

# FACULTAD DE ESTUDIOS ESTADÍSTICOS

## MÁSTER EN CIENCIA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2024/2025

---

### Trabajo de Fin de Máster

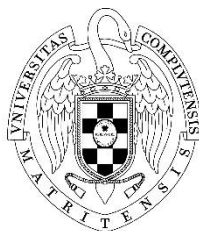
**TÍTULO:**

*Predicción del costo del seguro médico en el estado de Nueva York mediante técnicas de machine learning.*

**Alumno:** Wendy Johanna Camargo Orozco

**Tutores:** Antonio Jesús Guillén López y José Fernández Menéndez

Septiembre de 2025



UNIVERSIDAD COMPLUTENSE  
MADRID

**Declaración Responsable sobre Autoría y Uso Ético de Herramientas de  
Inteligencia Artificial (IA)**

Yo, Wendy Johanna Camargo Orozco con NIE: Y5473914C declaro de manera responsable que el presente:

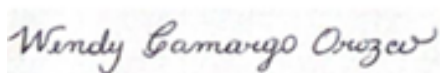
- ☐ Trabajo de Fin de Grado (TFG)
- ☒ Trabajo de Fin de Máster (TFM)
- ☐ Tesis Doctoral

Titulado **Predicción del costo del seguro médico en el estado de Nueva York mediante técnicas de machine learning** es el resultado de mi trabajo intelectual personal y creativo, y ha sido elaborado de acuerdo con los principios éticos y las normas de integridad vigentes en la comunidad académica y, más específicamente, en la Universidad Complutense de Madrid.

Soy, pues, autor del material aquí incluido y, cuando no ha sido así y he tomado el material de otra fuente, lo he citado o bien he declarado su procedencia de forma clara -incluidas, en su caso, herramientas de inteligencia artificial-. Las ideas y aportaciones principales incluidas en este trabajo, y que acreditan la adquisición de competencias, son mías y no proceden de otras fuentes o han sido reescritas usando material de otras fuentes.

Asimismo, aseguro que los datos y recursos utilizados son legítimos, verificables y han sido obtenidos de fuentes confiables y autorizadas. Además, he tomado medidas para garantizar la confidencialidad y privacidad de los datos utilizados, evitando cualquier tipo de sesgo o discriminación injusta en el tratamiento de la información.

En Madrid a 03 de septiembre de 2025.



Wendy Camargo Orozco.

## Resumen

Este trabajo tiene como objetivo predecir el coste total del seguro médico en pacientes con traumatismo del estado de Nueva York y analizar la contribución de variables clínicas, administrativas y demográficas en este coste. Se emplea la base de datos SPARCS (*Statewide Planning and Research Cooperative System*), publicada por el departamento de salud del estado de Nueva York a través de su portal abierto. Se siguen los pasos de la metodología CRISP-DM para abordar el problema y una vez preparados los datos se procede a la modelización para encontrar el algoritmo que se ajuste más a la situación abordada.

Seleccionado el set de variables más prometedor, se modeliza el coste del seguro a través de regresión lineal, modelos de aprendizaje automático basados en árboles, SVM y redes neuronales, evaluados mediante validación cruzada repetida con las métricas de rendimiento RMSE (Root Mean Square Error) y  $R^2$  (coeficiente de determinación).

Los resultados evidencian que el modelo XGBoost resulta ser el que ofrece mejor desempeño y, a través de la importancia de variables, permite interpretar la influencia de diferentes variables médicas, administrativas y demográficas en el coste, aportando evidencia útil para la toma de decisiones en contextos sanitarios y particulares como en Estados Unidos.

**Palabras claves:** Aprendizaje automático, validación cruzada, predicción de costes sanitarios; selección de variables, machine learning.

## Tabla de contenido

|                                                      |    |
|------------------------------------------------------|----|
| <b>1. INTRODUCCIÓN</b>                               | 7  |
| 1.1 Introducción del problema                        | 7  |
| 1.2 Objetivos                                        | 10 |
| 1.2.1 Objetivo principal                             | 10 |
| 1.2.2 Objetivos específicos                          | 10 |
| <b>2. METODOLOGÍA EMPLEADA</b>                       | 11 |
| 2.1 Metodología CRISP-DM                             | 11 |
| 2.2 Machine learning                                 | 12 |
| 2.2.1 Regresión lineal                               | 13 |
| 2.2.2 Bagging                                        | 14 |
| 2.2.3 Random Forest                                  | 14 |
| 2.2.4 Gradient Boosting                              | 15 |
| 2.2.5 XGBoost                                        | 15 |
| 2.2.6 SVM – Support Vector Machines                  | 16 |
| 2.2.7 Redes neuronales                               | 17 |
| 2.2.8 Selección de variables                         | 17 |
| <b>3. DESCRIPCIÓN Y PREPARACIÓN DE LOS DATOS</b>     | 19 |
| 3.1 Descripción de las variables                     | 19 |
| 3.2 Preparación de los datos                         | 20 |
| <b>4. APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING</b> | 31 |
| 4.1 Selección de variables                           | 31 |
| 4.2 Modelado                                         | 32 |
| 4.2.1 Regresión lineal                               | 32 |
| 4.2.2 Bagging                                        | 34 |
| 4.2.3 Random Forest                                  | 36 |
| 4.2.4 Gradient Boosting                              | 36 |
| 4.2.5 XGBoost                                        | 38 |
| 4.2.6 SVM – Support Vector Machines                  | 40 |
| 4.2.7 Redes Neuronales                               | 42 |
| 4.2.8 Comparación de modelos                         | 44 |
| 4.2.9 Selección del modelo ganador                   | 46 |
| 4.2.10 Comparación de resultados con SAS             | 48 |
| <b>5. CONCLUSIONES</b>                               | 51 |
| <b>BIBLIOGRAFIA</b>                                  | 53 |
| <b>ANEXOS</b>                                        | 55 |
| <i>Tablas complementarias de resultados en R</i>     | 55 |
| <i>Código en R</i>                                   | 60 |

## Índice de tablas

|                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------|----|
| Tabla 1. Descripción de las variables. ....                                                                         | 20 |
| Tabla 2. Nueva codificación variable Age Group. ....                                                                | 22 |
| Tabla 3. Coeficientes de V de Cramer por variable. ....                                                             | 29 |
| Tabla 4. Resumen de tipos de selección de variables empleados. ....                                                 | 31 |
| Tabla 5. Resultados del modelo de regresión lineal. ....                                                            | 33 |
| Tabla 6. Resultados con diferentes Nodsize - Bagging. ....                                                          | 34 |
| Tabla 7. Resultados con diferentes mtry - Random Forest. ....                                                       | 36 |
| Tabla 8. Resultados con diferentes parámetros – SVM lineal. ....                                                    | 40 |
| Tabla 9. Resultados con diferentes parámetros - SVM-Polinomial. ....                                                | 41 |
| Tabla 10. Métricas de rendimiento - Redes neuronales probadas. ....                                                 | 44 |
| Tabla 11. RMSE y $R^2$ para los modelos probados. ....                                                              | 44 |
| Tabla 12. Pesos de los modelos empleados en el ensamblado. ....                                                     | 46 |
| Tabla 13. Valores reales y predichos con el modelo XGBoost alpha 05 para las primeras cinco (5) observaciones. .... | 47 |
| Tabla 14. RMSE para modelos con SAS Enterprise Miner. ....                                                          | 50 |
| Tabla 15. Métricas de rendimiento con diferentes parámetros para los modelos Gradient Boosting. ....                | 56 |
| Tabla 16. Métricas de rendimiento con diferentes parámetros para los modelos XGBoost. ....                          | 57 |
| Tabla 17. Métricas de rendimiento con diferentes parámetros para los modelos SVM-RF. ....                           | 60 |

## Índice de imágenes

|                                                                                                          |    |
|----------------------------------------------------------------------------------------------------------|----|
| Imagen 1. Estructura del sistema de salud en Estados Unidos. ....                                        | 7  |
| Imagen 2. Comparación de desempeño en términos de $R^2$ de los diferentes modelos<br>empleados. ....     | 9  |
| Imagen 3. Fases de la metodología CRISP-DM. ....                                                         | 12 |
| Imagen 4. Representación gráfica de un modelo de regresión lineal con dos variables<br>predictoras. .... | 13 |
| Imagen 5. Esquema del funcionamiento de Bagging. ....                                                    | 14 |
| Imagen 6. Esquema del funcionamiento de Gradient Boosting. ....                                          | 15 |
| Imagen 7. Esquema del funcionamiento de SVM con función de Kernel Lineal. ....                           | 16 |
| Imagen 8. Esquema del funcionamiento de las Redes Neuronales. ....                                       | 17 |
| Imagen 9. Distribución de la variable "Hospital County" antes y después de los cambios. ....             | 21 |
| Imagen 10. Distribución de la variable "Facility Name" ....                                              | 22 |
| Imagen 11. Distribución de la variable "Gender" antes y después de los cambios. ....                     | 23 |
| Imagen 12. Distribución de la variable "Race" antes y después de los cambios. ....                       | 23 |
| Imagen 13. Histograma de la variable "Length of Stay". ....                                              | 24 |
| Imagen 14. Distribución de la variable "Patient disposition" antes y después de los cambios. ....        | 24 |
| Imagen 15. Distribución de la variable "APR MDC Description" antes y después de los<br>cambios. ....     | 25 |
| Imagen 16. Distribución de la variable "APR Severity of illness Description" ....                        | 25 |
| Imagen 17. Distribución de la variable "APR Risk of Mortality" ....                                      | 26 |
| Imagen 18. Distribución de la variable "APR Medical Surgical Description". ....                          | 26 |
| Imagen 19. Distribución de la variable " Payment Typology 1" antes y después de los cambios.<br>.....    | 26 |
| Imagen 20. Información relevante de la variable "Birth Weight". ....                                     | 27 |
| Imagen 21. Distribución de la variable "Emergency Department Indicator". ....                            | 27 |
| Imagen 22. Proporción de datos faltantes para las variables incluidas en el dataset. ....                | 27 |
| Imagen 23. Histograma de la variable "Total Costs". ....                                                 | 28 |
| Imagen 24. Gráfico de V de Cramer de las variables incluidas en el dataset. ....                         | 29 |
| Imagen 25. Matriz de correlación entre variables numéricas. ....                                         | 30 |
| Imagen 26. Gráfica del RMSE para cada conjunto de variables. ....                                        | 31 |
| Imagen 27. Gráfica del RMSE para cada conjunto de variables sin MXM y Boruta. ....                       | 31 |
| Imagen 28. MSE con nodelist=10. ....                                                                     | 34 |
| Imagen 29. RMSE para árbol simple y bagging base. ....                                                   | 35 |
| Imagen 30. RMSE para modelos de bagging con diferente sampsize. ....                                     | 35 |
| Imagen 31. RMSE para los diferentes modelos de Random Forest probados. ....                              | 36 |
| Imagen 32. Identificación de parámetros óptimos - Gradient Boosting. ....                                | 37 |
| Imagen 33. Early Stopping - Gradient Boosting. ....                                                      | 37 |
| Imagen 34. RMSE para los diferentes modelos de Gradient Boosting. ....                                   | 38 |
| Imagen 35. Identificación de parámetros óptimos - XGBoost. ....                                          | 38 |
| Imagen 36. Early Stopping - XGBoost. ....                                                                | 39 |
| Imagen 37. RMSE para los diferentes modelos de XGBoost. ....                                             | 39 |
| Imagen 38. Valor óptimo de C - SVM Lineal. ....                                                          | 40 |
| Imagen 39. RMSE para diferentes conjuntos de parámetros - SVM RFB. ....                                  | 41 |
| Imagen 40. Valor óptimo de C - SVM Polinomial. ....                                                      | 41 |
| Imagen 41. RMSE para los diferentes modelos de SVM. ....                                                 | 42 |
| Imagen 42. RMSE con diferentes parámetros - Red Neuronal. ....                                           | 43 |
| Imagen 43. RMSE para los diferentes modelos de redes neuronales. ....                                    | 43 |
| Imagen 44. RMSE de los mejores modelos por algoritmo probado. ....                                       | 45 |
| Imagen 45. RMSE de los mejores modelos por algoritmo probado sin la regresión lineal. ....               | 45 |
| Imagen 46. RMSE modelos empleados en el ensamblado. ....                                                 | 46 |
| Imagen 47. Importancia de variables del modelo ganador. ....                                             | 47 |
| Imagen 48. Gráfico de dispersión - Modelo ganador. ....                                                  | 48 |
| Imagen 49. Diagrama empleado en SAS Enterprise Miner. ....                                               | 49 |

## 1. INTRODUCCIÓN

### 1.1 Introducción del problema

El gasto sanitario en los Estados Unidos es de los más altos en el mundo, de hecho, el gasto *per capita* es significativamente más alto que otros países desarrollados. Según un artículo publicado en *JAMA*, los costos administrativos, los precios elevados de los servicios médicos y los medicamentos son algunos de los factores que explican este comportamiento (Papanicolas et al., 2018). Esta situación no sólo genera presión sobre el sistema de salud, sino también sobre los pacientes, quienes se ven afectados por los altos costos de un servicio que debería ser universal.

Antes de hablar de las afectaciones, es importante entender cómo funciona el sistema de salud en Estados Unidos. Este sistema está conformado por un componente público y otro privado. El público está a cargo del gobierno federal, estatal y local y es canalizado principalmente a través de medicare para mayores de 65 años y personas con algún tipo de discapacidad, Medicaid y CHIP que es para personas de bajos recursos y otros programas dirigidos a segmentos específicos como lo son los veteranos de guerra o militares activos. Sin embargo, la mayor parte de los estadounidenses cuentan con seguros privados, y estos son comúnmente suministrados por el empleador, contratados directamente por el individuo o gestionados por instituciones de atención médica (Aborode et al., 2025).

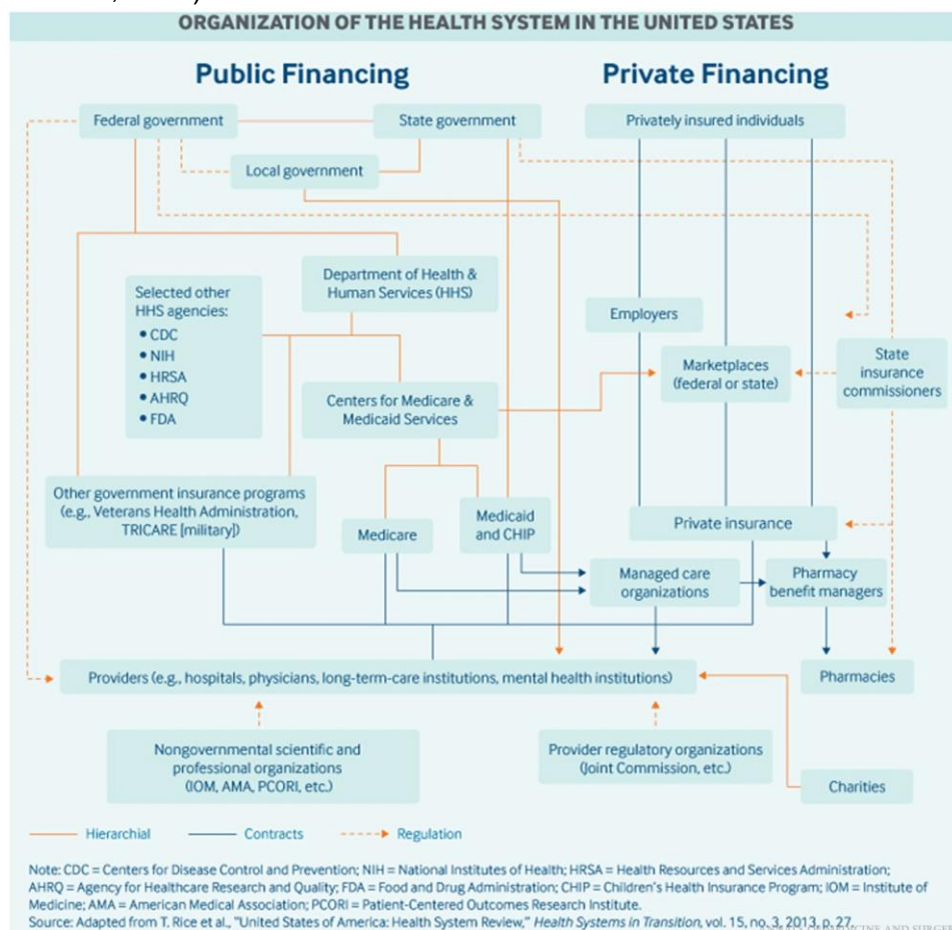


Imagen 1. Estructura del sistema de salud en Estados Unidos.

Tomado de: Aborode et al. (2025), *Healthcare debts in the United States: a silent fight*.

Esta estructura fragmentada y compleja del sistema, junto con la falta de cobertura universal, ha provocado que millones de personas se endeuden para cubrir sus gastos médicos. De hecho, personas que cuentan con seguros médicos, se enfrentan a situaciones como altos costos por emergencias, hospitalizaciones o enfermedades no cubiertas por la póliza y falta de transparencia en las facturaciones de los servicios. Esto ha generado un creciente malestar social sobre los derechos en materia de salud con numerosos casos de persona que han tenido que declararse en bancarrota a causas de sus tratamientos médicos.

Según Cunningham, el aumento de los costos médicos ha deteriorado la confianza en el sistema sanitario, generando la sensación que los médicos no actúan basados en lo que requiere el paciente sino en lo que le conviene a las aseguradoras, lo cual agrava la percepción de la ciudadanía (Cunningham, 2009).

Para poner en perspectiva esta problemática, resulta importante comparar el sistema sanitario estadounidense con otros modelos internacionales. En países como Canadá o Reino Unido, los sistemas de salud se basan en principios de cobertura universal financiada públicamente, lo que garantiza el acceso equitativo de la población sin exponerlos a gastos elevados e impagables y a pesar de operar con costos *per capita* más bajos, estos países logran indicadores de salud comparables en áreas como esperanza de vida, tasa de mortalidad evitable y satisfacción del cliente (Papanicolas et al., 2018).

El contraste del modelo estadounidense pone en evidencia no sólo asuntos de eficiencia económica, sino temas asociados a brechas en términos de justicia económica. De Acuerdo con Kaiser Family Foundation, alrededor del 24% de los encuestados dice que tiene facturas médicas o dentales que están vencidas o que no pueden pagar, uno de cada seis dice que tiene una deuda en un banco o agencias de cobranzas por préstamos utilizados para pagar estas facturas. La falta de dinero es la razón principal para endeudarse, pero asuntos asociados a la facturación y prácticas cuestionables de los seguros contribuyen a las deudas, siendo que el 67% de los encuestados que poseen deudas afirman que no han podido pagar o solo de manera parcial la factura médica porque no cuentan con el dinero suficiente (Lopes et al., 2022).

Teniendo todo este contexto, ha crecido el interés el uso de herramientas analíticas que permitan comprender y anticiparse al comportamiento de los costos sanitarios. Mas allá de optimizar la asignación de recursos, estas soluciones también buscan promover la transparencia en la gestión del sistema de salud, ofreciendo a todos los stakeholders (pacientes, aseguradoras, hospitales y gobierno) una visión más clara sobre los factores que influyen en el coste de la atención médica. Con este marco de referencia, el aprendizaje automático se presenta como una alternativa potente para identificar patrones complejos en grandes volúmenes de datos y generar modelos explicativos basados en variables clínicas, sociodemográficas y administrativas.

En los últimos años, diferentes estudios han demostrado la eficacia de los algoritmos de machine learning para abordar este problema. Por ejemplo, Morid y su equipo analizaron datos históricos de pacientes y comprobaron que utilizar información detallada sobre los

gastos mensuales y la frecuencia de uso de los servicios médicos mejora la precisión del modelo (Morid et al., 2019).

Por otra parte, Pfutzenreuter y Lima evaluaron diferentes modelos de predicción - como la regresión lineal, SVR, Random Forest, XGBoost, KNR y árboles simples- utilizando un conjunto de datos de seguros médicos. En su análisis lograron evidenciar que el modelo XGBoost obtuvo el mejor desempeño en términos de  $R^2$  (0,8873), seguido por Random Forest y el árbol simple. En este estudio se tuvieron en cuenta variables como la edad, el sexo, índice de masa muscular, número de hijo incluidos en el seguro, si eran o no fumadores y la región de residencia de los pacientes (Pfutzenreuter & Lima, 2022).

Otro estudio realizado en 2021 por Hassan, se aplican modelos de Random Forest, SVR, regresión lineal, XGBoost, arboles de decisión, KNN y Gradient Boosting, siendo este último el que mejor desempeño logró con un de  $R^2$  0,898595. Las variables empleadas fueron la edad, el sexo, el índice de masa corporal, número de hijos o personas a cargo, si son o no fumadores y la edad.

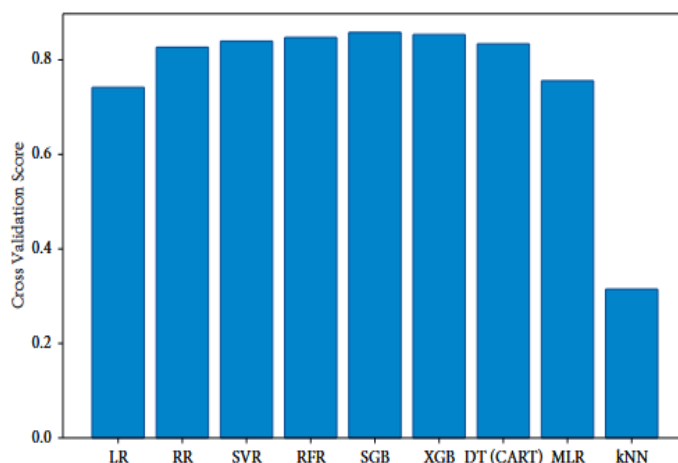


*Imagen 2. Comparación de desempeño en términos de  $R^2$  de los diferentes modelos empleados.*  
**Tomado de:** Hassan et al. (2021), A Computational Intelligence Approach for Predicting Medical Insurance Cost.

De forma complementaria, Langenberger desarrolló un modelo para identificar pacientes de alto coste en el sistema de salud alemán, utilizando también datos administrativos como números de diagnósticos principales, número de recetas, número de diagnósticos ambulatorios y proporción de gastos médicos. Sin dejar de lado variables más típicas como la edad. Como resultado final, se emplearon modelos de clasificación siendo el Random Forest el que mejor AUC obtuvo de 0,883 (Langenberger et al., 2023).

Teniendo en cuenta todos estos antecedentes, el presente trabajo de fin de máster plantea centrarse en la aplicación de diferentes algoritmos de aprendizaje automático con el objetivo de predecir el costo total de la atención médica en el estado de Nueva York. Se utilizarán un espectro de variables más amplio incluyendo sociodemográficas como edad, sexo y región, clínicas como diagnósticos principales, procedimientos clínicos, nivel de severidad del episodio clínico, riesgo de mortalidad y administrativas como forma de pago y aseguradora principal. Con este enfoque se pretende predecir el

costo total e identificar las variables que más influyen, contribuyendo a la toma de decisiones más informadas, sostenibles y transparentes.

## **1.2 Objetivos**

### **1.2.1 *Objetivo principal***

Desarrollar un modelo de predicción que permita estimar el coste total en pacientes ingresados en el estado de Nueva York a partir de características clínicas, administrativas y sociodemográficas, mediante la aplicación de técnicas de machine learning.

### **1.2.2 *Objetivos específicos***

- Identificar el set de variables más relevante para la construcción del modelo de predicción a través de diferentes técnicas de selección de variables.
- Entrenar y comparar modelos de regresión lineal y modelos de machine learning, con el propósito de predecir el costo de la atención médica de pacientes ingresados.
- Evaluar el rendimiento de los modelos empleados mediante métricas de error y ajuste que permita identificar el modelo con mejor capacidad predictiva.

## **2. METODOLOGÍA EMPLEADA**

### **2.1 Metodología CRISP-DM**

El presente trabajo se basa en la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) el cual es uno de los marcos de trabajo más ampliamente utilizados en proyectos de ciencia de datos (Chapman et al., 2000).

Esta metodología se compone de seis (6) fases principales, que se interrelacionan y no necesariamente se realizan de manera lineal, lo que convierte esta forma de trabajo en un modelo flexible y fácil de seguir. A continuación, una breve descripción de las fases y como se implementan en este trabajo de fin de máster.

#### **1. Comprensión del negocio (*Bussiness Understanding*).**

Esta etapa se centra en entender los objetivos y requerimientos del proyecto desde la perspectiva de negocio, definir los problemas que se desean resolver. En el caso de este trabajo, el objetivo principal es la predicción de los costos totales de seguro médico a partir de características demográficas, administrativas y clínicas de los pacientes. Para lo anterior, también se identificó la importancia de la industria aseguradora para el mercado analizado y las problemáticas actuales que presenta.

#### **2. Comprensión de los datos (*Data Understanding*).**

Esta segunda fase implica la recopilación de los datos y sus características principales como si cuentan con datos faltantes o atípicos, patrones de comportamiento y su calidad. En el presente trabajo se hace una identificación de todas las variables junto con su respectiva descripción, se revisan una a una si están correctamente clasificadas, presencia de NAs, errores o datos atípicos y la relación con la variable objetivo.

#### **3. Preparación de los datos (*Data Preparation*).**

En esta etapa se transforman los datos brutos en un conjunto limpio para el correcto funcionamiento de los modelos de predicción. Durante el proyecto se realizaron las imputaciones necesarias de los datos faltantes, se convirtieron las variables categóricas a dummies, se estandarizaron las variables numéricas, se transformó la variable objetivo total cost a logarítmica y, por último, se aplicaron técnicas de selección de variables para hacer los modelos de machine learning.

#### **4. Modelado (*Modeling*)**

Durante el modelado se entrenan los diferentes algoritmos de machine learning y se identifican los parámetros óptimos. También se validan las métricas de rendimiento con ayuda de técnicas robustas como validación cruzada.

Para este TFM se probaron modelos de Bagging, Random Forest, Gradient Boosting, XGBoost, SVM (Support Vector Machines) y redes neuronales, las validaciones de los modelos se hicieron con validación cruzada repetida para comprobar su robustez.

## 5. Evaluación (Evaluation)

Durante esta fase se revisan a detalle los modelos implementados para asegurarse que les dan cumplimiento a los objetivos definidos. No sólo debe evaluarse en función de métricas, también su relevancia en el contexto de la situación analizada. La decisión que se toma durante la evaluación es si con todo lo realizado ya es posible implementar el proyecto.

En este proyecto se utilizaron técnicas robustas de selección de variables y validación, haciendo uso de métricas de rendimiento como el RMSE (Raíz del Error Cuadrático Medio) y  $R^2$  (coeficiente de determinación) para identificar el modelo que más poder predictivo tiene y ayuda al cumplimiento de los objetivos definidos.

## 6. Despliegue (Deployment)

Una vez el modelo está construido y validado, se deben poner en práctica los resultados del proyecto, asegurando que existan lecciones aprendidas y que todo esté disponible para los usuarios finales.

Dado que el TFM es un ejercicio académico, el despliegue no se considerará el desarrollo en un sistema de producción automatizado, pero si en la presentación final de todos los resultados obtenidos, recomendaciones y cualquier otra información que se considere útil para el logro de los objetivos de esta investigación.

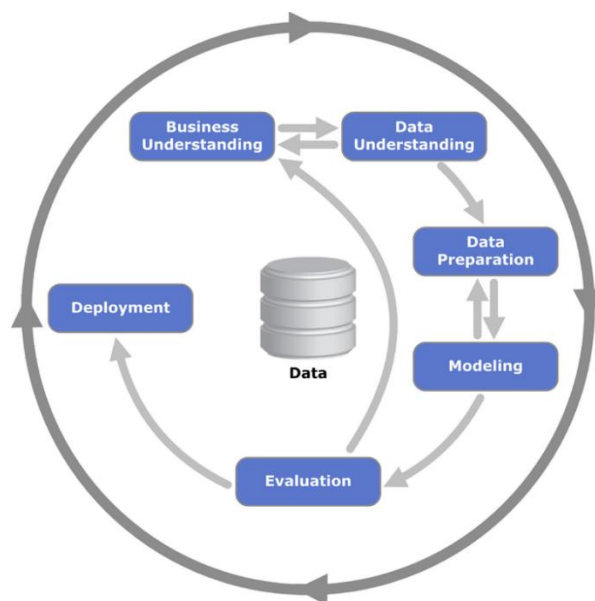


Imagen 3. Fases de la metodología CRISP-DM

Tomado de: Chapman et al. (2000), Metodología Crisp-DM Step by step data mining guide.

## 2.2 Machine learning

El Machine Learning es una disciplina que ha permitido que los sistemas aprendan patrones complejos a partir de un conjunto de datos. A través de diferentes algoritmos, estos modelos encuentran patrones y ayudan a hacer predicciones sobre nuevos casos, permitiéndonos anticipar y tomar decisiones de negocio. De acuerdo con Murdoch, “los

modelos de machine learning han demostrado gran éxito al aprender patrones complejos que les permiten realizar predicciones sobre datos no observados” (Murdoch et al., 2019). Pero estos modelos no sólo nos ofrecen solucionar problemas de predicción, también nos ayudan a entender y extraer información relevante de las variables involucradas en los casos de análisis, lo que resulta valioso en cualquier contexto de estudio.

Los modelos de machine learning se pueden clasificar, de acuerdo con los dos enfoques más comunes, en supervisado y no supervisados (Alpaydin, 2014).

El supervisado consiste en aprender una función a partir de un conjunto de datos en donde se conocen tanto las entradas (datos input) como las salidas (variable objetivo); y los no supervisados que trabaja con datos que no contienen etiquetas y su principal objetivo es encontrar estructuras o patrones entre ellos.

En este proyecto se hará uso de diferentes modelos de machine learning supervisados para predecir el costo total de las atenciones médicas.

### 2.2.1 Regresión lineal

Este es uno de los métodos estadísticos más empleados para modelizar la relación entre una variable de respuesta  $y$  y una o más variables predictoras  $(x_1, x_2, x_3, \dots, x_p)$  a través de la siguiente función:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon$$

Donde:

$y$  = Variable de respuesta (dependiente).

$\beta_0$  = Intercepto.

$\beta_x$  = Coeficiente.

$x_p$  = Variables predictoras (independientes).

$\varepsilon$  = Error aleatorio.

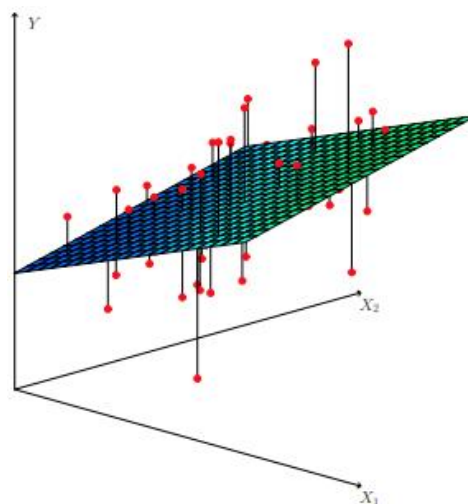


Imagen 4. Representación gráfica de un modelo de regresión lineal con dos variables predictoras.  
Tomado de: Sohil et al. (2022), An introduction to statistical learning with applications in R.

Una de las grandes ventajas de la regresión lineal frente a cualquier otro modelo de predicción es su fácil interpretabilidad, sin embargo, se limita solo a relaciones lineales entre sus variables.

### 2.2.2 Bagging

El Bagging (bootstrap aggregation) es uno de los primeros métodos de ensamblado desarrollados para mejorar la capacidad predictiva de los modelos mediante la combinación de múltiples predicciones individuales. Su principio consiste en generar múltiples muestras *bootstrap* del conjunto original de los datos, entrenar el modelo, por lo general con un árbol, sobre cada muestra y al final promediar sus predicciones. Esta estrategia lo que busca es reducir la varianza de los modelos inestables como los árboles, lo que conlleva a predicciones más estables y precisas.

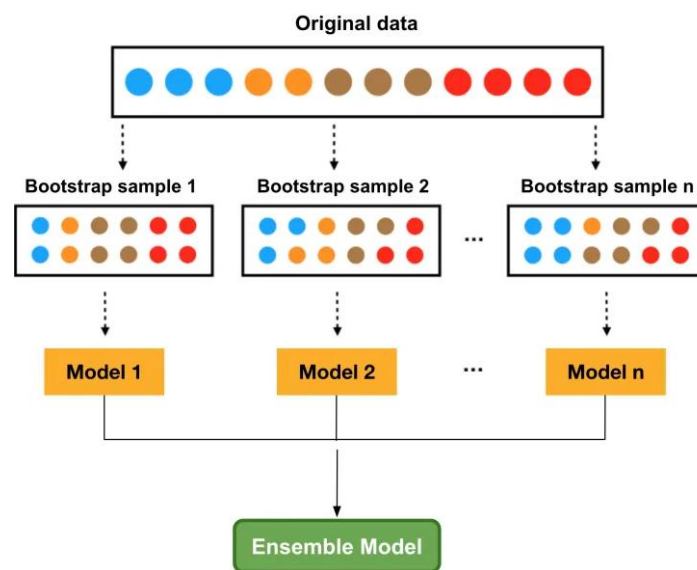


Imagen 5. Esquema del funcionamiento de Bagging.  
Tomado de: AIML.com (2022)

Este modelo es muy útil cuando se trabaja con algoritmos con alta varianza, pues su principal ventaja es que estabiliza las predicciones. Kun & Johnson indican que agregar múltiples versiones entrenadas sobre datos ligeramente distintos realmente reduce la varianza de la predicción, y por lo tanto, mejora el rendimiento (Kuhn & Johnson, 2013).

### 2.2.3 Random Forest

Este modelo es una evolución del Bagging diseñada para reducir la correlación entre los árboles generados en el ensamblado. Este algoritmo construye numerosos árboles no podados sobre muestras Bootstrap del conjunto de datos y, en cada nodo, restringe la búsqueda de la mejor partición a un subconjunto aleatorio de predictores (mtry), eligiendo la mejor división dentro de ese subconjunto. Esta aleatorización reduce la correlación entre árboles y, al promediar sus predicciones, reduce notablemente la varianza sin aumentar el sesgo y mejora la generalización (Kuhn & Johnson, 2013).

### 2.2.4 Gradient Boosting

El algoritmo de Gradient Boosting Machines (GBM) fue desarrollado por Jerome Friedman en 2001 como evolución de los métodos Boosting tradicionales.

El principio fundamental del Boosting es un ensamble que construye el modelo de forma incremental pues parte de una predicción inicial y, en cada iteración, entrena un árbol pequeño para corregir los errores que dejó el modelo previo. Esa corrección realizada se suma al modelo con un peso (tasa de aprendizaje), por lo que mejora la precisión sin caer en sobreajuste (Friedman, 2001).

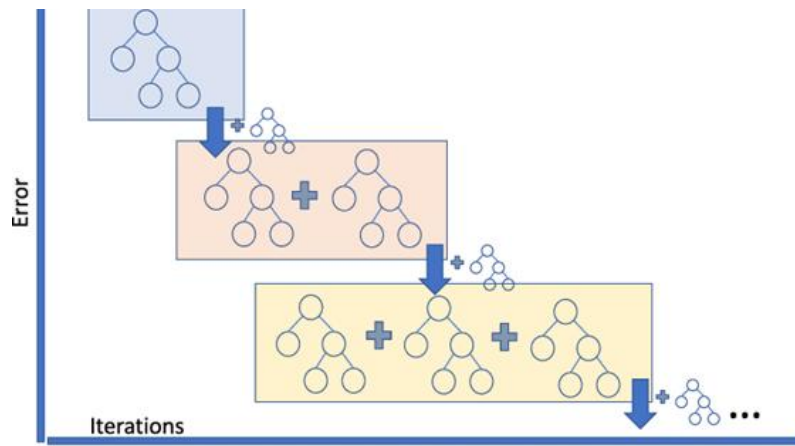


Imagen 6. Esquema del funcionamiento de Gradient Boosting.  
Tomado de: medium.com (2020)

El comportamiento de este algoritmo depende de tres decisiones:

- I. Cuantos árboles añadir: Se aumenta hasta que el error de validación deje de mejorar.
- II. El tamaño/profundidad de cada árbol: Determina cuanta interacción captura entre variables.
- III. La tasa de aprendizaje: Pesos más pequeños suelen generalizar mejor, pero requieren más árboles.

El resultado de este método es robusto, aunque menos interpretable que el desarrollo de un árbol único, sin embargo, es posible identificar la importancia de las variables.

### 2.2.5 XGBoost

El algoritmo es una versión optimizada de Gradient Boosting y mejora aspectos como la velocidad, regularización y manejo de grandes volúmenes de datos. Este modelo entrena un conjunto de árboles de forma aditiva pues parte de un modelo inicial y añade un árbol nuevo que es diseñado con el propósito de reducir al máximo la función de pérdida regularizada. Para tomar la decisión de cual árbol es más conveniente añadir, el método calcula para cada observación información de la pérdida: como cambiaría si se ajusta la predicción (gradiente de primer orden) y cuanta curvatura tiene alrededor de la predicción (información de segundo orden), lo que permite priorizar las divisiones con mayor reducción de error.

XGBoost incorpora penalización sobre el número y tamaño de las hojas y técnicas para evitar el sobreajuste como la tasa de aprendizaje y submuestreo de columnas/filas. Adicionalmente, ofrece encontrar divisiones eficaces en bases de datos grandes, lo que lo convierte en un modelo preciso y escalable (Chen & Guestrin, 2016).

### 2.2.6 SVM – Support Vector Machines

Este algoritmo tiene como objetivo encontrar una función  $f(x)$  de baja complejidad que su desviación con respecto a los valores observados no supere un margen  $\varepsilon$ , las diferencias con el valor absoluto  $\leq \varepsilon$  no se penalizan y sólo las superiores a  $\varepsilon$  contribuyen al ajuste. Su interés principal no es una predicción exacta de cada punto, sino que se encuentre dentro de un margen aceptable, lo que aporta al modelo robustez frente a valores atípicos (Cristianini & Shawe-Taylor, 2000).

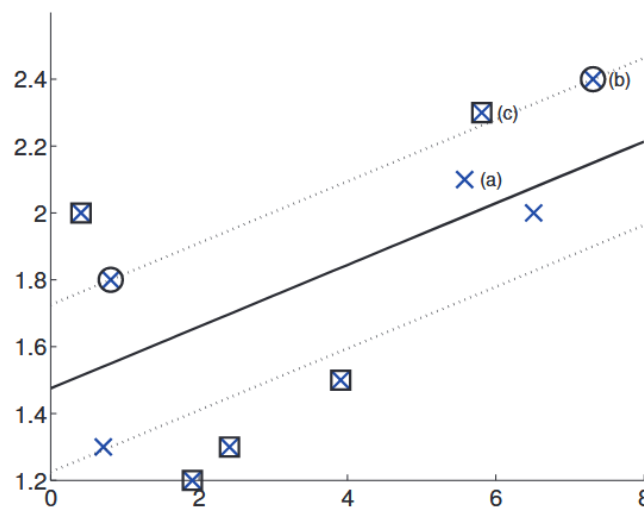


Imagen 7. Esquema del funcionamiento de SVM con función de Kernel Lineal.  
Tomado de: Alpaydin. (2014), Introduction to Machine Learning.

Este algoritmo se clasifica según el kernel empleado, que es una función de similitud que sustituye el producto escalar y permite trabajar en espacios de mayor dimensión, facilitando la captura tanto de relaciones lineales como no lineales entre variables. A continuación, una descripción de las diferentes funciones de Kernel:

- **Lineal:** Genera una función lineal de las variables. Este es adecuado cuando existe relación aproximadamente lineal entre las variables predictoras y la variable objetivo.
- **Polinomial:** Utiliza un Kernel polinomial que permite encontrar relaciones no lineales entre las variables. El modelo resultante puede representar curvas de diferente grado de complejidad, dependiendo del orden de polinomio utilizado.
- **RBF (Radial Basis Function):** Es el más flexible, transforma los datos en un espacio de características de dimensión infinita, permitiendo capturar relaciones altamente no lineales.

### 2.2.7 Redes neuronales

Es un modelo computacional que imita características del sistema nervioso humano. Una red neuronal es un sistema de procesamiento compuesto por unidades simples interconectadas o neuronas, que aprende a partir de datos ajustando los pesos sinápticos (fuerza de sus conexiones). En lugar de codificar reglas explícitas, adquiere conocimiento mediante un algoritmo de aprendizaje que modifica esos pesos para optimizar una tarea concreta, en el presente trabajo corresponde a una regresión. Gracias a su no linealidad y paralelismo, es capaz de modelar relaciones complejas y adaptarse a distintos entornos, imitando, de manera general, cómo el cerebro se adapta (Haykin & Haykin, 2009).

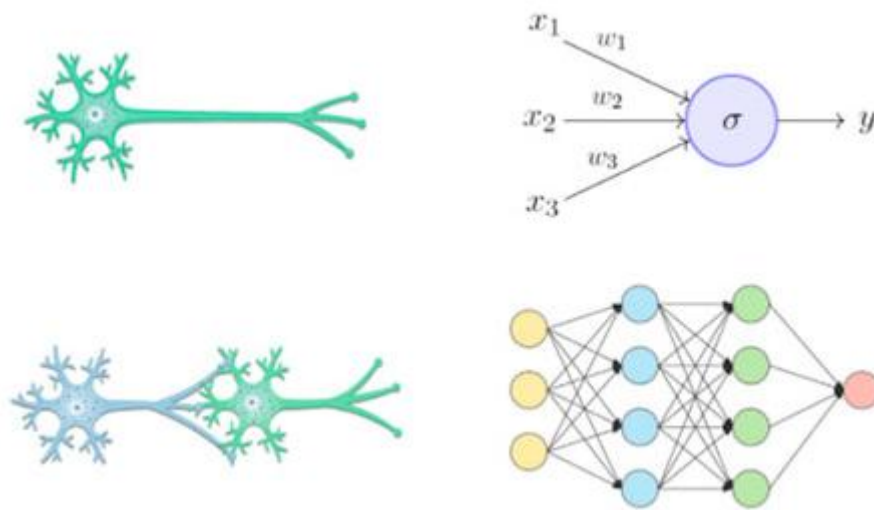


Imagen 8. Esquema del funcionamiento de las Redes Neuronales.

### 2.2.8 Selección de variables

La selección de variables es una etapa fundamental en la implementación de modelos para predicción debido a que permite disminuir dimensionalidad, mejorar la capacidad generalizadora y su interpretación.

Existen múltiples enfoques y algoritmos para seleccionar las variables. A continuación, una breve descripción de las técnicas empleadas en este trabajo.

- **SBF (Selection by filtering):** Este método consiste en aplicar un criterio estadístico para seleccionar cada variable de manera independiente. Por ejemplo, puede usarse una prueba t para comparar grupos a la correlación con la variable objetivo, y se conservan sólo aquellas que superan un umbral estadístico (Kuhn & Johnson, 2013).
- **RFE (Recursive Feature Elimination):** Este método entrena el modelo completo, calcula la importancia de cada variable y elimina las menos importantes. El proceso se repite hasta que encuentra el subconjunto óptimo (Kuhn & Johnson, 2013).

- **Stepwise AIC/BIC:** A través de estos métodos se evalúan modelos con distintos subconjuntos de variables, agregando o eliminando predictores en función de criterios de ajuste AIC (Akaike Information Criterion) o el BIC (Bayesian Information Criterion) (Kuhn & Johnson, 2013).
- **Boruta:** Este algoritmo utiliza Random Forest para medir la importancia de cada variable. Para cada predictor crea una copia aleatoria y entrena el bosque, si la importancia de la variable real es mayor que la de su copia aleatoria, se confirma como relevante; si es menor, se descarta. El objetivo de este método no es obtener un subconjunto óptimo, sino identificar todas las variables realmente relevantes (Kursa & Rudnicki, 2010).
- **MXM (Minimal Optimal Feature Selection):** Identifica aquellas variables más relevantes a través de métodos estadísticos que combinan pruebas condicionales y estrategias de búsqueda para seleccionar subconjuntos óptimos (Lagani et al., 2017).

### 3. DESCRIPCIÓN Y PREPARACIÓN DE LOS DATOS

Los datos empleados en el presente trabajo de fin de máster provienen de la base de datos SPARCS (*Statewide Planning and Research Cooperative System*), publicado por el departamento de salud del estado de Nueva York a través de su portal abierto ([https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/5dtw-tffi/about\\_data](https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/5dtw-tffi/about_data)). Este conjunto de datos incluye datos anonimizados de pacientes atendidos en diferentes hospitales del estado y su información demográfica, detalles clínicos y de la atención hospitalaria recibida y costos asociados.

#### 3.1 Descripción de las variables

La base de datos cuenta con un total de 2.000.000 de observaciones y la descripción de las variables se encuentra en la siguiente tabla.

| VARIABLE                     | NOMBRE DEL CAMPO             | DESCRIPCIÓN                                                                     |
|------------------------------|------------------------------|---------------------------------------------------------------------------------|
| Hospital Service Area        | hospital_service_area        | Área de servicio de salud (HSA) donde está ubicado el hospital.                 |
| Hospital County              | hospital_county              | Condado donde se encuentra el hospital.                                         |
| Operating Certificate Number | operating_certificate_number | Número de certificado de operación asignado por el Departamento de Salud de NY. |
| Permanent Facility Id        | permanent_facility_id        | Identificador permanente del centro hospitalario.                               |
| Facility Name                | facility_name                | Nombre del hospital según el identificador permanente (PFI).                    |
| Age Group                    | age_group                    | Grupo de edad del paciente al momento del alta.                                 |
| Zip Code - 3 digits          | zip_code_3_digits            | Tres primeros dígitos del código postal del paciente.                           |
| Gender                       | gender                       | Género del paciente.                                                            |
| Race                         | race                         | Raza del paciente.                                                              |
| Ethnicity                    | ethnicity                    | Etnicidad del paciente.                                                         |
| Length of Stay               | length_of_stay               | Número total de días de hospitalización (excluye días de ausencia).             |
| Type of Admission            | type_of_admission            | Tipo de ingreso al hospital (urgente, electivo, trauma, etc.).                  |
| Patient Disposition          | patient_disposition          | Estado o destino del paciente al momento del alta.                              |
| Discharge Year               | discharge_year               | Año del alta hospitalaria.                                                      |
| CCSR Diagnosis Code          | ccsr_diagnosis_code          | Código del diagnóstico principal (AHRQ CCSR).                                   |
| CCSR Diagnosis Description   | ccsr_diagnosis_description   | Descripción del diagnóstico principal según clasificación CCSR.                 |
| CCSR Procedure Code          | ccsr_procedure_code          | Código del procedimiento (AHRQ CCSR - ICD-10).                                  |
| CCSR Procedure Description   | ccsr_procedure_description   | Descripción del procedimiento (clasificación CCSR).                             |
| APR DRG Code                 | apr_drg_code                 | Código del grupo relacionado de diagnóstico (APR-DRG).                          |
| APR DRG Description          | apr_drg_description          | Descripción del grupo APR-DRG (versión 39, año 2022).                           |

|                                     |                                |                                                                                    |
|-------------------------------------|--------------------------------|------------------------------------------------------------------------------------|
| APR MDC Code                        | apr_mdc_code                   | Código de categoría diagnóstica mayor (MDC) refinada para todos los pacientes.     |
| APR MDC Description                 | apr_mdc_description            | Descripción de la categoría diagnóstica mayor APR.                                 |
| APR Severity of Illness Code        | apr_severity_of_illness_code   | Código del nivel de severidad del episodio clínico (APR SOI).                      |
| APR Severity of Illness Description | apr_severity_of_illness        | Descripción del nivel de severidad del episodio clínico.                           |
| APR Risk of Mortality               | apr_risk_of_mortality          | Nivel estimado de riesgo de mortalidad (APR ROM).                                  |
| APR Medical Surgical Description    | apr_medical_surgical           | Clasificación del caso como médico, quirúrgico o no aplicable según APR-DRG.       |
| Payment Typology 1                  | payment_typology_1             | Tipo principal de aseguradora o forma de pago (Medicare, Medicaid, privado, etc.). |
| Payment Typology 2                  | payment_typology_2             | Segundo tipo de forma de pago (si aplica).                                         |
| Payment Typology 3                  | payment_typology_3             | Tercer tipo de forma de pago (si aplica).                                          |
| Birth Weight                        | birth_weight                   | Peso al nacer del neonato (en gramos, redondeado a la centena más cercana).        |
| Emergency Department Indicator      | emergency_department_indicator | Indicador de si el paciente pasó por urgencias (basado en códigos de facturación). |
| Total Charges                       | total_charges                  | Total de cargos facturados al paciente.                                            |
| Total Costs                         | total_costs                    | Coste estimado total de la atención hospitalaria.                                  |

Tabla 1. Descripción de las variables.

En el presente trabajo de fin de máster se establece que la variable objetivo es **total cost**, por lo que el objetivo principal será predecir, mediante modelos de machine learning, el costo total de las atenciones hospitalarias en el estado de Nueva York.

Debido al gran volumen de datos y las limitaciones computacionales que esto conlleva, se trabajará únicamente con las observaciones correspondientes a atenciones clasificadas como **traumas** según la variable **type\_of\_admission** que corresponde a un total de 7.291 observaciones.

### 3.2 Preparación de los datos

Una vez importado el dataset, se procedió a hacer un análisis inicial que permita preparar los datos para su posterior análisis. Por lo anterior, se llevaron a cabo los siguientes puntos:

**A. Identificar los tipos de datos.** Se analizaron las características de las variables incluidas en el dataset para hacer las correcciones necesarias.

Las variables categóricas se convirtieron a factores para su mejor procesamiento.

- Hospital Service Area
- Hospital County
- Operating Certificate Number
- Permanent Facility Id
- Facility Name
- Zip Code - 3 digits

- Gender
- Race
- Ethnicity
- Type of Admission
- Patient Disposition
- CCSR Diagnosis Code
- CCSR Diagnosis Description
- CCSR Procedure Code
- CCSR Procedure Description
- APR DRG Code
- APR DRG Description
- APR MDC Code
- APR Severity of Illness Code
- APR MDC Description
- APR Severity of Illness Description
- APR Risk of Mortality
- APR Medical Surgical Description
- Payment Typology 2
- Payment Typology 3
- Payment Typology 1
- Emergency Department Indicator

**B. Análisis de cada variable.** Identificación de características de cada una de las variables incluidas en el dataset que incluye reagrupación de categorías poco representadas e identificación de valores atípicos.

### Hospital County

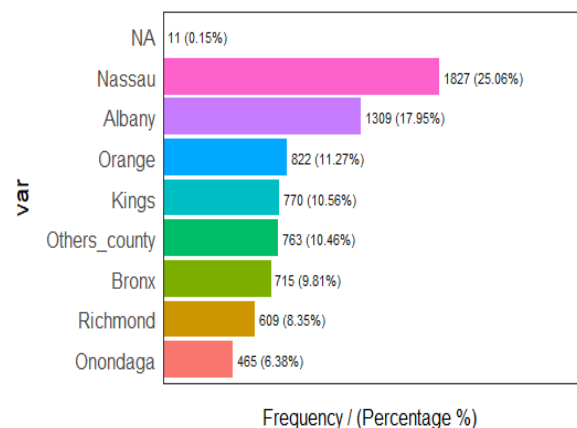
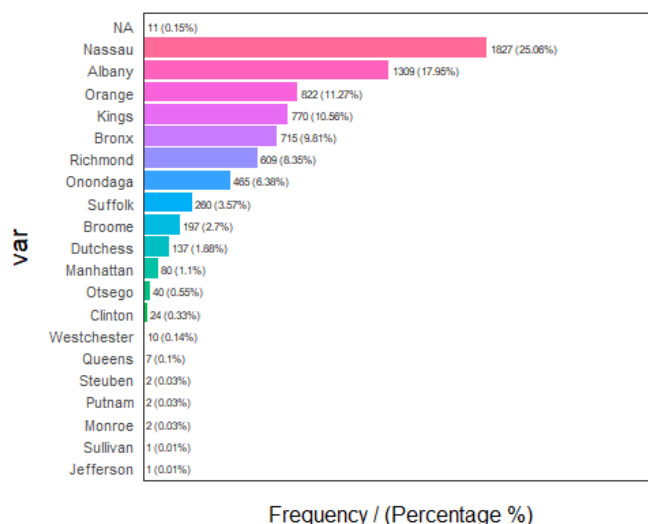


Imagen 9. Distribución de la variable "Hospital County" antes y después de los cambios.

Las categorías Suffolk, Broome, Dutchess, Manhattan, Otsego, Clinton, Westchester, Queens, Steuben, Putnam, Monroe, Sullivan y Jefferson al ser poco representadas se agrupan en una nueva categoría llamada "Others\_county". Esto con el propósito de

obtener mejores resultados cuando se implementen los diferentes modelos de machine learning al evitar el sobreajuste y reducir su complejidad.

**Hospital Service Area**

Esta categoría también representa información geográfica de los hospitales. Se toma la decisión de eliminarla y sólo conservar Hospital County porque la información proporcionada es más detallada y ampliamente conocida. Mantener ambas variables podría introducir redundancia y generar problemas de multicolinealidad al modelo.

**Operating Certificate Number - Permanent Facility Id**

Estas variables se eliminan del análisis por tratarse de identificadores que no aportan valor explicativo ni predictivo al modelo.

**Facility Name – Zip Code**

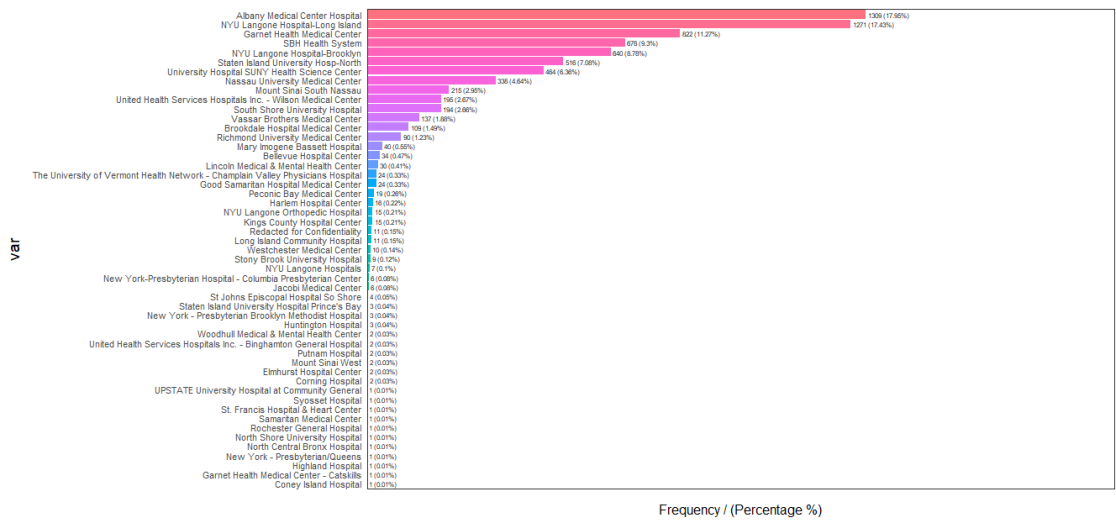


Imagen 10. Distribución de la variable "Facility Name"

Estas variables fueron excluidas del análisis debido a su alta cardinalidad, es decir, cuenta con más de 60 categorías, muchas de ellas con muy baja representación lo que pudiera introducir ruido en el modelo y afectar su capacidad predictiva. Adicionalmente, el conjunto de datos ya cuenta con información asociada a lugares de atención, por lo que este componente está presente en los modelos que se vayan a realizar.

**Age group**

Esta variable es categórica y representa los diferentes grupos de edad de los pacientes atendidos. Sin embargo, se toma la decisión de codificar cada una de estas categorías de forma numérica ordenada para explorar si se logra encontrar una relación con la variable objetivo.

| Clasificación inicial | Codificación |
|-----------------------|--------------|
| 0-17                  | 1            |
| 18-29                 | 2            |
| 30-49                 | 3            |
| 50-69                 | 4            |
| 70+                   | 5            |

Tabla 2. Nueva codificación variable Age Group.

## Gender

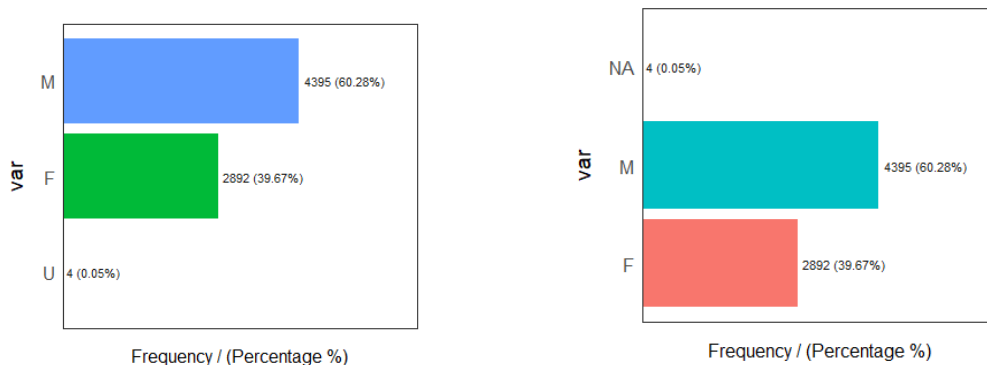


Imagen 11. Distribución de la variable "Gender" antes y después de los cambios.

Gender posee una categoría llamada "U" que es poco representada (4 observaciones), por lo que se decide convertirla sus observaciones a NAs para su posterior tratamiento.

## Race – Ethnicity

Ambas variables representan información asociada al origen del paciente, por lo que se decide conservar Race y eliminar Ethnicity para evitar redundancias en el modelo. Esta decisión se basa en que las categorías de la variable seleccionada cuentan con una distribución más balanceada y no contiene valores faltantes.

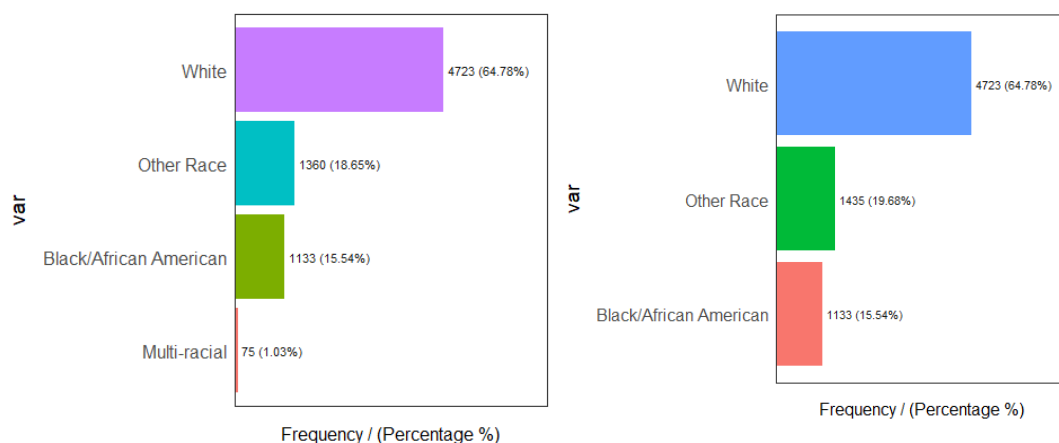


Imagen 12. Distribución de la variable "Race" antes y después de los cambios.

La categoría menos representada de Race (1,03%) es la Multi-racial, por lo que se decide agruparla con la categoría "Other race" y permitir mayor robustez en los modelos a emplear y evitar el sobreajuste.

## Length of Stay

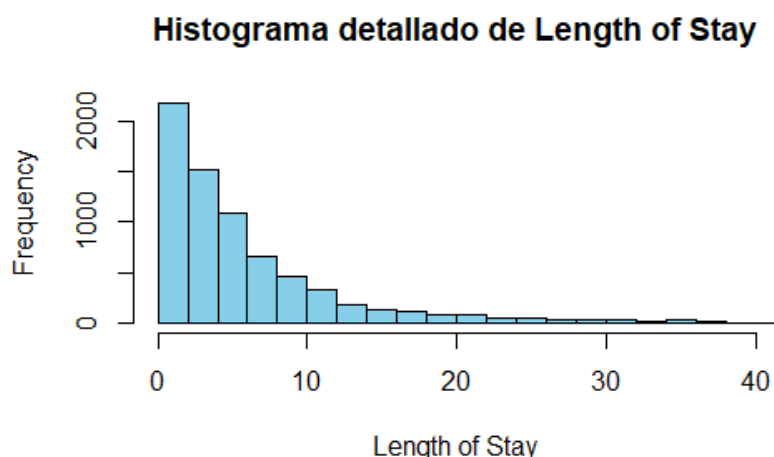


Imagen 13. Histograma de la variable "Length of Stay".

Length of stay representa el número total de días de hospitalización. Al revisar la distribución de los datos se puede observar que hay mayor concentración en valores menores que 20 y una menor proporción por encima de este umbral. Sin embargo, por la naturaleza del análisis, se decide no considerar estos últimos como valores atípicos, sino considerarlos como parte de la variabilidad esperada del proceso y como esta contribuye a la construcción del modelo de predicción.

## Type of Admission

Esta variable es eliminada porque ya fue elegida como criterio de filtrado para definir la muestra del presente trabajo, seleccionando únicamente las clasificadas como trauma.

## Patient Disposition

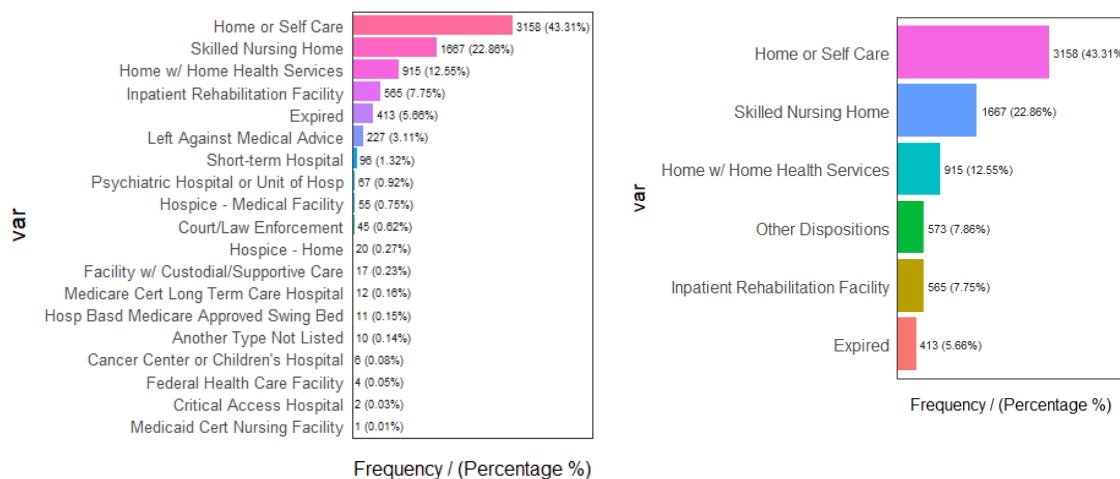


Imagen 14. Distribución de la variable "Patient disposition" antes y después de los cambios.

Esta variable contaba con muchas categorías poco representadas por lo que se toma la decisión de agruparlas como "Other Dispositions". Esto con el propósito de obtener mejores resultados cuando se implementen los diferentes modelos de machine learning al evitar el sobreajuste y reducir su complejidad.

### Discharge Year

Esta variable es eliminada debido a que todos los registros corresponden al mismo año (2022), lo que no contribuye variabilidad al modelo.

### CCSR Diagnosis Code - CCSR Diagnosis Description - CCSR Procedure Code - CCSR Procedure Description - APR DRG Code - APR DRG Description

Estas variables contienen información clínica detallada de los pacientes. Sin embargo, cuentan con muchos niveles (todas mayores a 320), por lo que podrían generar ruido, hacer más compleja la interpretabilidad del modelo y afectar su capacidad predictiva. Por lo anterior, se toma la decisión de eliminarlas del análisis y conservar otras variables con menos niveles que también aporten información clínica relevante para la predicción de los costos.

### APR MDC Code

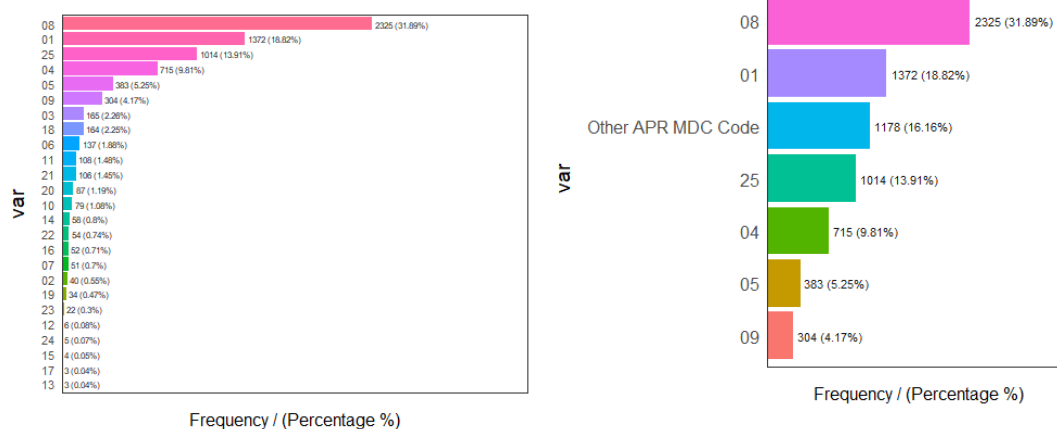


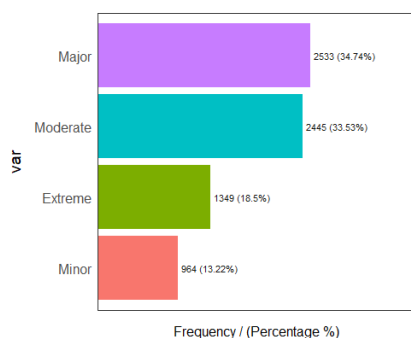
Imagen 15. Distribución de la variable "APR MDC Description" antes y después de los cambios.

Esta variable contaba con muchas categorías poco representadas por lo que se toma la decisión de agruparlas como "Other APR MDC Code". Esto con el propósito de obtener mejores resultados cuando se implementen los diferentes modelos de machine learning al evitar el sobreajuste y reducir su complejidad.

### APR MDC Description

La variable corresponde a la descripción de los códigos empleados en la variable APR MDC Code. Teniendo en cuenta que representan la misma información, se elimina del análisis para evitar redundancia en el modelo.

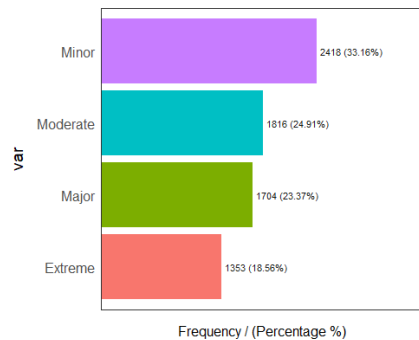
### APR Severity of Illness Code - APR Severity of Illness Description



Las variables APR Severity of Illness Code y APR Severity of Illness Description representan el mismo concepto clínico. Debido a que las categorías están bien representadas y puede permitir capturar posibles relaciones entre los costos de acuerdo con el nivel de severidad del episodio clínico, se decide trabajar sólo con la descripción para evitar redundancia en el modelo.

Imagen 16. Distribución de la variable "APR Severity of illness Description"

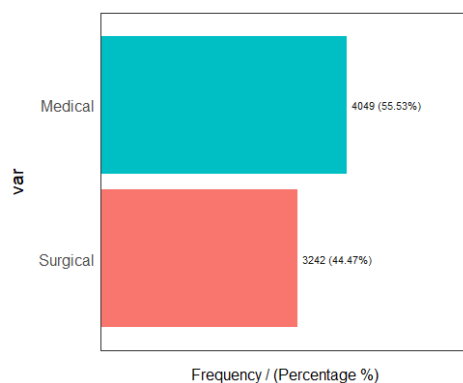
### APR Risk of Mortality



Esta variable se mantiene igual teniendo en cuenta que las categorías están bien representadas y puede permitir capturar posibles diferencias entre los costos de acuerdo con los niveles de riesgos de mortalidad.

Imagen 17. Distribución de la variable "APR Risk of Mortality".

### APR Medical Surgical Description



Esta variable se mantiene igual teniendo en cuenta que las categorías están bien representadas y puede permitir capturar posibles diferencias entre los costos de acuerdo con la clasificación del caso médico.

Imagen 18. Distribución de la variable "APR Medical Surgical Description".

### Payment Typology 1 - Payment Typology 2 - Payment Typology 3

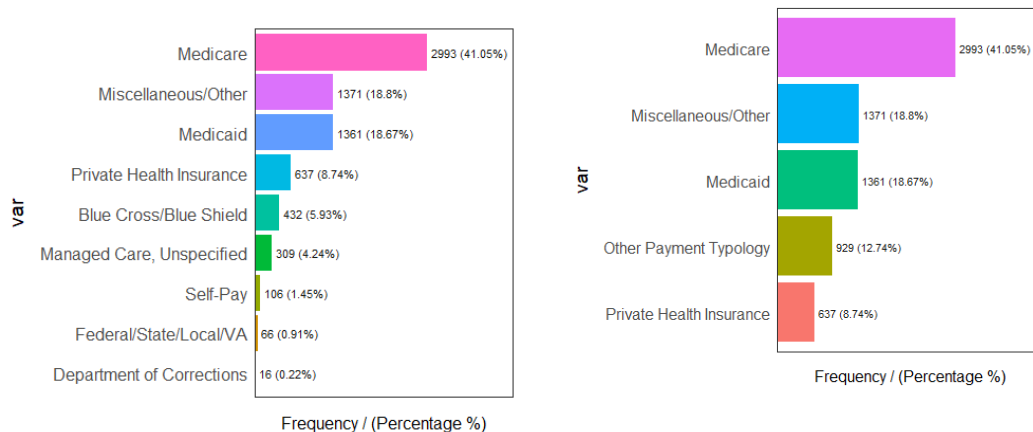


Imagen 19. Distribución de la variable "Payment Typology 1" antes y después de los cambios.

Payment Typology 1 representa el tipo principal de aseguradora o forma de pago, mientras que las otras dos variables corresponden a otras formas de pago si se utilizaron diferentes fuentes. En la mayoría de los casos, el pago se realizó con una sola fuente, por lo que las variables Payment Typology 2 y Payment Typology 3 cuentan con muchos valores faltantes, 3433 y 6483 respectivamente.

Por este motivo, se toma la decisión de eliminar Payment Typology 2 y Payment Typology 3 del análisis y reagrupar las categorías poco representadas del Payment

Typology 1 en una nueva categoría llamada “Other Payment Typology”, lo que permite facilitar la interpretabilidad del modelo y evitar el sobreajuste.

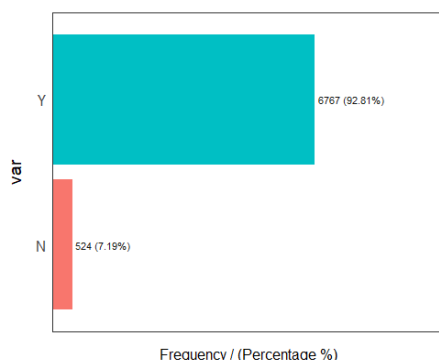
### Birth Weight

```
Birth weight
Min. : 600
1st Qu.: 2300
Median : 3400
Mean : 2780
3rd Qu.: 3700
Max. : 3900
NA's : 7286
```

Birth Weight corresponde al peso al nacer del neonato. Sin embargo, dado que el análisis se basa en atenciones hospitalarias por traumas, la gran mayoría de datos corresponden a pacientes no neonatales. Como consecuencia, esta variable representa un porcentaje muy alto de valores faltantes (99.93%), por lo que se decide eliminarla.

Imagen 20. Información relevante de la variable "Birth Weight".

### Emergency Department Indicator



Esta variable indica si el paciente ingresó o no por urgencias. Al foco del estudio estar en los atendidos por traumas, es razonable que la mayor proporción se encuentre en aquellos que si ingresaron por urgencias (Y) con un 92,87% del total de casos atendidos.

Imagen 21. Distribución de la variable "Emergency Department Indicator".

### Total Charges

Total Charges representa el total de cargos facturados al paciente antes de aplicar descuentos, coberturas de seguro u otros ajustes. Este valor se basa en tarifas internas que pueden variar entre instituciones (hospitales/aseguradoras).

Se decide eliminar esta variable ya que, aunque no equivale directamente el costo total de la atención, si guardan una relación estrecha, por lo que podría generar redundancia en el modelo y afectar su interpretabilidad.

**C. Identificación de valores faltantes e imputaciones.** Para cada variable incluida en el dataset se identifica la proporción de valores faltantes (NAs) y se realizan las imputaciones correspondientes.

```
> sort(round(colMeans(is.na(datos_trauma)), 4), decreasing = TRUE)
Hospital County Length of Stay
0.0015 0.0011
Total Costs Gender
0.0008 0.0005
Age Group Race
0.0000 0.0000
Patient Disposition APR MDC Code
0.0000 0.0000
APR Severity of Illness Description APR Risk of Mortality
0.0000 0.0000
APR Medical Surgical Description Payment Typology 1
0.0000 0.0000
Emergency Department Indicator
0.0000
```

Imagen 22. Proporción de datos faltantes para las variables incluidas en el dataset.

Al analizar la proporción de datos faltantes, se puede observar que ninguna supera el 1%, siendo Hospital County la variable con más NAs con un 0,15%.

Por lo anterior, se toma la decisión de trabajar con todas las variables y hacer el proceso de imputación haciendo uso del método aleatorio incluido en el script FuncionesMineria.R. Este método aplicado a las variables categóricas reemplaza los valores faltantes seleccionando aleatoriamente entre las categorías observadas en el conjunto de datos. Para la variable cuantitativa, la imputación se realiza generando valores a partir de la distribución de la variable, de modo que los valores imputados respetan su distribución original, esto para preservar la variabilidad de los datos y no incluir sesgos.

**D. Análisis de la variable objetivo.** Identificar distribución de los datos de la variable total costs.

Total Costs es el coste estimado total de la atención hospitalaria. Al analizar su distribución se pueden observar valores extremos, por lo que se toma la decisión de aplicar una transformación logarítmicamente para disminuir su influencia en los modelos de predicción.

**Histograma detallado de Total Costs**

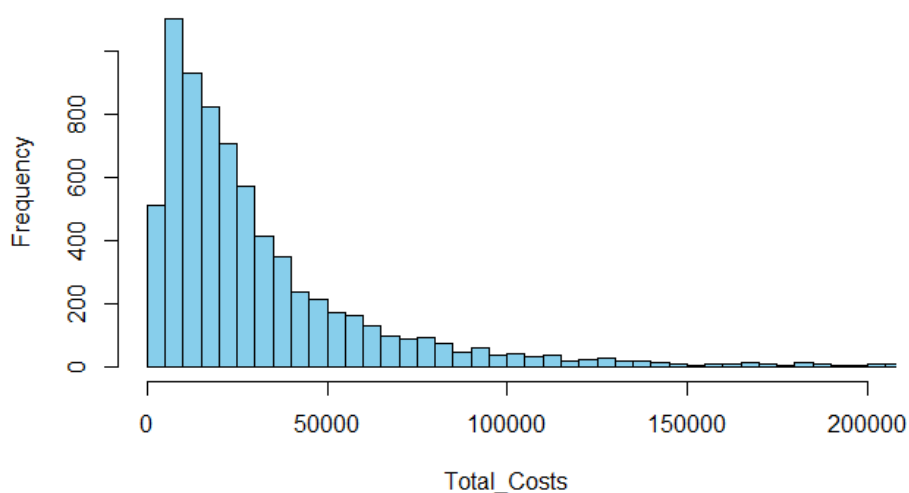


*Imagen 23. Histograma de la variable "Total Costs".*

**E. Identificación de relaciones entre variables.** Para identificar la relación que tienen la variable objetivo con las demás variables categóricas del dataset se hace el cálculo de la V de Cramer y para encontrar la relación con las variables numéricas se calcula el índice de correlación de Pearson.

## V de Cramer

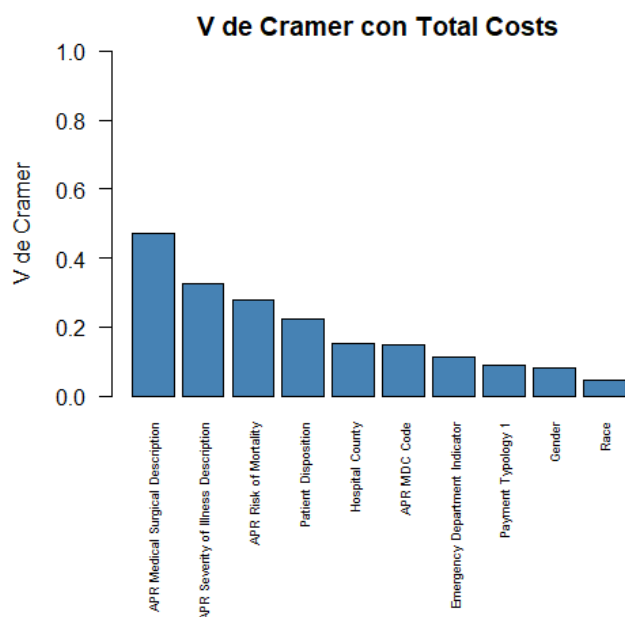


Imagen 24. Gráfico de V de Cramer de las variables incluidas en el dataset.

Al analizar la relación de las variables categóricas con la variable objetivo (total costs) mediante la V de Cramer, se puede identificar que la variable con mayor asociación es APR Medical Surgical Description, seguida de APR Severity of Illness Description y APR Risk of Mortality. Sin embargo, ninguno de los coeficientes es mayor a 0,5, lo que indica que no hay ninguna variable que cuente con una relación fuerte con los costos médicos totales, es decir, si bien estas variables pueden aportar valor predictivo al modelo, su influencia individual es limitada.

Los coeficientes de V de Cramer calculado para cada variable se muestran a continuación:

| VARIABLE                            | V DE CRAMER |
|-------------------------------------|-------------|
| APR Medical Surgical Description    | 0,4705      |
| APR Severity of Illness Description | 0,3260      |
| APR Risk of Mortality               | 0,2771      |
| Patient Disposition                 | 0,2238      |
| Hospital County                     | 0,1528      |
| APR MDC Code                        | 0,1501      |
| Emergency Department Indicator      | 0,1149      |
| Payment Typology 1                  | 0,0881      |
| Gender                              | 0,0809      |
| Race                                | 0,0480      |

Tabla 3. Coeficientes de V de Cramer por variable.

### Matriz de correlación

La gráfica muestra el coeficiente de correlación de Pearson entre las variables numéricas incluidas en el dataset y los coeficientes representan la fuerza y la dirección de cada par de variables.

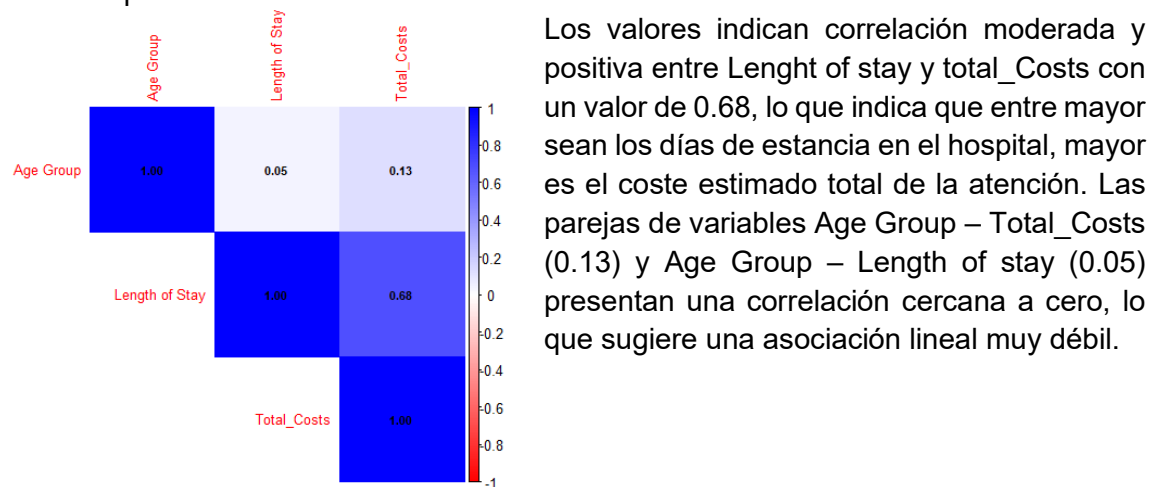


Imagen 25. Matriz de correlación entre variables numéricas.

- F. Creación de dummies y estandarización de variables.** Se crean las dummies para las variables categóricas y se estandarizan las variables continuas con el fin de normalizar su escala y facilitar el entrenamiento de los modelos.

Con todo el dataset preparado, se procede a la siguiente fase de modelización de los datos.

## 4. APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING

### 4.1 Selección de variables

Para la selección de variables se usan los siguientes métodos con validación cruzada repetida:

| MÉTODO DE SELECCIÓN | RMSE      | $R^2$     | NO. DE VARIABLES SELECCIONADAS |
|---------------------|-----------|-----------|--------------------------------|
| SBF                 | 0,4847916 | 0,753318  | 37                             |
| RFE                 | 0,4847916 | 0,753318  | 45                             |
| STEP-AIC            | 0,484299  | 0,7538227 | 26                             |
| STEP-BIC            | 0,4842837 | 0,753838  | 24                             |
| Boruta              | 0,6536051 | 0,5529307 | 4                              |
| MXM                 | 0,5022133 | 0,7353359 | 13                             |
| STEP-REP-AIC        | 0,484277  | 0,7538464 | 25                             |
| STEP-REP-BIC        | 0,484973  | 0,7531395 | 21                             |

Tabla 4. Resumen de tipos de selección de variables empleados.

A continuación, una representación gráfica del RMSE (Root Mean Squared Error), para los diferentes métodos empleados:

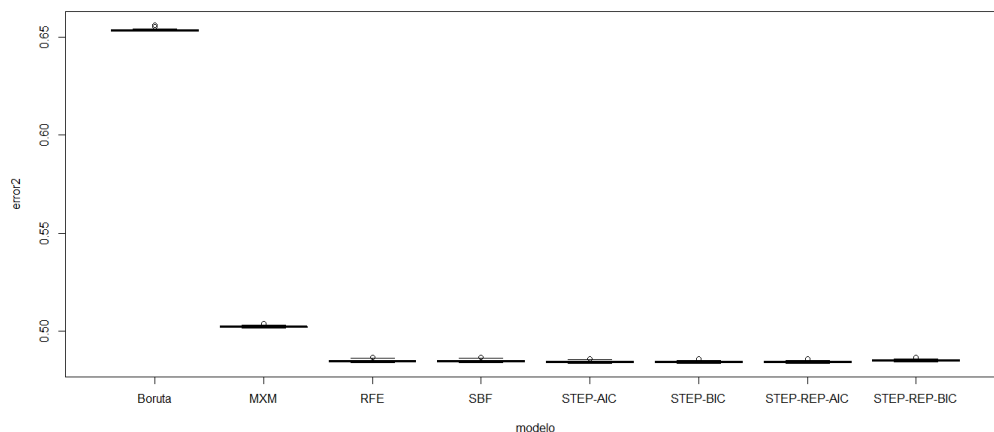


Imagen 26. Gráfica del RMSE para cada conjunto de variables.

Teniendo en cuenta que el MXM y Boruta tienen un RMSE alto, se eliminan del gráfico para poder determinar, con una mejor escala, el resultado de los otros métodos empleados:

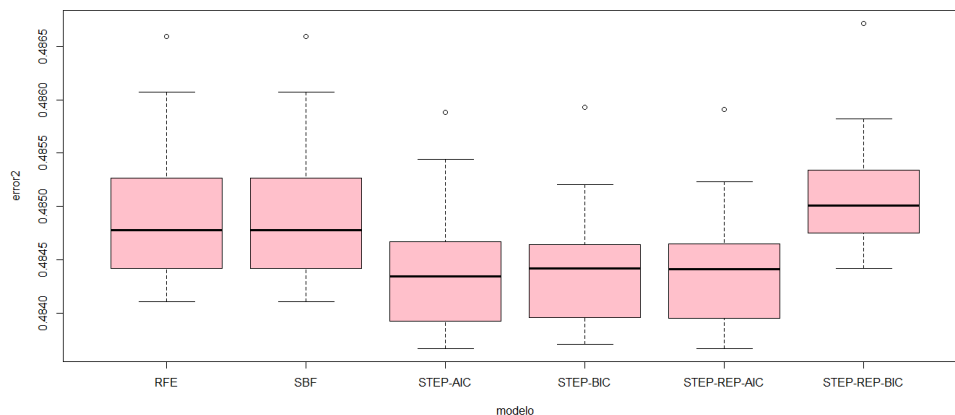


Imagen 27. Gráfica del RMSE para cada conjunto de variables sin MXM y Boruta.

Basado en el error, el  $R^2$  y el número de variables seleccionadas, se decide trabajar con las siguientes variables obtenidas con el método STEP-REP-BIC:

- Length of Stay
- APR Medical Surgical Description Medical
- APR Risk of Mortality Minor
- Hospital County Nassau
- Hospital County Kings
- Patient Disposition Home or Self Care
- Hospital County Onondaga
- APR Risk of Mortality Moderate
- APR MDC Code 09
- Emergency Department Indicator N
- Hospital County Albany
- APR Severity of Illness Description Minor
- APR Severity of Illness Description Moderate
- Patient Disposition Other Dispositions
- APR Severity of Illness Description Extreme
- Hospital County Bronx
- Patient Disposition Expired
- Race Other Race
- Hospital County Orange
- APR Risk of Mortality Extreme
- APR MDC Code 25

Este modelo presenta un RMSE,  $R^2$  y dispersión muy parecida al resto de set de variables, pero un número más reducido de variables en comparación con los otros modelos, lo que favorece su interpretabilidad sin comprometer su rendimiento predictivo.

## 4.2 Modelado

### 4.2.1 Regresión lineal

Como primer modelo a implementar, se procede con la definición de la fórmula de la regresión lineal para su análisis y comparación posterior con los diferentes modelos de machine learning empleados con el fin de determinar cuál es el que mejor puede predecir el comportamiento de los costos totales médicos.

Call:

```
lm(formula = Total_Costs ~ +`Hospital County` + `Age Group` + Gender + Race + `Length of Stay` + `Patient Disposition` + `APR MDC Code` + `APR Severity of Illness Description` + `APR Risk of Mortality` + `APR Medical Surgical Description` + `Payment Typology 1` + `Emergency Department Indicator`, data = datos_completo)
```

Residuals:

| Min     | 1Q      | Median | 3Q     | Max    |
|---------|---------|--------|--------|--------|
| -4.0884 | -0.2679 | 0.0227 | 0.2953 | 3.3252 |

Residual standard error: 0.4828 on 7247 degrees of freedom

Multiple R-squared: 0.7561, Adjusted R-squared: 0.7549

F-statistic: 641.8 on 35 and 7247 DF, p-value: < 2.2e-16

| <b>Coefficients:</b>                                    | <b>Estimate</b> | <b>Std. Error</b> | <b>t value</b> | <b>Pr(&gt; t )</b> |
|---------------------------------------------------------|-----------------|-------------------|----------------|--------------------|
| (Intercept)                                             | 8,9359054       | 0,0515426         | 173,369        | < 2e-16            |
| `Hospital County` Bronx                                 | 0,1408686       | 0,0260349         | 5,411          | 6,48E-08           |
| `Hospital County` Kings                                 | 0,7687103       | 0,023798          | 32,301         | < 2e-16            |
| `Hospital County` Nassau                                | 0,744191        | 0,0195358         | 38,094         | < 2e-16            |
| `Hospital County` Onondaga                              | 0,5277181       | 0,0267415         | 19,734         | < 2e-16            |
| `Hospital County` Orange                                | 0,1553189       | 0,0228902         | 6,785          | 1,25E-11           |
| `Hospital County` Others_county                         | 0,2895707       | 0,022845          | 12,675         | < 2e-16            |
| `Hospital County` Richmond                              | 0,2617027       | 0,025018          | 10,461         | < 2e-16            |
| `Age Group`                                             | 0,0146318       | 0,0068904         | 2,123          | 0,033747           |
| GenderM                                                 | -0,0063683      | 0,0123133         | -0,517         | 6,05E-01           |
| RaceOther Race                                          | 0,075813        | 0,0206378         | 3,673          | 0,000241           |
| RaceWhite                                               | 0,0077554       | 0,0181815         | 0,427          | 6,70E-01           |
| `Length of Stay`                                        | 0,0518184       | 0,0007782         | 66,591         | < 2e-16            |
| `Patient Disposition` Home or Self Care                 | -0,0559538      | 0,0298502         | -1,874         | 0,060904           |
| `Patient Disposition` Home w/ Home Health Services      | 0,1410923       | 0,0321413         | 4,39           | 1,15E-05           |
| `Patient Disposition` Inpatient Rehabilitation Facility | 0,1917902       | 0,0341083         | 5,623          | 1,95E-08           |
| `Patient Disposition` Other Dispositions                | -0,0590416      | 0,0337233         | -1,751         | 0,080028           |
| `Patient Disposition` Skilled Nursing Home              | 0,1630821       | 0,0301925         | 5,401          | 6,82E-08           |
| `APR MDC Code` 04                                       | -0,0270536      | 0,0227577         | -1,189         | 0,23457            |
| `APR MDC Code` 05                                       | 0,033886        | 0,0283165         | 1,197          | 0,231469           |
| `APR MDC Code` 08                                       | 0,0484026       | 0,0180627         | 2,68           | 7,39E-03           |
| `APR MDC Code` 09                                       | -0,2472102      | 0,031248          | -7,911         | 2,93E-15           |
| `APR MDC Code` 25                                       | 0,0991939       | 0,0216926         | 4,573          | 4,89E-06           |
| `APR MDC Code` Other APR MDC Code                       | 0,0003212       | 0,019661          | 0,016          | 0,986968           |
| `APR Severity of Illness Description` Major             | -0,1499108      | 0,0241151         | -6,216         | 5,37E-10           |
| `APR Severity of Illness Description` Minor             | -0,4882355      | 0,0334879         | -14,579        | < 2e-16            |
| `APR Severity of Illness Description` Moderate          | -0,3263808      | 0,0290559         | -11,233        | < 2e-16            |
| `APR Risk of Mortality` Major                           | -0,0958416      | 0,0237901         | -4,029         | 5,67E-05           |
| `APR Risk of Mortality` Minor                           | -0,2301993      | 0,0295587         | -7,788         | 7,76E-15           |
| `APR Risk of Mortality` Moderate                        | -0,1941205      | 0,0271479         | -7,15          | 9,49E-13           |
| `APR Medical Surgical Description` Surgical             | 0,6489322       | 0,013455          | 48,23          | < 2e-16            |
| `Payment Typology 1` Medicare                           | 0,0097759       | 0,0208552         | 0,469          | 0,639263           |
| `Payment Typology 1` Miscellaneous/Other                | -0,0337497      | 0,02051           | -1,646         | 0,099906           |
| `Payment Typology 1` Other Payment Typology             | -0,011875       | 0,0226071         | -0,525         | 0,599406           |
| `Payment Typology 1` Private Health Insurance           | 0,025968        | 0,0242289         | 1,072          | 0,283855           |
| `Emergency Department Indicator` Y                      | 0,2663954       | 0,0238201         | 11,184         | < 2e-16            |

Tabla 5. Resultados del modelo de regresión lineal.

Las categorías de referencia son: Hospital County (Albany), Gender (F), Race (Black/African American), Patient Disposition (Expired), APR MDC Code (01), APR Severity of Illness Description (Extreme), APR Risk of Mortality (Extreme), APR Medical Surgical Description (Medical), Payment Typology 1 (Medicaid) y Emergency Department Indicator (N).

El coeficiente negativo para las variables categóricas incluidas en el modelo indica que pertenecer a esa categoría específica reduce el valor que toma la variable objetivo en comparación con la categoría de referencia, contrario a lo que ocurre cuando el coeficiente es positivo. La variable numérica con el coeficiente negativo señala que tiene una relación inversamente proporcional con la variable objetivo, es decir, a medida que el valor que toma la variable aumenta, el valor de la variable objetivo disminuye.

Por ejemplo:

- La variable Length of Stay con coeficiente de 0,0518184 indica que, por cada unidad adicional en los días de estancia en el hospital, el logaritmo del costo total aumenta en 0,05 unidades logarítmicas.
- El coeficiente de APR Risk of Mortality\_Major (-0,0958416), indica que los pacientes categorizados con riesgo de mortalidad importante tienen una disminución de 0,0958 unidades en el logaritmo del costo total, en comparación la categoría de referencia (Extreme).

#### 4.2.2 Bagging

Con el grupo de variables seleccionadas en el ítem 4.1, se tunean los parámetros `mtry=21` teniendo en cuenta que ese es el número de variables independientes que tengo en mi dataset y se usa validación cruzada repetida.

Sabiendo que este modelo está basado en árboles, se evaluó el impacto del parámetro `nodesize` que representa el número mínimo de observaciones requeridas en una hoja terminal. Se prueban diferentes configuraciones con 1000 árboles para determinar el óptimo basado en el mejor desempeño del MSE, obteniendo los siguientes resultados:

| Nodesize | Ntrees | MSE       |
|----------|--------|-----------|
| 10       | 1000   | 0,1544514 |
| 20       | 1000   | 0,1712199 |
| 30       | 1000   | 0,1934678 |

Tabla 6. Resultados con diferentes Nodsize - Bagging

A medida que se incrementan los nodos, aumenta el error, y esto se debe a que se están creando modelos más simples pues tienen menor profundidad. Por lo anterior, se toma la decisión de trabajar Bagging con `Nodsize = 10` teniendo en cuenta que su error es el menor de los probados.

Para la elección del número mínimo de árboles se utiliza el MSE (Error Cuadrático Medio) y a través del gráfico se puede observar el error cometido sobre test a medida que avanzan las iteraciones.

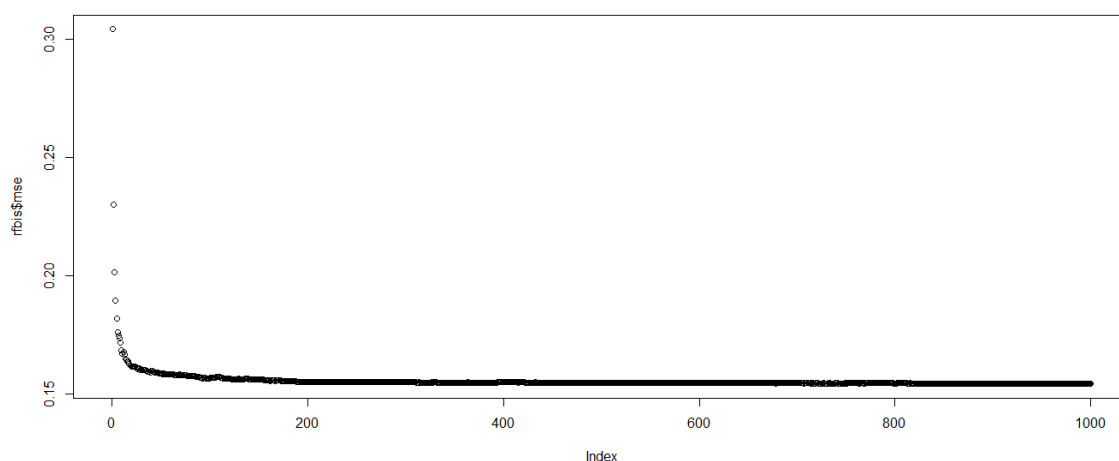


Imagen 28. MSE con `nodsize=10`.

En este caso se observa que a partir de los 200 árboles se estabiliza este valor, por lo que se toma este como su parámetro óptimo.

También se prueba con un árbol simple para comparar el rendimiento con bagging y obtenemos los siguientes resultados:

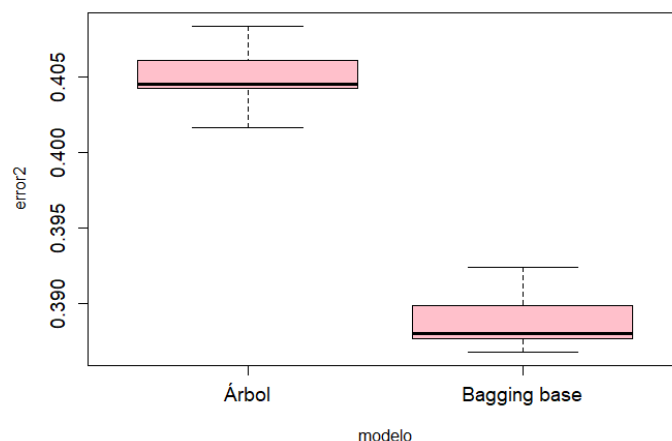


Imagen 29. RMSE para árbol simple y bagging base.

El árbol tiene peor rendimiento que el Bagging base, por lo que se seguirán explorando diferentes modelos de Bagging tuneando sus parámetros.

Para seguir analizando este modelo se tunea el parámetro sampsize con 10 grupos de validación cruzada y 5 número de repeticiones para lograr apreciar más el efecto. Adicionalmente, al tener 7283 observaciones con 10 grupos de validación cruzada, el máximo tamaño de muestra que se puede usar es  $0,9 \times 7283 = 6554$  observaciones, por lo que se construyeron modelos con sampsize de 1000, 3000 y 5000.

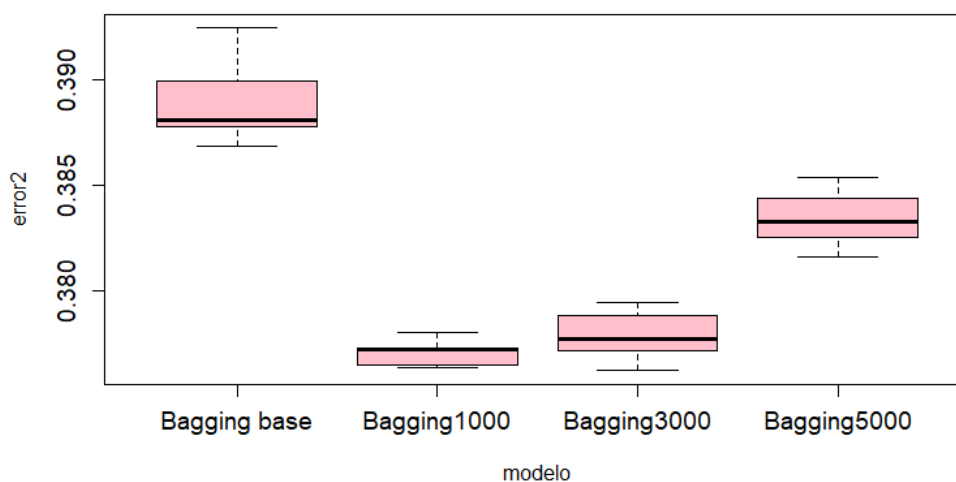


Imagen 30. RMSE para modelos de bagging con diferente sampsize.

El mejor Bagging fue el Bagging1000, configurado con nodesize=10, ntree=200 y sampsize=1000, ya que logró el menor RMSE entre todas las combinaciones evaluadas. Este será el modelo tomado como referencia para la comparación final con el resto de los algoritmos.

### 4.2.3 Random Forest

Para el modelo de Random Forest se prueban diferentes valores de mtry haciendo uso de validación cruzada repetida con 200 árboles y sampsize = 1000.

| mtry | RMSE      | $R^2$     | MAE       |
|------|-----------|-----------|-----------|
| 5    | 0,386019  | 0,8480565 | 0,28462   |
| 10   | 0,3772644 | 0,8505334 | 0,2727795 |
| 15   | 0,3824789 | 0,8463703 | 0,2761108 |
| 19   | 0,3853384 | 0,8441459 | 0,2781775 |
| 21   | 0,387029  | 0,8428491 | 0,2793794 |

Tabla 7. Resultados con diferentes mtry - Random Forest.

Con los diferentes mtry probados, el mejor rendimiento obtenido con respecto al RMSE es de 0,3772644 y  $R^2$  de 0,8505334, ambos corresponden a mtry=10, por lo que se selecciona como parámetro óptimo del modelo base.

Para seguir analizando este modelo se tunea el parámetro con 10 grupos de validación cruzada y 5 número de repeticiones para lograr apreciar más el efecto y se construyeron modelos con sampsize de 1000, 3000 y 5000, obteniendo los siguientes resultados.

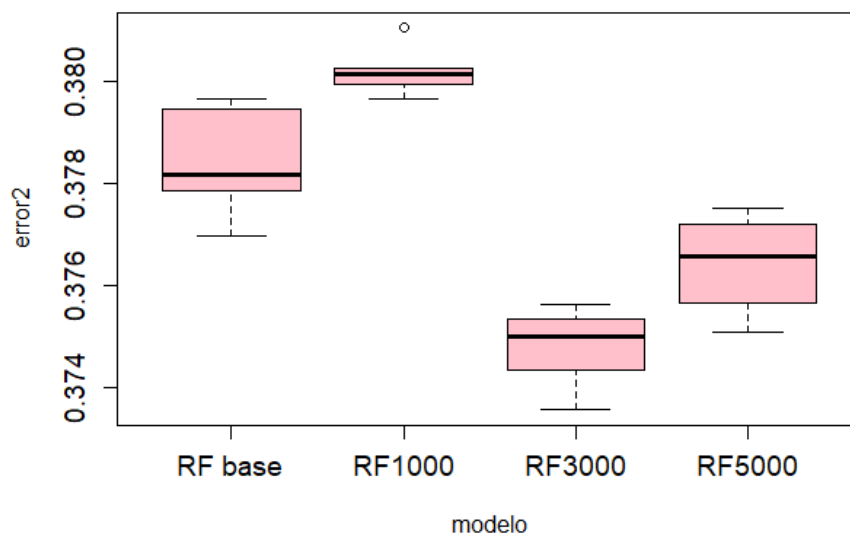


Imagen 31. RMSE para los diferentes modelos de Random Forest probados.

El modelo de Random Forest con sampsize=3000 es el que mejor resultados logra con un menor RMSE de 0,3743208 y un mayor  $R^2$  de 0,8529879.

### 4.2.4 Gradient Boosting

Se utiliza validación cruzada repetida y para tunear los parámetros de Gradient Boosting se evalúan los siguientes parámetros:

- Shrinkage= Parámetro  $\nu$  de regularización que mide la velocidad de ajuste.
- n.minobsnode: Tamaño máximo de nodos finales (el principal parámetro que mide la complejidad).

- n.trees= Número de iteraciones (árboles).
- Interaction.depth: Profundidad máximo de interacción.

Basado en las métricas de rendimiento, se obtienen mejores resultados con estos parámetros: n.trees= 5000, interaction.depth= 2, shrinkag =0.1 y n.minobsinnode=20.

Lo anterior se puede observar mejor en la siguiente gráfica:

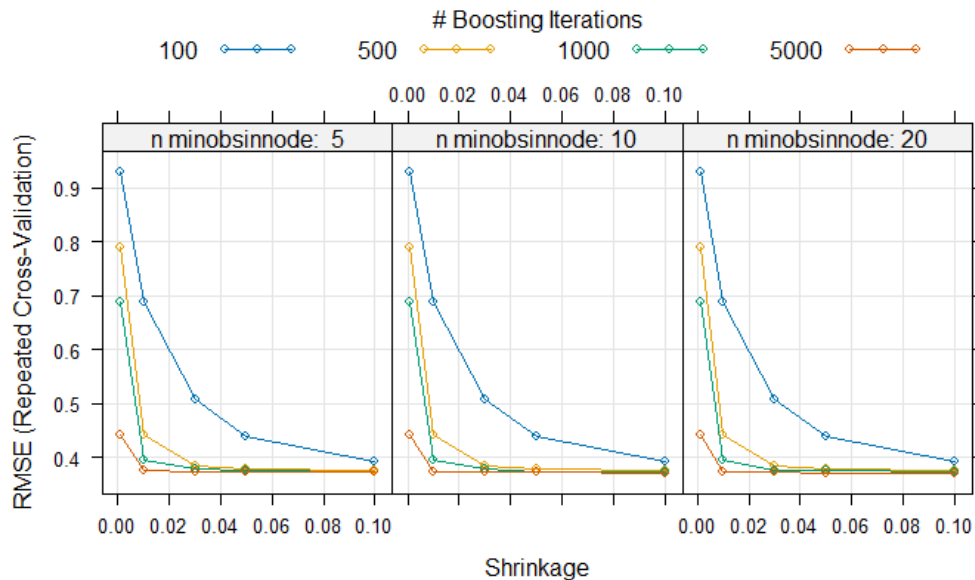


Imagen 32. Identificación de parámetros óptimos - Gradient Boosting.

Se prueba también con early stopping, evidenciando que se logra mejor rendimiento basado en el RMSE a partir de 5000 iteraciones y no varía desde este punto en adelante.

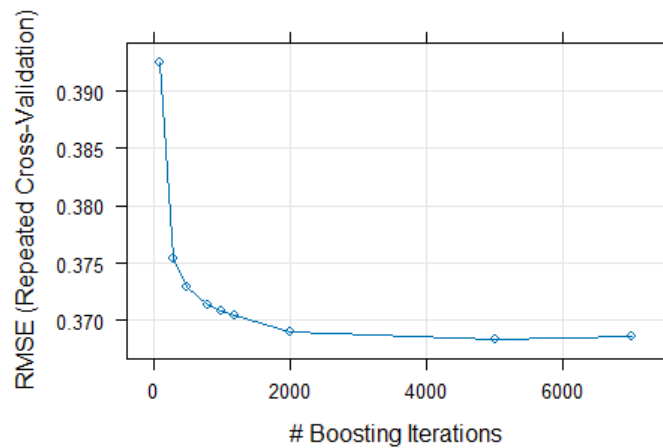


Imagen 33. Early Stopping - Gradient Boosting.

Definidos todos los parámetros óptimos y para seguir analizando este modelo, se tunean los parámetros con 10 grupos de validación cruzada y 5 número de repeticiones para lograr apreciar más el efecto y se tunea n.trees con 2000, 3000 y 4000 para comparar varios modelos de Gradient Boosting.

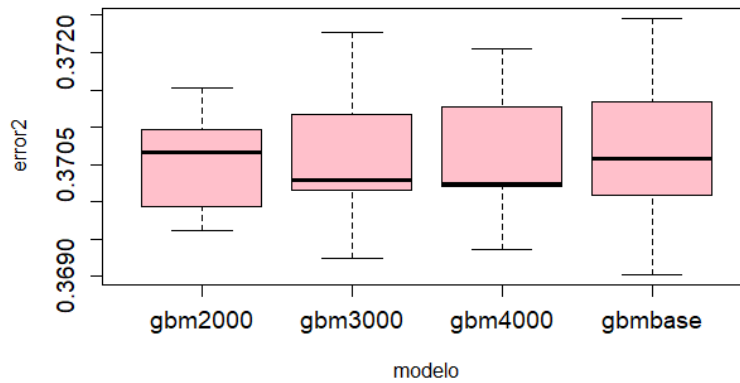


Imagen 34. RMSE para los diferentes modelos de Gradient Boosting.

Si bien las diferencias en el rendimiento entre las diferentes configuraciones son mínimas, el modelo de Gradient Boosting con  $n.trees=2000$  es el que mejor resultados logra con un menor RMSE de 0,3700605 y un mayor  $R^2$  de 0,8559541.

#### 4.2.5 XGBoost

Se utiliza validación cruzada repetida y para tunear los parámetros de XGBoost se evalúan los siguientes parámetros:

- rounds: Número de iteraciones de boosting (número total de árboles).
- max\_depth: Profundidad máxima permitida por cada árbol.
- eta: Tasa de aprendizaje o shrinkage (reduce el impacto de cada árbol).
- gamma: Reducción mínima de pérdida para hacer una partición.
- colsample\_bytree: Proporción de variables usadas por árbol.
- min\_child\_weight: Peso mínimo total en una hoja para permitir una división.
- subsample: Porcentaje de datos usados en cada iteración.

Basado en las métricas de rendimiento, se obtienen mejores resultados con estos parámetros:  $nrounds=100$ ,  $max\_depth=6$ ,  $eta=0.1$ ,  $gamma=0$ ,  $colsample\_bytree=1$ ,  $min\_child\_weight=10$  y  $subsample=1$ .

Lo anterior se puede observar mejor en la siguiente gráfica:

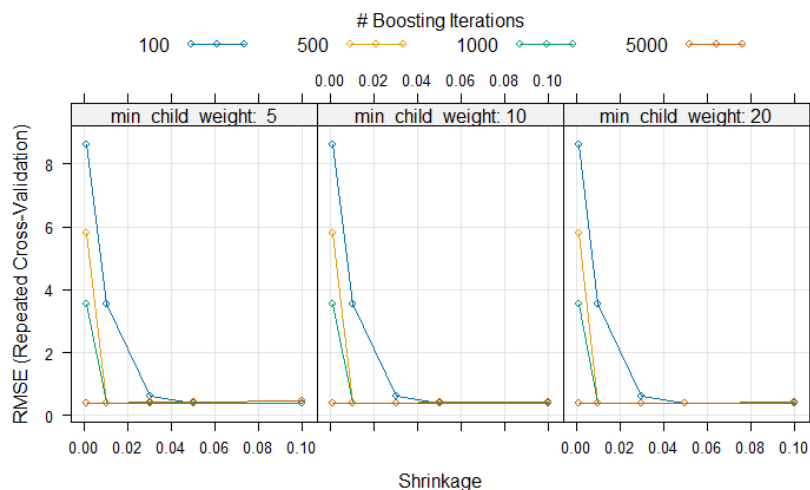


Imagen 35. Identificación de parámetros óptimos - XGBoost.

También se aplica early stopping, evidenciando que se logra el mejor rendimiento basado en el RMSE con 100 iteraciones y en adelante aumenta el error, lo que puede indicar un posible sobreajuste del modelo.

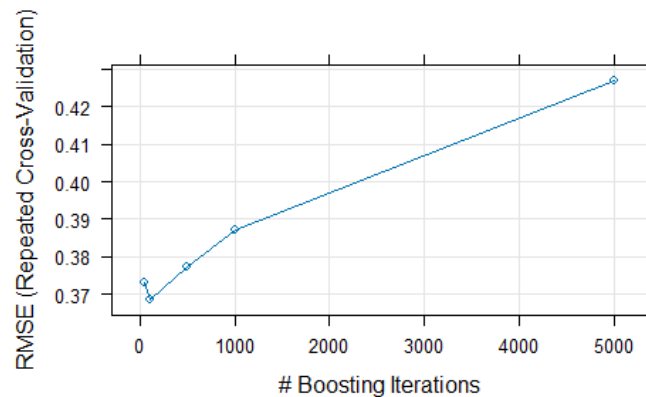


Imagen 36. Early Stopping - XGBoost.

Definidos todos los parámetros óptimos y con el propósito de seguir analizando este modelo, se tunean los parámetros con 10 grupos de validación cruzada y 5 número de repeticiones. Adicionalmente, se incorpora otro parámetro conocido como Alpha, utilizado como término de regularización L1, que favorece modelos más simples contribuyendo a reducir el riesgo de sobreajuste.

Se evaluaron distintos valores para Alpha (0.5, 0.7 y 0.3), obteniendo los siguientes resultados.

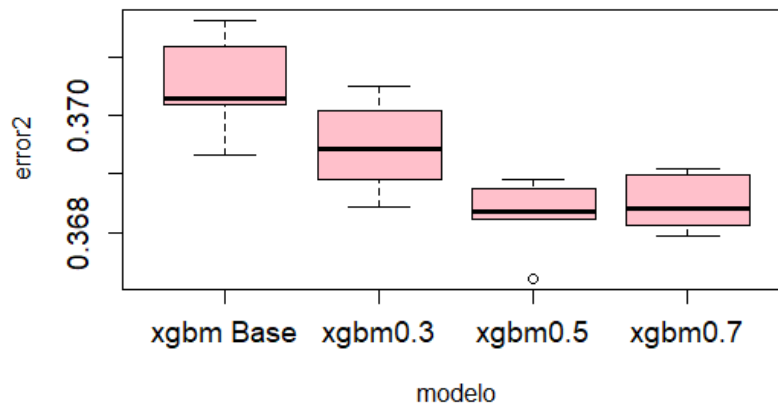


Imagen 37. RMSE para los diferentes modelos de XGBoost.

Se evalúan los modelos en términos de RMSE, evidenciando que aquellos en los que se tuneó el parámetro Alpha lograron mejor rendimiento que el modelo base. En particular, el xgbm0.5 es el que mejor resultados presenta al lograr el menor RMSE (0,3678241), un mayor  $R^2$  (0,8577018) y una menor dispersión de los datos, lo que indica mayor estabilidad y precisión en las predicciones.

#### 4.2.6 SVM – Support Vector Machines

Para los modelos SVM se identifican los parámetros óptimos con validación cruzada repetida y haciendo uso de diferentes funciones de Kernel. El Kernel actúa como una función de transformación que proyecta los datos originales a un espacio de mayor dimensión, lo que permite al modelo capturar relaciones complejas, lineales y no lineales, entre variables.

A continuación, se relacionan los parámetros óptimos identificados para cada una de las funciones Kernel consideradas en este proyecto.

- Lineal: El parámetro óptimo que se puede tunear es C que corresponde a una constante de regularización. Basado en las métricas de rendimiento, el parámetro óptimo es  $C=0,01$  con un RMSE de 0,4938197 y un  $R^2$  de 0,7479586

| C    | RMSE      | Rsquared  | MAE       | RMSESD     | RsquaredSD | MAESD      |
|------|-----------|-----------|-----------|------------|------------|------------|
| 0,01 | 0,4938197 | 0,7479586 | 0,3590335 | 0,01075173 | 0,00988157 | 0,00386601 |
| 0,05 | 0,495912  | 0,7469285 | 0,3589792 | 0,01140811 | 0,00991059 | 0,00380875 |
| 0,1  | 0,4962528 | 0,7467984 | 0,3589377 | 0,0113652  | 0,00985343 | 0,00382762 |
| 0,2  | 0,4964951 | 0,7466778 | 0,3589656 | 0,01150053 | 0,00989921 | 0,00383826 |
| 0,5  | 0,4964952 | 0,7466834 | 0,3589496 | 0,01152138 | 0,00991455 | 0,0038626  |
| 1    | 0,496626  | 0,7466302 | 0,3589466 | 0,01168623 | 0,00996598 | 0,00384832 |
| 2    | 0,4966052 | 0,7466343 | 0,3589476 | 0,01162133 | 0,00991094 | 0,00386976 |
| 5    | 0,4966134 | 0,7466452 | 0,358951  | 0,01166337 | 0,00991283 | 0,00385825 |
| 10   | 0,4966548 | 0,7466248 | 0,3589409 | 0,01169949 | 0,00994318 | 0,00384403 |

Tabla 8. Resultados con diferentes parámetros – SVM lineal.

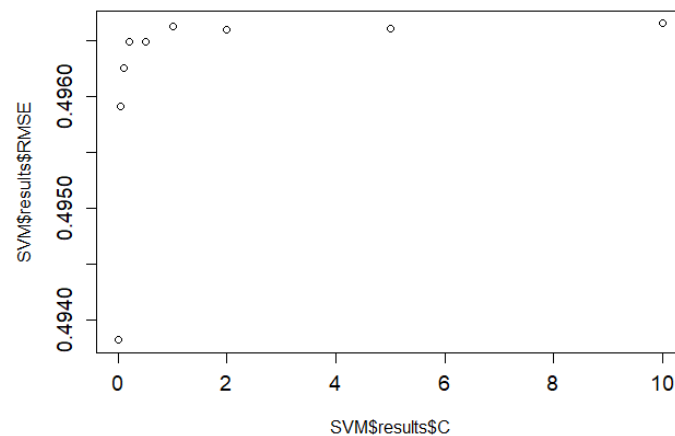


Imagen 38. Valor óptimo de C - SVM Lineal.

- RBF: Los parámetros que se pueden tunear son sigma y C, el primero controla la amplitud de Kernel y el segundo es la constante de regularización. Evaluadas diferentes combinaciones, se determinan como parámetros óptimo sigma = 0.01 y C = 10.

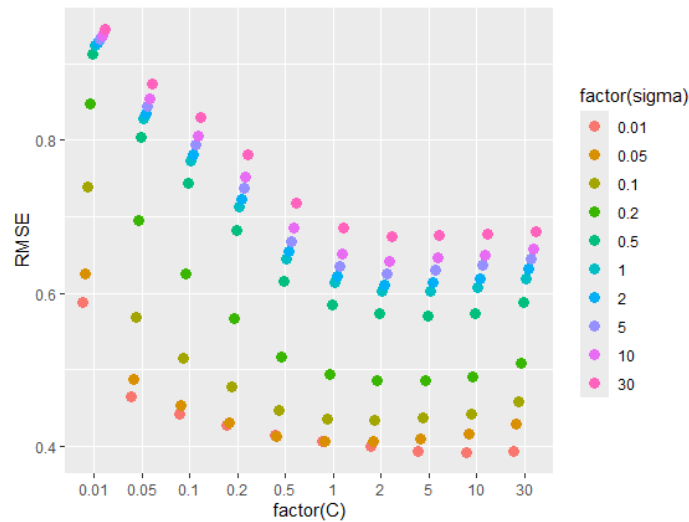


Imagen 39. RMSE para diferentes conjuntos de parámetros - SVM RFB.

- Polinomial: Para este tipo de SVM se pueden tunear los parámetros C que es la constante de regularización, scale que corresponde al factor de escalamiento del término polinomial y degree correspondiente al grado de polinomio utilizado por la función de Kernel. Este último parámetro sólo se probó degree=2 debido a la complejidad computacional que requiere este modelo. Se identifican como parámetros óptimos degree=2, scale=0.1 Y C=0.1.

| C    | scale | RMSE      | Rsquared  | MAE       |
|------|-------|-----------|-----------|-----------|
| 0.1  | 0.1   | 0.4119369 | 0.8220141 | 0.2938243 |
| 0.1  | 1.0   | 0.4149865 | 0.8196956 | 0.2948107 |
| 1.0  | 0.1   | 0.4146971 | 0.8199167 | 0.2946499 |
| 1.0  | 1.0   | 0.4158792 | 0.8189418 | 0.2949876 |
| 10.0 | 0.1   | 0.4151225 | 0.8196128 | 0.2947806 |
| 10.0 | 1.0   | 0.4296592 | 0.8066652 | 0.2969045 |

Tabla 9. Resultados con diferentes parámetros - SVM-Polinomial.

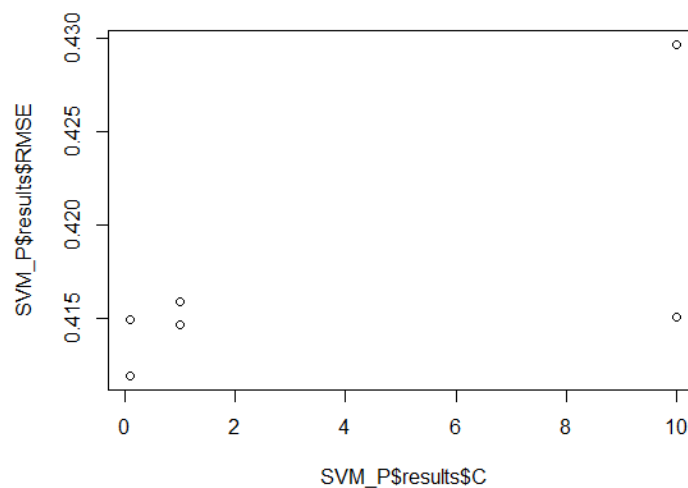


Imagen 40. Valor óptimo de C - SVM Polinomial.

Definidos todos los parámetros óptimos para cada uno de los modelos SVM, se tunea con 10 grupos de validación cruzada y 5 número de repeticiones logrando los siguientes resultados.

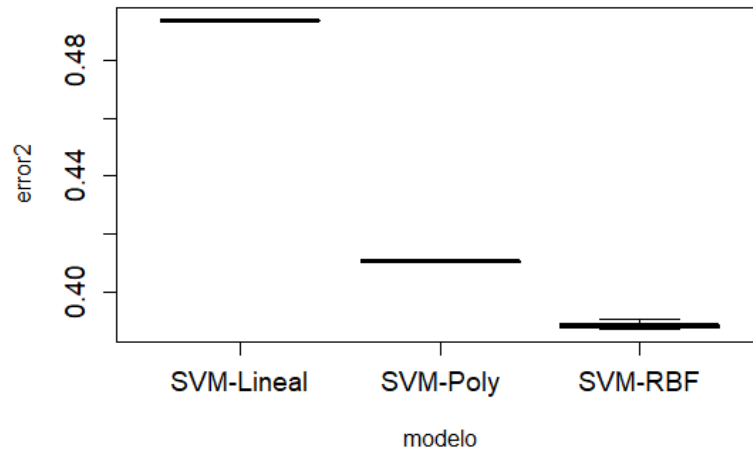


Imagen 41. RMSE para los diferentes modelos de SVM.

Al comparar el desempeño de los modelos SVM con diferentes funciones de Kernel, se observa que el modelo con Kernel RBF presenta el menor RMSE (0,3877647) y mayor  $R^2$  (0,8418016), por lo que es el que mejor se ajusta a los datos y es capaz de capturar de manera más efectiva las relaciones no lineales entre las variables.

#### 4.2.7 Redes Neuronales

Para ajustar modelos de redes neuronales, se empleó la validación cruzada repetida variando el número de nodos en la capa oculta (size), el parámetro de regularización (decay) y el número máximo de iteraciones (maxit). La elección final de los parámetros se basó en el menor valor de RMSE obtenido.

Para el cálculo de nodos de la red neuronal se tuvo en cuenta la siguiente fórmula (Portela, 2025):

$$h(k+1) + h + 1 = \frac{Obs}{p}$$

Donde:

- h es el número de nodos ocultos
- k es el número de nodos input
- Obs es el número total de observaciones
- p es el número de observaciones por parámetro (se recomienda al menos 30).

Reemplazando en la fórmula obtenemos:

$$h(21+1) + h + 1 = \frac{7283}{30}$$

$$h(22) + h + 1 = 242.76 \rightarrow 23h = 241.76 \rightarrow h = 10,51 \approx 10$$

La elección final de los parámetros se basó en el menor valor de RMSE obtenido, también utilizando la técnica de early stopping (detención temprana), que permite evitar el sobreajuste deteniendo el entrenamiento cuando el error deja de mejorar significativamente. Gráficamente se puede observar que lo anterior ocurre a partir de las 500 iteraciones.

Como resultado, los parámetros óptimos seleccionados para la red fueron: size = 6, decay = 0.01 e iteraciones = 500.

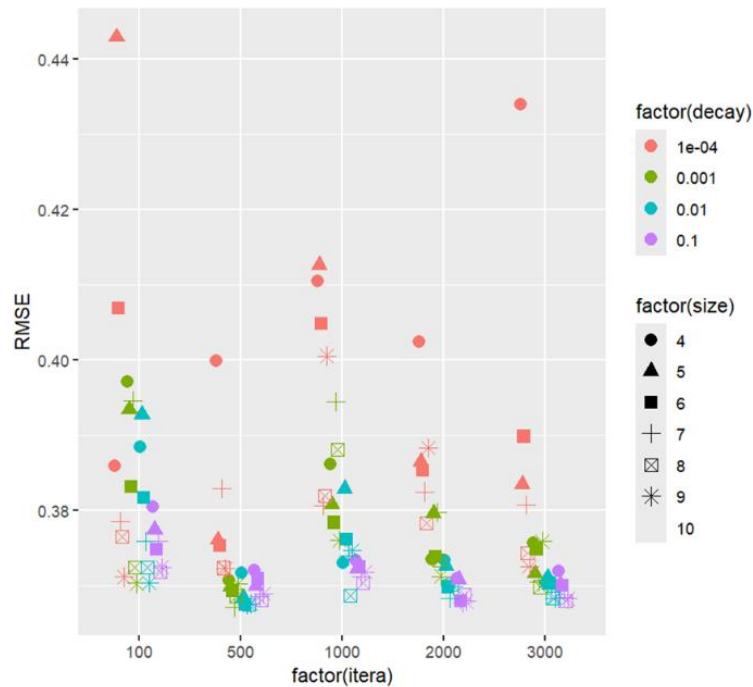


Imagen 42. RMSE con diferentes parámetros - Red Neuronal.

Definidos todos los parámetros óptimos y para seguir analizando este modelo, se construyeron tres (3) redes adicionales variando los parámetros de acuerdo con los resultados obtenidos previamente, con el objetivo de identificar si es posible mejorarlos haciendo uso de 10 grupos de validación cruzada y 5 número de repeticiones.

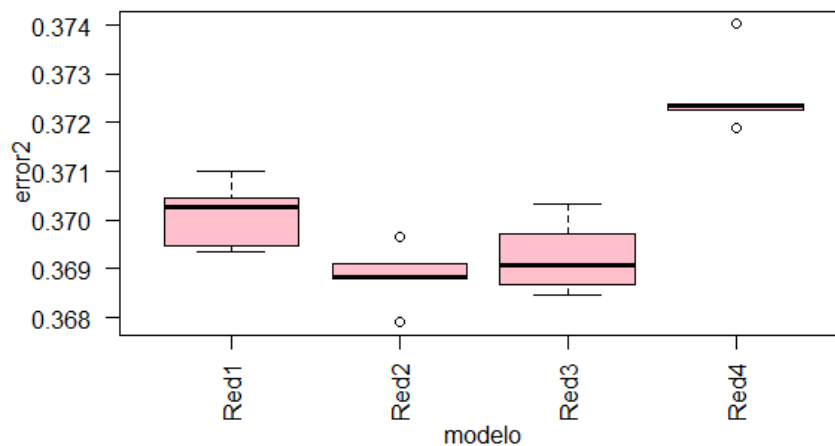


Imagen 43. RMSE para los diferentes modelos de redes neuronales.

| Size | Decay | Iteraciones | RMSE      | $R^2$     | Nombre red |
|------|-------|-------------|-----------|-----------|------------|
| 6    | 0.01  | 500         | 0,3696128 | 0,8563037 | Red1       |
| 7    | 0.001 | 500         | 0,3683308 | 0,8572693 | Red2       |
| 8    | 0.1   | 500         | 0,3687524 | 0,8569437 | Red3       |
| 5    | 0.01  | 400         | 0,3720609 | 0,8544515 | Red4       |

Tabla 10. Métricas de rendimiento - Redes neuronales probadas.

Al comparar los resultados de las redes, se puede observar que la red 2 es la que mejor rendimiento tiene con respecto a RMSE, sin embargo, la diferencia no es significativa ya que todas explican aproximadamente el 86% de la variabilidad del logaritmo del costo total.

#### 4.2.8 Comparación de modelos

Se evaluaron diferentes modelos de machine learning con diferentes configuraciones de sus parámetros, haciendo uso de la validación cruzada repetida para garantizar la robustez de los resultados, y se obtuvo los siguientes resultados en función de precisión predictiva (RMSE) y capacidad explicativa ( $R^2$ ).

| Modelo           | RMSE      | $R^2$     |
|------------------|-----------|-----------|
| Regresión lineal | 0,4828    | 0,7561    |
| Árbol            | 0,4048824 | 0,8282166 |
| Bagging Base     | 0,3888635 | 0,841339  |
| Bagging1000      | 0,3765036 | 0,8511738 |
| Bagging3000      | 0,3774098 | 0,8502566 |
| Bagging5000      | 0,3829799 | 0,8459344 |
| RF Base          | 0,3779898 | 0,8497955 |
| RF1000           | 0,3797353 | 0,8504612 |
| RF3000           | 0,3743208 | 0,8529879 |
| RF5000           | 0,3759552 | 0,8514569 |
| gbm Base         | 0,3702172 | 0,8558445 |
| gbm2000          | 0,3700605 | 0,8559541 |
| gbm3000          | 0,3701503 | 0,8558911 |
| gbm4000          | 0,3701391 | 0,8559033 |
| xgbm             | 0,3700595 | 0,8559935 |
| xgbm0,5          | 0,3678241 | 0,8577018 |
| xgbm0,3          | 0,3689729 | 0,8568107 |
| xgbm0,7          | 0,3680326 | 0,8575197 |
| SVM-Lineal       | 0,4934034 | 0,7493193 |
| SVM-RBF          | 0,3877647 | 0,8418016 |
| SVM-Polinomial   | 0,4099869 | 0,8236415 |
| Red1             | 0,3696128 | 0,8563037 |
| Red2             | 0,3683308 | 0,8572693 |
| Red3             | 0,3687524 | 0,8569437 |
| Red4             | 0,3720609 | 0,8544515 |

Tabla 11. RMSE y  $R^2$  para los modelos probados.

Posteriormente, se seleccionan los mejores modelos de todos los algoritmos probados para realizar un análisis gráfico de sus rendimientos.

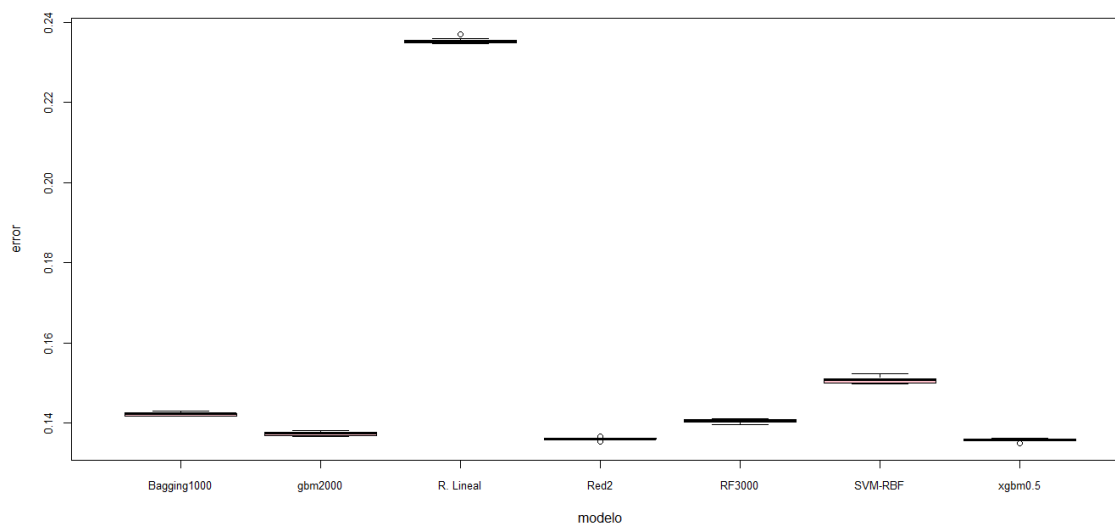


Imagen 44. RMSE de los mejores modelos por algoritmo probado.

Entendiendo que la regresión lineal es el modelo con peor rendimiento, con un RMSE de 0,4828 y  $R^2$  de 0,7561, se elimina de la gráfica para mejorar la escala.

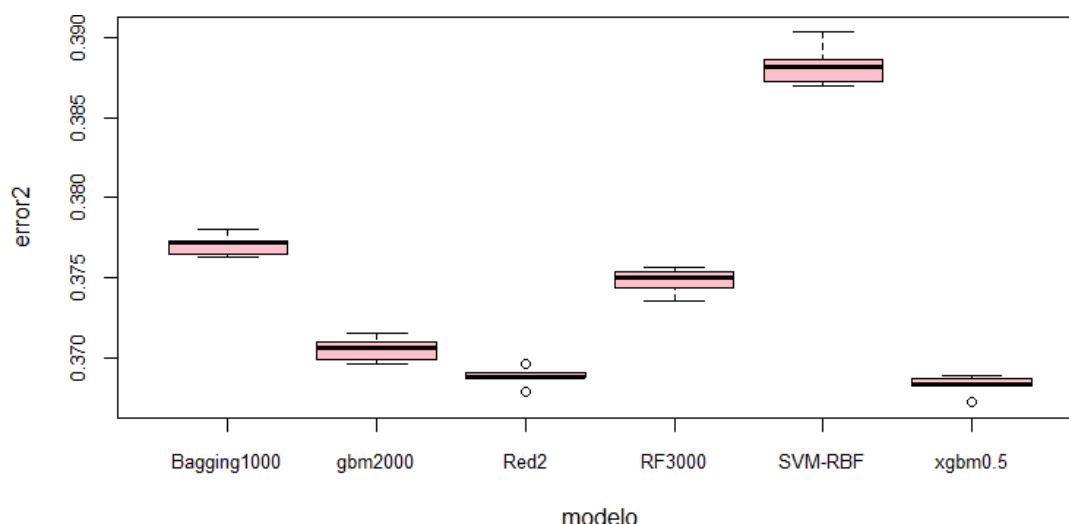


Imagen 45. RMSE de los mejores modelos por algoritmo probado sin la regresión lineal.

Los resultados más competitivos son los obtenidos a través de la red2 y xgbm0.5, siendo este último ligeramente mejor. El modelo de XGBoost con Alpha 0,5 es quién logró el mejor rendimiento general, no solo es el que menor RMSE tiene con un 0,3678241, también es el modelo que mayor  $R^2$  tiene, lo que indica que explica en un 85,77% la variabilidad del logaritmo del costo total de la atención médica en el estado de Nueva York de los pacientes que han ingresado por algún tipo de trauma.

Partiendo de los resultados preliminares, se toma la decisión de emplear técnicas de ensamblado que consiste en combinar diferentes modelos para ver si es posible mejorar las métricas de rendimiento, asumiendo como riesgo latente el posible sobreajuste. Para esto, seleccionamos los modelos más parecidos en términos de RSME que son Gradient Boosting, redes neuronales y XGBoost y utilizando la técnica Stacking que permite

combinar varios modelos distintos mediante un modelo final que aprende cual es la forma de mezclar sus predicciones. En este caso, se empleó una regresión lineal no negativa que asigna pesos a cada modelo para mejorar la precisión total.

| Xgbm  | Gbm    | Red    |
|-------|--------|--------|
| 0,543 | 0,3311 | 0,1259 |

Tabla 12. Pesos de los modelos empleados en el ensamblado.

Los pesos reflejan la contribución de cada modelo en el ensamblado final. En este caso, el XGBoost fue el más influyente en la predicción conjunta con más del 50% del peso asignado, lo que confirma que es un modelo que da respuestas con gran desempeño al problema abordado en esta investigación.

Con las predicciones de cada modelo se graficó el RMSE, observando que el ensamblado logró disminuir el error y aumentó el  $R^2$  hasta 0,8768. Sin embargo, aunque se puede comprobar que es posible mejorar las predicciones, la mejora no es significativa. Por este motivo, se decide dar prioridad a la interpretabilidad y simplicidad del modelo, eligiendo el XGBoost con Alpha 0.5 como el más adecuado.

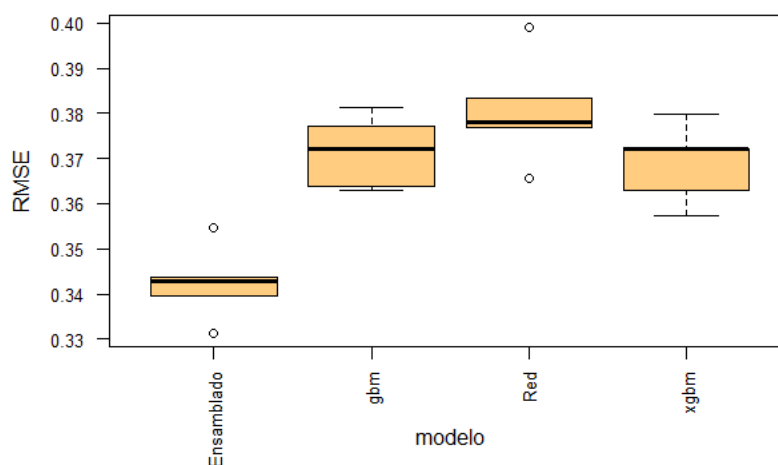


Imagen 46. RMSE modelos empleados en el ensamblado.

#### 4.2.9 Selección del modelo ganador

Aunque XGBoost es un modelo de caja negra y eso lo convierte en más complejo de explicar con respecto a un modelo más tradicional como la regresión lineal, se cuenta con herramientas que ayudan a identificar la importancia de las variables y su influencia en la variable objetivo, que en este caso es el logaritmo del costo total de la atención.

La función varImp() incluida en el paquete caret es utilizada para mostrar gráficamente el efecto general de las variables incluidas en un modelo. En el caso de los boosted tree como XGBoost, la importancia se calcula de forma acumulativa a lo largo de las iteraciones del boosting, sumando la contribución de cada variable en cada árbol del conjunto (Kuhn, 2008).

A continuación, se presenta el gráfico de importancia de las variables para el modelo xgbm0.5, seleccionado como el ganador.

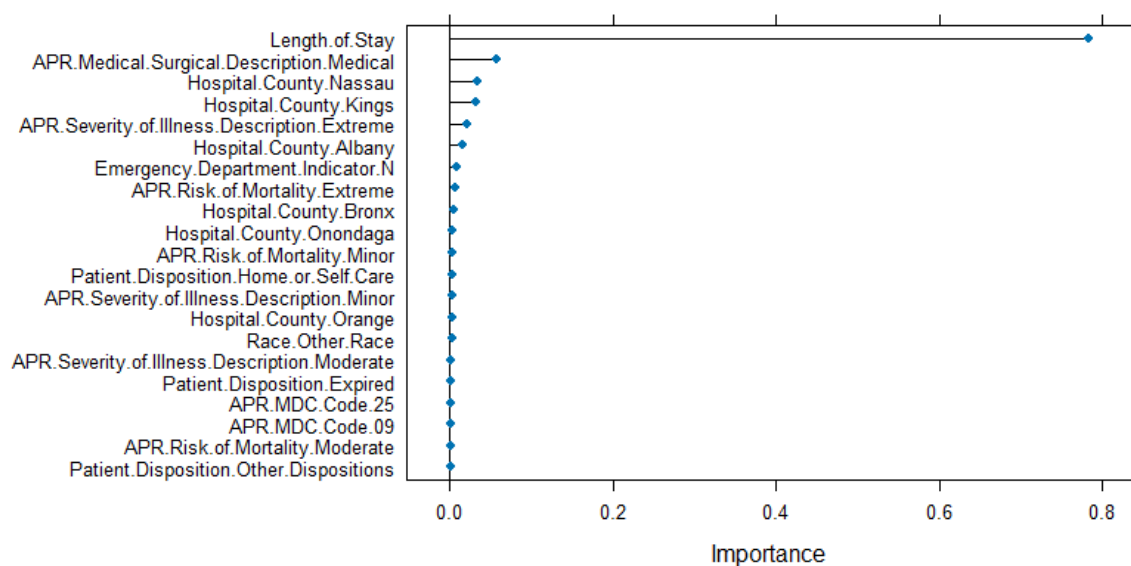


Imagen 47. Importancia de variables del modelo ganador.

De acuerdo con los resultados, la variable más importante es length of stay que representa los días de estancia en el hospital, lo que resulta coherente con el problema abordado, ya que incluso desde la evaluación inicial de la correlación se evidenció que el coeficiente de Pearson entre esta variable y la objetivo era 0,68. La segunda variable más importante es APR Medical Surgical Description que es la clasificación del caso, específicamente para los clasificados como casos médicos y Hospital County correspondiente a la ubicación geográfica del hospital, en particular para los ubicados en los condados de Nassau y Kings. Otras variables como la descripción del nivel de severidad del episodio clínico, concretamente a los clasificados como extremos, contribuyen significativamente en el modelo.

Para validar el modelo, se utiliza para predecir el logaritmo del costo total y observar cómo opera. A continuación, se relacionan los resultados de las primeras 5 observaciones, mostrando los valores reales (log-transformados) y las predicciones del modelo.

| Real      | Predicho  |
|-----------|-----------|
| 11,888619 | 11,502980 |
| 8,830787  | 8,861979  |
| 10,219386 | 10,012684 |
| 11,750869 | 10,962914 |
| 9,173953  | 8,944303  |
| 10,270845 | 9,914974  |

Tabla 13. Valores reales y predichos con el modelo XGBoost alpha 05 para las primeras cinco (5) observaciones.

El modelo obtuvo un RMSE de 0,3197013 y el  $R^2$  de 0,8925845, lo que demuestra que se ajusta muy bien a los datos originales y que sus resultados son consistentes.

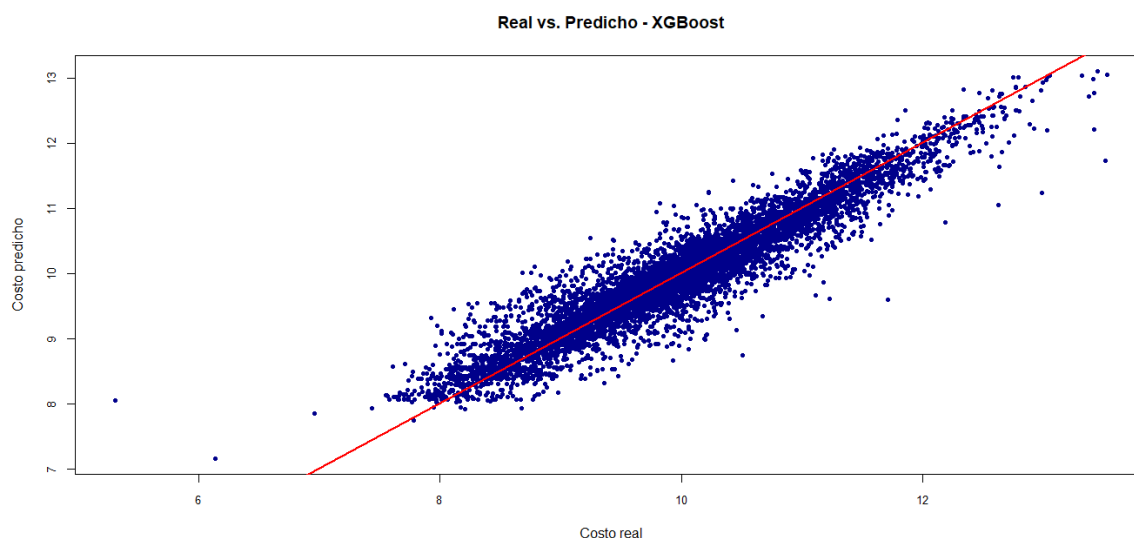


*Imagen 48. Gráfico de dispersión - Modelo ganador.*

En la imagen anterior se muestra a través de un gráfico de dispersión el valor real y el predicho del modelo, observando que los puntos de encuentran muy cercanos a la diagonal (línea roja), lo que demuestra que el modelo logra predecir con gran precisión el logaritmo del costo total de la atención médica.

Estos resultados son consistentes con estudios previos incluidos en el ítem 1.1 introductorio al caso de estudio en donde modelos de XGBoost fueron elegidos para predecir costos médicos con valores de  $R^2$  de 0,8873 o modelos como Gradient Boosting que también lograron  $R^2$  de 0,8986.

Con todo lo anterior, podemos concluir que el modelo XGBoost con Alpha=5 demostró ser el más apropiado para predecir el comportamiento de los costos totales en la atención médica por traumas en el estado de Nueva York. Su combinación de alta precisión, capacidad explicativa y consistencia en sus resultados lo establece como el mejor algoritmo evaluado.

#### **4.2.10 Comparación de resultados con SAS**

Se lleva a cabo el proceso de selección de variables e implementación de diferentes modelos en SAS Enterprise Miner con el propósito de comparar los resultados obtenidos con R. Si bien no fue posible replicar exactamente los mismos modelos en ambos entornos debido a las limitaciones propias del software, la comparación permite tener una visión complementaria de algunos modelos en sus métricas de rendimiento.

A continuación, el esquema empleado en SAS que incluye los modelos probados.

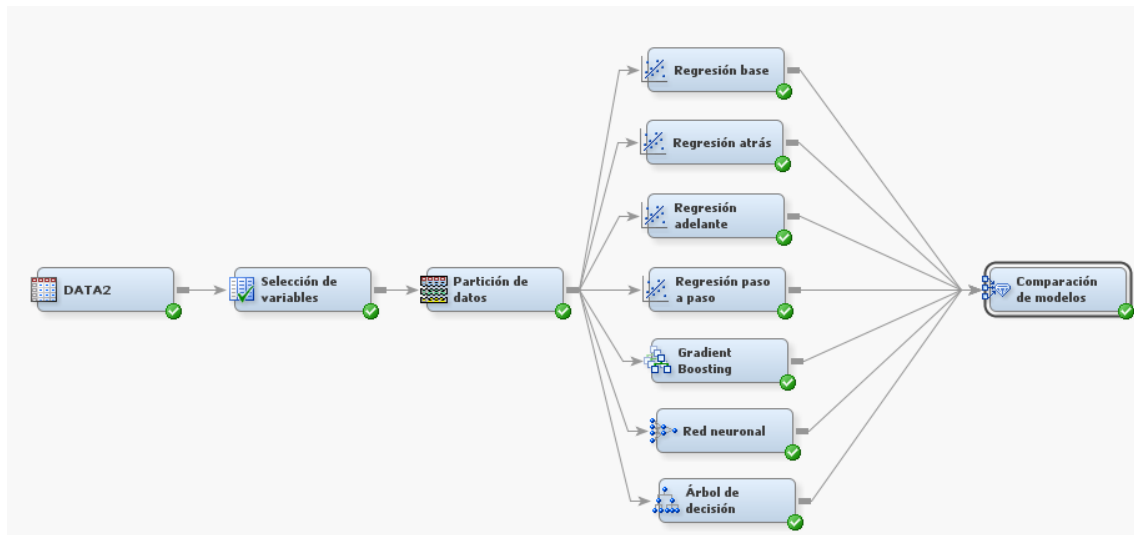


Imagen 49. Diagrama empleado en SAS Enterprise Miner.

La selección de variables se hizo a través del nodo disponible en el software, las variables seleccionadas (20) son:

- APR MDC Code 09
- APR MDC Code 25
- APR Medical Surgical Description - Medical
- APR Risk of Mortality - Major
- APR Risk of Mortality - Minor
- APR Risk of Mortality - Moderate
- Hospital County Albany
- Hospital County Bronx
- Hospital County Kings
- Hospital County Nassau
- Hospital County Onondaga
- Length of stay
- Patient Disposition - Home or selfcare
- Patient Disposition - Home w/ home health services
- Patient Disposition – Inpatient rehabilitation facility
- Patient Disposition – Skilled Nursing Home
- APR Severity of Illness Description – Major
- APR Severity of Illness Description – Minor
- APR Severity of Illness Description – Moderate
- Emergency Department Indicator - Y

Del anterior set de variables, 14 fueron seleccionadas tanto por el nodo de selección de SAS miner como por R con el STEP repetido bajo el criterio BIC, que fue el empleado en el desarrollo del trabajo.

Al comparar el rendimiento de los modelos empleados basados en el RMSE, se obtuvieron los siguientes resultados:

| Modelo                         | RMSE train | RMSE test |
|--------------------------------|------------|-----------|
| Gradient Boosting              | 0,4922     | 0,5035    |
| Regresión lineal - atrás       | 0,4856     | 0,4847    |
| Regresión lineal - adelante    | 0,4856     | 0,4847    |
| Regresión lineal - paso a paso | 0,4856     | 0,4847    |
| Regresión lineal - base        | 0,4856     | 0,4847    |
| Árbol de decisión              | 0,4270     | 0,4456    |
| Red neuronal                   | 0,3771     | 0,3782    |

*Tabla 14. RMSE para modelos con SAS Enterprise Miner.*

La red neuronal fue el modelo con mejor rendimiento tanto en datos test como en datos train en comparación con el resto de los modelos.

Aunque en R el proceso se hizo a través de validación cruzada repetida y con SAS Miner se empleó una división simple de datos con train y test, se puede observar que las métricas de rendimiento en términos de error son coherentes en ambos entornos. En R, la red neuronal que mejor desempeño presentó logró un RMSE de 0,3683 vs. el 0,3782 que se obtuvo en la red de SAS, caso similar a la regresión lineal que en R el RMSE fue de 0,4856 vs R con 0,4828.

Caso contrario al experimentado por el modelo de Gradient Boosting, con un RMSE en SAS de 0,4922 vs. 0,37 en R. Estas discrepancias se deben probablemente debido a las diferencias de la configuración predeterminada del algoritmo, así como la facilidad de RStudio de tunear sus parámetros, lo que permite una mayor optimización del modelo.

## 5. CONCLUSIONES

El objetivo principal de este trabajo de fin de máster ha sido la construcción de un modelo de predicción capaz de estimar el costo total de la atención médica para pacientes ingresados por traumas en el estado de Nueva York.

Inicialmente se hizo una revisión del comportamiento de cada una de las variables incluidas en el dataset, seguido de un proceso de preparación y tratamiento de esos datos brutos para poder hacer las correcciones necesarias para el correcto funcionamiento de los modelos de machine learning. Una vez superado la etapa de comprensión y preparación de datos, se emplearon diferentes métodos de selección de variables (SBF, Boruta, STEP AIC/BIC, MXM, entre otros), lo que permitió reducir el número de variables predictoras sin comprometer la capacidad predictiva de los modelos. La combinación de variables seleccionadas fue la realizada a través del método de selección stepwise repetido bajo el criterio BIC, siendo 21 las variables seleccionadas y la decisión se basó en el menor valor de RMSE junto con el número de variables y la dispersión de los datos.

A partir de ese conjunto de variables seleccionadas se entrenaron diferentes modelos de machine learning, desde el más tradicional como la regresión lineal a modelos más avanzados como Bagging, Random Forest, Gradient Boosting, XGBoost, SVM y redes neuronales, tuneando sus parámetros y empleando validación cruzada repetida con 10 grupos y 5 repeticiones para garantizar robustez en los resultados.

Del grupo de algoritmos seleccionados, el que se define como ganador es el XGBoost con Alpha igual a 0,5, siendo el que mejor rendimiento obtuvo con un RMSE de 0,3678241 y un  $R^2$  de 0,8577018, lo cual indica que explica el 85,77% de la variabilidad del costo total de la atención médica (en su forma logarítmica), a través de ese conjunto de variables. Si bien este algoritmo se considera de caja negra, se empleó el análisis de importancia de variables a través de la función `varImp()` de `caret` para analizar la importancia relativa de las variables, encontrando que mayor incidencia tienen en el logaritmo del costo total son los días de estancia, el tipo de clasificación del caso y el condado donde se encuentra ubicado el hospital.

El gráfico de dispersión sobre los valores predicho vs. los valores reales muestra una alineación cercana con la diagonal, lo que confirma una alta capacidad predictiva del modelo sobre los datos disponibles.

También se emplearon técnica de ensamblado combinando XGBoost, Gradient Boosting y redes neuronales. Si bien el ensamblado logró una ligera mejora en RMSE y  $R^2$ , la ganancia no fue significativa y no justifica la pérdida de interpretabilidad. Por esta razón el modelo seleccionado como ganador es el XGBoost con Alpha=0.5, pues presenta equilibrio entre precisión, estabilidad y capacidad de interpretación.

Para validar la consistencia de los resultados, se realizó una comparación con SAS Enterprise Miner. Si bien el entorno de SAS tiene limitaciones, los resultados fueron coherentes, la red neuronal obtuvo un RMSE de 0,3782 muy cercano al valor calculado en R de 0,3683. La regresión lineal también mostró resultados consistentes en ambos

entornos. Sin embargo, en Gradient Boosting si se apreciaron diferencias más marcadas con un RMSE en SAS de 0,4922 vs. 0,37 en R, lo que es atribuible a las limitaciones en la configuración en SAS frente a la flexibilidad de tunear los hiperparámetros en R.

De manera conjunta, los resultados confirman que el enfoque de machine learning puede aportar valor a escenarios complejos como en este caso la predicción de los costos médicos, facilitando que se tomen decisiones más informadas. Desde un punto de vista operativo, este tipo de modelos pudieran ayudar a mejorar la planificación financiera y generar mayor transparencia en la gestión de la salud en los hospitales, especialmente en un entorno como Estados Unidos donde existe desconfianza de los usuarios, el sistema es complejo y la salud no tiene cobertura universal.

En conclusión, este tipo de proyectos demuestra que el uso modelos de machine learning aplicados a situaciones reales y complejas pueden aportar a dar soluciones con información precisa y útil para tomar decisiones más transparentes y humanas.

## BIBLIOGRAFIA

- Aborode, A. T., Oginni, O., Abacheng, M., Edima, O., Lamunu, E., Folorunso, T. N., Oko, C. I., Iretiayo, A. R., Lawal, L., Amarachi, R., Badri, R., Bamigbade, G. B., Olanrewaju, O. F., Agwuna, F. O., & Adesola, R. O. (2025). Healthcare debts in the United States: A silent fight. *Annals of Medicine & Surgery*, 87(2), 663-672. <https://doi.org/10.1097/ms9.0000000000002865>
- AIML.com. (2022, junio 11). What is Bagging? How do you perform bagging and what are its advantages? *AIML.Com*. <https://aiml.com/what-is-bagging/>
- Alpaydin, E. (2014). *Introduction to Machine Learning*. MIT Press.
- Chapman, P., Clinton, J., Kever, R., & Khabaza, T. (2000). *Metodología Crisp-DM 1—Step-by-step data mining guide*. Studocu. <https://www.studocu.com/cl/document/pontificia-universidad-catolica-de-chile/business-analytics/metodologia-crisp-dm-1/8584569>
- Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. 785-794. <https://doi.org/10.1145/2939672.2939785>
- Choudhury, A. (2020, noviembre 27). What is Gradient Boosting? How is it different from Ada Boost? *Analytics Vidhya*. <https://medium.com/analytics-vidhya/what-is-gradient-boosting-how-is-it-different-from-ada-boost-296f055ecacd>
- Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines. *Cambridge University Press*.
- Cunningham, P. J. (2009). High Medical Cost Burdens, Patient Trust, and Perceived Quality of Care. *Journal of General Internal Medicine*, 24(3), 415-420. <https://doi.org/10.1007/s11606-008-0879-3>
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5). <https://doi.org/10.1214/aos/1013203451>
- Haykin, S. S., & Haykin, S. S. (2009). *Neural networks and learning machines* (3rd ed). Prentice Hall.
- Kuhn, M. (2008). Building Predictive Models in R Using the **caret** Package. *Journal of Statistical Software*, 28(5). <https://doi.org/10.18637/jss.v028.i05>
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York. <https://doi.org/10.1007/978-1-4614-6849-3>
- Kursa, M. B., & Rudnicki, W. R. (2010). Feature Selection with the **Boruta** Package. *Journal of Statistical Software*, 36(11). <https://doi.org/10.18637/jss.v036.i11>

- Lagani, V., Athineou, G., Farcomeni, A., Tsagris, M., & Tsamardinos, I. (2017). Feature Selection with the R Package **MXM**: Discovering Statistically Equivalent Feature Subsets. *Journal of Statistical Software*, 80(7). <https://doi.org/10.18637/jss.v080.i07>
- Langenberger, B., Schulte, T., & Groene, O. (2023). The application of machine learning to predict high-cost patients: A performance-comparison of different models using healthcare claims data. *PLOS ONE*, 18(1), e0279540. <https://doi.org/10.1371/journal.pone.0279540>
- Lopes, L., Kearney, A., Montero, A., Hamel, L., & Brodie, M. (2022, junio 16). Health Care Debt In The U.S.: The Broad Consequences Of Medical And Dental Bills - Main Findings - 9957. *KFF*. <https://www.kff.org/report-section/kff-health-care-debt-survey-main-findings/>
- Morid, M. A., Sheng, O. R. L., Kawamoto, K., Ault, T., Dorius, J., & Abdelrahman, S. (2019). Healthcare cost prediction: Leveraging fine-grain temporal patterns. *Journal of Biomedical Informatics*, 91, 103113. <https://doi.org/10.1016/j.jbi.2019.103113>
- Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Interpretable machine learning: Definitions, methods, and applications. *Proceedings of the National Academy of Sciences*, 116(44), 22071-22080. <https://doi.org/10.1073/pnas.1900654116>
- Papanicolas, I., Woskie, L. R., & Jha, A. K. (2018). Health Care Spending in the United States and Other High-Income Countries. *JAMA*, 319(10), 1024. <https://doi.org/10.1001/jama.2018.1150>
- Pfutzenreuter, T. C., & Lima, E. P. D. (2022). MACHINE LEARNING IN HEALTHCARE MANAGEMENT FOR MEDICAL INSURANCE COST PREDICTION. En *Open Science Research II* (1.<sup>a</sup> ed., pp. 1323-1334). Editora Científica Digital. <https://doi.org/10.37885/220207863>
- Portela, J. (2025). *Apuntes Machine Learning redes neuronales*.
- Sohil, F., Sohali, M. U., & Shabbir, J. (2022). An introduction to statistical learning with applications in R. *Statistical Theory and Related Fields*, 6(1), 87-87. <https://doi.org/10.1080/24754269.2021.1980261>

## ANEXOS

### Tablas complementarias de resultados en R

| shrinkage | n.minobsinnode | n.trees | RMSE      | Rsquared  | MAE       |
|-----------|----------------|---------|-----------|-----------|-----------|
| 0.001     | 5              | 100     | 0.9286356 | 0.5378417 | 0.7398729 |
| 0.001     | 5              | 500     | 0.7898058 | 0.6024880 | 0.6299907 |
| 0.001     | 5              | 1000    | 0.6879299 | 0.6437566 | 0.5526324 |
| 0.001     | 5              | 5000    | 0.4422711 | 0.8147564 | 0.3445037 |
| 0.001     | 10             | 100     | 0.9286356 | 0.5378417 | 0.7398729 |
| 0.001     | 10             | 500     | 0.7898058 | 0.6024880 | 0.6299907 |
| 0.001     | 10             | 1000    | 0.6879299 | 0.6437566 | 0.5526324 |
| 0.001     | 10             | 5000    | 0.4422711 | 0.8147564 | 0.3445037 |
| 0.001     | 20             | 100     | 0.9286356 | 0.5378417 | 0.7398729 |
| 0.001     | 20             | 500     | 0.7898058 | 0.6024880 | 0.6299907 |
| 0.001     | 20             | 1000    | 0.6879299 | 0.6437566 | 0.5526324 |
| 0.001     | 20             | 5000    | 0.4422711 | 0.8147564 | 0.3445037 |
| 0.010     | 5              | 100     | 0.6872811 | 0.6440867 | 0.5521613 |
| 0.010     | 5              | 500     | 0.4418384 | 0.8148852 | 0.3441022 |
| 0.010     | 5              | 1000    | 0.3950065 | 0.8397857 | 0.2949625 |
| 0.010     | 5              | 5000    | 0.3746917 | 0.8525990 | 0.2725951 |
| 0.010     | 10             | 100     | 0.6872811 | 0.6440867 | 0.5521613 |
| 0.010     | 10             | 500     | 0.4418384 | 0.8148852 | 0.3441022 |
| 0.010     | 10             | 1000    | 0.3950115 | 0.8397706 | 0.2949493 |
| 0.010     | 10             | 5000    | 0.3745490 | 0.8527049 | 0.2724396 |
| 0.010     | 20             | 100     | 0.6872811 | 0.6440867 | 0.5521613 |
| 0.010     | 20             | 500     | 0.4418384 | 0.8148852 | 0.3441022 |
| 0.010     | 20             | 1000    | 0.3949236 | 0.8398189 | 0.2948053 |
| 0.010     | 20             | 5000    | 0.3744149 | 0.8528037 | 0.2723606 |
| 0.030     | 5              | 100     | 0.5085315 | 0.7771178 | 0.4059781 |
| 0.030     | 5              | 500     | 0.3842298 | 0.8460096 | 0.2833010 |
| 0.030     | 5              | 1000    | 0.3773961 | 0.8505192 | 0.2756483 |
| 0.030     | 5              | 5000    | 0.3733997 | 0.8536216 | 0.2708653 |
| 0.030     | 10             | 100     | 0.5085315 | 0.7771178 | 0.4059781 |
| 0.030     | 10             | 500     | 0.3843592 | 0.8458767 | 0.2833446 |
| 0.030     | 10             | 1000    | 0.3774724 | 0.8504480 | 0.2756008 |
| 0.030     | 10             | 5000    | 0.3730397 | 0.8538822 | 0.2705598 |
| 0.030     | 20             | 100     | 0.5085315 | 0.7771178 | 0.4059781 |
| 0.030     | 20             | 500     | 0.3842784 | 0.8459487 | 0.2833234 |
| 0.030     | 20             | 1000    | 0.3771525 | 0.8506897 | 0.2755002 |
| 0.030     | 20             | 5000    | 0.3727695 | 0.8541010 | 0.2702394 |
| 0.050     | 5              | 100     | 0.4397952 | 0.8156838 | 0.3419787 |
| 0.050     | 5              | 500     | 0.3781958 | 0.8499243 | 0.2766037 |
| 0.050     | 5              | 1000    | 0.3748403 | 0.8524838 | 0.2727608 |
| 0.050     | 5              | 5000    | 0.3731956 | 0.8537795 | 0.2704110 |

|       |    |      |           |           |           |
|-------|----|------|-----------|-----------|-----------|
| 0.050 | 10 | 100  | 0.4397952 | 0.8156838 | 0.3419787 |
| 0.050 | 10 | 500  | 0.3783345 | 0.8498095 | 0.2766156 |
| 0.050 | 10 | 1000 | 0.3745209 | 0.8527170 | 0.2724182 |
| 0.050 | 10 | 5000 | 0.3727569 | 0.8541058 | 0.2700159 |
| 0.050 | 20 | 100  | 0.4397952 | 0.8156838 | 0.3419787 |
| 0.050 | 20 | 500  | 0.3782465 | 0.8498680 | 0.2766719 |
| 0.050 | 20 | 1000 | 0.3747777 | 0.8525080 | 0.2726628 |
| 0.050 | 20 | 5000 | 0.3717673 | 0.8548769 | 0.2689975 |
| 0.100 | 5  | 100  | 0.3934239 | 0.8403422 | 0.2932861 |
| 0.100 | 5  | 500  | 0.3754050 | 0.8520215 | 0.2732454 |
| 0.100 | 5  | 1000 | 0.3739922 | 0.8531337 | 0.2717364 |
| 0.100 | 5  | 5000 | 0.3721310 | 0.8545755 | 0.2690144 |
| 0.100 | 10 | 100  | 0.3934025 | 0.8403632 | 0.2932445 |
| 0.100 | 10 | 500  | 0.3753281 | 0.8520746 | 0.2730259 |
| 0.100 | 10 | 1000 | 0.3732554 | 0.8537004 | 0.2709039 |
| 0.100 | 10 | 5000 | 0.3717290 | 0.8548659 | 0.2684102 |
| 0.100 | 20 | 100  | 0.3933662 | 0.8404300 | 0.2931977 |
| 0.100 | 20 | 500  | 0.3749821 | 0.8523370 | 0.2727509 |
| 0.100 | 20 | 1000 | 0.3735840 | 0.8534433 | 0.2712954 |
| 0.100 | 20 | 5000 | 0.3715699 | 0.8550127 | 0.2684732 |

Tabla 15. Métricas de rendimiento con diferentes parámetros para los modelos Gradient Boosting.

| eta   | min_child_weight | nrounds | RMSE       | Rsquared  | MAE        |
|-------|------------------|---------|------------|-----------|------------|
| 0,001 | 5                | 100     | 86.335.737 | 0,6907929 | 85.848.684 |
| 0,001 | 5                | 500     | 58.033.875 | 0,753322  | 57.577.168 |
| 0,001 | 5                | 1000    | 35.426.166 | 0,7934438 | 34.956.642 |
| 0,001 | 5                | 5000    | 0,3759388  | 0,8563417 | 0,2728936  |
| 0,001 | 10               | 100     | 86.335.737 | 0,6907929 | 85.848.684 |
| 0,001 | 10               | 500     | 58.033.875 | 0,753322  | 57.577.168 |
| 0,001 | 10               | 1000    | 35.426.166 | 0,7934438 | 34.956.642 |
| 0,001 | 10               | 5000    | 0,3756306  | 0,8565893 | 0,2726506  |
| 0,001 | 20               | 100     | 86.335.737 | 0,6907929 | 85.848.684 |
| 0,001 | 20               | 500     | 58.033.875 | 0,753322  | 57.577.168 |
| 0,001 | 20               | 1000    | 35.426.166 | 0,7934438 | 34.956.642 |
| 0,001 | 20               | 5000    | 0,3763433  | 0,8559648 | 0,2730568  |
| 0,01  | 5                | 100     | 35.269.720 | 0,7931424 | 34.799.231 |
| 0,01  | 5                | 500     | 0,3757402  | 0,8563062 | 0,272683   |
| 0,01  | 5                | 1000    | 0,3699974  | 0,8562138 | 0,265865   |
| 0,01  | 5                | 5000    | 0,3821266  | 0,8469373 | 0,2752874  |
| 0,01  | 10               | 100     | 35.269.720 | 0,7931424 | 34.799.231 |
| 0,01  | 10               | 500     | 0,3755463  | 0,856486  | 0,2724351  |
| 0,01  | 10               | 1000    | 0,368959   | 0,857033  | 0,2650232  |
| 0,01  | 10               | 5000    | 0,3769954  | 0,8509447 | 0,2708948  |
| 0,01  | 20               | 100     | 35.269.720 | 0,7931424 | 34.799.231 |
| 0,01  | 20               | 500     | 0,3761175  | 0,8559597 | 0,2727639  |

|      |    |      |           |           |           |
|------|----|------|-----------|-----------|-----------|
| 0,01 | 20 | 1000 | 0,3687548 | 0,85717   | 0,2652851 |
| 0,01 | 20 | 5000 | 0,3729549 | 0,8540104 | 0,2681651 |
| 0,03 | 5  | 100  | 0,5979168 | 0,8500576 | 0,5031239 |
| 0,03 | 5  | 500  | 0,370407  | 0,8559405 | 0,2662884 |
| 0,03 | 5  | 1000 | 0,3748373 | 0,8525573 | 0,2702307 |
| 0,03 | 5  | 5000 | 0,4039049 | 0,8299179 | 0,2928526 |
| 0,03 | 10 | 100  | 0,5974195 | 0,8502975 | 0,5027434 |
| 0,03 | 10 | 500  | 0,3696655 | 0,8565197 | 0,2653827 |
| 0,03 | 10 | 1000 | 0,3726692 | 0,8542646 | 0,2676883 |
| 0,03 | 10 | 5000 | 0,3923854 | 0,8389963 | 0,2843675 |
| 0,03 | 20 | 100  | 0,5973346 | 0,8500323 | 0,5024448 |
| 0,03 | 20 | 500  | 0,3695893 | 0,8565669 | 0,2654891 |
| 0,03 | 20 | 1000 | 0,3712164 | 0,8553292 | 0,2666155 |
| 0,03 | 20 | 5000 | 0,3841775 | 0,8453702 | 0,2775981 |
| 0,05 | 5  | 100  | 0,374757  | 0,8562247 | 0,271675  |
| 0,05 | 5  | 500  | 0,3740991 | 0,8531101 | 0,2690809 |
| 0,05 | 5  | 1000 | 0,3810439 | 0,8477828 | 0,2750012 |
| 0,05 | 5  | 5000 | 0,4182038 | 0,8187245 | 0,3037931 |
| 0,05 | 10 | 100  | 0,3746951 | 0,8562909 | 0,2714317 |
| 0,05 | 10 | 500  | 0,3717375 | 0,8549595 | 0,2669459 |
| 0,05 | 10 | 1000 | 0,3762373 | 0,8515544 | 0,2707936 |
| 0,05 | 10 | 5000 | 0,4055841 | 0,8286237 | 0,2943786 |
| 0,05 | 20 | 100  | 0,3751353 | 0,8558794 | 0,2715876 |
| 0,05 | 20 | 500  | 0,3705262 | 0,8558573 | 0,2660157 |
| 0,05 | 20 | 1000 | 0,3742159 | 0,8530278 | 0,269594  |
| 0,05 | 20 | 5000 | 0,3927668 | 0,8387291 | 0,2846423 |
| 0,1  | 5  | 100  | 0,3696293 | 0,8565124 | 0,2656004 |
| 0,1  | 5  | 500  | 0,3836375 | 0,8457663 | 0,2766507 |
| 0,1  | 5  | 1000 | 0,395784  | 0,8363408 | 0,286305  |
| 0,1  | 5  | 5000 | 0,4436142 | 0,7987656 | 0,3223175 |
| 0,1  | 10 | 100  | 0,3685534 | 0,8573547 | 0,2648151 |
| 0,1  | 10 | 500  | 0,3772087 | 0,8507996 | 0,2716218 |
| 0,1  | 10 | 1000 | 0,3872533 | 0,8429789 | 0,2802212 |
| 0,1  | 10 | 5000 | 0,4269916 | 0,8117968 | 0,3115944 |
| 0,1  | 20 | 100  | 0,3686009 | 0,8573012 | 0,2655424 |
| 0,1  | 20 | 500  | 0,3752234 | 0,8522269 | 0,2701843 |
| 0,1  | 20 | 1000 | 0,3812413 | 0,8475955 | 0,2753388 |
| 0,1  | 20 | 5000 | 0,4104174 | 0,8248572 | 0,2993734 |

Tabla 16. Métricas de rendimiento con diferentes parámetros para los modelos XGBoost.

| C    | sigma | RMSE      | Rsquared  | MAE       |
|------|-------|-----------|-----------|-----------|
| 0,01 | 0,01  | 0,587646  | 0,7315466 | 0,4573748 |
| 0,01 | 0,05  | 0,6246583 | 0,6711932 | 0,4706775 |
| 0,01 | 0,1   | 0,7383878 | 0,5440573 | 0,5591305 |
| 0,01 | 0,2   | 0,8483284 | 0,3849375 | 0,65381   |

|      |      |           |           |           |
|------|------|-----------|-----------|-----------|
| 0,01 | 0,5  | 0,9124818 | 0,2638602 | 0,7144448 |
| 0,01 | 1    | 0,923784  | 0,2421507 | 0,7253941 |
| 0,01 | 2    | 0,9272488 | 0,2367698 | 0,7282997 |
| 0,01 | 5    | 0,9320677 | 0,2324836 | 0,7329317 |
| 0,01 | 10   | 0,9368813 | 0,2299787 | 0,7380111 |
| 0,01 | 30   | 0,9455776 | 0,2210311 | 0,7472756 |
| 0,05 | 0,01 | 0,4646961 | 0,7844076 | 0,3491818 |
| 0,05 | 0,05 | 0,4871727 | 0,7655421 | 0,3531102 |
| 0,05 | 0,1  | 0,569054  | 0,6929101 | 0,4110051 |
| 0,05 | 0,2  | 0,6948193 | 0,5563222 | 0,5105104 |
| 0,05 | 0,5  | 0,8046958 | 0,3969231 | 0,6048532 |
| 0,05 | 1    | 0,8284719 | 0,3536881 | 0,6238177 |
| 0,05 | 2    | 0,8352624 | 0,3408792 | 0,6275368 |
| 0,05 | 5    | 0,8446123 | 0,3230148 | 0,6351889 |
| 0,05 | 10   | 0,854135  | 0,306845  | 0,645228  |
| 0,05 | 30   | 0,8732976 | 0,2782813 | 0,6678369 |
| 0,1  | 0,01 | 0,4425998 | 0,7988123 | 0,328645  |
| 0,1  | 0,05 | 0,4535896 | 0,7901828 | 0,3253443 |
| 0,1  | 0,1  | 0,5150774 | 0,738007  | 0,3662106 |
| 0,1  | 0,2  | 0,6252909 | 0,627804  | 0,4489771 |
| 0,1  | 0,5  | 0,7439757 | 0,4736542 | 0,5461838 |
| 0,1  | 1    | 0,7726009 | 0,4286295 | 0,5680978 |
| 0,1  | 2    | 0,7812608 | 0,4150052 | 0,5726838 |
| 0,1  | 5    | 0,7937062 | 0,394863  | 0,5823831 |
| 0,1  | 10   | 0,8063403 | 0,3736029 | 0,5947698 |
| 0,1  | 30   | 0,8298271 | 0,3336046 | 0,6197981 |
| 0,2  | 0,01 | 0,4276572 | 0,8096991 | 0,3136982 |
| 0,2  | 0,05 | 0,4302323 | 0,8081496 | 0,3069695 |
| 0,2  | 0,1  | 0,4773916 | 0,7679539 | 0,3362957 |
| 0,2  | 0,2  | 0,5664581 | 0,6828403 | 0,3991114 |
| 0,2  | 0,5  | 0,6821912 | 0,5456988 | 0,4888841 |
| 0,2  | 1    | 0,7137678 | 0,5020098 | 0,5119217 |
| 0,2  | 2    | 0,7231183 | 0,4894418 | 0,5162921 |
| 0,2  | 5    | 0,7371051 | 0,4689007 | 0,5268496 |
| 0,2  | 10   | 0,7521647 | 0,4459037 | 0,5403883 |
| 0,2  | 30   | 0,7809375 | 0,40082   | 0,5692845 |
| 0,5  | 0,01 | 0,4136566 | 0,8207479 | 0,2994806 |
| 0,5  | 0,05 | 0,412909  | 0,8214111 | 0,2939681 |
| 0,5  | 0,1  | 0,4469951 | 0,7919855 | 0,3151182 |
| 0,5  | 0,2  | 0,515826  | 0,7270992 | 0,3591976 |
| 0,5  | 0,5  | 0,6148992 | 0,6157795 | 0,4289997 |
| 0,5  | 1    | 0,6456257 | 0,5765637 | 0,4491458 |
| 0,5  | 2    | 0,6538922 | 0,5662433 | 0,4531364 |
| 0,5  | 5    | 0,668011  | 0,5484908 | 0,4635585 |
| 0,5  | 10   | 0,6852981 | 0,5252998 | 0,4786815 |
| 0,5  | 30   | 0,7180896 | 0,4788272 | 0,5097465 |
| 1    | 0,01 | 0,405327  | 0,8274963 | 0,2920593 |

|    |      |           |           |           |
|----|------|-----------|-----------|-----------|
| 1  | 0,05 | 0,4065916 | 0,8264148 | 0,2899365 |
| 1  | 0,1  | 0,4355894 | 0,8011719 | 0,3074552 |
| 1  | 0,2  | 0,4935535 | 0,7462069 | 0,3431363 |
| 1  | 0,5  | 0,5845531 | 0,6451625 | 0,4027939 |
| 1  | 1    | 0,6136247 | 0,6089627 | 0,4204814 |
| 1  | 2    | 0,6213272 | 0,599494  | 0,4245654 |
| 1  | 5    | 0,6351789 | 0,582137  | 0,4359443 |
| 1  | 10   | 0,6513758 | 0,5612225 | 0,4504136 |
| 1  | 30   | 0,6848352 | 0,5152958 | 0,4814797 |
| 2  | 0,01 | 0,39876   | 0,8328854 | 0,2866334 |
| 2  | 0,05 | 0,4052823 | 0,8274604 | 0,2889464 |
| 2  | 0,1  | 0,4332168 | 0,8029371 | 0,3061287 |
| 2  | 0,2  | 0,4862842 | 0,7520948 | 0,3368211 |
| 2  | 0,5  | 0,5726279 | 0,6562229 | 0,3919894 |
| 2  | 1    | 0,6021019 | 0,6197772 | 0,4090478 |
| 2  | 2    | 0,6109272 | 0,6089437 | 0,4145634 |
| 2  | 5    | 0,6258059 | 0,5901118 | 0,4278499 |
| 2  | 10   | 0,6423496 | 0,5685197 | 0,4427238 |
| 2  | 30   | 0,6739031 | 0,5256716 | 0,4716953 |
| 5  | 0,01 | 0,3935084 | 0,8371151 | 0,2818575 |
| 5  | 0,05 | 0,4100322 | 0,8236756 | 0,2926511 |
| 5  | 0,1  | 0,4364763 | 0,800129  | 0,3077468 |
| 5  | 0,2  | 0,4858383 | 0,7522146 | 0,3352447 |
| 5  | 0,5  | 0,5703121 | 0,6586971 | 0,3876773 |
| 5  | 1    | 0,602617  | 0,6192374 | 0,4087465 |
| 5  | 2    | 0,6137567 | 0,6052667 | 0,4170948 |
| 5  | 5    | 0,6305562 | 0,5836414 | 0,4324926 |
| 5  | 10   | 0,64625   | 0,5628705 | 0,4470344 |
| 5  | 30   | 0,6754544 | 0,5227648 | 0,4729772 |
| 10 | 0,01 | 0,3915741 | 0,838693  | 0,2800586 |
| 10 | 0,05 | 0,4159693 | 0,8190726 | 0,2964501 |
| 10 | 0,1  | 0,441941  | 0,7954699 | 0,3108484 |
| 10 | 0,2  | 0,4902961 | 0,7477895 | 0,3379603 |
| 10 | 0,5  | 0,5734597 | 0,6553308 | 0,389447  |
| 10 | 1    | 0,6066645 | 0,6148835 | 0,4117954 |
| 10 | 2    | 0,6186694 | 0,5995695 | 0,4211901 |
| 10 | 5    | 0,6365363 | 0,5761978 | 0,4381075 |
| 10 | 10   | 0,6497441 | 0,5582334 | 0,4500835 |
| 10 | 30   | 0,6769826 | 0,5204694 | 0,4737847 |
| 30 | 0,01 | 0,3931663 | 0,8376145 | 0,2803361 |
| 30 | 0,05 | 0,4279968 | 0,8095761 | 0,3042784 |
| 30 | 0,1  | 0,4586297 | 0,7812437 | 0,3205412 |
| 30 | 0,2  | 0,5083472 | 0,7306644 | 0,3469361 |
| 30 | 0,5  | 0,5873427 | 0,6407095 | 0,3978201 |
| 30 | 1    | 0,6195239 | 0,6008272 | 0,4200804 |
| 30 | 2    | 0,6316426 | 0,5848515 | 0,4304549 |
| 30 | 5    | 0,6456343 | 0,5648301 | 0,4446879 |

|    |    |           |           |           |
|----|----|-----------|-----------|-----------|
| 30 | 10 | 0,6575838 | 0,5477802 | 0,4551359 |
| 30 | 30 | 0,6803507 | 0,5154891 | 0,474706  |

Tabla 17. Métricas de rendimiento con diferentes parámetros para los modelos SVM-RF.

## Código en R

```
setwd("C:\\Users\\Flia Barrios Camargo\\OneDrive\\MASTER CIENCIA DE DATOS\\10. TFM")
source("FuncionesMineria.R")
library(readr)
library(caret)

datos <- read_csv("C:\\Users\\Flia Barrios Camargo\\OneDrive\\MASTER CIENCIA DE
DATOS\\10. TFM\\Hospital_Inpatient_Discharges.csv")

#Tipo de datos
str(datos)
summary(datos)

#Convertir datos de caracteres a factores para su análisis
cols_factor <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16, 17, 18, 19, 20, 21, 23,
22, 24, 25, 26, 28, 29, 27, 31)
datos[cols_factor] <- lapply(datos[cols_factor], as.factor)

# Convertir caracteres a numéricos
datos$`Length of Stay` <- as.numeric(datos$`Length of Stay`)
datos$`Birth weight` <- as.numeric(datos$`Birth weight`)

sum(is.na(datos$`Birth weight`))
sum(is.na(datos$`Length of Stay`))

# Obtener los nombres de las variables que convertiste a factor
nombres_factor <- names(datos)[cols_factor]

# Ver el listado
nombres_factor

library(dplyr)

# Filtrar correctamente la variable
datos_trauma <- datos %>%
  filter(`Type of Admission` == "Trauma")

# Ver cuántas filas tienes
nrow(datos_trauma)

str(datos_trauma)
summary(datos_trauma)

#Determinar número de NAs
colSums(is.na(datos_trauma))

#Análisis de cada variable
#Hospital Service Area
funModeling::freq(datos_trauma$`Hospital Service Area`)

#Hospital County
funModeling::freq(datos_trauma$`Hospital County`)

#Pasar a caracter para poder agruparlo
datos_trauma$`Hospital County` <- as.character(datos_trauma$`Hospital County`)

#Agrupar categorías poco representadas
datos_trauma$`Hospital County` <- ifelse(datos_trauma$`Hospital County` %in%
c("Suffolk", "Broome", "Dutchess", "Manhattan", "Otsego", "Clinton",
"Westchester", "Queens", "Monroe", "Putnam", "Steuben", "Jefferson", "Sullivan" ),
"Others_county", datos_trauma$`Hospital
County`))

#Convertir a factor
datos_trauma$`Hospital County` <- as.factor(datos_trauma$`Hospital County`))
```

```

#Operating Certificate Number
funModeling::freq(datos_trauma$'Operating Certificate Number')
funModeling::freq(datos_trauma$'Permanent Facility Id')

#Facility Name
funModeling::freq(datos_trauma$'Facility Name')

#Age Group
funModeling::freq(datos_trauma$'Age Group')

#Cambiar los nombres de las categorías
datos_trauma <- datos_trauma %>%
  mutate('Age Group' = case_when(
    `Age Group` == "0 to 17" ~ 1,
    `Age Group` == "18 to 29" ~ 2,
    `Age Group` == "30 to 49" ~ 3,
    `Age Group` == "50 to 69" ~ 4,
    `Age Group` == "70 or older" ~ 5
  ))

#Convertir a numérica
datos_trauma$`Age Group` <- as.numeric(datos_trauma$`Age Group`)

#Gender
funModeling::freq(datos_trauma$'Gender')

datos_trauma$Gender <- as.character(datos_trauma$Gender)
datos_trauma$Gender[datos_trauma$Gender == "U"] <- NA
datos_trauma$Gender <- as.factor(datos_trauma$Gender)

#Race
funModeling::freq(datos_trauma$'Race')

# Agrupar 'Multi-racial' como 'Other Race'
datos_trauma$Race <- as.character(datos_trauma$Race)
datos_trauma$Race[datos_trauma$Race == "Multi-racial"] <- "Other Race"
datos_trauma$Race <- as.factor(datos_trauma$Race)

#Ethnicity
funModeling::freq(datos_trauma$'Ethnicity')

#Patient Disposition
funModeling::freq(datos_trauma$'Patient Disposition')

# Convertir a carácter para modificar
datos_trauma$`Patient Disposition` <- as.character(datos_trauma$`Patient Disposition`)

# Definir las categorías más representativas que se conservarán
principales <- c("Home or Self Care", "Skilled Nursing Home", "Home w/ Home Health
Services",
                "Inpatient Rehabilitation Facility", "Expired")

# Reemplazar las categorías poco frecuentes
datos_trauma$`Patient Disposition`[!datos_trauma$`Patient Disposition` %in% principales]
<- "Other Dispositions"

# Convertir de nuevo a factor
datos_trauma$`Patient Disposition` <- as.factor(datos_trauma$`Patient Disposition`)

#APR MDC Code
funModeling::freq(datos_trauma$'APR MDC Code')

# Convertir a carácter para modificar
datos_trauma$`APR MDC Code` <- as.character(datos_trauma$`APR MDC Code`)

# Definir las categorías más representativas que se conservarán
principales2 <- c("03", "18", "06", "11", "21", "20", "10", "14", "22", "16", "07",
                 "02", "19", "23", "12", "24", "15", "13", "17")

# Reemplazar las categorías poco frecuentes
datos_trauma$`APR MDC Code`[datos_trauma$`APR MDC Code` %in% principales2] <- "Other APR
MDC Code"

# Convertir de nuevo a factor
datos_trauma$`APR MDC Code` <- as.factor(datos_trauma$`APR MDC Code`)

```

```

#APR Severity of Illness Description
funModeling::freq(datos_trauma$'APR Severity of Illness Description')

#APR Risk of Mortality
funModeling::freq(datos_trauma$'APR Risk of Mortality')

#APR Medical Surgical Description
funModeling::freq(datos_trauma$'APR Medical Surgical Description')

#Payment Typology 1
funModeling::freq(datos_trauma$'Payment Typology 1')

# Convertir a carácter para modificar
datos_trauma$`Payment Typology 1` <- as.character(datos_trauma$`Payment Typology 1`)

# Definir las categorías más representativas que se conservarán
principales3 <- c("Blue Cross/Blue Shield", "Managed Care, Unspecified", "Self-Pay",
                  "Federal/State/Local/VA", "Department of Corrections")

# Reemplazar las categorías poco frecuentes
datos_trauma$`Payment Typology 1`[datos_trauma$`Payment Typology 1` %in% principales3]
<- "Other Payment Typology"

# Convertir de nuevo a factor
datos_trauma$`Payment Typology 1` <- as.factor(datos_trauma$`Payment Typology 1`)

#Emergency Department Indicator
funModeling::freq(datos_trauma$'Emergency Department Indicator')

install.packages("psych")
library(psych)

#Para otros estadísticos
describe(Filter(is.numeric, datos_trauma))

library(ggplot2)

#Length of Stay
hist(datos_trauma$"Length of Stay",
      breaks = 50,
      main = "Histograma detallado de Length of Stay",
      xlab = "Length of Stay",
      xlim = c(0, 40),
      col = "skyblue")

#Eliminación de variables que no usaré
datos_trauma <- datos_trauma[, -c(1, 3, 4, 5, 7, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22,
23, 28, 29, 30, 32)]

#Determinar número de NAs
colSums(is.na(datos_trauma))
# Mostrar proporción de NAs por variable (en porcentaje)
round(colMeans(is.na(datos_trauma)), 4)
sort(round(colMeans(is.na(datos_trauma)), 4), decreasing = TRUE)

#Variable objetivo
varObj <- datos_trauma$`Total Costs`

# Crear copia del dataset sin la variable objetivo
datos_imputar <- datos_trauma[, setdiff(names(datos_trauma), "Total Costs")]

#Imputaciones
# Imputo todas las variables cuantitativas (numéricas) usando el método "aleatorio".
datos_imputar[,as.vector(which(sapply(datos_imputar, class) == "numeric"))] <-
sapply(Filter(is.numeric, datos_imputar), function(x) ImputacionCuant(x, "aleatorio"))

# Imputo todas las variables cualitativas (categóricas) usando el método "aleatorio".
datos_imputar[,as.vector(which(sapply(datos_imputar, class) == "factor"))] <-
sapply(Filter(is.factor, datos_imputar), function(x) ImputacionCuali(x, "aleatorio"))

#A veces, al imputar variables categóricas, el tipo de dato puede cambiar de factor a
character.
datos_imputar[,as.vector(which(sapply(datos_imputar, class) == "character"))] <-
lapply(datos_imputar[,as.vector(which(sapply(datos_imputar, class) == "character"))],
factor)

```

```

#Reviso que no queden valores faltantes (NA) después de la imputación.
summary(datos_imputar)

#Unir nuevamente los datos
datos_completo <- cbind(datos_imputar, Total_Costs = varObj)

#Eliminar errores detectados
datos_completo <- datos_completo[datos_completo$Total_Costs <= 1000000, ]
datos_completo <- datos_completo[datos_completo$Total_Costs >= 10, ]
summary(datos_completo)

#Análisis variable obj
hist(datos_completo$"Total_Costs",
      breaks = 300,
      main = "Histograma detallado de Total Costs",
      xlab = "Total_Costs",
      xlim = c(0, 200000),
      col = "skyblue")

#Transformación logarítmica
datos_completo$Total_Costs <- log(datos_completo$Total_Costs)

#Volver a separar variable objetivo
varObj_limpio <- datos_completo$Total_Costs

#Dataset sin la variable objetivo
datos_limpios <- datos_completo[, setdiff(names(datos_completo), "Total_Costs")]

#Calcular la v de cramer
library(DescTools)
library(Hmisc)

variables_categoricas <- names(Filter(is.factor, datos_limpios))
v_cramer_resultados <- c()

for (var in variables_categoricas) {
  tabla <- table(datos_limpios[[var]], cut2(varObj_limpio, g = 5))
  if (min(dim(tabla)) > 1) {
    v_cramer_resultados[var] <- round(CramerV(tabla), 4)
  }
}

#Mostrar ordenado
sort(v_cramer_resultados, decreasing = TRUE)

#resultados ordenados
v_cramer_ordenado <- sort(v_cramer_resultados, decreasing = TRUE)

#Gráfico V de Cramer
par(mar = c(9, 4, 2, 2)) # abajo, izquierda, arriba, derecha

barplot(
  v_cramer_ordenado,
  las = 2,
  col = "steelblue",
  main = "V de Cramer con Total Costs",
  ylab = "V de Cramer",
  cex.names = 0.55,
  ylim = c(0, 1)
)

#Correlación
install.packages("corrplot")
install.packages("Hmisc")

# Cargar librerías
library(corrplot)
library(Hmisc)

# Seleccionar solo las numéricas (incluyendo la variable objetivo)
numericas <- datos_completo[, sapply(datos_completo, is.numeric)]

# Calcular matriz de correlación (usando cor o rcorr para más detalles)
matriz_cor <- cor(numericas, use = "complete.obs")

# Graficar

```

```

corrplot(matriz_cor, method = "color", type = "upper",
          addCoef.col = "black", tl.cex = 0.8,
          number.cex = 0.7, col = colorRampPalette(c("red", "white", "blue"))(200),
          mar = c(0,0,1,0), title = "Matriz de correlación entre variables numéricas")

#Quitar la diagonal (convirtiéndola en NA)
matriz_cor_sin_diag <- matriz_cor
diag(matriz_cor_sin_diag) <- NA

#Convertir en data frame largo para inspeccionar pares de variables
cor_sin_diag_df <- as.data.frame(as.table(matriz_cor_sin_diag))

#Eliminar NA (la diagonal) y duplicados (ya que es simétrica)
cor_sin_diag_df <- cor_sin_diag_df[!is.na(cor_sin_diag_df$Freq), ]
cor_sin_diag_df <- cor_sin_diag_df[as.character(cor_sin_diag_df$Var1) <
as.character(cor_sin_diag_df$Var2), ]

#Ordenar por magnitud de la correlación
cor_sin_diag_df <- cor_sin_diag_df[order(-abs(cor_sin_diag_df$Freq)), ]

#Ver resultados
print(cor_sin_diag_df)
summary(datos_completo)

#Crear dummies para variables categóricas
# Asegurar que los datos están como data.frame
datos_limpios <- as.data.frame(datos_limpios)

#Separar variables categóricas y numéricas
vars_cat <- names(Filter(is.factor, datos_limpios))
vars_num <- names(Filter(is.numeric, datos_limpios))

# Cargar librería
library(ggplot2)

cat_vars <- names(Filter(is.factor, datos_completo))

for (var in cat_vars) {
  p <- ggplot(datos_completo, aes(x = Total_Costs, fill = .data[[var]])) +
    geom_density(alpha = 0.4) +
    labs(title = paste("Distribución de Total Costs por", var),
         x = "Total Costs", fill = var) +
    xlim(quantile(datos_completo$Total_Costs, probs = c(0.01, 0.99), na.rm = TRUE)) +
    theme_minimal()

  print(p)
  readline(prompt = "Presiona Enter para continuar...")
}

#Crear dummies SOLO para las categóricas
library(dummies)
datos_dummies <- dummy.data.frame(datos_limpios, names = vars_cat, sep = ".")

#Identificar las columnas dummies creadas (que ahora son 0 y 1)
cols_dummies <- setdiff(names(datos_dummies), vars_num)

#Estandarizar solo las variables numéricas originales
means <- apply(datos_dummies[, vars_num], 2, mean, na.rm = TRUE)
sds <- apply(datos_dummies[, vars_num], 2, sd, na.rm = TRUE)

datos_escalados <- as.data.frame(scale(datos_dummies[, vars_num], center = means, scale
= sds))

#Unir escaladas (numéricas) + dummies (sin escalar)
datos_modelo <- cbind(datos_escalados, datos_dummies[, cols_dummies])

#Unir la variable objetivo
datos_modelo <- cbind(datos_modelo, Total_Costs = varObj_limpio)

summary(datos_modelo)

#Guardar los datos finales
saveRDS(data.frame(datos_modelo), "Datos_TFM")
datos_modelo <- readRDS("Datos_TFM")

```

```

summary(datos_modelo)

#SELECCIÓN DE VARIABLES
#-----
-----
source("funcion steprepetido.R")
archivo1<-datos_modelo
dput(names(archivo1))
vardep<-"Total_Costs"
nombres1<-
  c("Age.Group", "Length.of.Stay", "Hospital.County.Albany", "Hospital.County.Bronx",
    "Hospital.County.Kings", "Hospital.County.Nassau", "Hospital.County.Onondaga",
    "Hospital.County.Orange", "Hospital.County.Others_county",
    "Hospital.County.Richmond",
    "Gender.F", "Gender.M", "Race.Black.African.American", "Race.Other.Race",
    "Race.White", "Patient.Disposition.Expired",
    "Patient.Disposition.Home.or.Self.Care",
    "Patient.Disposition.Home.w..Home.Health.Services",
    "Patient.Disposition.Inpatient.Rehabilitation.Facility",
    "Patient.Disposition.Other.Dispositions",
    "Patient.Disposition.Skilled.Nursing.Home",
    "APR.MDC.Code.01", "APR.MDC.Code.04", "APR.MDC.Code.05", "APR.MDC.Code.08",
    "APR.MDC.Code.09", "APR.MDC.Code.25", "APR.MDC.Code.Other.APR.MDC.Code",
    "APR.Severity.of.Illness.Description.Extreme",
    "APR.Severity.of.Illness.Description.Major",
    "APR.Severity.of.Illness.Description.Minor",
    "APR.Severity.of.Illness.Description.Moderate",
    "APR.Risk.of.Mortality.Extreme", "APR.Risk.of.Mortality.Major",
    "APR.Risk.of.Mortality.Minor", "APR.Risk.of.Mortality.Moderate",
    "APR.Medical.Surgical.Description.Medical",
    "APR.Medical.Surgical.Description.Surgical",
    "Payment.Typology.1.Medicare", "Payment.Typology.1.Medicare",
    "Payment.Typology.1.Miscellaneous.Other",
    "Payment.Typology.1.Other.Payment.Typology",
    "Payment.Typology.1.Private.Health.Insurance", "Emergency.Department.Indicator.N",
    "Emergency.Department.Indicator.Y")

y<-archivo1[,vardep]
x<-archivo1[,nombres1]

#Con SBF
library(caret)
filtro<-sbf(x,y,sbfControl = sbfControl(functions = rfsbf,
                                         method = "repeatedcv", verbose = FALSE))

a<-dput(filtro$optVariables)

length(a)

#Toma 47 variables
#c("Age.Group", "Length.of.Stay", "Hospital.County.Albany", "Hospital.County.Bronx",
#"Hospital.County.Kings", "Hospital.County.Nassau", "Hospital.County.Onondaga",
#"Hospital.County.Orange", "Hospital.County.Others_county", "Hospital.County.Richmond",
#"Race.Black.African.American", "Race.Other.Race", "Patient.Disposition.Expired",
#"Patient.Disposition.Home.or.Self.Care",
#"Patient.Disposition.Home.w..Home.Health.Services",
#"Patient.Disposition.Inpatient.Rehabilitation.Facility",
#"Patient.Disposition.Other.Dispositions",
#"Patient.Disposition.Skilled.Nursing.Home", "APR.MDC.Code.04",
#"APR.MDC.Code.08", "APR.MDC.Code.09", "APR.MDC.Code.25",
#"APR.MDC.Code.Other.APR.MDC.Code",
#"APR.Severity.of.Illness.Description.Extreme",
#"APR.Severity.of.Illness.Description.Major",
#"APR.Severity.of.Illness.Description.Minor",
#"APR.Severity.of.Illness.Description.Moderate",
#"APR.Risk.of.Mortality.Extreme", "APR.Risk.of.Mortality.Major",
#"APR.Risk.of.Mortality.Minor", "APR.Risk.of.Mortality.Moderate",
#"APR.Medical.Surgical.Description.Medical",
#"APR.Medical.Surgical.Description.Surgical",
#"Payment.Typology.1.Medicare", "Payment.Typology.1.Other.Payment.Typology",
#"Emergency.Department.Indicator.N", "Emergency.Department.Indicator.Y")

#WRAPPERS
#Con RFE
control <- rfeControl(functions=rffuncs, method="repeatedcv", number=10)

```

```

#Corre el algoritmo RFE
results <- rfe(x, y, sizes=c(1:8), rfeControl=control)

selecrfe<-results$optVariables
length(selecrfe)
dput(selecrfe)

#45 variables c("Length.of.Stay", "Hospital.County.Nassau", "Hospital.County.Kings",
#"Hospital.County.Albany", "Hospital.County.Onondaga", "Hospital.County.Orange",
#"APR.MDC.Code.08", "Emergency.Department.Indicator.N",
"Emergency.Department.Indicator.Y",
#"APR.Medical.Surgical.Description.Surgical", "Age.Group",
"APR.Medical.Surgical.Description.Medical",
#"Patient.Disposition.Home.or.Self.Care", "APR.Severity.of.Illness.Description.Extreme",
#"Hospital.County.Bronx", "Hospital.County.Richmond",
"APR.Severity.of.Illness.Description.Major",
#"APR.Risk.of.Mortality.Extreme", "Payment.Typeology.1.Medicare",
#"APR.Risk.of.Mortality.Minor", "Race.Other.Race", "APR.MDC.Code.25",
#"APR.Severity.of.Illness.Description.Moderate", "Hospital.County.Others_county",
#"APR.Risk.of.Mortality.Major", "Patient.Disposition.Skilled.Nursing.Home",
#"APR.Severity.of.Illness.Description.Minor", "APR.MDC.Code.04",
#"Patient.Disposition.Expired", "APR.MDC.Code.01", "APR.MDC.Code.09",
#"APR.MDC.Code.Other.APR.MDC.Code", "Patient.Disposition.Home.w.Home.Health.Services",
#"APR.Risk.of.Mortality.Moderate", "Payment.Typeology.1.Miscellaneous.Other",
#"Race.White", "Payment.Typeology.1.Other.Payment.Typeology", "Gender.M",
#"Gender.F", "Patient.Disposition.Other.Dispositions", "Race.Black.African.American",
#"APR.MDC.Code.05", "Patient.Disposition.Inpatient.Rehabilitation.Facility",
#"Payment.Typeology.1.Medicaid", "Payment.Typeology.1.Private.Health.Insurance")

#STEPWISE, BACKWARD, FORWARD
library(MASS)

# CON AIC
full<-glm(Total_Costs~.,data=archivo1,family = gaussian(link = "identity"))
null<-glm(Total_Costs~1,data=archivo1,family = gaussian(link = "identity"))

selec1<-stepAIC(null,scope=list(upper=full),
                direction="both",family = gaussian(link = "identity"),trace=FALSE)

vec<-(names(selec1[[1]]))

length(vec)
dput(vec)

#27 variables menos el intercepto
#(Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
# "Hospital.County.Nassau", "Hospital.County.Kings", "APR.Risk.of.Mortality.Minor",
# "APR.Risk.of.Mortality.Moderate", "Patient.Disposition.Home.or.Self.Care",
# "Hospital.County.Onondaga", "APR.Severity.of.Illness.Description.Minor",
# "APR.Severity.of.Illness.Description.Moderate", "Hospital.County.Albany",
# "Emergency.Department.Indicator.Y", "Patient.Disposition.Other.Dispositions",
# "APR.Severity.of.Illness.Description.Extreme", "APR.MDC.Code.09",
# "Hospital.County.Bronx", "Patient.Disposition.Expired", "Hospital.County.Orange",
# "Race.Other.Race", "APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25",
# "APR.MDC.Code.08", "Age.Group", "Payment.Typeology.1.Miscellaneous.Other",
# "APR.MDC.Code.04", "Patient.Disposition.Inpatient.Rehabilitation.Facility")

#Con BIC ponemos k=log(n) en stepAIC, en este caso n=7283 observaciones, k=3.86
full<-glm(Total_Costs~.,data=archivo1,family = gaussian(link = "identity"))
null<-glm(Total_Costs~1,data=archivo1,family = gaussian(link = "identity"))

selec1<-stepAIC(null,scope=list(upper=full),
                direction="both",family = gaussian(link = "identity"),trace=FALSE,
k=3.86)

vec<-(names(selec1[[1]]))

length(vec)
dput(vec)
#25 menos el intercepto
#c("Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
#"Hospital.County.Nassau", "Hospital.County.Kings", "APR.Risk.of.Mortality.Minor",
#"APR.Risk.of.Mortality.Moderate", "Patient.Disposition.Home.or.Self.Care",
#"Hospital.County.Onondaga", "APR.Severity.of.Illness.Description.Minor",
#"APR.Severity.of.Illness.Description.Moderate", "Hospital.County.Albany",

```

```

#"Emergency.Department.Indicator.Y", "Patient.Disposition.Other.Dispositions",
#"APR.Severity.of.Illness.Description.Extreme", "APR.MDC.Code.09",
#"Hospital.County.Bronx", "Patient.Disposition.Expired", "Hospital.County.Orange",
#"Race.Other.Race", "APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25",
#"APR.MDC.Code.08", "Age.Group", "Payment.Typeology.1.Miscellaneous.Other")

#con BORUTA
library(Boruta)
out.boruta <- Boruta(Total_Costs~., data = archivo1)

#beep("fanfare")
print(out.boruta)

boruta.cosa <- TentativeRoughFix(out.boruta)
print(boruta.cosa)

plot(boruta.cosa, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(boruta.cosa$ImpHistory),function(i)
  boruta.cosa$ImpHistory[is.finite(boruta.cosa$ImpHistory[,i]),i])
names(lz) <- colnames(boruta.cosa$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
      at = 1:ncol(boruta.cosa$ImpHistory), cex.axis = 0.7)

sal2<-as.data.frame(Labels)
sal2$var<-rownames(sal2)

sal2<-sal2[order(-sal2$Labels),]

selectBoruta<-sal2[1:4,2]
sal<-data.frame(out.boruta$finalDecision)
sal2<-sal[which(sal$out.boruta.finalDecision=="Confirmed"),,drop=FALSE]
dput(row.names(sal2))
length(selectBoruta)
dput(selectBoruta)

#4 variables
#"Length.of.Stay", "Hospital.County.Nassau", "Hospital.County.Kings",
#"Hospital.County.Albany")

#Con MXM
library(MXM)
mmpc1 <- MMPCC(vardep, archivo1, max_k = 3, hash = TRUE,
               test = "testIndFisher")

mmpc1@selectedVars

a<-dput(names(archivo1[,c(mmpc1@selectedVars)]))
length(a)
a
#13 variables
#"Length.of.Stay" "Hospital.County.Albany" "Hospital.County.Kings"
"Hospital.County.Nassau"
#"Patient.Disposition.Home.or.Self.Care" "Patient.Disposition.Skilled.Nursing.Home"
#"APR.MDC.Code.09" "APR.Severity.of.Illness.Description.Extreme"
"APR.Severity.of.Illness.Description.Minor"
#"APR.Severity.of.Illness.Description.Moderate" "APR.Risk.of.Mortality.Minor"
#"APR.Medical.Surgical.Description.Medical" "APR.Medical.Surgical.Description.Surgical"

source("funcion steprepetido.R")
#STEP REPETIDO CON AIC
lista<-steprepetido(data=archivo1,vardep=c("Total_Costs"),
                  listconti=nombres1,
                  sinicio=12345,sfinal=12385,porcen=0.8,criterio="AIC")

tabla<-lista[[1]]
dput(lista[[2]][[1]])

#25 variables
#c("Length.of.Stay", "APR.Medical.Surgical.Description.Surgical",
#"Hospital.County.Nassau", "Hospital.County.Kings", "APR.Risk.of.Mortality.Minor",
#"APR.Risk.of.Mortality.Moderate", "Patient.Disposition.Home.or.Self.Care",
#"Hospital.County.Onondaga", "Emergency.Department.Indicator.N",
#"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
#"Hospital.County.Albany", "Patient.Disposition.Other.Dispositions",

```

```

#"APR.Severity.of.Illness.Description.Extreme", "APR.MDC.Code.09",
#"Hospital.County.Bronx", "Patient.Disposition.Expired", "Hospital.County.Orange",
#"Race.Other.Race", "APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25",
#"APR.MDC.Code.08", "Age.Group", "Payment.Typeology.1.Miscellaneous.Other",
#"Patient.Disposition.Inpatient.Rehabilitation.Facility")

#STEP REPETIDO BIC
lista<-steprepetido(data=archivo1,vardep=c("Total_Costs"),
                    listconti=nombres1,
                    sinicio=12345,sfinal=12385,porcen=0.8,criterio="BIC")

tabla<-lista[[1]]
dput(lista[[2]][[1]])
dput(lista[[2]][[2]])

#22 variables
#c("Length.of.Stay", "APR.Medical.Surgical.Description.Surgical",
#"Hospital.County.Nassau", "Hospital.County.Kings", "APR.Risk.of.Mortality.Minor",
#"APR.Risk.of.Mortality.Moderate", "Patient.Disposition.Home.or.Self.Care",
#"Hospital.County.Onondaga", "Emergency.Department.Indicator.N",
#"APR.Severity.of.Illness.Description.Minor",
#"APR.Severity.of.Illness.Description.Moderate",
#"Hospital.County.Albany", "Patient.Disposition.Other.Dispositions",
#"APR.Severity.of.Illness.Description.Extreme", "APR.MDC.Code.09",
#"Hospital.County.Bronx", "Patient.Disposition.Expired", "Hospital.County.Orange",
#"Race.Other.Race", "APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25",
#"APR.MDC.Code.08")

#21 variables
#c("Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
#"APR.Risk.of.Mortality.Minor", "Hospital.County.Nassau", "Hospital.County.Kings",
#"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
#"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09",
#"Emergency.Department.Indicator.N",
#"Hospital.County.Albany", "APR.Severity.of.Illness.Description.Minor",
#"APR.Severity.of.Illness.Description.Moderate",
#"Patient.Disposition.Other.Dispositions",
#"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
#"Patient.Disposition.Expired", "Race.Other.Race", "Hospital.County.Orange",
#"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25")

#Comparaciones de modelos de selección de variables
source("cruzadas avnnet y lin.R")

data<-datos_modelo

medias1<-cruzadalin(data=data,
                    vardep="Total_Costs",listconti=
                    c("Age.Group", "Length.of.Stay", "Hospital.County.Albany",
"Hospital.County.Bronx",
                    "Hospital.County.Kings", "Hospital.County.Nassau",
"Hospital.County.Onondaga",
                    "Hospital.County.Orange", "Hospital.County.Others_county",
"Hospital.County.Richmond",
                    "Race.Black.African.American", "Race.Other.Race",
"Patient.Disposition.Expired",
                    "Patient.Disposition.Home.or.Self.Care",
"Patient.Disposition.Home.w..Home.Health.Services",
                    "Patient.Disposition.Inpatient.Rehabilitation.Facility",
"Patient.Disposition.Other.Dispositions",
                    "Patient.Disposition.Skilled.Nursing.Home", "APR.MDC.Code.04",
"APR.MDC.Code.08", "APR.MDC.Code.09", "APR.MDC.Code.25",
"APR.MDC.Code.Other.APR.MDC.Code",
                    "APR.Severity.of.Illness.Description.Extreme",
"APR.Severity.of.Illness.Description.Major",
                    "APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
                    "APR.Risk.of.Mortality.Extreme", "APR.Risk.of.Mortality.Major",
"APR.Risk.of.Mortality.Minor", "APR.Risk.of.Mortality.Moderate",
                    "APR.Medical.Surgical.Description.Medical",
"APR.Medical.Surgical.Description.Surgical",
                    "Payment.Typeology.1.Medicare",
"Payment.Typeology.1.Other.Payment.Typeology",
                    "Emergency.Department.Indicator.N",
"Emergency.Department.Indicator.Y"),
                    listclass=c(""),grupos=4,sinicio=1234, repe=25)

```

```

medias1$modelo="SBF"

medias2<-cruzadalin(data=data,
                    vardep="Total_Costs",listconti=
                    c("Age.Group", "Length.of.Stay", "Hospital.County.Albany",
"Hospital.County.Bronx", "Hospital.County.Kings", "Hospital.County.Nassau",
"Hospital.County.Onondaga", "Hospital.County.Orange", "Hospital.County.Others_county",
"Hospital.County.Richmond", "Race.Black.African.American", "Race.Other.Race",
"Patient.Disposition.Expired", "Patient.Disposition.Home.or.Self.Care",
"Patient.Disposition.Home.w..Home.Health.Services",
"Patient.Disposition.Inpatient.Rehabilitation.Facility",
"Patient.Disposition.Other.Dispositions",
"Patient.Disposition.Skilled.Nursing.Home", "APR.MDC.Code.04",
"APR.MDC.Code.08", "APR.MDC.Code.09", "APR.MDC.Code.25",
"APR.MDC.Code.Other.APR.MDC.Code",
"APR.Severity.of.Illness.Description.Extreme",
"APR.Severity.of.Illness.Description.Major",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"APR.Risk.of.Mortality.Extreme", "APR.Risk.of.Mortality.Major",
"APR.Risk.of.Mortality.Minor", "APR.Risk.of.Mortality.Moderate",
"APR.Medical.Surgical.Description.Medical",
"APR.Medical.Surgical.Description.Surgical",
"Payment.Typology.1.Medicare",
"Payment.Typology.1.Other.Payment.Typology",
"Emergency.Department.Indicator.N",
"Emergency.Department.Indicator.Y"),
                    listclass=c(""),grupos=4,sinicio=1234,repe=25)

medias2$modelo="RFE"

medias3<-cruzadalin(data=data,
                    vardep="Total_Costs",listconti=
                    c("Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"APR.Risk.of.Mortality.Minor",
"APR.Risk.of.Mortality.Moderate",
"Patient.Disposition.Home.or.Self.Care",
"Hospital.County.Onondaga",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Hospital.County.Albany",
"Emergency.Department.Indicator.Y",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme",
"APR.MDC.Code.09",
"Hospital.County.Bronx", "Patient.Disposition.Expired",
"Hospital.County.Orange",
"Race.Other.Race", "APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25",
"APR.MDC.Code.08", "Age.Group",
"Payment.Typology.1.Miscellaneous.Other",
"APR.MDC.Code.04",
"Patient.Disposition.Inpatient.Rehabilitation.Facility"),
                    listclass=c(""),grupos=4,sinicio=1234,repe=25)

medias3$modelo="STEP-AIC"

medias4<-cruzadalin(data=data,
                    vardep="Total_Costs",listconti=
                    c("Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"APR.Risk.of.Mortality.Minor",
"APR.Risk.of.Mortality.Moderate",
"Patient.Disposition.Home.or.Self.Care",
"Hospital.County.Onondaga",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Hospital.County.Albany",
"Emergency.Department.Indicator.Y",
"Patient.Disposition.Other.Dispositions",

```

```

"APR.MDC.Code.09", "APR.Severity.of.Illness.Description.Extreme",
"Hospital.County.Bronx", "Patient.Disposition.Expired",
"Hospital.County.Orange", "Race.Other.Race", "APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25", "APR.MDC.Code.08", "Age.Group",
"Payment.Typology.1.Miscellaneous.Other"),
listclass=c(""), grupos=4, sinicio=1234, repe=25)

medias4$modelo="STEP-BIC"

medias5<-cruzadalin(data=data,
vardep="Total_Costs", listconti=
c("Length.of.Stay", "Hospital.County.Nassau",
"Hospital.County.Kings",
"Hospital.County.Albany"),
listclass=c(""), grupos=4, sinicio=1234, repe=25)

medias5$modelo="Boruta"

medias6<-cruzadalin(data=data,
vardep="Total_Costs", listconti=
c("Length.of.Stay", "Hospital.County.Albany",
"Hospital.County.Kings", "Hospital.County.Nassau",
"Patient.Disposition.Home.or.Self.Care",
"Patient.Disposition.Skilled.Nursing.Home",
"APR.MDC.Code.09",
"APR.Severity.of.Illness.Description.Extreme",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"APR.Risk.of.Mortality.Minor",
"APR.Medical.Surgical.Description.Medical",
"APR.Medical.Surgical.Description.Surgical"),
listclass=c(""), grupos=4, sinicio=1234, repe=25)

medias6$modelo="MXM"

medias7<-cruzadalin(data=data,
vardep="Total_Costs", listconti=
c("Length.of.Stay", "APR.Medical.Surgical.Description.Surgical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"APR.Risk.of.Mortality.Minor",
"APR.Risk.of.Mortality.Moderate",
"Patient.Disposition.Home.or.Self.Care",
"Hospital.County.Onondaga", "Emergency.Department.Indicator.N",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Hospital.County.Albany",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme",
"APR.MDC.Code.09",
"Hospital.County.Bronx", "Patient.Disposition.Expired",
"Hospital.County.Orange",
"Race.Other.Race", "APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25",
"APR.MDC.Code.08", "Age.Group",
"Payment.Typology.1.Miscellaneous.Other",
"Patient.Disposition.Inpatient.Rehabilitation.Facility"),
listclass=c(""), grupos=4, sinicio=1234, repe=25)

medias7$modelo="STEP-REP-AIC"

medias8<-cruzadalin(data=data,
vardep="Total_Costs", listconti=
c("Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor", "Hospital.County.Nassau",
"Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care",
"Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09",
"Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",

```

```

"APR.Severity.of.Illness.Description.Extreme",
"Hospital.County.Bronx", "Patient.Disposition.Expired", "Race.Other.Race",
"Hospital.County.Orange", "APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25"),
listclass=c(""), grupos=4, sinicio=1234, repe=25)

medias8$modelo="R. Lineal"

union1<-rbind(medias1,medias2,medias3,medias4,medias5, medias6, medias7, medias8)
par(cex.axis=0.7)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1)
boxplot(data=union1,col="pink",error2~modelo)

#Mejorar escala
union1<-rbind(medias1,medias2,medias3,medias4,medias7, medias8 )
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1)
boxplot(data=union1,col="pink",error2~modelo)

#Obtención de los parámetros de la regresión
summary(datos_completo)
modelo_lineal <- lm(Total_Costs ~ + `Hospital County` + `Age Group` + `Gender` + `Race`
+ `Length of Stay` +
`Patient Disposition` + `APR MDC Code` + `APR Severity of Illness
Description` + `APR Risk of Mortality` +
`APR Medical Surgical Description` + `Payment Typology 1` +
`Emergency Department Indicator`,
data = datos_completo)

summary(modelo_lineal)

#Bagging
library(randomForest)
rfgrid<-expand.grid(mtry=c(21))
set.seed(1234)

str(datos_modelo$Total_Costs)

rfbis<-randomForest(Total_Costs ~ Length.of.Stay +
APR.Medical.Surgical.Description.Medical +
APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor
+
APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
APR.Severity.of.Illness.Description.Extreme +
Hospital.County.Bronx +
Patient.Disposition.Expired + Race.Other.Race +
Hospital.County.Orange +
APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
data=data,
mtry=21,ntree=1000,samplesize=300,nodesize=10,replace=TRUE)

plot(rfbis$mse)

rfbis1<-randomForest(Total_Costs ~ Length.of.Stay +
APR.Medical.Surgical.Description.Medical +
APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga
+
APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
Hospital.County.Albany +
APR.Severity.of.Illness.Description.Minor +

```

```

APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
APR.Severity.of.Illness.Description.Extreme +
Hospital.County.Bronx +
Patient.Disposition.Expired + Race.Other.Race +
Hospital.County.Orange +
APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
data=data,
mtry=21,ntree=1000,samplesize=300,nodesize=20,replace=TRUE)

plot(rfbis1$mse)

rfbis2<-randomForest(Total_Costs ~ Length.of.Stay +
APR.Medical.Surgical.Description.Medical +
APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga
+
APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
Hospital.County.Albany +
APR.Severity.of.Illness.Description.Minor +
APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
APR.Severity.of.Illness.Description.Extreme +
Hospital.County.Bronx +
Patient.Disposition.Expired + Race.Other.Race +
Hospital.County.Orange +
APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
data=data,
mtry=23,ntree=1000,samplesize=300,nodesize=30,replace=TRUE)

plot(rfbis2$mse)

source ("cruzadas avnnet y lin.R")
source ("cruzada arbol continua.R")
source ("cruzada rf continua.R")

medias9<-cruzadaarbol(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
listclass=c(""),
grupos=4,sinicio=1234,pepe=5,cp=0,minbucket=5)

medias9$modelo="Arbol"

medias10<-cruzadarf(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",

```

```

"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",

"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
  listclass=c(""),
  grupos=4,sinicio=1234, repe=5,
  nodesize=10,replace=TRUE,ntree=200,mtry=21)

medias10$modelo="Bagging base"

# Calcular MSE promedio del modelo
mse_gbm <- mean(medias18$error)

# Calcular la varianza de la variable objetivo
var_total_costs <- var(data$Total_Costs)

# Calcular R²
r2_gbm <- 1 - (mse_gbm / var_total_costs)
r2_gbm

union1<-rbind(medias9, medias10)
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1.2)
boxplot(data=union1,col="pink",error2~modelo)

medias11<-cruzararf(data=data,
                    vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
  listclass=c(""),
  grupos=10,sinicio=1234, repe=5, sampsize= 1000,
  nodesize=10,replace=TRUE,ntree=200,mtry=21)

medias11$modelo="Bagging1000"

medias12<-cruzararf(data=data,
                    vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",

```

```

"Race.Other.Race", "Hospital.County.Orange", "Patient.Disposition.Expired",
"APR.MDC.Code.25"), "APR.Risk.of.Mortality.Extreme",
listclass=c(""),
grupos=10,sinicio=1234,pepe=5, sampsize= 3000,
nodesize=10,replace=TRUE,ntree=200,mtry=21)

medias12$modelo="Bagging3000"

medias13<-cruzadarf(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
listclass=c(""),
grupos=10,sinicio=1234,pepe=5, sampsize= 5000,
nodesize=10,replace=TRUE,ntree=200,mtry=21)

medias13$modelo="Bagging5000"

union1<-rbind(medias10, medias11, medias12, medias13 )
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1.2)
boxplot(data=union1,col="pink",error2~modelo)

#Random Forest
library(randomForest)
set.seed(12345)
rfgrid<-expand.grid(mtry=c(5,10,15,19,21))
control<-trainControl(method = "repeatedcv",number=4,savePredictions = "all",
classProbs=TRUE)

rf<- train(Total_Costs ~ Length.of.Stay + APR.Medical.Surgical.Description.Medical +
APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange +
APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
data=data,
method="rf",trControl=control,tuneGrid=rfgrid,
ntree=200,sampsize=5000,nodesize=10,replace=TRUE,
importance=TRUE)

rf

medias14<-cruzadarf(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",

```

```

"APR.MDC.Code.09", "Emergency.Department.Indicator.N", "APR.Risk.of.Mortality.Moderate",
"APR.Severity.of.Illness.Description.Minor", "Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Race.Other.Race", "Hospital.County.Orange", "Patient.Disposition.Expired",
"APR.MDC.Code.25"), "APR.Risk.of.Mortality.Extreme",
listclass=c(""),
grupos=10,sinicio=1234, repe=5,
nodesize=10,replace=TRUE,ntree=200,mtry=10)

medias14$modelo="RF base"

medias15<-cruzadarf(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
listclass=c(""),
grupos=10,sinicio=1234, repe=5,sampsize= 1000,
nodesize=10,replace=TRUE,ntree=200,mtry=10)

medias15$modelo="RF1000"

medias16<-cruzadarf(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
listclass=c(""),
grupos=10,sinicio=1234, repe=5,sampsize= 3000,
nodesize=10,replace=TRUE,ntree=200,mtry=10)

medias16$modelo="RF3000"

medias17<-cruzadarf(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",

```

```

"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
                                         "APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
                                         "Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
                                         "Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
                                         "APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
      listclass=c(""),
      grupos=10,sinicio=1234, repe=5,sampsize= 5000,
      nodesize=10,replace=TRUE,ntree=200,mtry=10)

medias17$modelo="RF5000"

union1<-rbind(medias14, medias15, medias16, medias17)
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1.2)
boxplot(data=union1,col="pink",error2~modelo)

#Gradient boosting
gbmgrid<-expand.grid(shrinkage=c(0.1,0.05,0.03,0.01,0.001),
                     n.minobsinnode=c(5,10,20),
                     n.trees=c(100,500,1000,5000),
                     interaction.depth=c(2))

control<-trainControl(method = "repeatedcv",number=4,savePredictions = "all")

gbm<- train(Total_Costs ~ Length.of.Stay + APR.Medical.Surgical.Description.Medical +
            APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
            Hospital.County.Kings +
            Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
            APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
            Emergency.Department.Indicator.N +
            Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
            APR.Severity.of.Illness.Description.Moderate +
            Patient.Disposition.Other.Dispositions +
            APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
            Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange +
            APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
            method="gbm",trControl=control,tuneGrid=gbmgrid,data=data,
            distribution="gaussian", bag.fraction=1,verbose=FALSE)

gbm
plot(gbm)

#Con early stopping
gbmgrid1<-expand.grid(shrinkage=c(0.1),
                      n.minobsinnode=c(20),
                      n.trees=c(100,300,500,800,1000,1200,2000,5000,7000),
                      interaction.depth=c(2))

control_gbm1<-trainControl(method = "repeatedcv",number=4,savePredictions = "all")

gbm1<- train(Total_Costs ~ Length.of.Stay + APR.Medical.Surgical.Description.Medical +
            APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
            Hospital.County.Kings +
            Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
            APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
            Emergency.Department.Indicator.N +
            Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
            APR.Severity.of.Illness.Description.Moderate +
            Patient.Disposition.Other.Dispositions +
            APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
            Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange +
            APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
            data=data, method="gbm",trControl=control_gbm1,tuneGrid=gbmgrid1,

```

```

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

gbm1
plot(gbm1)

source("cruzada gbm continua.R")

medias18<-cruzadagbm(data=data,
                      vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Race.Other.Race", "Hospital.County.Orange",
"APR.MDC.Code.25"),
                      listclass=c(""),
                      grupos=10,sinicio=1234,repe=5,
                      n.minobsinnode=20,shrinkage=0.1,n.trees=2000,interaction.depth=2)

medias18$modelo="gbm2000"

medias19<-cruzadagbm(data=data,
                      vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Race.Other.Race", "Hospital.County.Orange",
"APR.MDC.Code.25"),
                      listclass=c(""),
                      grupos=10,sinicio=1234,repe=5,
                      n.minobsinnode=20,shrinkage=0.1,n.trees=5000,interaction.depth=2)

medias19$modelo="gbmbase"

medias20<-cruzadagbm(data=data,
                      vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",

```

```

"Race.Other.Race", "Hospital.County.Orange", "Patient.Disposition.Expired",
"APR.MDC.Code.25"), "APR.Risk.of.Mortality.Extreme",
listclass=c(""),
grupos=10,sinicio=1234,repo=5,
n.minobsinnode=20,shrinkage=0.1,n.trees=3000,interaction.depth=2)

medias20$modelo="gbm3000"

medias21<-cruzadagbm(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
listclass=c(""),
grupos=10,sinicio=1234,repo=5,
n.minobsinnode=20,shrinkage=0.1,n.trees=4000,interaction.depth=2)

medias21$modelo="gbm4000"

union1<-rbind(medias18, medias19, medias20, medias21)
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1.2)
boxplot(data=union1,col="pink",error2~modelo)

#XGBoost
set.seed(12345)

xgbmgrid<-expand.grid(
min_child_weight=c(5,10,20),
eta=c(0.1,0.05,0.03,0.01,0.001),
nrounds=c(100,500,1000,5000),
max_depth=6,gamma=0,colsample_bytree=1,subsample=1)

control_xgb<-trainControl(method = "repeatedcv",number=4,savePredictions = "all")

xgbm<- train(Total_Costs ~ Length.of.Stay + APR.Medical.Surgical.Description.Medical +
APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange +
APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
data=data,
method="xgbTree",trControl=control_xgb,
tuneGrid=xgbmgrid,verbose=FALSE)

xgbm
plot(xgbm)

#Con early stopping
xgbmgrid1<-expand.grid(eta=c(0.1),

```

```

min_child_weight=c(10),
nrounds=c(50, 100, 500, 1000, 5000),
max_depth=6,gamma=0,colsample_bytree=1,subsample=1)
set.seed(12345)
control_xgb1<-trainControl(method = "repeatedcv",number=4,savePredictions = "all")
xgbm1<- train(Total_Costs ~ Length.of.Stay + APR.Medical.Surgical.Description.Medical +
APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange +
APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
data=data,
method="xgbTree",trControl=control_xgb1,
tuneGrid=xgbmgrid1,verbose=FALSE)

varImp(xgbm1, scale = FALSE)
plot(varImp(xgbm1, scale = FALSE))

source("cruzada xgboost continua.R")
medias22<-cruzadaxgbm(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
listclass=c(""),
grupos=10,sinicio=1234,repe=5,
min_child_weight=10,eta=0.1,nrounds=100,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1)

medias22$modelo="xgbm Base"

medias23<-cruzadaxgbm(data=data,
vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
listclass=c(""),
grupos=10,sinicio=1234,repe=5, alpha=0.5,
min_child_weight=10,eta=0.1,nrounds=100,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1)

```

```

medias23$modelo="xgbm0.5"

varImp(xgbm1)
plot(varImp(xgbm1))

medias24<-cruzadaxgbm(data=data,
                      vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Race.Other.Race", "Hospital.County.Orange",
"APR.MDC.Code.25"),
                      listclass=c(""),
                      grupos=10,sinicio=1234, repe=5, alpha=0.3,
                      min_child_weight=10, eta=0.1, nrounds=100, max_depth=6,
                      gamma=0, colsample_bytree=1, subsample=1)

medias24$modelo="xgbm0.3"

medias25<-cruzadaxgbm(data=data,
                      vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Race.Other.Race", "Hospital.County.Orange",
"APR.MDC.Code.25"),
                      listclass=c(""),
                      grupos=10,sinicio=1234, repe=5, alpha=0.7,
                      min_child_weight=10, eta=0.1, nrounds=100, max_depth=6,
                      gamma=0, colsample_bytree=1, subsample=1)

medias25$modelo="xgbm0.7"

union1<-rbind(medias22, medias23, medias24, medias25)
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1.2)
boxplot(data=union1,col="pink",error2~modelo)

#SVM
source("cruzada SVM continua lineal.R")
source("cruzada SVM continua polinomial.R")
source("cruzada SVM continua RBF.R")

SVMgrid<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10))

control_SVM<-trainControl(method = "repeatedcv",number=4,

```

```

savePredictions = "all")

SVM<- train(data=data,Total_Costs ~ Length.of.Stay +
APR.Medical.Surgical.Description.Medical +
      APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
      Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
      APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
      Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
      APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
      APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
      Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange +
      APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
method="svmLinear",trControl=control_SVM,
tuneGrid=SVMgrid,verbose=FALSE)

SVM$results
plot(SVM$results$C,SVM$results$RMSE)

medias26<-cruzadaSVM(data=data,
      vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
      "APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
      "Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
      "APR.Risk.of.Mortality.Moderate",
"APR.MDC.Code.09", "Emergency.Department.Indicator.N",
      "Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
      "APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
      "APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
      "Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
      "APR.Risk.of.Mortality.Extreme",
"APR.MDC.Code.25"),
      listclass=c(""),
      grupos=10,sinicio=1234,repe=5,C=0.01)

medias26$modelo="SVM-Lineal"

#SVM CON RF
SVMgrid_RF<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10,30),
      sigma=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10,30))

control_SVMgrid_RF<-trainControl(method = "repeatedcv",
      number=4,savePredictions = "all")

SVMgrid_RF<- train(data=data,Total_Costs ~ Length.of.Stay +
APR.Medical.Surgical.Description.Medical +
      APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
Hospital.County.Kings +
      Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
      APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
Emergency.Department.Indicator.N +
      Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor
+
      APR.Severity.of.Illness.Description.Moderate +
Patient.Disposition.Other.Dispositions +
      APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx
+
      Patient.Disposition.Expired + Race.Other.Race +
Hospital.County.Orange +
      APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
method="svmRadial",trControl=control_SVMgrid_RF,
tuneGrid=SVMgrid_RF,verbose=FALSE)

SVMgrid_RF

dat<-as.data.frame(SVMgrid_RF$results)

```

```

ggplot(dat, aes(x=factor(C), y=RMSE,
                 color=factor(sigma)))+
  geom_point(position=position_dodge(width=0.5),size=3)

medias27<-cruzadaSVMRBF(data=data,
                        vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25"),
listclass=c("")),
grupos=10,sinicio=1234, repe=5,C=10,sigma=0.01)

medias27$modelo="SVM-RBF"

#Polinomeal
library(doParallel)

#Activar procesamiento en paralelo con 4 núcleos (ajusta según tu equipo)
cl <- makePSOCKcluster(4)
registerDoParallel(cl)

#Definir la parrilla de parámetros más reducida
SVMgrid_P <- expand.grid(C = c(0.1, 1, 10),
                        degree = 2,
                        scale = c(0.1, 1))

#Control de validación cruzada
control_SVM_P <- trainControl(method = "repeatedcv",
                             number = 4,
                             repeats = 2,
                             savePredictions = "all")

#Entrenamiento del modelo SVM con kernel polinomial
set.seed(1234)
SVM_P <- train(Total_Costs ~ Length.of.Stay + APR.Medical.Surgical.Description.Medical +
               APR.Risk.of.Mortality.Minor + Hospital.County.Nassau +
               Hospital.County.Kings +
               Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
               APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 +
               Emergency.Department.Indicator.N +
               Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
               APR.Severity.of.Illness.Description.Moderate +
               Patient.Disposition.Other.Dispositions +
               APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
               Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange
               +
               APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
               data = data,
               method = "svmPoly",
               trControl = control_SVM_P,
               tuneGrid = SVMgrid_P,
               verbose = FALSE)

print(SVM_P)
print(SVM_P$results)

#Apagar clúster paralelo
stopCluster(cl)
registerDoSEQ()

```

```

medias28<-cruzadaSVMpoly(data=data,
                          vardep="Total_Costs",listconti=c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25"),
                          listclass=c(""),
                          grupos=10,sinicio=1234, repe=5,C=0.1,degree=2,scale=0.1)

medias28$modelo="SVM-Poly"

union1<-rbind(medias26, medias27, medias28)
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1)
boxplot(data=union1,col="pink",error2~modelo)

#Red neuronal
#Calculos: 23 variables, 7283 obs , con 30 obs por parametro implica 7283/30=242.76
(242) parámetros max
#241/24 = 10.11 nodos (10)

vardep="Total_Costs"
variables<-c("Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor", "Hospital.County.Nassau",
"Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09",
"Emergency.Department.Indicator.N",
"Hospital.County.Albany", "APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired", "Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25")

data2<-data[,c(variables,vardep)]

control_rn<-trainControl(method = "cv",
                          number=4,savePredictions = "all")

set.seed(123)
nnetgrid <- expand.grid(size=c(5, 6, 7, 8, 9, 10, 11, 12,
13),decay=c(0.01,0.1,0.001,0.0001),bag=F)

completo<-data.frame()
listaite<-c(100, 500, 1000, 2000, 3000)

for (iter in listaite)
{
  rednnet<- train(Total_Costs~.,
                  data=data2,
                  method="avNNet",linout = TRUE,maxit=iter,
                  trControl=control,repeats=5,tuneGrid=nnetgrid,trace=F)
  # Añado la columna del parametro de iteraciones
  rednnet$results$itera<-iter
  # Voy incorporando los resultados a completo
  completo<-rbind(completo,rednnet$results)
}

```

```

}

completo<-completo[order(completo$RMSE),]

ggplot(completo, aes(x=factor(itera), y=RMSE,
                     color=factor(decay),pch=factor(size))) +
  geom_point(position=position_dodge(width=0.5),size=3)
#----
completo2<-completo[completo$size==6,]

ggplot(completo2, aes(x=factor(itera), y=RMSE,
                     color=factor(decay))) +
  geom_point(position=position_dodge(width=0.5),size=3)

#size = 6, decay = 0.01 e iteraciones = 500.
medias29<-cruzadaavnnnet(data=data2,
                        vardep="Total_Costs",listconti= c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor",
"Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09", "Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired",
"Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25"),
listclass=c(""),grupos=10,sinicio=1234, repe=5, itera=500,
size=c(6),decay=c(0.01))

medias29$modelo="Red1"

#size = 7, decay = 0.001 e iteraciones = 500.
medias30<-cruzadaavnnnet(data=data2,
                        vardep="Total_Costs",listconti=
c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor", "Hospital.County.Nassau",
"Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care",
"Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09",
"Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme",
"Hospital.County.Bronx",
"Patient.Disposition.Expired", "Race.Other.Race",
"Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25"),
,
listclass=c(""),grupos=10,sinicio=1234, repe=5, itera=500,
size=c(7),decay=c(0.001))

medias30$modelo="Red2"

#size = 8, decay = 0.1 e iteraciones = 500
medias31<-cruzadaavnnnet(data=data2,
                        vardep="Total_Costs",listconti=
c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor", "Hospital.County.Nassau",
"Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care",
"Hospital.County.Onondaga",

```

```

"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09",
"Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme",
"Hospital.County.Bronx",
"Patient.Disposition.Expired", "Race.Other.Race",
"Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25")
',
listclass=c(""), grupos=10, sinicio=1234, repe=5, itera=500,
size=c(8), decay=c(0.1))
medias31$modelo="Red3"

#size = 5, decay = 0.01 e iteraciones = 400
medias32<-cruzadaavnnnet(data=data2,
vardep="Total_Costs", listconti=
c("Length.of.Stay",
"APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor", "Hospital.County.Nassau",
"Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care",
"Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09",
"Emergency.Department.Indicator.N",
"Hospital.County.Albany",
"APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme",
"Hospital.County.Bronx",
"Patient.Disposition.Expired", "Race.Other.Race",
"Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25")
',
listclass=c(""), grupos=10, sinicio=1234, repe=5, itera=400,
size=c(5), decay=c(0.01))
medias32$modelo="Red4"

union1<-rbind(medias29, medias30, medias31, medias32)
par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
par(cex.axis=1)
boxplot(data=union1,col="pink",error2~modelo)

#Comparación de todos los modelos
union1<-rbind(medias11, medias16, medias18, medias23, medias27, medias30)

par(cex.axis=0.8)
boxplot(data=union1,col="pink",error~modelo)

union1$error2<-sqrt(union1$error)
boxplot(data=union1,col="pink",error2~modelo)

library(xgboost)
dput(names(data2))
vardep_xgb<-"Total_Costs"
nombres_xgb<- c("Length.of.Stay", "APR.Medical.Surgical.Description.Medical",
"APR.Risk.of.Mortality.Minor", "Hospital.County.Nassau", "Hospital.County.Kings",
"Patient.Disposition.Home.or.Self.Care", "Hospital.County.Onondaga",
"APR.Risk.of.Mortality.Moderate", "APR.MDC.Code.09",
"Emergency.Department.Indicator.N",
"Hospital.County.Albany", "APR.Severity.of.Illness.Description.Minor",
"APR.Severity.of.Illness.Description.Moderate",
"Patient.Disposition.Other.Dispositions",
"APR.Severity.of.Illness.Description.Extreme", "Hospital.County.Bronx",
"Patient.Disposition.Expired", "Race.Other.Race", "Hospital.County.Orange",
"APR.Risk.of.Mortality.Extreme", "APR.MDC.Code.25")
y_xgb<-data2[,vardep_xgb]
x_xgb<-data2[,nombres_xgb]

#Convertir a matriz y dmatrix

```

```

X_matrix <- as.matrix(x_xgb)
dtrain <- xgb.DMatrix(data = X_matrix, label = y_xgb)

# Paso 2: entrenar modelo final
modelo_xgb <- xgboost(data = dtrain,
                      eta = 0.01,
                      max_depth = 6,
                      nrounds = 5000,
                      min_child_weight = 20,
                      subsample = 1,
                      colsample_bytree = 1,
                      gamma = 0,
                      alpha = 0.5,
                      objective = "reg:squarederror",
                      verbose = 0)

# Paso 3: predecir sobre los mismos datos
predicciones <- predict(modelo_xgb, newdata = dtrain)

# Paso 4: evaluar
RMSE <- sqrt(mean((y_xgb - predicciones)^2))
R2 <- cor(y_xgb, predicciones)^2

cat("RMSE:", RMSE, "\n")
cat("R2:", R2, "\n")

#Crear dataframe con valores reales y predichos
resultados_modelo_ganador <- data.frame(Real = y_xgb,
                                         Predicho = predict(modelo_xgb, newdata = as.matrix(x_xgb)))

#Ver las primeras filas
head(resultados_modelo_ganador)

plot(resultados_modelo_ganador$Real, resultados_modelo_ganador$Predicho,
     xlab = "Costo real",
     ylab = "Costo predicho",
     main = "Real vs. Predicho - XGBoost",
     col = "darkblue", pch = 20)
abline(a = 0, b = 1, col = "red", lwd = 2)

#Ensamblado
library(caretEnsemble)
library(nnls)

set.seed(1234)

#Variable dependiente
vardep <- "Total_Costs"

#Control para regresión
control <- trainControl(
  method = "cv",
  number = 5,
  savePredictions = "final"
)

#Grids de los modelos mejores (con los hiperparámetros seleccionados)
grid_xgb <- expand.grid(
  nrounds = 100,
  max_depth = 6,
  eta = 0.1,
  gamma = 0,
  colsample_bytree = 1,
  min_child_weight = 10,
  subsample = 1
)

grid_gbm <- expand.grid(
  n.trees = 2000,
  interaction.depth = 2,
  shrinkage = 0.1,
  n.minobsinnode = 20
)

grid_red <- expand.grid(
  size = 7,

```

```

    decay = 0.001,
    bag = FALSE
  )

#Entrenamiento de modelos base
modelos <- caretList(
  Total_Costs ~ Length.of.Stay + APR.Medical.Surgical.Description.Medical +
    APR.Risk.of.Mortality.Minor + Hospital.County.Nassau + Hospital.County.Kings +
    Patient.Disposition.Home.or.Self.Care + Hospital.County.Onondaga +
    APR.Risk.of.Mortality.Moderate + APR.MDC.Code.09 + Emergency.Department.Indicator.N
+
  Hospital.County.Albany + APR.Severity.of.Illness.Description.Minor +
  APR.Severity.of.Illness.Description.Moderate +
  Patient.Disposition.Other.Dispositions +
  APR.Severity.of.Illness.Description.Extreme + Hospital.County.Bronx +
  Patient.Disposition.Expired + Race.Other.Race + Hospital.County.Orange +
  APR.Risk.of.Mortality.Extreme + APR.MDC.Code.25,
  data = data2,
  trControl = control,
  metric = "RMSE",
  tuneList = list(
    xgb = caretModelSpec(method = "xgbTree", tuneGrid = grid_xgb),
    gbm = caretModelSpec(method = "gbm", tuneGrid = grid_gbm, verbose = FALSE),
    red = caretModelSpec(method = "avNNet", tuneGrid = grid_red, linout = TRUE, trace =
FALSE)
  )
)

#Ensamblado usando regresión no negativa (NNLS)
ensamble <- caretEnsemble(
  modelos,
  metric = "RMSE",
  modelFunction = "nnls",
  trControl = trainControl(method = "cv", number = 5)
)

#Ver resultados
summary(ensamble)

library(dplyr)

#Para ensamblado
pred_ensamble <- data.frame(
  pred = predict(ensamble, newdata = data2),
  obs = data2$Total_Costs,
  Resample = modelos$xgb$pred$Resample
)

#Funciones
rmse_func <- function(obs, pred) sqrt(mean((obs - pred)^2))
r2_func <- function(obs, pred) cor(obs, pred)^2

#Crear tabla para guardar resultados
metricas <- data.frame()

#Función auxiliar para un modelo
extraer_metricas <- function(pred_df, modelo_name) {
  pred_df %>%
    group_by(Resample) %>%
    summarise(
      RMSE = rmse_func(obs, pred),
      R2 = r2_func(obs, pred),
      modelo = modelo_name,
      .groups = "drop"
    )
}

#Modelos
m1 <- extraer_metricas(modelos$xgb$pred, "xgbm")
m2 <- extraer_metricas(modelos$gbm$pred, "gbm")
m3 <- extraer_metricas(modelos$red$pred, "Red")
m4 <- extraer_metricas(pred_ensamble, "Ensamblado")

#Unión de todos
metricas <- bind_rows(m1, m2, m3, m4)

```

```

#Boxplot RMSE
boxplot(RMSE ~ modelo, data = metricas,
        col = "#FFCC80",
        main = "RMSE por modelo",
        ylab = "RMSE",
        cex.axis = 0.8, las = 2)

#Boxplot R2
boxplot(R2 ~ modelo, data = metricas,
        col = "#90CAF9",
        main = "R² por modelo",
        ylab = "R²",
        cex.axis = 0.8, las = 2)

#Comparaciones con SAS
install.packages("haven")
# Cargar librería
library(haven)

#Crear copia del objeto original
data_clean <- data

#Guardar nombres originales
original_names <- names(data_clean)

#Crear nombres limpios y acortados
new_names <- make.names(original_names)
new_names <- gsub("\\\\.", "_", new_names)
new_names <- substr(new_names, 1, 32)

#Asignar nuevos nombres a la tabla
names(data_clean) <- new_names

#Crear diccionario de nombres
diccionario <- data.frame(
  original = original_names,
  nuevo = new_names,
  stringsAsFactors = FALSE
)

library(haven)
write_sas(data_clean, "data_clean.sas7bdat")

Funciones minería
# Imputación variables cuantitativas
ImputacionCuant <- function(vv, tipo) {
  # Entrada:
  # - vv: Una variable numérica que puede contener valores faltantes (NA) y a la que se
  le quiere aplicar una imputación.
  # - tipo: Un argumento de texto que debe tomar uno de los siguientes valores:
  #         - "media": Para imputar los valores faltantes con la media de la variable.
  #         - "mediana": Para imputar los valores faltantes con la mediana de la
  variable.
  #         - "aleatorio": Para imputar los valores faltantes con un valor aleatorio
  generado a partir de la distribución de la variable.
  #
  # Salida:
  # - La variable 'vv' con los valores faltantes (NA) imputados según el tipo
  especificado.

  if (tipo == "media") {
    # Imputación por media:
    # Los valores faltantes (NA) en 'vv' se reemplazan con la media de la variable,
    redondeada a 4 decimales.
    vv[is.na(vv)] <- round(mean(vv, na.rm = TRUE), 4)

  } else if (tipo == "mediana") {
    # Imputación por mediana:
    # Los valores faltantes (NA) en 'vv' se reemplazan con la mediana de la variable,
    redondeada a 4 decimales.
    vv[is.na(vv)] <- round(median(vv, na.rm = TRUE), 4)

  } else if (tipo == "aleatorio") {
    # Imputación por valores aleatorios:
    # Los valores faltantes (NA) se reemplazan con valores aleatorios generados a partir
    de la densidad de la distribución de 'vv'.

```

```

# Se crea una densidad con la función density() y se utiliza una aproximación para
generar valores aleatorios basados en dicha distribución.
dd <- density(vv, na.rm = TRUE, from = min(vv, na.rm = TRUE), to = max(vv, na.rm =
TRUE))
vv[is.na(vv)] <- round(approx(cumsum(dd$y) / sum(dd$y), dd$x,
runif(sum(is.na(vv))))$y, 4)
}

# Devolver la variable 'vv' con los valores imputados
return(vv)
}

# Imputación variables cualitativas
ImputacionCuali <- function(vv, tipo) {
# Entrada:
# - vv: Una variable categórica que puede contener valores faltantes (NA) y a la que
se le quiere aplicar una imputación.
# - tipo: Un argumento de texto que debe tomar uno de los siguientes valores:
#       - "moda": Para imputar los valores faltantes con la categoría más frecuente
(modas).
#       - "aleatorio": Para imputar los valores faltantes con valores aleatorios
seleccionados de los valores no faltantes de la variable.
#
# Salida:
# - La variable 'vv' con los valores faltantes (NA) imputados según el tipo
especificado.

if (tipo == "moda") {
# Imputación por moda:
# Los valores faltantes (NA) se reemplazan por la categoría más frecuente (moda) de
la variable.
# sort(table(vv), decreasing = TRUE)[1] encuentra la categoría con mayor frecuencia.
vv[is.na(vv)] <- names(sort(table(vv), decreasing = TRUE))[1]

} else if (tipo == "aleatorio") {
# Imputación por valores aleatorios:
# Los valores faltantes (NA) se reemplazan con valores seleccionados aleatoriamente
de las observaciones no faltantes.
# sample() selecciona aleatoriamente valores de 'vv' que no son NA, con reemplazo.
vv[is.na(vv)] <- sample(vv[!is.na(vv)], sum(is.na(vv)), replace = TRUE)
}

# Devolver la variable 'vv' con los valores imputados
return(vv)
}

# Busca la transformación de una variable numérica que maximiza la correlación con una
variable objetivo continua
mejorTransfCorr <- function(vv, target) {
# Entrada:
# - vv: Variable numérica que se desea transformar.
# - target: Variable objetivo continua con la que se quiere maximizar la correlación.
#
# Salida:
# - Una lista que contiene:
#   1. El nombre de la transformación que maximiza la correlación con 'target'.
#   2. La variable transformada correspondiente.

# Paso 1: Estandarizar la variable 'vv' para tener media 0 y desviación estándar 1
vv <- scale(vv)

# Paso 2: Asegurar que todos los valores de 'vv' sean positivos
# Esto es necesario para algunas transformaciones (como logaritmo y raíz) que no están
definidas para valores negativos
vv <- vv + abs(min(vv, na.rm = TRUE)) * 1.0001

# Paso 3: Generar diferentes transformaciones de 'vv'
posiblesTransf <- data.frame(
  x = vv,                                # Variable original estandarizada
  logx = log(vv),                        # Logaritmo natural
  expx = exp(vv),                        # Exponencial
  sqrx = vv^2,                           # Cuadrado
  sqrtx = sqrt(vv),                     # Raíz cuadrada
  cuartax = vv^4,                        # Cuarta potencia
  raiz4 = vv^(1/4)                       # Raíz cuarta

```

```

)

# Paso 4: Calcular la correlación entre 'target' y cada transformación de 'vv'
# Se usa 'use = "complete.obs"' para manejar valores faltantes
correlaciones <- abs(cor(target, posiblesTransf, use = "complete.obs"))

# Paso 5: Identificar la transformación que maximiza la correlación
idx_max_cor <- which.max(correlaciones)

# Paso 6: Retornar el nombre de la mejor transformación y la variable transformada
correspondiente
nombre_mejor_transf <- colnames(posiblesTransf)[idx_max_cor]
mejor_variable <- posiblesTransf[, idx_max_cor]

return(list(nombre_mejor_transf, mejor_variable))
}

```