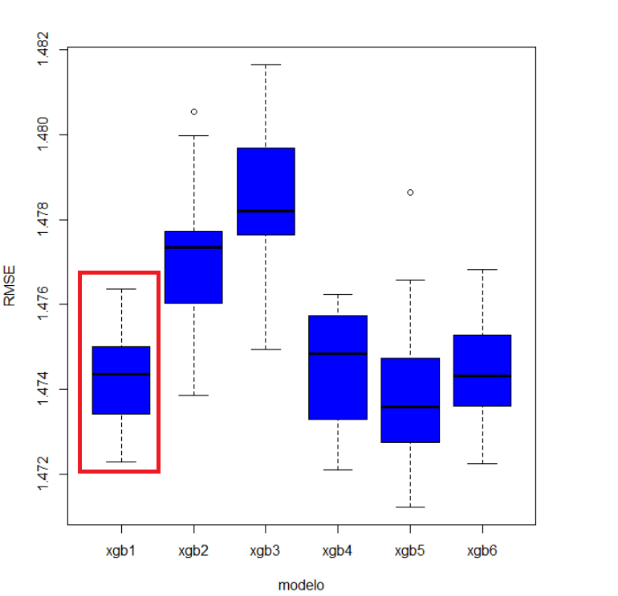


Figura 4.34: XGboost. Validación cruzada repetida

La varianza de los modelos realizados varía en gran medida dependiendo de la configuración realizada. El modelo *xgb5* presenta un error ligeramente superior al resto de modelos, pero la altura de la caja (varianza) es bastante superior al resto de casos. Esto se debe a que el tamaño del nodo final es inferior aumentando el sobreajuste y perdiendo parte de representatividad futura de nuestros datos. Por lo tanto, se ha decidido seleccionar el modelo *xgb1 (ntrees = 500, eta =0.01, min\_child\_weight* = 50) que presenta un error ligeramente superior pero una varianza más reducida.

Por último se obtiene la importancia del modelo óptimo de XGboost

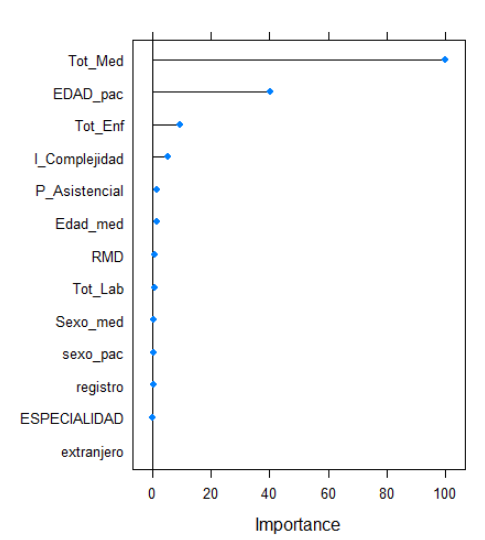


Figura 4.35: Importancia de las variables. XGboost

Este algoritmo ofrece unos resultados en la importancia de muy similares al Gradient Boosting. La variable registro no presenta ninguna importancia sobre el modelo, por lo que todas aquellas variables por debajo de estas resultarían inservibles a la hora de predecir el gasto en farmacia.

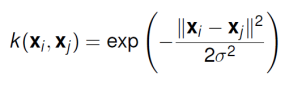
### 4.4.8.- Support Vector Machine

Como se ha comentado en anteriores puntos, la idea del Support Vector Machine (SVM) consiste en determinar los vectores soporte que determinan en cada caso el margen máximo a ambos lados del hiperplano que equidista al resto de puntos.

De forma análoga al modelo de redes neuronales, se trabajará con el conjunto de datos depurado y estandarizado, que utiliza como método de selección de variables los predictores obtenidos por consenso de 4/7 algoritmos de selección (ver Tabla 4.3).

**Parámetros a optimizar:**

* **Constante de penalización**: la distancia de los puntos respecto del margen máximo se determinan a través de añadir al problema de optimización una constante de penalización *C* (coste*),* que se encuentra relacionada inversamente con la anchura del margen máximo. Se utilizará utilizarán diferentes tamaños de esta constante para determinar el nivel de coste que genera un sesgo y varianza menor. Dependiendo de los valores de *C* se estará penalizando mucho o poco la mala clasificación de dichos puntos.
* **Valores de escala:** valores que permiten a la función kernel trabajar computacionalmente en una dimensión controlada a través de la función Kernel.
* **Función *Kernel* lineal:** se ha utilizado la función lineal para determinar el margen máximo que equidista los datos en cada lado del hiperplano.
* **Función *Kernel* polinomial**: se ha utilizado la función polinomial El parámetro *p* (2-3)determina la dimensión (grado del polinomio) de la función.
* **Función *Kernel* radial (RBF):** se ha utilizado la función radial con el parámetro σ que permite controlar el comportamiento del Kernel, de manera que valores altos de gamma implican menor sesgo y mayor sobreajuste.

******

A partir de estos parámetros, se ha especificado un bucle utilizando la librería *Caret* para la función *Kernel* lineal por validacíon cruzada simple que contempla los valores de la constante de penalización (*C)* comprendidos entre (0.01-50). De manera que tengamos un rango lo suficientemente amplio para comprobar el comportamiento del modelo.

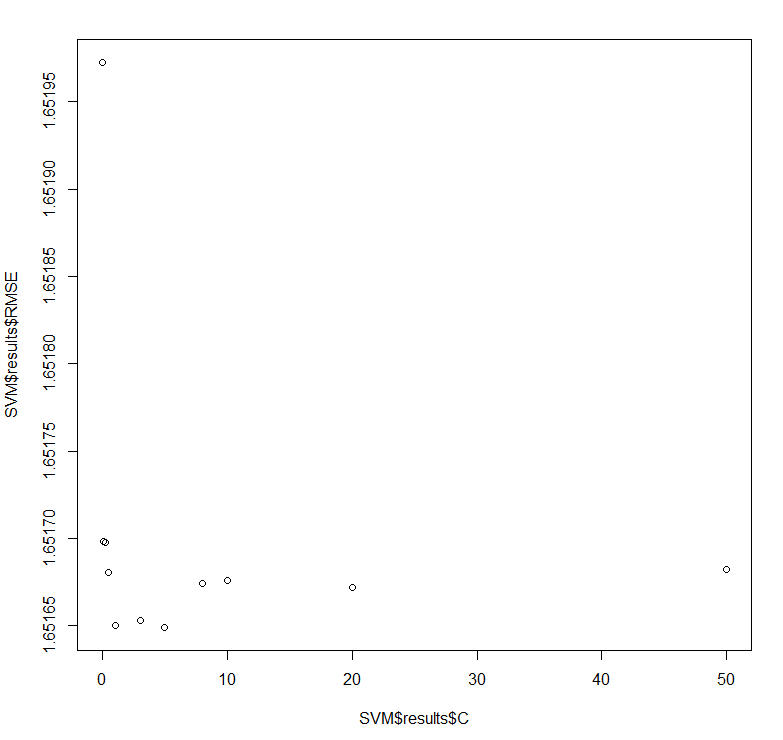


Figura 4.36: Nivel de penalización óptimo. SVM lineal

En la Figura 4.36 se puede observar que las diferencias en términos de error para los distintos tipos de *C* son mínimas. No obstante, se puede afirmar que para valores más bajos de la constante de penalización, se obtiene un error ligeramente inferior. Esto se debe a que en el conjunto *train*, los valores superiores de esta constante determinan que la separación que equidista de los datos más cercanos a ambos lados del hiperplano es menor. Este criterio más estricto reduce los residuos sobre *train,* lo que genera que el sesgo sea más reducido pero el sobreajuste es mayor (mayor varianza) lo que empeora los resultados en *test*. Se han seleccionado los tres modelos que disponen de un RMSE menor, para realizar posteriores comprobaciones por validación cruzada repetida.

A continuación se ha especificado otro bucle para la función *Kernel* polinomial que contempla valores de la constante de penalización comprendidos entre (0.1-10), valores de *P (2-3)* y finalmente valores de escala comprendidos entre (0.1-5). Es importante destacar que se ha reducido el rango de estos parámetros para controlar el tiempo computacional. Se obtienen los siguientes resultados.

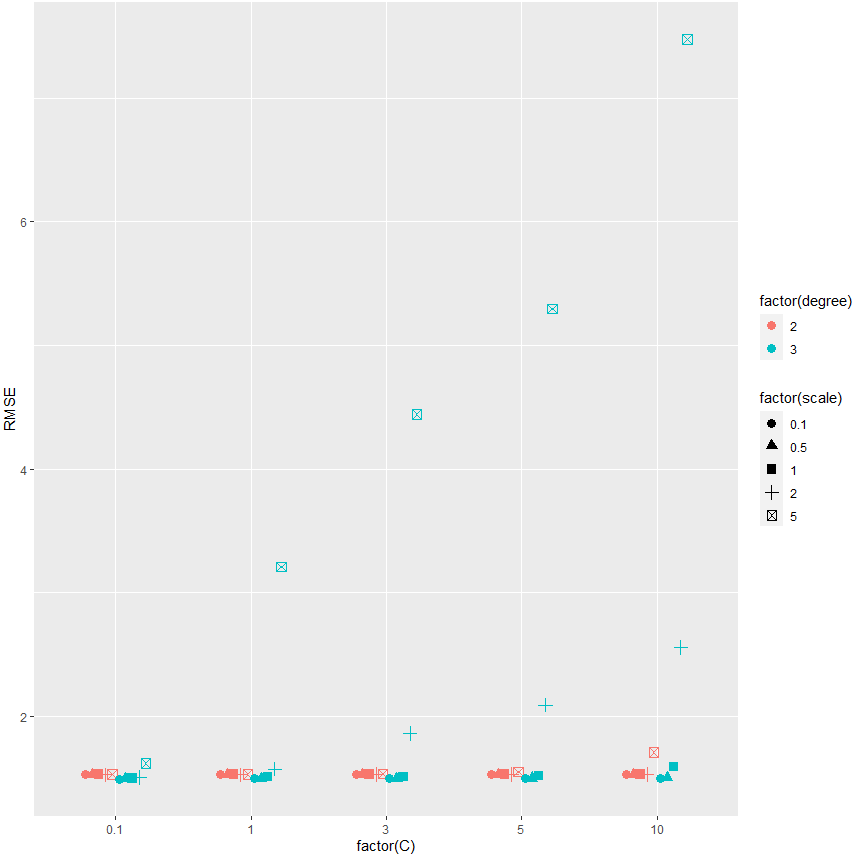


Figura 4.37: Nivel de penalización óptimo. SVM lineal

En esta imagen se observa que cuando el grado del polinomio es igual a 3, el sesgo es más reducido cuando el valor de escala es más bajo, aunque estos valore se incrementan de manera exponencial cuando el coste y la escala son superiores a 2. Por otro lado, el RMSE se mantiene relativamente bajo cuando el coste y la escala presentan valores muy reducidos, independientemente del grado del polinomio. Sobre los valores escogidos, parece alcanzarse un valor mínimo cuando el modelo viene expresado por una función kernel polinomial de grado 3, con factor de escala = 0.1 y constante de penalización = 0.1. Se guardan las especificaciones óptimas de los tres mejores modelos.

De manera análoga al *Kernel* lineal y polinomial, se ha especificado otro bucle para la función radial que recoge valores de *C y* sigmacomprendidos entre (0.01-20).

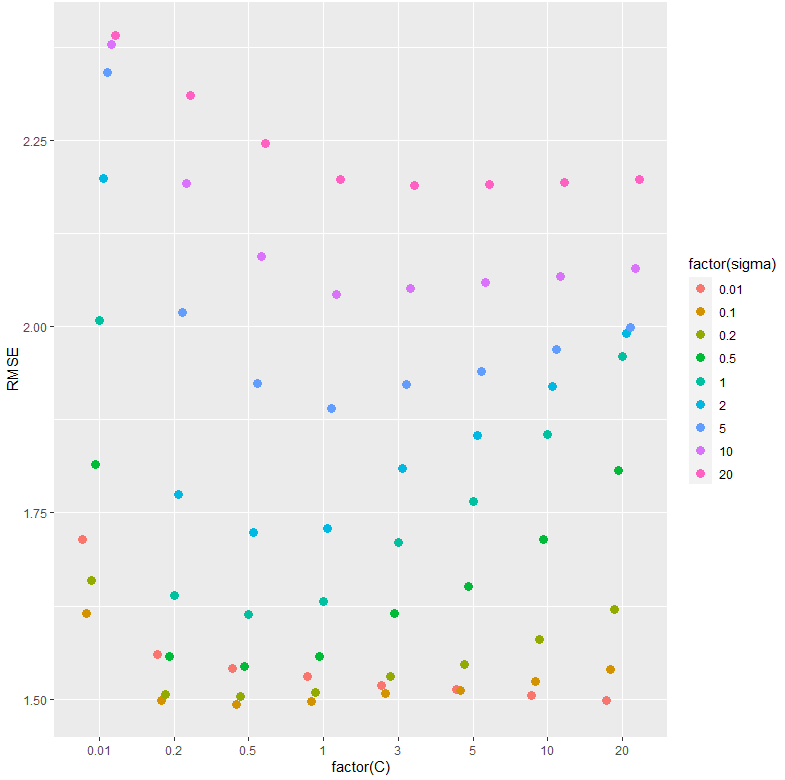


Figura 4.38: Parametización óptima. SVM RBF

En este gráfico se puede ver que el comportamiento que siguen ambos parámetros se encuentra en la misma dirección, de manera que para para valores altos de *C*  se requieren altos niveles de sigma para estabilizar el error y reducir el error sobre el conjunto de *train*. No obstante, para los datos *test* se observa que el sesgo de estos valores es muy superior al caso donde se utilizan valores reducidos de estos dos parámetros, debido al sobreajuste producido. Se guardan las especificaciones óptimas de los tres mejores modelos.

Finalmente, a partir de las especificaciones óptimas obtenidas en cada bucle, se ha construido por validación cruzada repetida con nfolds=10 y 20 repeticiones (semilla1234) el siguiente *boxplot.*

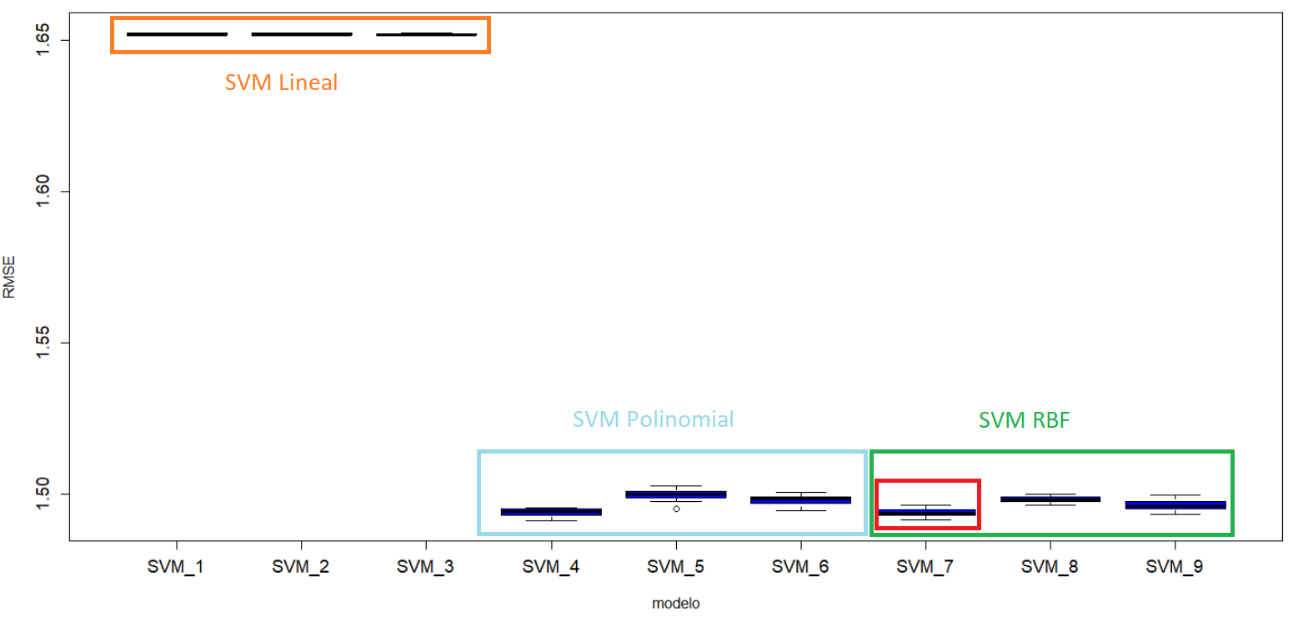


Figura 4.39: Comparación de modelos SVM. Validación cruzada repetida

En la Figura 4.39 se observa que la varianza en cada modelo es muy similar. Por otro lado, en relación al sesgo, es importante destacar que dada la configuración de nuestros datos se observa que la separación entre clases por los hiperplanos no es lineal, lo que permite al SVM polinomial y RBF captar mejor el comportamiento de nuestros datos y reducir el sesgo. Entre estos dos últimos modelos los resultados son muy similares, no obstante el error en el modelo SVM\_7 compuesto por los parámetros *C = 0.5 y* σ = 0.1 presenta valores del RMSE y varianza (altura de la caja) ligeramente inferiores.

## 4.5.- Modelos de ensamblado

En el presente trabajo, a partir de las técnicas de *machine Learning* se han podido construir distintos modelos que contemplan diferentes relaciones lineales, no lineales, técnicas geométricas, funciones de aproximación entre otras técnicas. Cada modelo asume una configuración determinada de los datos disponibles, por lo que resulta razonable aprovechar cada una de las formas de resolver un problema para llegar a una posición común.

Como se ha comentado en anteriores apartados, los modelos de ensamblado son aquellos que pretenden predecir el comportamiento de una variable *output* a partir de la combinación de predicciones de varios modelos. A lo largo del trabajo, se han utilizado algunas técnicas de ensamblado esencialmente basadas en árboles. No obstante es posible, combinar todos los modelos realizados hasta ahora, con el objetivo de elaborar un modelo conjunto, que en algunos casos resulte más robusto y presente un menor sesgo observado.

En la tabla 4.4 se han construido distintas combinaciones de modelos de ensamblado a partir de los mejores obtenidos por validación cruzada repetida en anteriores apartados (salvo el Bagging y los modelos de árboles simples ya que el Random Forest supone una particularización de ambos).

|  |  |
| --- | --- |
| Modelo | Ensamblaje |
| Predi1 | xgbm + rf |
| Predi2 | avnnet + svmRadial |
| Predi3 | xgbm + rf + gbm |
| Predi4 | avnnet + rf + gbm |
| Predi5 | avnnet +gbm + svmRadial |
| Predi6 | rf + gbm + svmRadial |
| Predi7 | xgbm + svmRadial + rf |
| Predi8 | gbm + svmRadial + rf + avnnet |
| Predi9 | reg + svmRadial + gbm + rf |
| Predi10 | gbm + xgbm + rf + avnnet |
| Predi11 | xgbm + avnnet + rf + svmRadial + gbm |
| Predi12 | xgbm + reg + rf + svmRadial + gbm |
| Predi13 | reg+avnnet +rf +gbm + svmRadial+xgbm |

Tabla 4.4: Posibles combinaciones de modelos de ensamblado

Para cada combinación se ha construido un modelo nuevo cuya predicción final es obtenida por el método *averaging.* Esto es elpromedio de cada una de las predicciones.

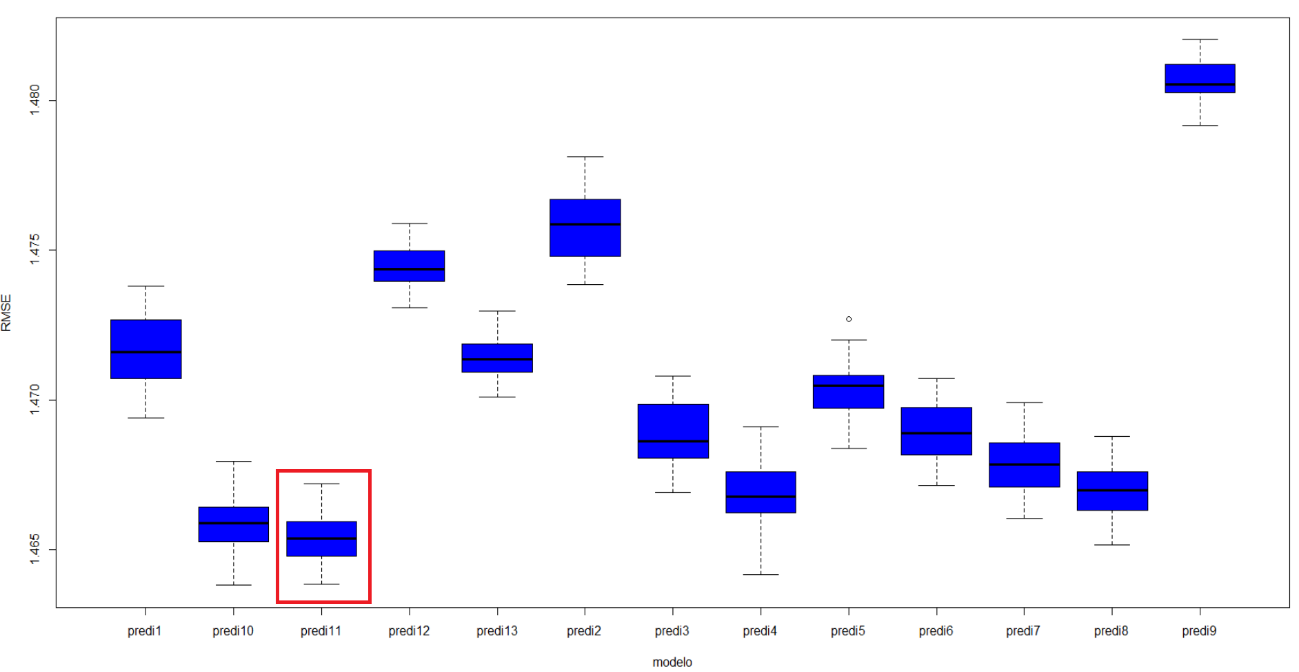


Figura 4.40: Comparación modelos de ensamblado

En la Figura 4.40 se observa que todos los modelos presentan una varianza muy similar. En relación al sesgo (RMSE) los modelos *predi10* y *predi 5* presentan valores muy similares de error y relativamente inferiores al resto de modelos. No obstante, la varianza (altura de la caja) del modelo 10 conformado por la combinación del gbm + svmRadial + rf + avnnet es ligeramente inferior.

## 4.6.- Comparación de resultados

En esta última fase de la metodología SEMMA, se han discutido los resultados de la parte de modelización con el objetivo de comparar la calidad de los modelos realizados y establecer conclusiones (*Access).* Para ello, se han comparado los mejores modelos de manera conjunta por cada técnica aplicada, se han analizado sus resultados y finalmente se ha seleccionado el mejor modelo en términos de menor sesgo y varianza por validación cruzada repetida.

En la Tabla 4.5 se ha recogido la configuración paramétrica óptima y el proceso de selección de variables de cada técnica aplicada.

|  |  |
| --- | --- |
| Técnica | Características |
| Regresión lineal | -Selección de variables: Consenso métodos de selección (ver Tabla 4.3) |
| Árboles de regresión | -Selección de variables: Criterio de punto de corte óptimo. Varianza  -Tamaño del nodo final: 85 |
| Redes neuronales | -Selección de variables: Consenso métodos de selección (ver Tabla 4.3)  - Nodos en la capa oculta: 10  -Número de iteraciones : 250  -Learning rate : 0.0001  -Algotirmo optimización: Gradient descent |
| *Bagging* | - Selección de variables: Bagging  - Mtry = 13  - Número de iteraciones: 130  -Tamaño de la muestra: 3500  -Tamaño del nodo final: 85 |
| *Random forest* | - Selección de variables: Remuestreo de variables  -Mtry = 7 |
| Gradient Boosting | Selección de variables: Gradient Boosting  Constante de regularización: 0.05  Número de iteraciones: 1000  Tamaño del nodo final: 50 |
| Extreme Gradient Boosting | Selección de variables: colsample\_bytree=1  Número de iteraciones: 500  Eta: 0.01  Tamaño del nodo final: 50 |
| SVM RBF | -Selección de variables: Consenso métodos de selección (ver Tabla 4.3)  -Constante de penalización: 0.5  -σ = 0.1  -Kernel : Radial |
| Ensamblaje | xgbm + avnnet + rf + svmRadial + gbm |

Tabla 4.5: Características de los mejores modelos obtenidos

En la Figura 4.41 se recoge la configuración óptima de cada modelo realizado por validación cruzada repetida con nfolds=10 y 20 repeticiones (semilla1234).

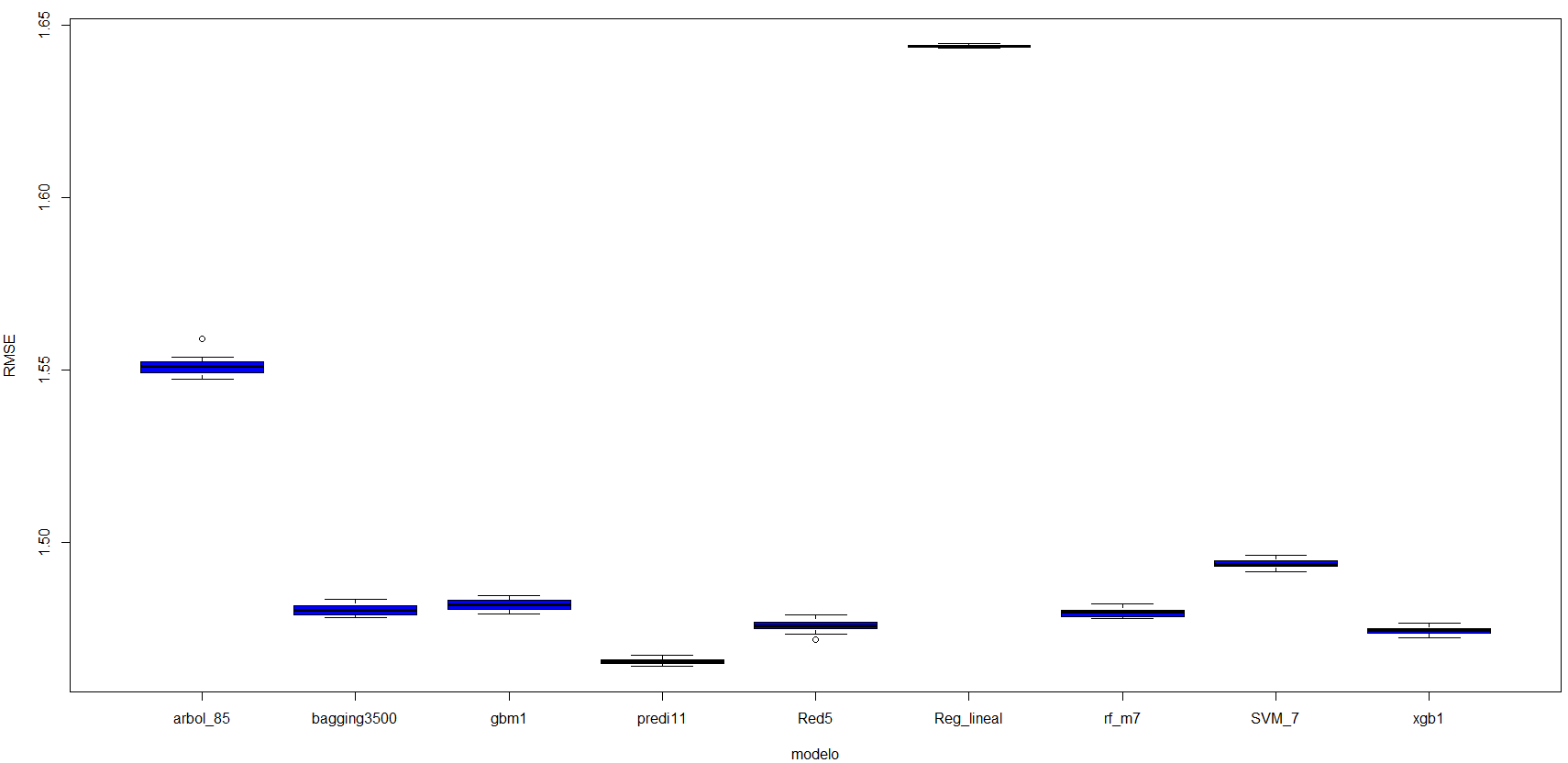


Figura 4.41: Comparación modelos finales. Validación cruzada repetida

En este gráfico destacan los modelos de redes neuronales, *random forest, extreme gradient boosting* y el modelo de ensamblado, que han obtenido resultados sensiblemente mejores que el resto tanto en términos de varianza como en sesgo. A su vez, es importante resaltar el mal funcionamiento relativo al resto de modelos que tiene la regresión lineal. Esto nos confirma la existencia de relaciones no lineales y complejas que dificultan a este modelo obtener una predicción demasiado precisa.

Para analizar con mayor detalle las diferencias en términos de precisión y varianza, se ha ampliado el gráfico anterior (ver Figura 4.41).

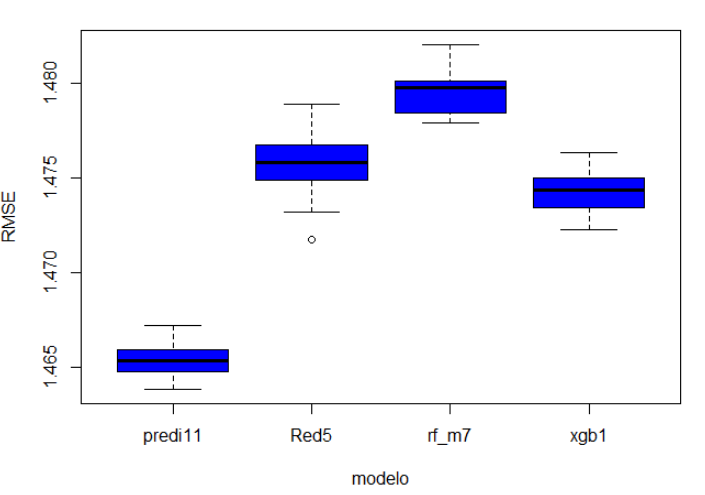


Figura 4.42: Comparación ampliada de los modelos finales. Validación cruzada repetida

En la Figura 4.42 se ha observado que las predicciones del modelo de ensamblado son las más precisas y robustas entre los modelos analizados. Cabe destacar que este ha sido construido a partir de la combinación de los modelos que de manera individual presentan el menor nivel de error, lo que nos puede indicar que en este caso la agrupación de los modelos óptimos con RMSE inferior ha sido una buena elección para obtener una predicción con bajo sesgo y varianza.

Para finalizar con este apartado de resultados, se ha procedido a comentar los valores predictivos obtenidos en función de si los modelos son interpretables, o bien se desconocen las relaciones que han conformado su predicción.

**Interpretación de los parámetros estimados**

Regresión lineal

Como se ha comentado anteriormente, el modelo de regresión lineal no ha aportado valor predictivo demasiado preciso debido a las relaciones no lineales y complejas de los datos (ver Figura 4.19). No obstante, una de las ventajas que nos proporciona el modelo de regresión lineal es su capacidad de interpretabilidad de los resultados.

En la siguiente expresión se obtiene el valor de los coeficientes estimados de cada variable independiente. Es importante destacar que el gasto en farmacia está en escala logarítmica, por lo que para realizar cualquier interpretación es necesario calcular *exp(gasto en farmacia).*

*farmacia = -1.81 + 0.46EDAD\_pac + 0.142\*Tot\_Med + 0.065 \* Tot\_Enf*

*0.103\*sexo\_pac + 1.64\*I\_Compjejidad +0.07 \*Tot\_Lab +0.0057 P\_Asistencial*

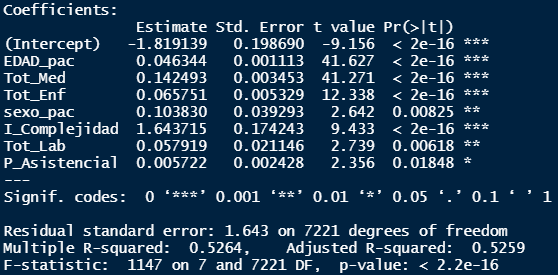
**

Figura 4.43. Coeficientes modelo de regresión lineal óptimo

En la Figura 4.43 se ha observado que el modelo en conjunto tiene un p-valor muy bajo y las variables de manera individual son significativas para un nivel de significación del 99%, además dado que se cumplían en gran medida las hipótesis paramétricas (ver Figura 4.14) la interpretación de los coeficientes procede. En este sentido, se puede observar que todas las variables incrementan el gasto sanitario, y que aquellas que pueden tomar un rango de valores más elevado (edad y el número de citas) son las que incrementan en mayor medida el gasto farmacéutico. Por otro lado, variables como la complejidad del departamento del médico solo pueden tomar valores entre (0-1.1) incrementan el gasto sanitario de manera significativa, pero en menor medida. Respecto al sexo del paciente, cuando toma el valor 1 (hombre), el gasto en farmacia tiene a incrementarse ligeramente.

Árboles de regresión

El árbol de regresión aporta un sesgo relativamente inferior al modelo de regresión lineal (algo poco frecuente), pero una varianza superior. En comparación al resto de modelos más complejos, los árboles se caracterizan por tener poca eficacia predictiva y mala generalización, ya que al ser cada hoja un parámetro es más propenso a generar modelos sobreajustados e inestables a la predicción.

No obstante, este modelo presenta una gran utilidad descriptiva en cuanto a que permite representar gráficamente las divisiones y los criterios de corte sobre los que se han construido los árboles. Esto permite comprender en gran medida la naturaleza de la variable respuesta a predecir y permite identificar qué variables pueden estar aportando más o menos información para el análisis. Respecto a la regresión lineal, los árboles no se ven afectados en gran medida por las relaciones no lineales de las variables input, lo que ha permitido descubrir patrones e interacciones complejas entre nuestros datos que no pueden ser captadas por la regresión lineal.

En relación a la interpretabilidad de este modelo (ver Figura 4.17), es importante tener en cuenta que los resultados de cada nodo están expresados en escala logarítmica, de manera que el nodo raíz representa la media de los logaritmos del gasto en farmacia. En este sentido si analizamos la rama final que determina un mayor gasto en farmacia (ver Figura 4.44), si la edad es superior a 67 años, para deshacer el promedio de logaritmos se busca sobre los datos originales el gasto medio de los pacientes cuando la edad era superior a este valor (gasto en farmacia = 770€). Atendiendo a este razonamiento, observamos que cuando la edad es superior a 67 años y las citas en la consulta del médico son superiores a 16, se llega a un nodo hoja con 450 observaciones y un gasto medio en farmacia = 1237. Este razonamiento es idéntico para cada división dentro del árbol.

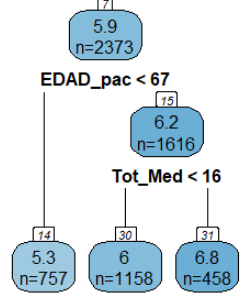


Figura 4.44: Nodos finales árbol de regresión simplificado

**Modelos cuyos parámetros y relaciones no son explicables**

Este conjunto de modelos se ha caracterizado por el desconocimiento de las transformaciones internas realizadas para la elaboración de los valores predictivos, renunciando a la interpretabilidad que proporcionan modelos más simples, pero ganando una mayor eficacia en la obtención de predicciones más precisas.

A pesar de que algunos de que en algunos de estos modelos se han podido obtener las estimaciones de los pesos de los parámetros (redes neuronales), se desconoce la forma de interpretar estos pesos de una manera generalizable a distintos escenarios. En este sentido, a modo de ejemplo, variar mínimamente la construcción de la red o incluso variar la semilla, reporta pesos completamente distintos tanto en signo con en valor.

No obstante, tanto las redes neuronales, como los métodos de ensamblado y el SVM, nos han permitido estudiar las relaciones complejas presentes en nuestros datos, construir modelos más robustos a partir del remuestreo de observaciones y variables (*random forest*), y en gran parte de los casos han aportado información acerca de las variables más relevantes sobre las que se han construido los modelos (ver Figuras 4.18, 4.27, 4.31, 4.35). Esto último, responde a uno de los objetivos principales de este trabajo, ya que hemos podido observar que pese a que las variables relacionadas con el paciente tienen un claro peso en la predicción de este gasto sanitario, otros *inputs* como la presión asistencial o la complejidad del departamento del médico tienen una relevancia significativa.

# 5.- Conclusiones

Tras finalizar el desarrollo del proyecto, se han llegado a las siguientes conclusiones que pueden tener un aspecto relevante para el desarrollo de futuros análisis.

1. La alta dispersión de los datos y la presencia de un gran número de observaciones atípicas ha justificado la transformación logarítmica sobre el gasto en farmacia.
2. La transformación logarítmica ha permitido obtener modelos más precisos y robustos. Esto ha incrementado el R2 inicial desde 0.3 hasta llegar a valores de 0.62
3. El proceso de selección de variables construido a partir del consenso de diferentes algoritmos de selección ha permitido reducir el tiempo computacional y el sobreajuste de los modelos realizados.
4. Las variables relacionadas con el paciente, concretamente la edad y las citas en las consultas del médico de medicina y enfermería determinan en mayor medida el gasto en farmacia.
5. Las variables relacionadas con el médico tienen un impacto más reducido para determinar el gasto en farmacia. La presión asistencial y la complejidad del departamento del médico son los factores más significativos.
6. La renta media disponible del municipio tiene un impacto reducido para explicar el gasto en farmacia. No obstante, dado el periodo (1997-1999) de ligero crecimiento económico, se espera que este factor adquiera una mayor relevancia en periodos de desaceleración económica.
7. Las relaciones complejas y no lineales entre los datos justifican la aplicación de modelos complejos capaces de captar dicha estructura.
8. El modelo de ensamblado compuesto por el promedio de las predicciones de las técnicas *extreme gradient boosting*, red neuronal, *random forest, support mector machine* con función radial y el *gradient boosting,* obtiene los mejores resultados en términos de sesgo y varianza.

# Anexos

# 

Anexo 1: Diagrama SAS Enterprise Miner

# Bibliografía

Alberquilla, A., Feliz, L. V., de Francisco, A. G., Vásquez, A. R., del Mar Rodríguez, M., & de Francisco, S. G. (2017). Análisis de la hiperfrecuentación ajustando por comorbilidad según ACG (Adjusted Clinical Groups) en una Zona Básica de Salud. *Medicina general*, *6*(6), 2.

Andersen, R. (1968). A behavioral model of families' use of health services. *A behavioral model of families' use of health services.*, (25).

Ávila-Tomás, J. F., Mayer-Pujadas, M. A., & Quesada-Varela, V. J. (2021). La inteligencia artificial y sus aplicaciones en medicina II: Importancia actual y aplicaciones prácticas. *Atención Primaria*, *53*(1), 81-88.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152).

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

Brownlee, J. (2016). *XGBoost With python: Gradient boosted trees with XGBoost and scikit-learn*. Machine Learning Mastery.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

de Salud, S. E. Características, perspectivas y retos de la atención primaria de salud: financiación y necesidades. La visión de un médico asistencial de un centro de salud urbano. Francisco Buitrago. Especialista en Medicina Familiar y Comunitaria. Doctor en Medicina. *DOCUMENTOS DEL I SEMINARIO*, 37.

Fernández, R. R. Métodos de ensamblado en Machine Learning.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.".

Gómez, A. A. R., & Bautista, D. W. R. (2010). Inteligencia de negocios: Estado del arte. *Scientia et technica*, *1*(44), 321-326.

Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11). London: Chapman and Hall.

López-Montes, M. T. M. C. (2007). *Modelo multinivel explicativo de la utilización de las consultas de atención primaria en Andalucía* (Doctoral dissertation, Universidad de Granada).

Moine, J. M., Haedo, A. S., & Gordillo, S. E. (2011). Estudio comparativo de metodologías para minería de datos. In *XIII Workshop de Investigadores en Ciencias de la Computación*.

Palacios-Cruz, L., Pérez, M., Rivas-Ruiz, R., & Talavera, J. O. (2013). Investigación clínica XVIII. Del juicio clínico al modelo de regresión lineal. *Revista Médica del Instituto Mexicano del Seguro Social*, *51*(6), 656-661.

Servicio Andaluz de Salud/EASP (2016). Los ACG\* en la Gestión Clínica en Atención Primaria. Recuperado de: https://www.sspa.juntadeandalucia.es/servicioandaluzdesalud/sites/default/files/sincfiles/wsas-media-mediafile\_sasdocumento/2019/que\_son\_los\_ACGs.pdf

Sicras-Mainar, A., & Navarro-Artieda, R. (2009). Validación retrospectiva del Johns-Hopkins ACG Case-Mix System en la población española. *Gaceta Sanitaria*, *23*, 228-231.

Starfield, B., Weiner, J., Mumford, L., & Steinwachs, D. (1991). Ambulatory care groups: a categorization of diagnoses for research and management. *Health services research*, *26*(1), 53.

Suárez, E. J. C. (2014). Tutorial sobre máquinas de vectores soporte (sVM). *Tutorial sobre Máquinas de Vectores Soporte (SVM)*, 1-12.

Vapnik, V. (1964). A note one class of perceptrons. *Automation and remote control*.