FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2023/2024

Trabajo de Fin de Máster

TITULO: Análisis de riesgo de padecer enfermedad cardíaca mediante técnicas de Machine Learning

***Alumno*: Alejandro Calvo Robas**

***Tutor*: Javier Castro Cantalejo**

Septiembre de 2024

UNIVERSIDAD COMPLUTENSE
MADRID

RESUMEN

Las enfermedades cardíacas son una de las enfermedades más mortales en el mundo, por lo que resulta de vital importancia estudiar todos los factores que pueden afectar a la mortalidad por este tipo de enfermedades.

A través de las respuestas a las preguntas de una encuesta telefónica, se define como objetivo principal estudiar qué factores pueden suponer un mayor riesgo de padecer una enfermedad cardíaca y la muerte por ello. Para conocer estos factores se utilizan técnicas de Machine Learning y se extrae el mejor modelo a través del cual se puede conocer el efecto de cada factor de riesgo. Con ello se pueden extraer conclusiones y dar recomendaciones al paciente.

ABSTRACT

Heart Disease is one of the most lethal diseases in the world, so it is important to study all the factors that can affect de mortality from this type of diseases.

Through the answers to the questions in a telephone survey, the main purpose of this project is to study which factors increase the risk of suffering from heart disease and death from it. To find out these factors, Machine Learning techniques are used, and the best model is extracted to know the effect of each risk factor. Therefore, conclusions are drawn, and recommendations can also be given to the patient.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

1.1. ¿QUÉ SON LAS ENFERMEDADES CARDÍACAS? IMPORTANCIA DE SU ESTUDIO.

Las enfermedades cardíacas son aquellas que afectan al corazón y a los vasos sanguíneos. Son enfermedades que no producen dolor y en la mayor parte de las ocasiones son pasadas por alto. Si no se detectan y se tratan a tiempo, pueden ocasionar problemas más graves como ataques al corazón, embolia cerebral o incluso el deterioro de la funcionalidad de los riñones.

Además, es de vital importancia tener en cuenta que hay diversas enfermedades que se pueden padecer a la vez, por lo que es necesario estar en alerta por la posibilidad de padecer alguna de las enfermedades.

Entre estas enfermedades se pueden destacar algunas como:

- Insuficiencia cardíaca. Se produce cuando el músculo del corazón no bombea sangre de forma correcta, provocando que el líquido se quede atrapado en los pulmones, provocando graves problemas.

- Hipertensión arterial. Aumenta el riesgo de sufrir un ataque cardíaco y se produce por un estrechamiento de las arteriolas, que se encargan de regular el flujo sanguíneo.

- Infarto de miocardio. Se produce por la obstrucción de una arteria, provocando la disminución del oxígeno que se pasan a las células del corazón, de forma que se destruye poco a poco el músculo.

Hay otras enfermedades cardíacas que no han sido mencionadas, pero todas tienen en común la afectación del corazón.

Para prevenir estas enfermedades y detectarlas a tiempo es importante observar los cambios que se producen en los latidos del corazón y acudir al especialista tan pronto como se tengan dudas. (Blog de Cardiología de los Hospitales Quirónsalud Alicante, Murcia, Torrevieja y Valencia)

En cuanto a la situación de las enfermedades cardiovasculares en España, se trata de la primera causa de muerte. Al margen de que todos los factores de riesgo son conocidos y, por tanto, evitables, los hábitos de las personas han ido empeorando hasta el punto de que en el año 2006 hubo 120.760 fallecidos, lo que supuso el 32.5% del total de ingresados en los hospitales.

Tal y como afirman Bertomeu y Castillo – Castillo (Bertomeu, Castillo-Castillo 2008), las causas más frecuentes de ingresos en hospitales fueron la insuficiencia cardíaca y la enfermedad cerebrovascular, representando el 24% de los ingresos cada una de ellas.

Los principales factores de riesgo relacionados con el desarrollo de estas enfermedades son el consumo de tabaco o la dislipemia (alteración en los niveles de grasas en sangre).

1.2. OBJETIVOS

Como se ha comentado anteriormente, las enfermedades cardíacas son un problema muy serio en la sociedad de hoy en día. Por ello es importante estudiar todos los factores de riesgo que puedan estar afectando para que aparezcan cada vez más enfermedades cardíacas.

Este trabajo tiene como objetivo principal abordar la problemática de las enfermedades cardíacas mediante el análisis de la base de datos proporcionada. Por tanto, se quiere predecir si un individuo va a tener o no una enfermedad cardíaca (infarto de miocardio o enfermedad coronaria) a partir de una serie de indicadores.

Para ello se utilizarán técnicas de aprendizaje automático para desarrollar modelos predictivos capaces de predecir la probabilidad de desarrollar enfermedades cardíacas en función de diversos factores de riesgo e identificar y analizar estos con el objetivo de proporcionar información relevante para la prevención y el tratamiento de estas afecciones.

Por otra parte, se pueden desarrollar así herramientas de apoyo a la decisión clínica basadas en los modelos predictivos desarrollados, con el propósito de ayudar a los profesionales de la salud en la identificación temprana y eficaz de los riesgos de enfermedades cardíacas en sus pacientes.

Por tanto, estos objetivos, se plantean con la finalidad de contribuir al avance del conocimiento en el campo de la salud cardiovascular, así como ofrecer diferentes herramientas y recursos que puedan ser útiles para la clínica y la toma de decisiones en el ámbito médico.

2. DESCRIPCIÓN DE LA BASE DE DATOS

La base de datos con la que se va a trabajar ha sido obtenida de "Kaggle" (https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease/data). Este conjunto de datos contiene información acerca de 320.000 personas y se ha recogido para cada una de ellas una serie de atributos.

Se trata de un conjunto de datos proveniente de CDC (Centers for Disease Control and Prevention) y es una parte importante del Sistema de Vigilancia de Factores de Riesgo del Comportamiento (BRFSS) que se dedican a realizar encuestas telefónicas cada año para recopilar información sobre la salud de los residentes de Estados Unidos.
BRFSS está considerado como el mayor programa de salud realizado de forma continua, ya que cada año se recopilan datos de 400.000 entrevistas.

Se recogen indicadores como la diabetes, el grado de obesidad, no tener una buena forma física o beber mucho alcohol.

2.1. DESCRIPCIÓN DE LAS VARIABLES

La mayor parte de las preguntas realizadas en las entrevistas se responden con un sí o no. Se muestra a continuación una tabla con todas las variables presentes en la base de datos, que son 18.

Tabla 1:Descripción de las variables

VARIABLE	DESCRIPCIÓN	TIPO
HeartDisease	¿Has tenido alguna vez enfermedad coronaria o infarto de miocardio? SI/NO	Cualitativa
BMI	Índice de masa corporal	Cuantitativa
Smoking	¿Has fumado al menos 100 cigarros en tu vida? SI/NO	Cualitativa
AlcoholDrinking	Hombres: ¿Bebes más de 14 copas a la semana? SI/NO Mujeres: ¿Bebes más de 7 copas a la semana? SI/NO	Cualitativa
Stroke	¿Has sufrido un ataque cerebral? SI/NO	Cualitativa
PhysicalHealth	Durante cuántos días de los últimos 30 su salud física no fue buena	Cuantitativa
MentalHealth	Durante cuántos días de los últimos 30 su salud mental no fue buena	Cuantitativa
DiffWalking	¿Tienes dificultad para caminar o subir escaleras? SI/NO	Cualitativa
Sex	Hombre o mujer	Cualitativa
AgeCategory	Edad (14 intervalos de edad)	Cualitativa
Race	¿Cuál es tu raza?	Cualitativa
Diabetes	¿Has tenido diabetes? SI/NO	Cualitativa
PyhisicalActivity	¿Has hecho ejercicio aparte de tu trabajo habitual? SI/NO	Cualitativa
GenHealth	En general, tu salud es…	Cualitativa
SleepTime	Número de horas de sueño durante las 24 horas de un día	Cuantitativa
Asthma	¿Has tenido asma? SI/NO	Cualitativa
KidneyDisease	¿Has tenido alguna enfermedad renal? SI/NO	Cualitativa
SkinCancer	¿Has tenido cáncer de piel? SI/NO	Cualitativa

Se observa que se tienen 14 variables categóricas, de las cuales la mayoría de ellas son contestadas con sí o no, y 4 variables cuantitativas.

La variable objetivo del estudio será HeartDisease, a partir de la cual se predecirá si se sufrirá un accidente cardíaco o no a partir del resto de variables anteriormente mencionadas.

En primer lugar, se realizará un análisis exploratorio de las variables para familiarizarse con el conjunto de datos y entender las características de los individuos encuestados.

En segundo lugar, se procederá a hacer una serie de modificaciones en la base de datos con el fin de trabajar de manera más cómoda con los datos.

2.2. ANÁLISIS EXPLORATORIO

A continuación, se hará un análisis exploratorio tanto de las 14 variables cualitativas como de las 4 variables cuantitativas.

Se distinguirán 3 subapartados: análisis univariante, análisis bivariante y análisis multivariante.

2.2.1. ANÁLISIS UNIVARIANTE

El análisis univariante es una técnica utilizada para explorar variables de forma individual. Su objetivo es entender la distribución y las características de estas.
Es de gran utilidad para un primer acercamiento a la base de datos y proporciona medidas como la media, mediana, cuartiles, permite identificar valores extremos que puedan afectar de forma negativa o sesgada a los resultados

Para las variables cualitativas se sacará el porcentaje de cada categoría para ver cuál es la más predominante en cada una de ellas.

Variable Smoking

Esta variable recoge información acerca de si el individuo ha fumado más de 100 cigarros durante su vida, sabiendo que 100 cigarros son unos 5 paquetes de tabaco.

Tabla 2:Frecuencia Smoking

	Frecuencia absoluta	**Porcentaje**
SI	131908	41.25
NO	187887	58.75

El 41.25% de los individuos afirma haber fumado al menos 100 cigarros en toda su vida, mientras que el 58.75% no lo ha hecho.

Variable AlcoholDrinking

Esta variable mide el número de copas bebidas a la semana. Para las mujeres se fija el umbral en 7 copas y para los hombres en 14 copas.

Tabla 3: Frecuencia AlcoholDrinking

	Frecuencia absoluta	Porcentaje
SI	21777	6.81
NO	298018	93.19

Casi la totalidad de las personas afirman no beber más de esa cantidad fija de copas.

Stroke

Los individuos deben responder si han sufrido algún ataque cerebral o no.

Tabla 4: Frecuencia Stroke

	Frecuencia absoluta	Porcentaje
SI	12069	3.77
NO	307726	96.23

El 96.23% de las personas no han sufrido ningún ataque cerebral. Sin embargo, el 3.77% si lo ha sufrido por lo que habrá que estar pendientes de estas personas, serán más propensas a sufrir un accidente cardiovascular.

DiffWalking

Se recogen datos acerca de si se tienen problemas para andar o subir escaleras o no.

Tabla 5: Frecuencia DiffWalking

	Frecuencia absoluta	Porcentaje
SI	44410	13.89
NO	275385	86.11

Sexo

Se debe responder si la persona es hombre o mujer.

Tabla 6: Frecuencia Sexo

	Frecuencia absoluta	Porcentaje
HOMBRE	151990	47.53
MUJER	167805	52.47

AgeCategory

Esta variable recoge la edad de las personas, pero no se recoge la edad exacta sino clasificada por tramos. Se tienen 14 intervalos desde los 18-24 hasta +80. Se muestra qué intervalos de edad son los más frecuentes.

Tabla 7: Frecuencia AgeCategory

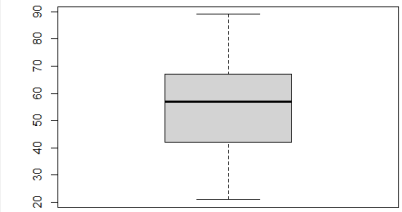
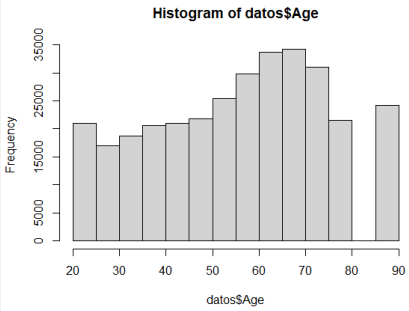
	Frecuencia absoluta	Porcentaje
18 – 24	21064	6.59
25 – 29	16955	5.30
30 – 34	18753	5.86
35 – 39	20550	6.43
40 – 44	21006	6.57
45 – 49	21791	6.81
50 – 54	25382	7.94
55 – 59	29757	9.31
60 – 64	33686	10.53
65 – 69	34151	10.68
70 – 74	31065	9.71
75 – 79	21482	6.72
80 +	24153	7.55

Se observa que el rango de edad que más responde a las preguntas de esta entrevista son personas de entre 50 a 74 años. Los que menos responden son aquellos individuos con 25-45 años.

Con el fin de mejorar los análisis predictivos, se hará una transformación de esta variable de forma que sea cuantitativa. Para ello se usa la marca de clase para cada uno de los intervalos y para el grupo +80 se utiliza la esperanza de vida del año 2020 proporcionada por el INE, que son 89.

Por tanto, tratando esta variable como cunatitativa se tienen los siguientes descriptivos y el histograma:

Tabla 8: Descriptivos, boxplot, histograma. Age

DESCRIPTIVOS		BOXPLOT	HISTOGRAMA
Min	**21**		
Max	**89**		
Mediana	**57**		
Media	**55.04**		

Se observa que el rango de edad varía desde los 21 hasta los 89 años. Donde hay una mayor respuesta a las encuestas telefónicas es alrededor de los 60 -70 años y donde menos entre los 20 y los 30 años.

Race

Se debe responder de qué raza son: nativo, asiático, negro, hispánico, blanco u otra.

Tabla 9: Frecuencia Race

	Frecuencia absoluta	Porcentaje
Indio	5202	1.63
Asiático	8068	2.52
Negro	22939	7.17
Blanco	245212	76.68
Otra	10928	3.42
Hispánico	27446	8.58

Casi el 80% de las personas son de raza blanca mientras que la raza india es la menos representada.

PhysicalActivity

Esta variable recoge información sobre si se ha hecho ejercicio físico o no durante los últimos 30 días además del trabajo propio.

Tabla 10: Frecuencia PhysicalActivity

	Frecuencia absoluta	Porcentaje
SI	247957	77.54
NO	71838	22.46

El 77.54% de las personas afirman haber realizado ejercicio físico durante los últimos 30 días.

Diabetic

La persona entrevistada debe responder si ha sufrido diabetes o no.

Tabla 11: Frecuencia Diabetic

	Frecuencia absoluta	Porcentaje
SI	43361	13.56
NO	276434	86.44

Cerca del 14% de las personas sí han sufrido diabetes, mientras que el resto no.

GenHealth

Tabla 12: Frecuencia GenHealth

	Frecuencia absoluta	Porcentaje
Excellent	66842	20.90
Fair	34677	10.84
Good	93129	29.12
Poor	11289	3.53
Very Good	113858	35.60

Más del 80% de las personas consideran que su salud es suficientemente buena. Y un 3.53% afirma que es muy pobre.

Asthma

Se debe responder si se tiene o se ha tenido asma.

Tabla 13: Frecuencia Asthma

	Frecuencia absoluta	Porcentaje
SI	42872	13.41
NO	276923	86.59

El 13.41% de los entrevistados tienen o han tenido asma.

KidneyDisease

Es una variable binaria que recoge si se ha tenido alguna enfermedad renal o no.

Tabla 14: Frecuencia KidneyDisease

	Frecuencia absoluta	Porcentaje
SI	11779	3.68
NO	308016	96.32

Tan sólo el 3.68% de la muestra ha tenido alguna enfermedad renal.

SkinCancer

Esta variable indica si se ha tenido o no cáncer de piel.

Tabla 15: Frecuencia SkinCancer

	Frecuencia absoluta	Porcentaje
SI	29819	9.32
NO	289976	90.68

El 9.32% tienen cáncer de piel.

HeartDisease

Esta variable es la variable objetivo del estudio y recoge si se ha tenido alguna enfermedad coronaria o infarto de miocardio o no.

Tabla 16:Frecuencia HeartDisease

	Frecuencia absoluta	Porcentaje
SI	27373	8.56
NO	292422	91.44

Menos del 10% ha sufrido un ataque cardíaco, mientras que el 90% restante no ha sufrido.

Se observa que es una variable que está muy desbalanceada. La categoría objetivo (Sí) está muy poco representada y esto puede traer problemas a la hora de aplicar técnicas de Machine Learning.

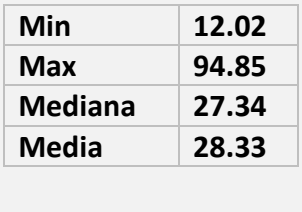
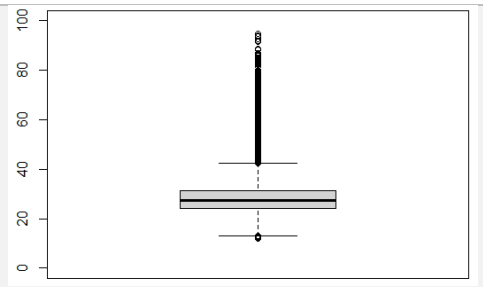
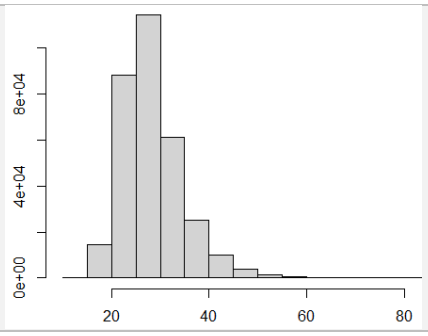
Para evitar que los modelos estén sesgados y que surjan muchos falsos positivos, es muy recomendable la aplicación de técnicas de submuestreo que se verán en el apartado 3.

Una vez exploradas las variables categóricas, se explorarán las variables cuantitativas que se tienen en la base de datos que son: BMI, PhysicalHealth, MentalHealth y SleepTime.
Para todas estas variables se sacarán unos descriptivos e histogramas. Para la variable BMI (índice de masa corporal) se mostrará además un boxplot para observar los outliers.

BMI

El índice de masa corporal es el peso de una persona en kilogramos dividido por el cuadrado de la estatura en metro.

Tabla 17: Descriptivos, boxplot, histograma. BMI

DESCRIPTIVOS	BOXPLOT	HISTOGRAMA
		

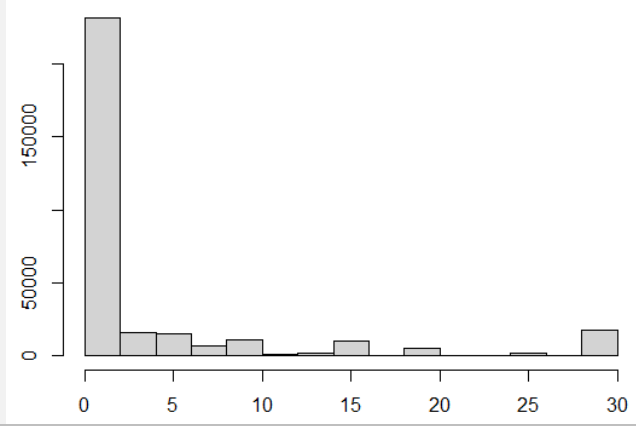
Según Centros para el Control y la Prevención de enfermedades (CDC), con este índice se pueden clasificar a las personas en 4 grupos según el nivel de peso:

- Bajo peso → < 18.5
- Normal → 18.5 – 24.9
- Sobrepeso → 25 – 29.9
- Obesidad → 30 o más.

Se puede ver que hay valores muy grandes, por lo que podría ser que fueran outliers o simplemente que en Estados Unidos se tienen muchos problemas de obesidad mórbida y, por tanto, algo más normal. (Rodrigo 2013)

MentalHealth

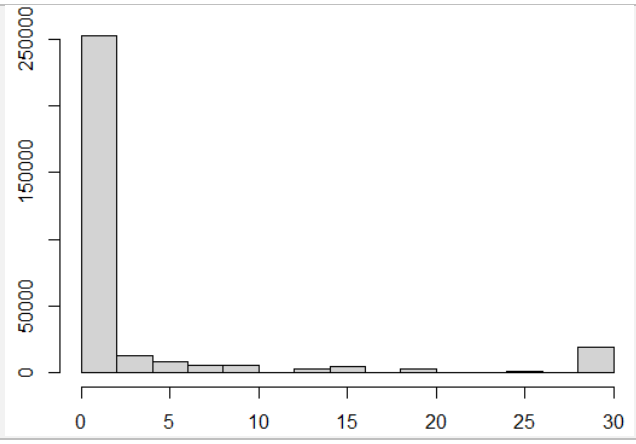
Tabla 18: Descriptivos, histograma. MentalHealth

DESCRIPTIVOS		HISTOGRAMA
Mínimo	**0**	
Máximo	**30**	
Mediana	**0**	
Media	**3.898**	

Esta variable cuenta el número de días que una persona no se encuentra mentalmente bien.
Se puede ver que hay personas que no se han encontrado bien durante cada uno de los últimos 30 días. No se tomarán como valores extremos, ya que es normal que haya gente que no se encuentre mentalmente mal.

PhysicalHealth

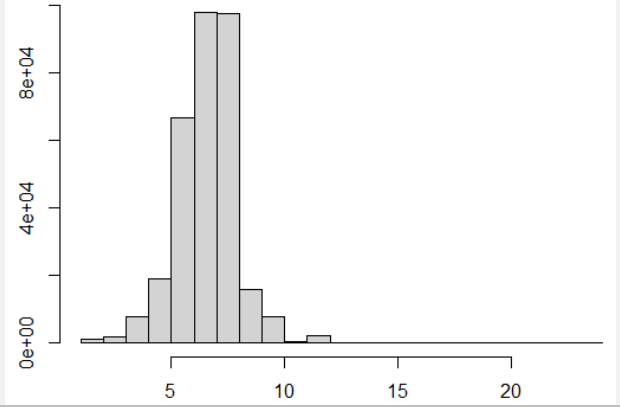
Tabla 19: Descriptivos, histograma. PhysicalHealth

DESCRIPTIVOS		HISTOGRAMA
Mínimo	**0**	
Máximo	**30**	
Mediana	**0**	
Media	**3.372**	

Esta variable cuenta el número de días que los individuos no se han encontrado físicamente bien. Como en el caso de la variable anterior, se ve que la mayor parte de ellos no han tenido problemas, mientras que hay otros que declaran haber tenido 30 días físicamente mal.
Estos últimos no deberían ser considerados como valores extremos al igual que pasa con la anterior variable.

SleepTime

Tabla 20: Descriptivos, histograma. SleepTime

DESCRIPTIVOS		HISTOGRAMA
Mínimo	**1**	
Máximo	**24**	
Mediana	**7**	
Media	**7.097**	

Esta variable hace referencia al número de horas de sueño que se duerme durante las 24 horas que tiene un día.
Se observa que la mayoría de la gente duerme 7-8 horas diarias, mientras que se observan casos extremos como dormir tan solo 1 o 2 horas como 24 horas, ya que es bastante improbable que se duerma todo el día.

2.2.2. ANÁLISIS BIVARIANTE

El análisis bivariante se utiliza para estudiar las relaciones entre dos variables a la vez. Es de gran utilidad para examinar si hay asociaciones entre variables.

La variable respuesta es HeartDisease que es dicotómica, por lo que se medirá la relación que existe entre esta variable y todas las demás.

Para medir la relación entre dos variables cualitativas se acudirá al test Chi – Cuadrado en el que se tienen las siguientes hipótesis:

$$\boldsymbol{H_0 = Independencia}$$
vs
$$\boldsymbol{H_1: Asociación}$$

El estadístico a utilizar será el χ^2. Este contraste se basa en que, bajo la hipótesis nula, el estadístico se distribuye según una Chi – Cuadrado con (h-1)*(k-1) grados de libertad, siendo h y k el número de categorías de cada una de las variables categóricas de las que está siendo estudiada su relación.

$$\chi^2 = \sum_{i=1}^{h} \sum_{j=1}^{k} \frac{\left(n'_{ij} - n_{ij}\right)^2}{n'_{ij}}$$

Para ello se utiliza una tabla de contingencia, que tendrá el siguiente aspecto:

Tabla 21: Tabla de contingencia

Variable Y \ Variable X	Categoría 1 (X)	Categoría 2 (X)
Categoría 1 (Y)	(1,1)	(1,2)
Categoría 2 (Y)	(2,1)	(2,2)

(1,1) → Frecuencia conjunta de la modalidad 1 de la variable X y de la modalidad 1 de la variable Y.

En términos estadísticos, cuando no existe ningún tipo de influencia de una variable en otra, se dice que son independientes. Si son independientes, la frecuencia relativa conjunta que se menciona en la tabla será igual al producto de las frecuencias marginales respectivas.

Resultados:

Tabla 22: Test Chi-Cuadrado

VARIABLES	χ^2	P – valor
HeartDisease – Smoking	3713.816	0.000
HeartDisease – AlcoholDrinking	329.104	< 0.001
HeartDisease – Stroke	12390.181	0.000
HeartDisease – DiffWalking	12953.233	0.000
HeartDisease – Sex	1568.808	0.000
HeartDisease – Race	844.315	< 0.001
HeartDisease – Diabetic	9769.372	0.000
HeartDisease – PhysicalActivity	3199.865	0.000
HeartDisease – GenHealth	21542.177	0.000
HeartDisease – Asthma	549.286	< 0.001
HeartDisease – KidneyDisease	6741.981	0.000
HeartDisease – SkinCancer	2784.788	0.000

Atendiendo a los resultados, al p-valor que es de 0 y que el estadístico Chi -Cuadrado es muy grande para todas las relaciones, se puede afirmar que para cualquier nivel de significación se rechaza la hipótesis nula de que las variables son independientes. Esto significa que todas las variables categóricas están asociadas con la variable HeartDisease, es decir, con haber sufrido un infarto de miocardio o enfermedad coronaria.

Una vez estudiada la relación entre la variable respuesta y las variables cualitativas, se procederá al estudio de la relación entre la variable respuesta y las cinco variables cuantitativas que se tienen en la base de datos.

Para ello se utiliza el contraste de comparación de medias de dos grupos independientes, mediante la t de student.

Para poder llevar a cabo este contraste es necesario que se cumplan las hipótesis de que los datos en cada uno de los dos grupos siguen una distribución normal y que tengan la misma varianza. (Domingo F. Rasilla)

Las hipótesis y el estadístico del contraste son los siguientes:

$$H_0: \mu_A = \mu_B \quad \text{vs} \quad H_1: \mu_A \neq \mu_B$$

$$t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \sim t_{n_1+n_2-2}$$

En el caso de que la hipótesis de normalidad no se cumpla, habría que recurrir a la prueba de Wilcoxon, que es un contraste no paramétrico.

En primer lugar, se ejecuta el contraste de normalidad de cada variable cuantitativa en cada uno de los dos grupos de la variable categórica dependiente:

Tabla 23: Pruebas de normalidad

Pruebas de normalidad				
	HeartDisease	Kolmogorov-Smirnov[a]		
		Estadístico	gl	Sig.
BMI	No	,085	292422	,000
	Yes	,078	27373	,000
PhysicalHealth	No	,381	292422	,000
	Yes	,285	27373	,000
MentalHealth	No	,328	292422	,000
	Yes	,359	27373	,000
Age	No	,092	292422	,000
	Yes	,119	27373	,000
SleepTime	No	,177	292422	,000
	Yes	,185	27373	,000
a. Corrección de significación de Lilliefors				

Atendiendo al resultado de la *Tabla 23*, se rechaza la hipótesis nula de que los datos en los dos grupos siguen una distribución normal. Por tanto, se recurra al contraste no paramétrico de Wilcoxon para dos muestras independientes.

Se hará este contraste para cada una de las variables cuantitativas con la variable objetivo, y se acompañará el resultado de un diagrama de caja y bigotes para saber cómo es la distribución de las personas que declaran haber tenido algún problema cardíaco y las que no.

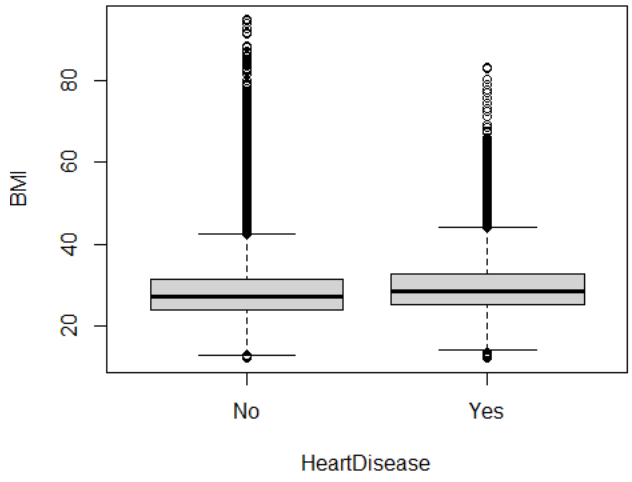
Tabla 24: Test Wilcoxon 2 muestras independientes

VARIBLES	ESTADÍSTICO W	P – valor
HeartDisease – BMI	3528521417	< 2.2e-16
HeartDisease – PhysicalHealth	3054079283	< 2.2e-16
HeartDisease – MentalHealth	4026615485	0.05143
HeartDisease – Age	2032559629	< 2.2e-16
HeartDisease – SleepTime	3942216716	2.11e-05

Para todas las combinaciones de variables cuantitativas con la variable respuesta, la hipótesis nula de que las medias en ambos grupos son iguales, queda rechazada, lo que quiere decir que las variables están relacionadas ya que sus medias son diferentes.

HeartDisease – BMI

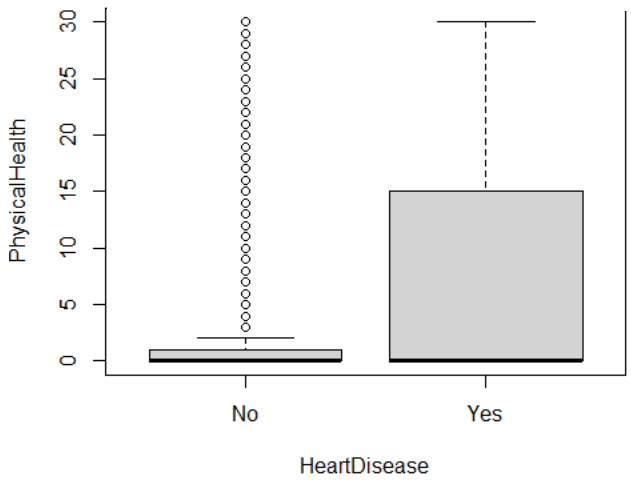
Ilustración 1: Boxplot. HeartDisease - BMI



El índice medio de masa corporal de una persona que declara haber tenido una enfermedad cardíaca es diferente con respecto a aquellas personas que no han tenido. Las personas que lo han tenido tienen un índice de masa corporal mayor.

HeartDisease – PhysicalHealth

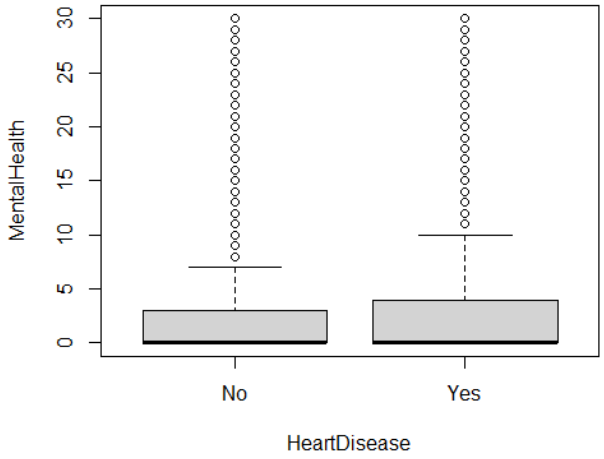
Ilustración 2: Boxplot. HeartDisease - PhysicalActivity



El número medio de días que una persona no se encuentra físicamente bien es claramente diferente en cada uno de los dos grupos.
Esta diferencia es grande entre ambos grupos. Las personas que declaran haber tenido más días físicamente mal, son aquellas que han sufrido alguna vez una enfermedad cardíaca.

HeartDisease – MentalHealth

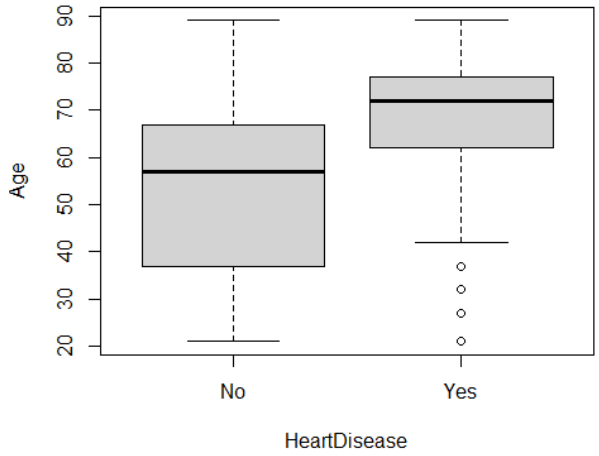
Ilustración 3: Boxplot. HeartDisease - MentalHealth



De manera similar a la variable anterior, el número medio de días que una persona no se ha encontrado mentalmente sana es diferente en los dos grupos.
En este caso la diferencia es menor en cuanto a si se sufre una enfermedad cardíaca o no, pero es ligeramente superior el sí sufrir una enfermedad cardíaca en cuanto a número de días mentalmente mal.

HeartDisease – Age

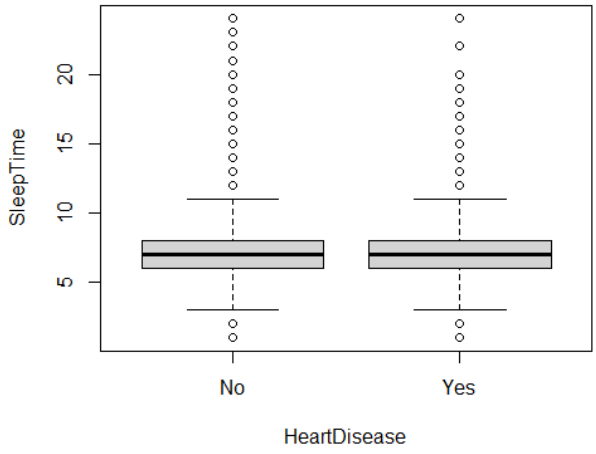
Ilustración 4: Boxplot. HeartDisease - Age



La edad media de las personas que han sufrido alguna enfermedad cardíaca es significativamente distinta a la de las personas que no la han sufrido.
Aquellas personas con más edad son las que más sufren enfermedades cardíacas, mientras que las que no lo sufren tienen edades más bajas.

HeartDisease – SleepTime

Ilustración 5:Boxplot. HeartDisease - SleepTime



A pesar de que visualmente parece no haber muchas diferencias, atendiendo al p – valor, el número medio de horas de sueño es distinto para cada uno de los grupos.
Esta discrepancia entre la información visual y el p – valor se debe al alto número de observaciones, 319.795, que hace que la varianza de la media muestral sea muy pequeña y, por tanto, tienda a rechazar la hipótesis de igualdad de medias.

2.2.3. ANÁLISIS MULTIVARIANTE

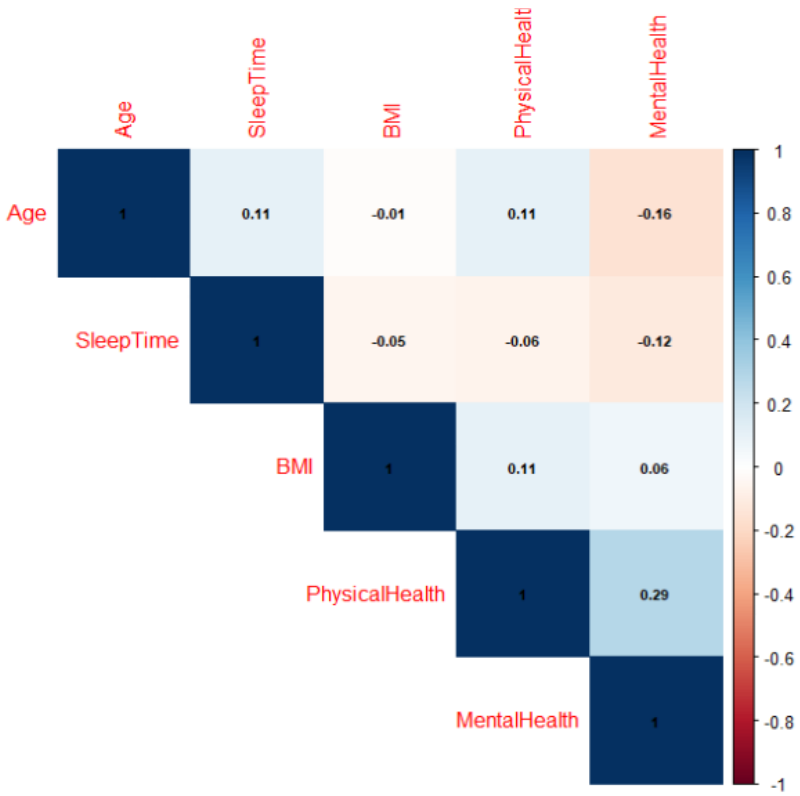
El análisis multivariante es una técnica empleada para entender relaciones complejas que se producen entre varias variables a la vez. Su intención es revelar patrones o asociaciones que podrían pasar desapercibidos al analizar las variables de forma individual.

Se analizarán las relaciones que puedan existir entre todas las variables cuantitativas mediante una matriz de correlaciones.

La matriz de correlaciones muestra los valores de correlación de Pearson, que miden el grado de correlación lineal para cada par de variables. Puede tomar valores entre -1 y +1, de forma que un valor cercano al +1, indica una correlación positiva, es decir, las dos variables tenderían a ascender o descender a la misma vez.

Se muestra la matriz de correlaciones:

Ilustración 6: Matriz de correlaciones.



Los resultados muestran correlaciones generalmente bajas entre las variables. La correlación más alta es entre la PhysicalHealth y MentalHealth con un valor de 0.29, sugiriendo que cuantos más días pase una persona físicamente mal, más días va a estar una persona mentalmente mal y viceversa. La relación entre Age y MentalHealth es moderadamente negativa indicando que, con la edad, el número de días que una persona está mentalmente mal puede aumentar. Las otras correlaciones son débiles, indicando relaciones poco significativas.

3. METODOLOGÍA

3.1. METODOLOGÍA SEMMA

En este trabajo se va a utilizar la metodología SEMMA que se define como el proceso de selección, exploración y modelado de conjuntos de datos. Esta metodología se compone de 5 fases:

► **MUESTREO (SAMPLE)**: En esta fase se selecciona una muestra representativa de los datos que contenga información relevante pero lo suficientemente pequeña como para ser manejada eficazmente.

► **EXPLORAR (EXPLORE):** Se realiza un análisis exploratorio de los datos para obtener información relevante, detectar valores atípicos, patrones y entender las relaciones que haya entre variables.

► **MODIFICAR (MODIFY):** En esta etapa, los datos se transforman y ajustan para mejorar la calidad del modelo. Esto incluye la transformación de los datos a través de la estandarización de variables cuantitativas, la creación de nuevas variables y la transformación de variables existentes.

► **MODELIZAR (MODEL):** Se desarrollan y entrenan modelos predictivos utilizando diversas técnicas estadísticas y de machine Learning. El objetivo principal de esta fase es encontrar el modelo que mejor se ajuste a los datos y sea capaz de hacer predicciones precisas.

► **EVALUAR (ASSES):** Se trata de la última etapa de la metodología SEMMA. En ella se evalúan los modelos construidos utilizando diferentes métricas como la sensibilidad, especificidad, tasa de acierto o AUC. A través de estas se compara el desempeño de diferentes modelos y se selecciona el mejor.

Con respecto a la **fase de muestreo**, se tiene que la variable objetivo está desbalanceada, con un 9% de observaciones en la categoría “Yes”. En primera instancia, se estudió la posibilidad de utilizar técnicas de oversampling, pero debido a la gran cantidad de datos existente (N = 319795), aumentar las observaciones de la clase minoritaria para igualarlo con la mayoritaria provocaba grandes problemas computacionales.

Por tanto, se opta por utilizar técnicas de undersampling que consiste en reducir el número de observaciones de la clase mayoritaria para igualarla con la clase minoritaria. Con esto se tendrían 27.373 observaciones de cada una de las categorías. Sin embargo, computacionalmente hablando no es posible procesar esta cantidad de datos, por lo que se sortean de cada una de las categorías 7500 observaciones.

Para llevar a cabo este balanceo se tienen unos pesos de balanceo:

- Para la categoría “Yes” se tienen 27373 observaciones y se sortean 7500, por tanto, el peso es 0.274.
- Para la categoría “No” se tienen 292422 observaciones y se sortean 7500, por tanto, el peso es 0.02565.

Obviamente para que los datos finales resultantes tengan la distribución original de la población que es sobre la que han de tomarse conclusiones y decisiones, se deben volver a utilizar los pesos descritos anteriormente para recuperar la situación original de la población.

Las medidas sobre las que se van a evaluar y comparar los diferentes algoritmos de Machine Learning serán la tasa de fallos y el área bajo la curva (AUC). Sin embargo, no se utilizará una tasa de fallos normal, sino que se utilizará una tasa de fallos ponderada, ya que se le dará más error al hecho de decirle a un paciente que no va a sufrir una enfermedad cardíaca cuando en realidad sí que la va a sufrir. Por tanto, se le dará una ponderación de 3 veces más grave que el otro error.

A continuación, se muestra un ejemplo de lo que se ha contado en este apartado para que sea más representativo y visual.

Póngase que se tiene una población de 500 individuos, de los cuales 120 pertenecen a la categoría “Yes” y 380 a la categoría “No”. Se toma una muestra de 120 individuos, 60 pertenecen a “Yes” y 60 pertenecen a “No”. Los pesos en este caso serían:

- Para “yes”: 60/120 = 0.5 (p1 = 0.5)
- Para “no”: 60/380 = 0.158 (p0 = 0.158)

Para la muestra de 120 individuos se obtiene la matriz de confusión siguiente:

Tabla 25: Ejemplo matriz de confusión. Undersampling balanceado

	0 (NO)	**1 (SI)**	**TOTAL**
0(NO)	65	12	77
1 (SI)	9	34	43
TOTAL	74	46	120

El cálculo de la tasa de fallos ponderada, como se ha comentado anteriormente, ha de hacerse sobre la población original, por lo que para llevar a cabo esta transformación habría que hacer lo siguiente:

Tabla 26: Ejemplo matriz de confusión. Transformación deshecha.

	0 (NO)	1 (SI)	TOTAL
0(NO)	$\frac{65}{N * p0}$	$\frac{12}{N * p1}$	X
1 (SI)	$\frac{9}{N * p0}$	$\frac{34}{N * p1}$	Y
TOTAL	Z	K	1

Donde N = 500 y p0 y p1 los pesos.
Por último, la tasa de fallos ponderada se calcularía de la siguiente forma:

$$Tasafallos_{pond} = 3 * \left(\frac{12}{N * p1}\right) + (\frac{9}{N * p0})$$

Con respecto a la **fase de modificar** se hará un estudio de los valores missing, se estandarizarán las variables cuantitativas y se crearán variables dummies.

En primer lugar, se comprueba que en ninguna variable se encuentran valores perdidos, por lo que con respecto a esto no hay que hacer ninguna modificación.

En segundo lugar, se estandarizan las variables cuantitativas. De esta forma se consigue una mejora del rendimiento del modelo, se consiguen modelos menos sesgados y más precisos, facilita la convergencia en modelos de redes neuronales y ayuda a que se mantengan los valores en un rango similar, evitando problemas de estabilidad numérica en los algoritmos de optimización.

Por último, se procede a la creación de variables dummy. Se tienen 12 variables categóricas independientes, por lo que se construye una variable dummy por cada una de las categorías. Para las variables con dos categorías, se crean dos variables dummy, pero como con una se tendría información de la otra, se elimina una categoría de esas variables. Por otra parte, para las variables con tres o más categorías se crean tres o más dummies y no se elimina ninguna, aunque con una menos se habría tenido información implícita de la que se elimina.

De esta forma se pasa de tener 12 variables categóricas a tener 21 variables dummies.
Estas 21 variables sumadas a las 5 variables cuantitativas y a la variable objetivo, hacen un total de 27 variables. Son las siguientes: : BMI, PhysicalHealth, MentalHealth, Age, SleepTime, Smoking.Yes, AlcoholDrinking.Yes, Stroke.Yes, DiffWalking.Yes, Sex.Female, Race.Indian, Race.Asian, Race.Black, Race.Hispanic, Race.Other, Race.White, Diabetic.Yes, PhysicalActivity.Yes, GenHealth.Excellent, GenHealth.Fair, Genhealth.Good, GenHealth.Poor, GenHealth.VeryGood, Asthma.Yes, KidneyDisease.Yes, SkincCancer.Yes, HeartDisease.

3.2. REGRESIÓN LOGÍSTICA

El objetivo de los modelos de regresión es tratar de predecir una variable Y en función de una serie de variables independientes X.
A diferencia de los modelos de regresión lineales, en la logística binaria se tiene la particularidad de que la variable respuesta Y es dicotómica y suele tomar valores 0 y 1.

En regresión logística, como función de enlace, se utiliza la función de distribución logística principalmente porque sus parámetros pueden ser interpretados a partir de los odds – ratio.

$$\boldsymbol{P(Y = 1|X_1, X_2, \dots, X_n) = \frac{e^{(\beta_0+\beta_1+\beta_2X_2+\cdots+\beta_nX_n)}}{1 + e^{(\beta_0+\beta_1+\beta_2X_2+\cdots+\beta_nX_n)}}}$$

Dentro de la regresión logística, uno de los pasos más importantes es la selección de variables, de forma que permitirá saber qué variables son las más relevantes a la hora de dar estimaciones acerca de la variable de interés.

En primer lugar, mejora la interpretabilidad del modelo y disminuye el ruido, aumentando la precisión. También previene el sobreajuste, evitando que el modelo se ajuste demasiado a los datos de entrenamiento y mejorando su rendimiento en datos nuevos. Además, aumenta la eficiencia computacional. Por último, ayuda al modelo a generalizar mejor a nuevos datos y facilita su adaptación a cualquier tipo de cambio. Por otra parte, evita los problemas de multicolinealidad.

La selección de variables bajo regresión logística se puede hacer con diferentes métodos:

- ***SBF***: Selecciona variables filtrando aquellas que cumplen ciertos criterios estadísticos, como la correlación con la variable objetivo. Es una técnica eficiente para reducir la dimensionalidad, mejorar la interpretabilidad y el rendimiento del modelo.

- ***AIC***: Evalúa modelos basándose en su ajuste y complejidad. Selecciona variables minimizando el AIC, que equilibra la calidad del modelo y el número de parámetros, favoreciendo modelos simples y efectivos.

- ***BIC***: Es un método similar al anterior, pero en este caso minimiza el BIC. Es un criterio que penaliza más que el AIC.

- ***Stepwise Repetido AIC***: Iterativamente, ajusta el modelo añadiendo o eliminando variables para minimizar el AIC y mejorar el rendimiento del modelo.

- ***Stepwise Repetido BIC***: Igual que el anterior, pero en este caso con el criterio BIC.

- ***RFE:*** Elimina recursivamente las variables menos importantes basándose en la importancia asignada por un modelo, como una regresión o un árbol. Se Obtiene así un modelo óptimo en cuanto a variables elegidas.

- ***MMPC:*** Identifica las variables relevantes utilizando pruebas de independencia con respecto a la variable objetivo. Selecciona aquellas que tienen una relación directa con esa variable.

- ***SES:*** Es un método iterativo que elimina y selecciona variables en cada iteración según su rendimiento en el modelo. Se repite hasta encontrar el conjunto óptimo que maximiza la precisión del modelo.

3.3. REDES NEURONALES

Las redes neuronales son modelos computacionales inspirados en el funcionamiento del cerebro humano. Estas redes se utilizan en gran variedad de ocasiones, incluyendo la clasificación de datos en diferentes categorías, la predicción de valores numéricos, o el reconocimiento de patrones complejos en datos no estructurados.

En este trabajo se utilizarán redes neuronales para clasificación binaria, al tener la variable respuesta dicotómica. Por tanto, se utilizará para predecir la probabilidad de que ocurra un evento o para clasificar las observaciones en una de las dos categorías.

Las redes neuronales son adecuadas para datos complejos y no lineales ya que capturan este tipo de relaciones de manera muy eficaz, pueden aprender y modelar relaciones complejas, son robustas, es decir, manejan grandes cantidades de datos de manera efectiva.

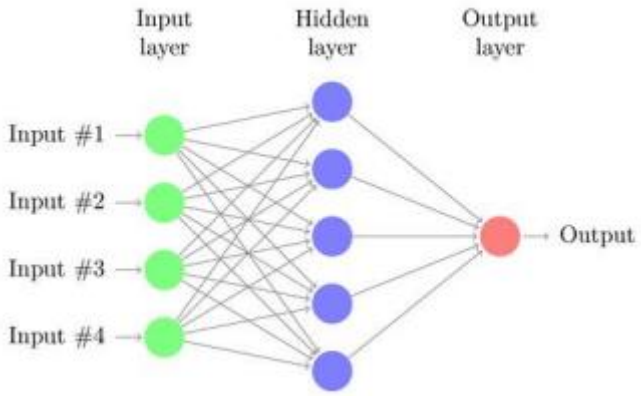
Al utilizar redes neuronales, es importante controlar diferentes hiperparámetros para optimizar el rendimiento del modelo. Uno de los hiperparámetros más importantes es el número de nodos en la capa oculta, esta capa permite que la red aprenda características más complejas de los datos. El número de nodos en esta capa influye a la hora de identificar patrones en los datos, pero también aumenta el riesgo de sobreajuste si se elige un gran número de estos.

El sobreajuste ocurre muy a menudo en la modelización de redes neuronales y ocurre cuando el modelo se ajusta muy bien a los datos de entrenamiento, pero no a los datos de prueba. Para evitar este problema se pueden recurrir a soluciones como la validación cruzada repetida.

Otro factor importante para tener en cuenta es el número de observaciones en los datos. Si se tiene un gran volumen de datos es posible que la red aprenda patrones más robustos y generalice mejor a datos nuevos. Pero también tiene los inconvenientes de que aumenta el tiempo de entrenamiento y la complejidad. (Bishop 2006)

Tal y como se puede ver en la *Figura 7,* las redes neuronales tienen una capa de entrada formada por las variables independientes del modelo, una o varias capas ocultas con un número de nodos ocultos a determinar, y una capa de salida en la que se encuentra la variable respuesta.

Ilustración 7: Red neuronal



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

A continuación, se aplica a cada nodo oculta la función de activación, que suele ser la tangente hiperbólica y, por último, se aplica una función de activación de la capa oculta a la capa output, de modo que se queda una expresión como la siguiente:

$$\begin{aligned} \boldsymbol{Y} = {} & \boldsymbol{w_{1,out}}(\boldsymbol{tanh}(\boldsymbol{w_{11}X1} + \boldsymbol{w_{21}X2} + \boldsymbol{w_{31}X3} + \boldsymbol{w_{41}X4} + \boldsymbol{b_1})) \\ & + \boldsymbol{w_{2,out}}(\boldsymbol{tanh}(\boldsymbol{w_{12}X1} + \boldsymbol{w_{22}X2} + \boldsymbol{w_{32}X3} + \boldsymbol{w_{42}X4} + \boldsymbol{b_2})) \\ & + \boldsymbol{w_{3,out}}(\boldsymbol{tanh}(\boldsymbol{w_{13}X1} + \boldsymbol{w_{23}X2} + \boldsymbol{w_{33}X3} + \boldsymbol{w_{43}X4} + \boldsymbol{b_3})) \\ & + \boldsymbol{w_{4,out}}(\boldsymbol{tanh}(\boldsymbol{w_{14}X1} + \boldsymbol{w_{24}X2} + \boldsymbol{w_{34}X3} + \boldsymbol{w_{44}X4} + \boldsymbol{b_4})) + \boldsymbol{b_{out}} \end{aligned}$$

El número de parámetros de una red neuronal viene dado por:

$$\boldsymbol{h(k+1)+h+1}$$

Donde h hace referencia al número de nodos ocultos y k al número de variables independientes.

3.4. ÁRBOL DE CLASIFICACIÓN

Los árboles de clasificación son de gran utilidad cuando la variable dependiente es cualitativa, como es el caso de la variable en estudio.

Consiste en una segmentación de los datos siguiendo unas reglas que se aplican de forma secuencial, de forma que se van obteniendo nodos a partir de otros. Una vez finalizada esta segmentación en forma de nodos, a los nodos que ya no se segmentan más, se asigna un valor de predicción.

Los árboles tienen grandes ventajas como el tratamiento automático de valores missings, el tratamiento automático de categorías poco representadas, detección automática de regiones y puntos de corte, resultados a menudo fáciles de comprender o adaptabilidad a la forma funcional entre la variable objetivo y predictoras.

Así como estas ventajas, se tienen desventajas como que tienen poca capacidad predictiva y gran varianza y son muy sensibles a cambios en los datos, tienden a la inestabilidad y a la falta de robustez.

3.5. BAGGING

El algoritmo bagging (Boostrap Averaging) fue introducido por Leo Breiman en su artículo “Bagging Predictors” publicado en 1996. (Breiman 1996)

Breiman propuso esta técnica como una forma de mejorar la precisión de los modelos de predicción en el contexto de árboles de decisión. El método Bagging consiste en entrenar múltiples modelos de manera independiente utilizando conjuntos de datos generados mediante muestreo con reemplazamiento, y luego promediar las predicciones de estos modelos para obtener una predicción final.

Al utilizarse diferentes submuestras, se reduce la dependencia de la estructura de los datos para construir el modelo y, por tanto, se reduce la varianza, lo que resulta en predicciones más estables y precisas.

Entre los principales parámetros a controlar en Bagging:

- Tamaño de las muestras (n) y si se va a utilizar Boostrap o no.
- Número de iteraciones (m) a promediar.
- Número de hojas final.
- Número de observaciones en cada nodo. Se puede ampliar para evitar el sobreajuste y reducir la varianza o reducir para ajustar mejor y se reduce el sesgo.

3.6. RANDOM FOREST

El algoritmo de Random Forest es una modificación del Bagging en el que además de sortear las muestras, incorpora la aleatoriedad en las variables utilizadas para segmentar cada nodo del árbol.

Este algoritmo trata de solventar el problema de selección de variables, evitando utilizar siempre un único set de variables y aprovecha a la vez las ventajas del Bagging, por lo que se puede decir que el Bagging es un caso particular de Random Forest.

De esta forma se busca la reducción del sobreajuste a la vez que se pretende que se ajusten bien las relaciones particulares en los datos, mediante el remuestreo de observaciones y de variables. A medida que la varianza disminuye, el sesgo tiende a aumentar un poco pero este algoritmo controla muy bien este aspecto.

Principales parámetros a controlar:

- Tamaño (n) o % de las muestras y si se utiliza Boostrap o no.
- Número de iteraciones (m) a promediar.
- Número de variables (p) a muestrear en cada nodo.
- Número de hojas final.
- Número mínimo de observaciones en cada nodo.

3.7. GRADIENT BOOSTING

Gradient Boosting fue el tercer algoritmo planteado a partir de árboles. Fue desarrollado y formalizado por Jerome H. Friedman en su artículo “Greedy Function Approximation: A Gradient Boosting Machine” (Friedman 2001) . Se buscaba reducir la varianza y sesgo, aplicar más suavidad y solventar el problema de la falta de robustez.

Es una técnica de Machine Learning que consiste en construir una serie de modelos de manera secuencial de forma que en cada uno de ellos va corrigiendo los errores cometidos. De esta forma se van modificando ligeramente las predicciones iniciales y se va minimizando el error. Las predicciones se ajustan cada vez más a los datos por lo que constituye una mejora importante con respecto a la construcción de un solo árbol.

La principal diferencia entre esta técnica y Random Forest es que Gradient Boosting reduce el sesgo y Random Forest reduce varianza.

Por tanto, es un algoritmo en el que hay que minimizar una función de error y no tiene sentido evaluarlo mediante datos train ya que su error va a ser cero, por lo que con los datos test es sobre los que hay que evaluar las predicciones.

Principales parámetros del algoritmo:

- Número de árboles (m) a promediar.
- Constante de regularización (shrinkage)
- Número de hojas final.
- Número mínimo de observaciones en cada nodo.

3.8. XGBOOST

Xgboost es un algoritmo basado en boosting, construyendo un modelo predictivo a partir de árboles de decisión. Destaca por su eficiencia y escalabilidad y entre sus características destacan el manejo de datos faltantes, regularización para evitar el sobreajuste y una estructura de árbol optimizada que mejora la velocidad y rendimiento.

A diferencia del gradient boosting, Xgboost introduce mejoras como la regularización mediante parámetros L1 y L2 para prevenir el sobreajuste. Utiliza una técnica de aproximación para buscar mejores puntos de división en los árboles, reduciendo el tiempo de entrenamiento.

Las ventajas del Xgboost incluyen su capacidad para manejar grandes volúmenes de datos y su eficiencia en el uso de recursos computacionales. La regularización y el ajuste de hiperparámetros permiten alta precisión sin incurrir en sobreajuste.

Sin embargo, Xgboost tiene desventajas como su complejidad. El ajuste de los hiperparámetros puede ser costoso y se requiere mucha potencia de procesamiento.

Los hiperparámetros a controlar en Xgboost son los mismos que en gradient boosting, pero en este caso se puede hacer una selección de un porcentaje de observaciones y de variables.

3.9. SVM LINEAL

El Support Vector Machine lineal consiste en encontrar un hiperplano que separe los datos de dos clases de manera que la separación sea lo más amplia posible. El hiperplano es una frontera que separa las muestras de diferentes clases y hay que buscar el óptimo.

3.10. SVM POLINOMIAL

El Support Vector Machine Polinomial es similar al SVM Lineal en términos de optimización con la diferencia de que se busca un hiperplano en el espacio transformado por el kernel polinomial que maximice el margen entre las clases.

3.11. SVM RADIAL

El Support Vector Machine Radial (RBF) es una herramienta que maneja problemas de clasificación no lineales y busca un hiperplano transformado por el Kernel RBF que maximice el margen entre clases.

3.12. MÉTODOS DE ENSAMBLADO

Los métodos de ensamblado consisten en la construcción de predicciones a partir de la combinación de varios modelos. Con esto se busca mejorar le precisión, mejoran la varianza del error (no suelen empeorar los modelos) y la robustez de las predicciones.

El propósito de estos métodos es alcanzar un rendimiento superior al que se obtendría con un modelo simple.

Por otra parte, los métodos de ensamblado tienen algunas desventajas como que se aumenta la complejidad de los modelos, se puede llegar muchas veces al sobreajuste y los modelos no son interpretables.

Hay varias formas de llevar a cabo ensamblado de modelos. Una de ellas es promediando las predicciones de varios modelos, lo que suele reducir el error de predicción. Otra forma es el bagging, que implica entrenar muchas veces el mismo modelo con diferentes subconjuntos de datos generados por muestreo con reemplazamiento. Y otra forma puede ser el boosting de manera que se va reduciendo secuencialmente el error de predicción.

4. APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING

En este apartado se procederá a la construcción de los distintos modelos de Machine Learning. Para ello es importante ajustar los hiperparámetros disponibles en cada uno de ellos para obtener resultados óptimos. Posteriormente, se comparará el rendimiento de cada uno de los modelos generados con cada técnica mediante métricas de evaluación.

En cuanto al tuneo de los hiperparámetros, se probarán con diferentes combinaciones para ver con cuál de ellas se alcanza unos buenos resultados. Aquella combinación que alcance una tasa de aciertos más alta será considerada como óptima.

Por otro lado, se aplicará validación cruzada repetida con cuatro grupos y diez repeticiones en cada uno de los modelos. Con la validación cruzada repetida se divide el conjunto de datos en dos subconjuntos: entrenamiento y prueba. De esta forma con el conjunto de datos de entrenamiento se estiman los parámetros y con el conjunto de datos de prueba se evalúa el modelo. Una de las ventajas es que todas las observaciones se predicen una vez sin ser incluidas en la creación del modelo, pero aun así contribuyen al modelo en las iteraciones posteriores. Para evitar los efectos de la aleatoriedad, se repite este proceso con diferentes semillas.

A partir de la validación cruzada repetida, se construirán unos gráficos de cajas y bigotes en cuanto a AUC y tasa de fallos ponderada, a partir de los cuales se tomará la decisión acerca de cuál es el modelo óptimo de cada uno de los algoritmos.

4.1. REGRESIÓN LOGÍSTICA

A partir de cada set de variables seleccionado con cada uno de los métodos mencionados en la parte de metodología, se construyen modelos de regresión logística. A continuación, se muestran los boxplots en cuanto a AUC y tasa de fallos ponderada y se muestra en un cuadrado el número de variables de cada uno de los modelos.

Ilustración 8: AUC. Regresión Logística. Selección de variables.

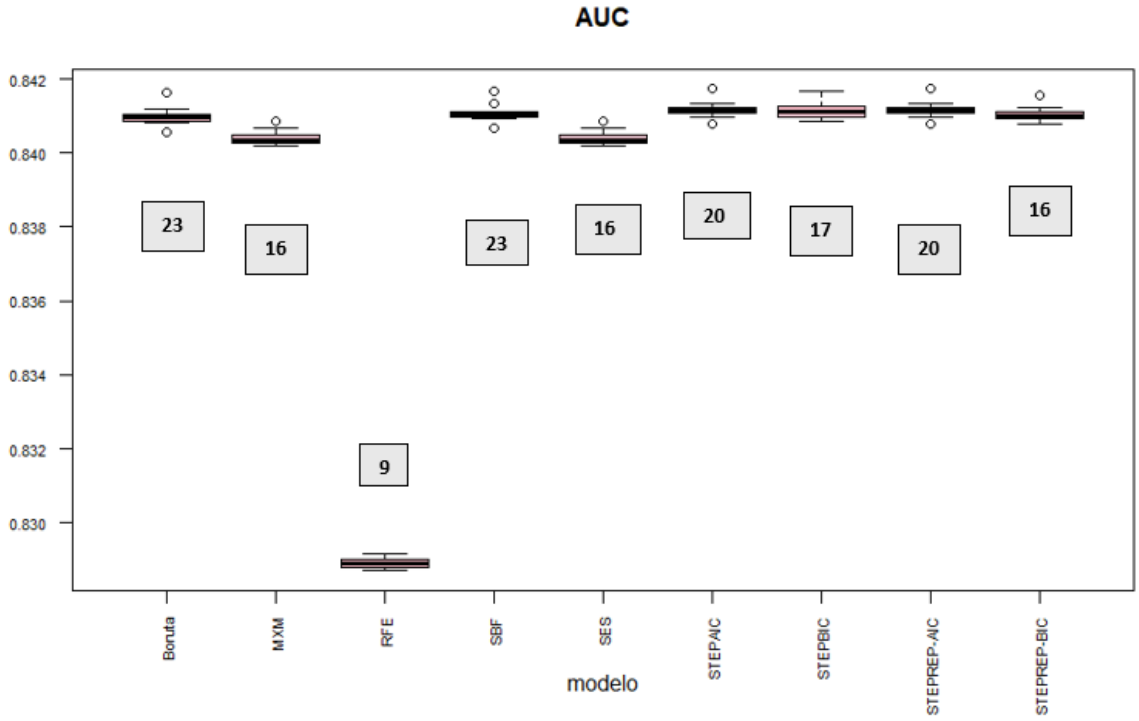
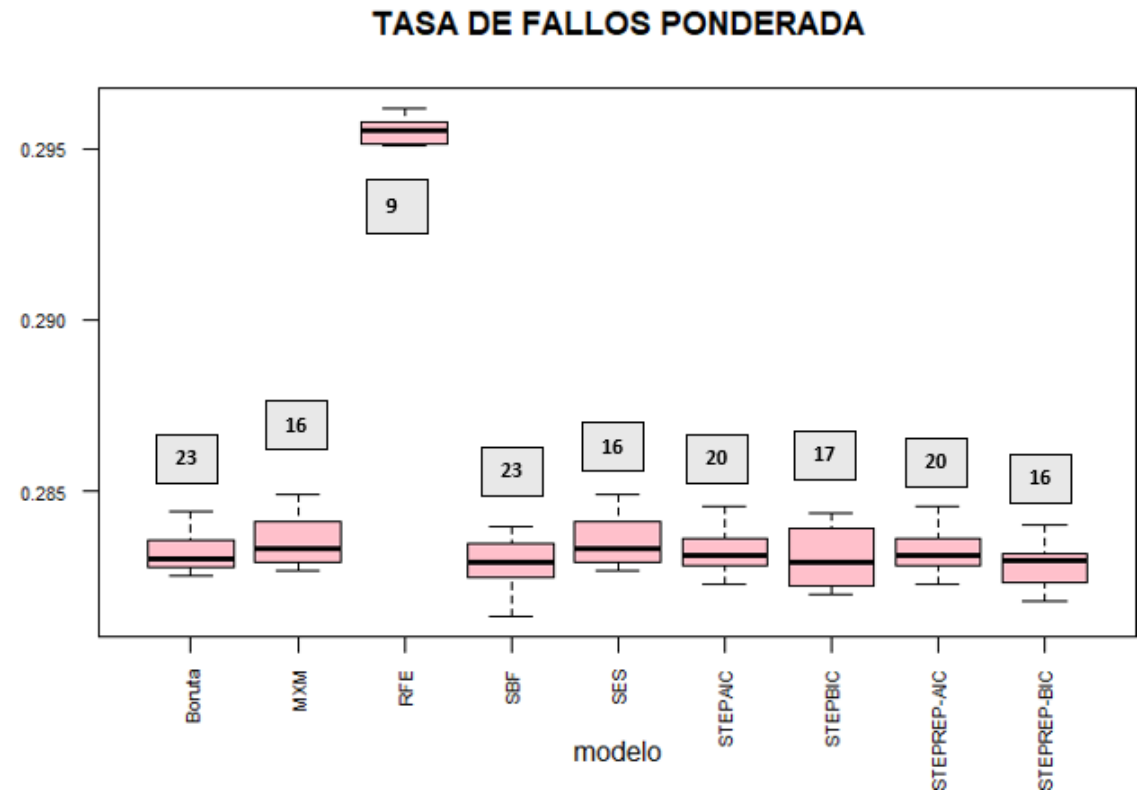


Ilustración 9: Tasa de fallos ponderada. Regresión Logística. Selección de variables.



Inicialmente los métodos de Boruta y SBF quedan descartados debido a que seleccionan una gran cantidad de variables y no mejoran demasiado a los otros métodos en AUC ni en tasa de fallos ponderada. También se descarta RFE ya que es el que mayor tasa de fallos ponderada y menor AUC presenta.

De entre el resto de métodos se selecciona el STEPREP-BIC que es un stepwise repetido bajo el criterio BIC que selecciona 16 variables y tiene un AUC alto y es el que menos tasa de fallos ponderada presenta.

4.2. REDES NEURONALES

Para la correcta modelización de una red neuronal es necesario el tuneo de los siguientes hiperparámetros:

- **Size**: número de nodos ocultos en la capa oculta.
- **Decay**: Hiperparámetro que reduce la magnitud de los pesos o disminuye la tasa de aprendizaje durante el entrenamiento para mejorar el rendimiento y evitar el sobreajuste.
- **Maxit**: Número máximo de iteraciones durante el entrenamiento, estableciendo un límite para evitar entrenamientos excesivamente largos.

A la hora de tunear el número de nodos ocultos es recomendable tener en cuenta un aspecto. Lo habitual es fijar unas 25 – 30 observaciones por parámetro para hacer un buen ajuste de la red y no incurrir en sobreajuste. Por tanto, si se fija 30 observaciones por parámetro, se tendría el siguiente número de nodos en la capa oculta de forma orientativa:

$$h(k+1)+h+1=\frac{7500(obs\ clase\ de\ interés)}{30\ (obs*parametro)}$$

$$h(16+1)+h+1=250\rightarrow 18h=249\rightarrow \boldsymbol{h=13.83}$$

Donde k es el número de variables independientes y h el número de nodos ocultos.
Esto indica que como máximo el número de nodos en la capa oculta estará entre 13 – 14.

Tras probar valores entre 8 y 14 nodos en la capa oculta, se llegó a la conclusión de que era necesario probar con valores más pequeños, ya que cuanto más pequeño era el número de nodos más aumentaba la tasa de aciertos.

Por tanto, se prueba con los siguientes valores de hiperparámetros:

- Size = 1, 3, 4, 6 y 8
- Decay = 0.001, 0.01 y 0.1.
- Maxit = 10, 20, 50, 100, 200, 300, 500, 1000, 2000, 3000, 4000, 5000 y 6000

Se muestra este tuneo en la *Figura 10*:

Ilustración 10: Tuneo hiperparámetros. Red Neuronal.

A la vista del gráfico se extraen cuatro posibles modelos candidatos a ser el óptimo de red neuronal. Estos modelos son los siguientes:

Tabla 27: Modelos de Redes Neuronales

MODELO	SIZE	DECAY	MAXIT
Red6 0.01 20	6	0.01	20
Red3 0.001 90	3	0.001	90
Red4 0.01 300	4	0.01	300
Red3 0.01 5000	3	0.01	5000

Estos modelos se comparan gráficamente a través de boxplots del AUC y de la tasa de fallos ponderada tras la aplicación de validación cruzada repetida:

Ilustración 11: AUC. Red neuronal.

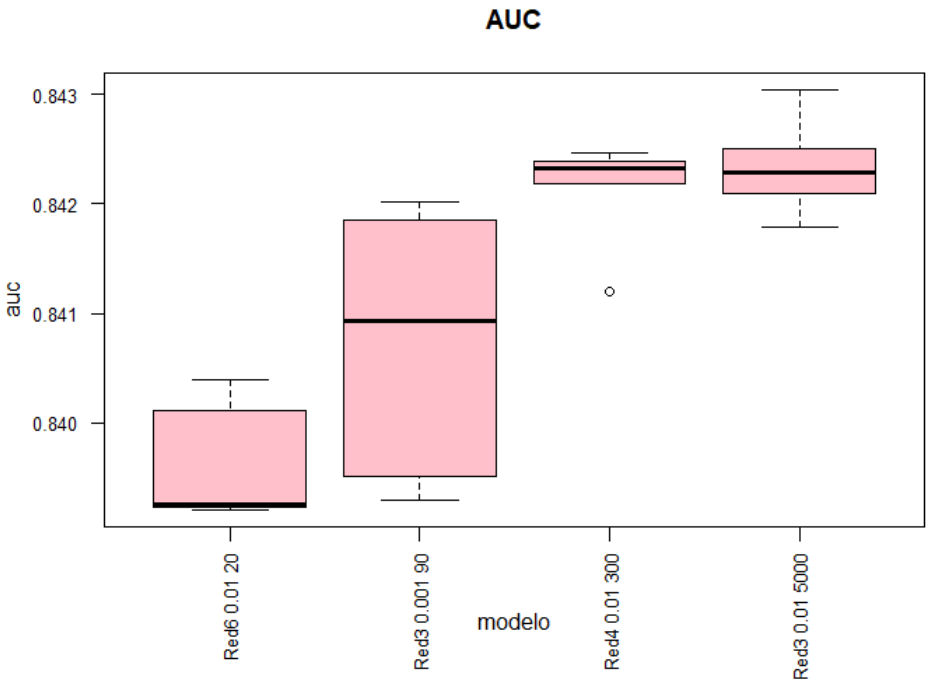
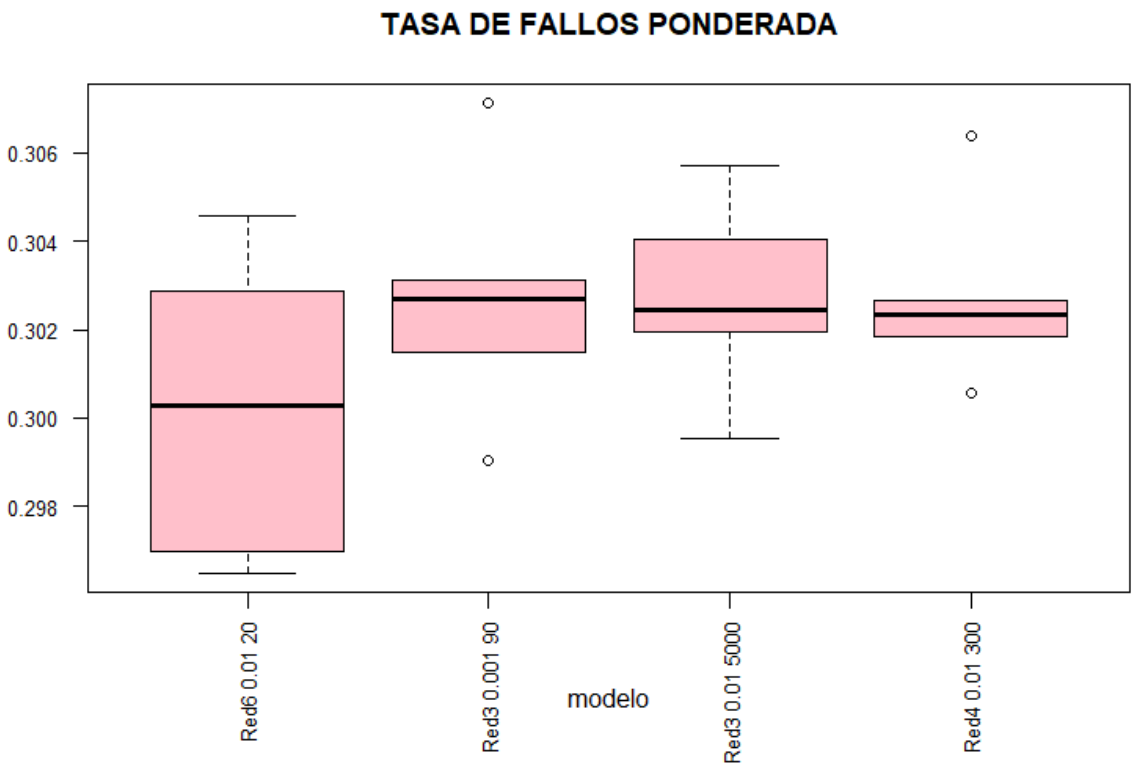


Ilustración 12: Tasa de fallos ponderada. Red neuronal.



Atendiendo a ambos gráficos, aunque el de 6 nodos ocultos con un decay de 0.01 y 20 iteraciones es el que menor tasa de fallos ponderada presenta, es el que menor AUC tiene, por lo que queda descartado. En este caso se busca un modelo intermedio que funcione bien en términos de AUC y de tasa de fallos ponderada. Estas características las reúne el modelo de red con 4 nodos en la capa oculta, un decay de 0.01 y 300 iteraciones, por lo que será el modelo elegido.

4.3. ÁRBOL DE CLASIFICACIÓN

En este trabajo se va a utilizar el árbol de clasificación para decidir el valor que tomará el minibucket o nodesize (número mínimo de observaciones en las hojas de un árbol) en los modelos de Bagging, Random Forest, Gradient Boosting y Xgboost.

Esto se hace ya que por motivos computacionales resulta muy costoso, por lo que se fija este valor para el resto de algoritmos, aunque también se podría en un futuro y si se tuviera más capacidad de computación, incluirlo como hiperparámetro en todos estos modelos.

Para fijar este hiperparámetro, se probarán valores de entre 100 y un 10% del tamaño de la muestra, es decir, 1500. Para cada uno de estos valores se sacarán la tasa de acierto y el AUC y donde se alcance un valor máximo tanto en tasa de aciertos como AUC querrá decir que es el valor óptimo.

Se muestra la tabla:

Tabla 28:Tuneo Minibucket

MINIBUCKET	TASA DE ACIERTOS	AUC
100	0.7445	0.8204
200	0.7407	0.8182
300	0.7434	0.8186
400	0.746	0.8204
500	0.7389	0.8192
600	0.7315	0.8103
700	0.729	0.8087
800	0.7276	0.8063
900	0.7235	0.8018
1000	0.7197	0.7996
1100	0.7121	0.7884
1200	0.71	0.7822
1300	0.7058	0.7803
1400	0.7013	0.7787
1500	0.6947	0.7759

Se observa que desde un tamaño mínimo de hoja de 100 hasta 400 tanto la tasa de aciertos como el AUC aumentan y a partir de este tamaño, ambas métricas de evaluación comienzan a disminuir. Por tanto, se fija el minibucket en 400.

4.4. BAGGING

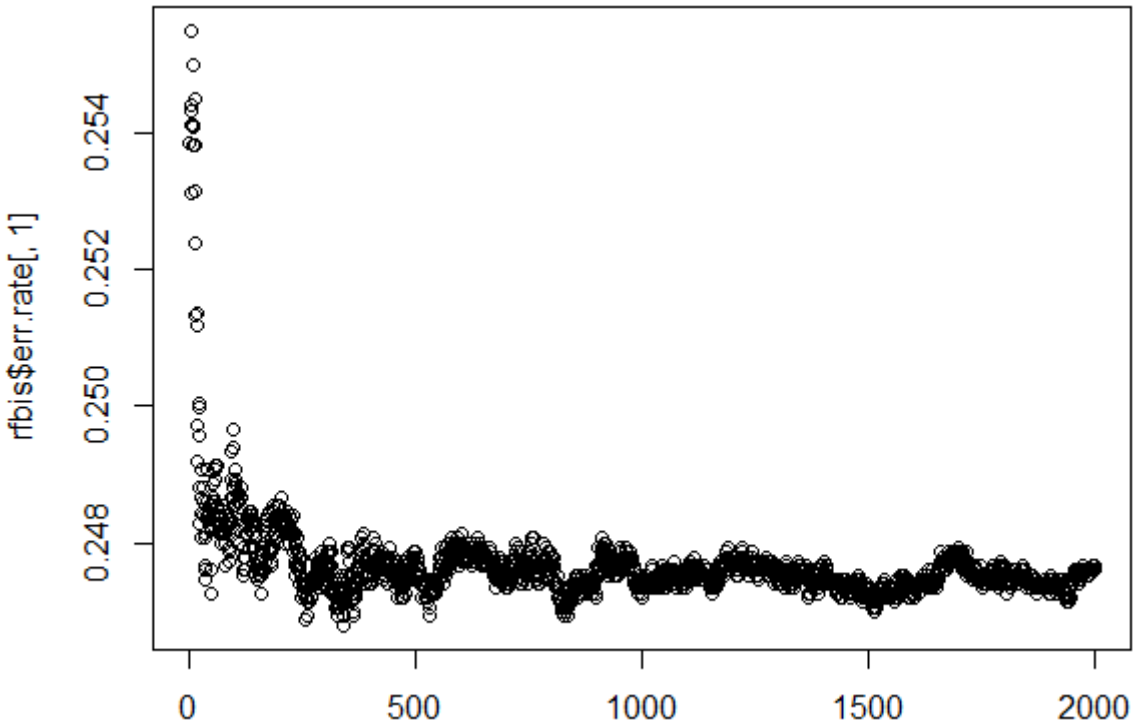
Los hiperparámetros que se van a tunear en este algoritmo son:

- **Ntree**: número de árboles de decisión que se modelizan.
- **Sampsize**: tamaño de cada una de las submuestras

Se llevará a cabo un proceso de tuneo secuencial. Aunque lo suyo sería hacer el grid completo de todos con todos, por problemas computacionales, se llevará a cabo una estructura secuencial donde primero se va a mantener fijo el **Nodesize** (número mínimo de observaciones en las hojas de cada árbol) en 400, (atendiendo al tuneo del minibucket en el árbol simple) y se cerrará el número de árboles (**Ntree**) que se modelizarán. Una vez fijados estos dos hiperparámetros se tuneará el **Sampsize**.
Además, otro hiperparámetro de este algoritmo es **Mtry**, que en este caso permanecerá fijado en el número de variables independientes que se tienen en el modelo, 16.

A continuación, se muestra en la *Figura 15* la evolución del error en cuanto al número de árboles a construir:

Ilustración 13: Tuneo Ntree. Bagging



Se sabe que en los modelos de Bagging cualquier número de árboles no va a empeorar el correcto funcionamiento del modelo, por lo que poner más árboles no es malo, pero en este caso basta con fijar el **Ntree en 300**, que a pesar de que a partir de ahí sigue habiendo fluctuaciones, estas no son pronunciadas y está más o menos estabilizado en torno a 0.248.

A continuación, se procede a tunear el Sampsize. Como para todos los algoritmos, los modelos se comparan a través de validación cruzada repetida con cuatro grupos (k=4), por lo que el número máximo de observaciones en cada muestra debe ser inferior a 0.75 * 15000, que es 11250, atendiendo a la siguiente fórmula:

$$nºmax.sampsize = \frac{k-1}{k} * nºtotal\ obs$$

Por lo tanto, se prueban valores de Sampsize entre 200 y 11250 y mediante validación cruzada repetida se comparan todos los modelos obtenidos y se elegirá aquel considerado óptimo teniendo en cuenta la tasa de fallos ponderada y el AUC.

Se comparan los siguientes modelos:

Tabla 29: Modelos de Bagging

MODELO	SAMPSIZE	NODESIZE	NTREE	MTRY
Bagging1000	1000	400	300	16
Bagging11250	11250	400	300	16
Bagging10000	10000	400	300	16
Bagging8000	8000	400	300	16

Bagging7000	7000	400	300	16
Bagging5000	5000	400	300	16
Bagging3000	3000	400	300	16

Ilustración 14: AUC. Bagging.

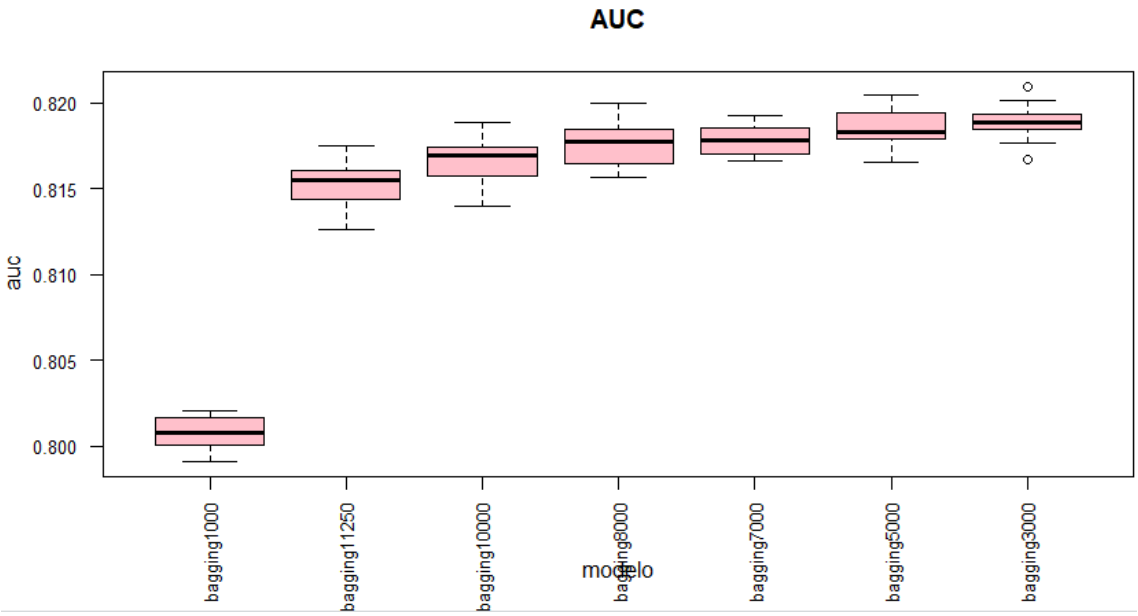
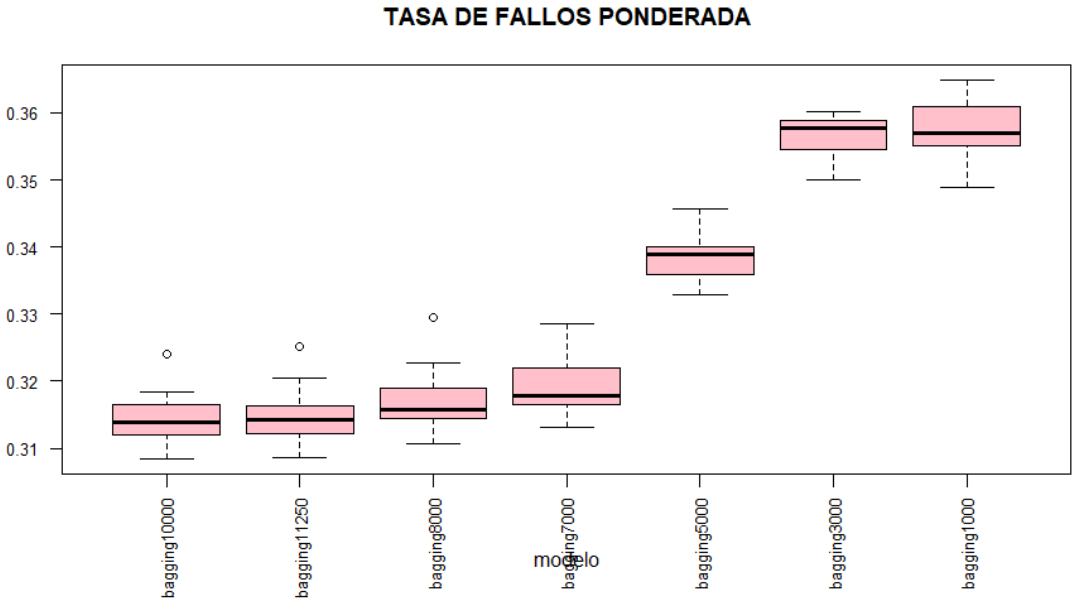


Ilustración 15: Tasa de fallos ponderada. Bagging.



El modelo Bagging1000 queda de primeras descartado debido a ser el peor en cuanto a tasa de fallos ponderada y en cuanto a AUC. El resto de modelos en cuanto a AUC muestran un rendimiento parecido, pero mirando la tasa de fallos se pueden descartar los modelos de Bagging3000 y Bagging5000.

Por tanto, entre los modelos restantes, tienen valores similares en cuanto a tasa de fallos ponderada y AUC, por lo que se optará por elegir el modelo Bagging7000 que es el tercer

mejor en cuanto a AUC y el cuarto mejor en tasa de fallos ponderada. Este modelo se caracteriza por: Sampsize = 7000, Nodesize = 400, Ntree = 300 y Mtry = 16.

4.5. RANDOM FOREST

En este algoritmo se van a tunear los siguientes hiperparámetros:

- **Mtry**: número de variables predictoras entre las que elige la mejor para la partición de cada uno de los nodos.
- **Ntree**: número de árboles de clasificación a construir.
- **Sampsize**: Tamaño de cada submuestra

Por motivos computacionales, como en el caso de Bagging, se llevará a cabo un tuneo secuencial, aunque cabe destacar que también se podrían tunear todos con todos en forma de parrilla. Un tuneo de todos con todos daría mejores resultados a la hora de encontrar el mejor modelo, pero por optimalidad en cuanto a velocidad de conseguir el modelo, se opta por un tuneo secuencial.

En primer lugar, se fija el Nodesize en 400, siguiendo los resultados que se habían obtenido en el árbol de clasificación.

A continuación, se procede a tunear el Mtry para determinar cuántas variables se seleccionan antes de cada partición del nodo y cuál se elige de entre esas.

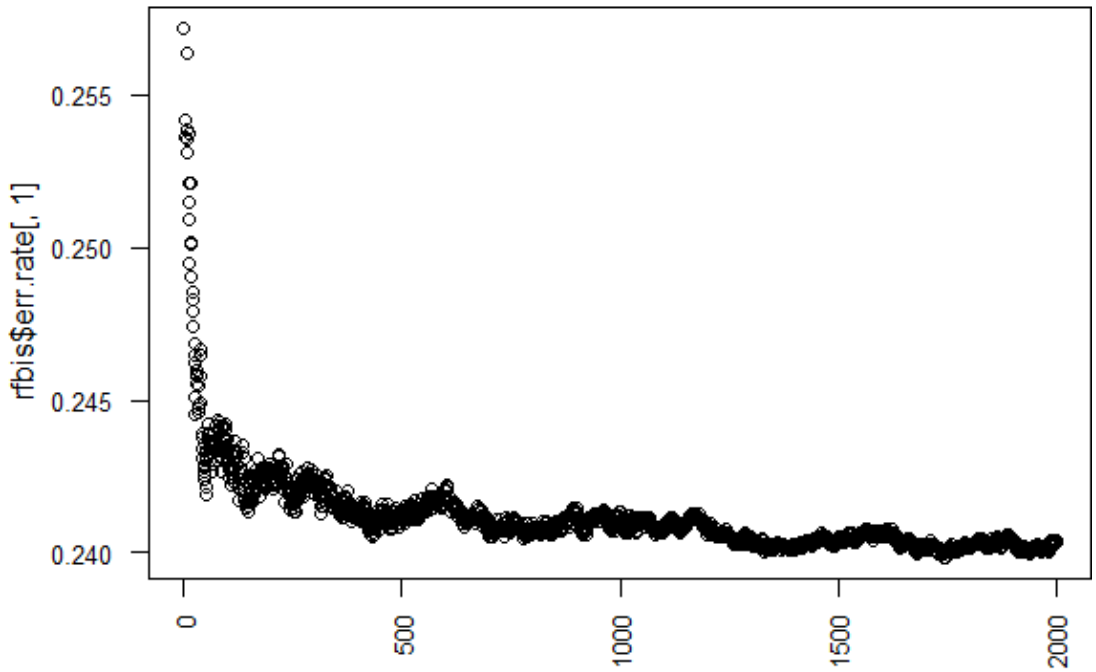
Tabla 30: Tuneo Mtry. Random Forest

MTRY	ACCURACY
3	0.7566667
4	0.7569333
5	0.7567333
6	0.7554000
7	0.7544000
8	0.7537333
9	0.7523333
10	0.7519333
11	0.7507333
12	0.7509333
13	0.7512667
14	0.7494000
15	0.7494667
16	0.7487333

En la *Tabla 30* se muestran los valores de Mtry que se prueban (entre 3 y 16 debido a que 16 es el número de variables seleccionadas bajo regresión logística en el apartado de selección de variables) junto con la tasa de acierto.
El valor de Mtry para el que la tasa de aciertos alcanza un máximo, es 4, por lo que se fijará este hiperparámetro **Mtry = 4**.

A continuación, al igual que en Bagging, es necesario determinar el número de árboles que han de construirse.

Ilustración 16: Tuneo Ntree. Random Forest.



Atendiendo a la *Figura 16,* se fija el Ntree en 500 ya que es un valor a partir del cual el error alcanza una estabilidad y fluctúa ligeramente poco.

Por último, se tunea el último hiperparámetro disponible, el Sampsize. Al igual que en bagging y por la misma razón alegada en ese apartado, se probarán valores de Sampsize de hasta 11250 observaciones. Se prueban los siguientes modelos y se comparan bajo validación cruzada repetida con 4 grupos a partir del AUC y la tasa de fallos ponderada.

Tabla 31: Modelos de Random Forest

MODELO	SAMPSIZE	NODESIZE	NTREE	MTRY
Rf4 1000	1000	400	500	4
Rf4 3000	3000	400	500	4
Rf4 5000	5000	400	500	4
Rf4 7000	7000	400	500	4
Rf4 8000	8000	400	500	4
Rf4 10000	10000	400	500	4
Rf4 11250	11250	400	500	4

Ilustración 17: AUC. Random Forest.

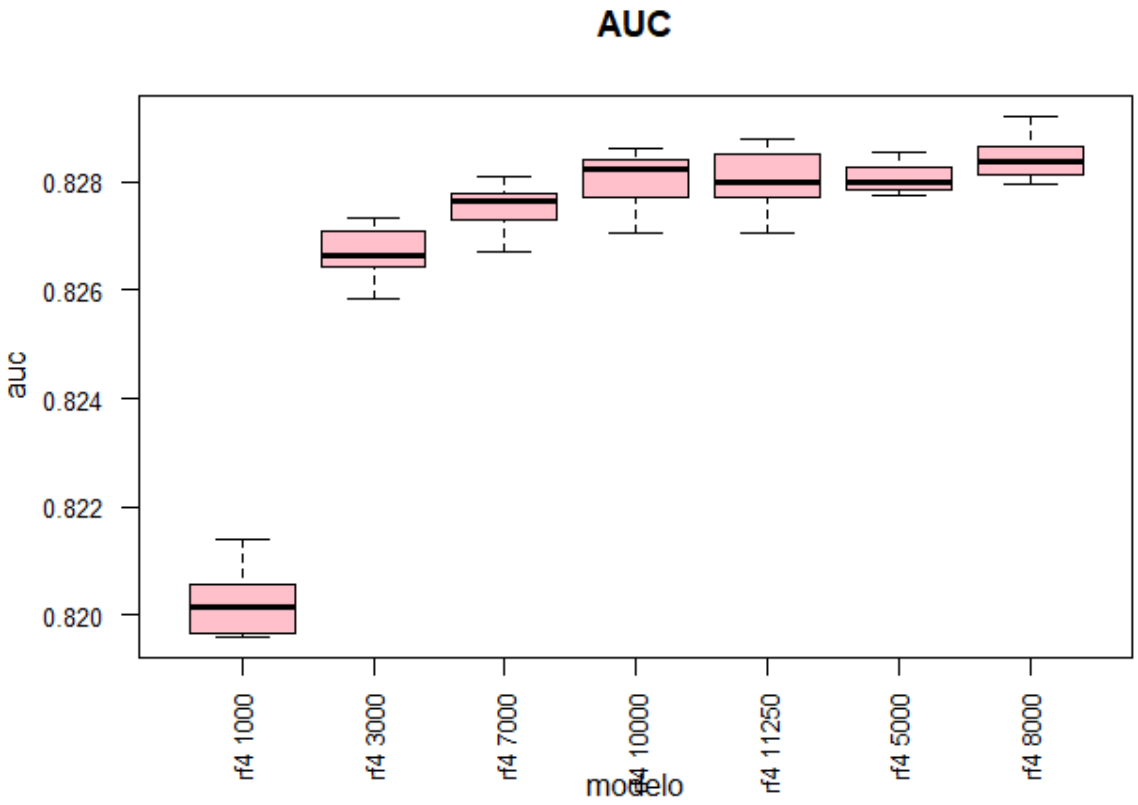
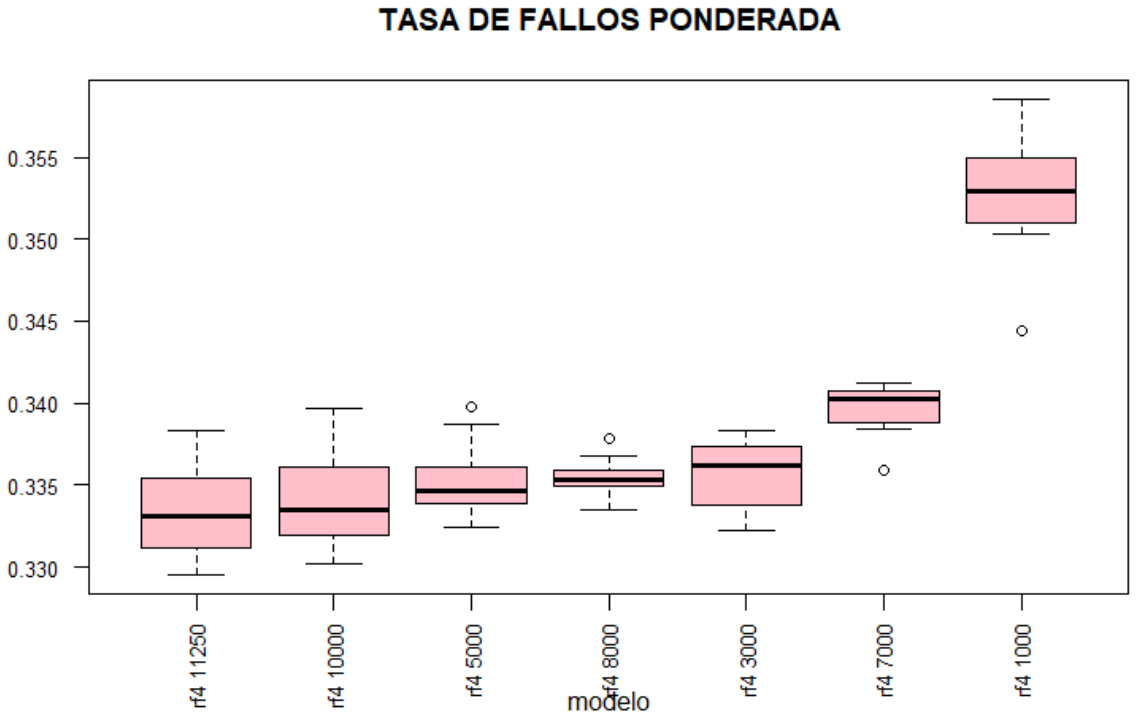


Ilustración 18: Tasa de fallos ponderada. Random Forest.



Atendiendo a los gráficos anteriores, se puede descartar inicialmente el modelo rf4 1000 ya que es el peor en cuanto a tasa de fallos ponderada y AUC. Además, los modelos rf4 7000 y rf4 3000 también se descartan por ser los siguientes peores en tanto AUC como tasa de fallos ponderada.

Entre los modelos restantes se elige mejor modelo el rf4 5000 que es el segundo mejor en cuanto a AUC y el tercer mejor en cuanto a tasa de fallos ponderada, aunque también habría sido buena elección el modelo rf4 8000, ya que es el mejor en cuanto a AUC y el cuarto mejor en cuanto a tasa de fallos ponderada.

El modelo rf4 5000 se caracteriza por los siguientes valores de los hiperparámetros: Sampsize = 5000, Nodesize = 400, Ntree = 500, Mtry = 4.

4.6. GRADIENT BOOSTING

En Gradient Boosting se van a tunear los siguientes parámetros:

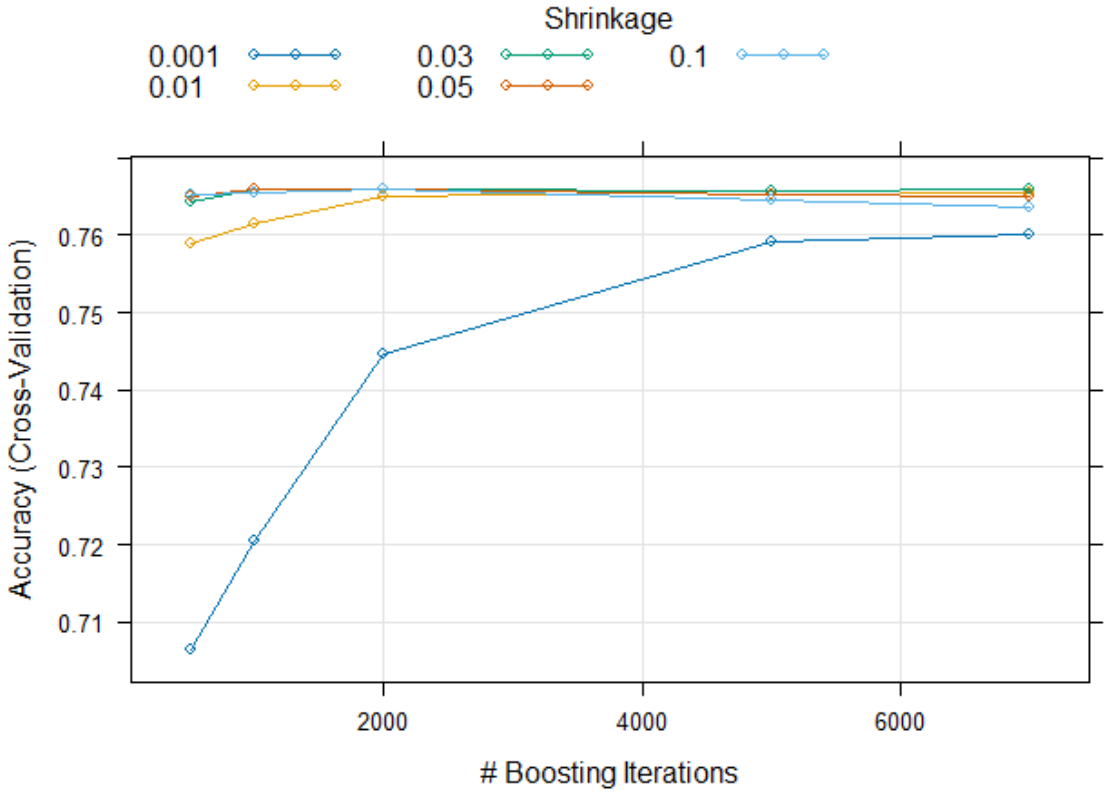
- **N.trees**: Número de árboles de clasificación a construir.
- **Shrinkage**: Parámetro que reduce la contribución de cada árbol en el modelo final. Esto se logra multiplicando las predicciones del árbol por un factor pequeño, mejorando la estabilidad del modelo y reduciendo el riesgo de sobreajuste.

Por otra parte, hay otros hiperparámetros que se van a fijar de antemano:

- **N.minobsinnode**: Número mínimo de observaciones en las hojas de cada árbol. Como en Bagging y Random Forest se utiliza el valor que se obtuvo con el árbol de clasificación construido inicialmente en el que se tuneó este valor. Por lo tanto, se fija en 400.
- **Bag.fraction**: Por defecto se dejará en 1, se utilizará todo el conjunto de datos.
- **Interaction.depth**: Profundidad de cada árbol. Por defecto se fija en 2.

En este algoritmo se tunearán los dos hiperparámetros mencionados al principio de forma simultánea. A continuación, se muestra el gráfico de a partir del cual se determinan varios valores a probar:

Ilustración 19: Tuneo hiperparámetros. Gradient Boosting.



A la vista de la *Figura 19,* se prueban los siguientes modelos atendiendo a la tasa de aciertos que se alcanza con cada combinación de hiperparámetros y luego se comparan los modelos bajo validación cruzada repetida a través del AUC y la tasa de fallos ponderada.

Tabla 32: Modelos de Gradient Boosting

MODELO	NTREES	NODESIZE	SHRINKAGE
Gbm0.1 2000	2000	400	0.1
Gbm0.05 1000	1000	400	0.05
Gbm0.01 5000	5000	400	0.01
Gbm0.03 2000	2000	400	0.03

Ilustración 20: AUC. Gradient Boosting.

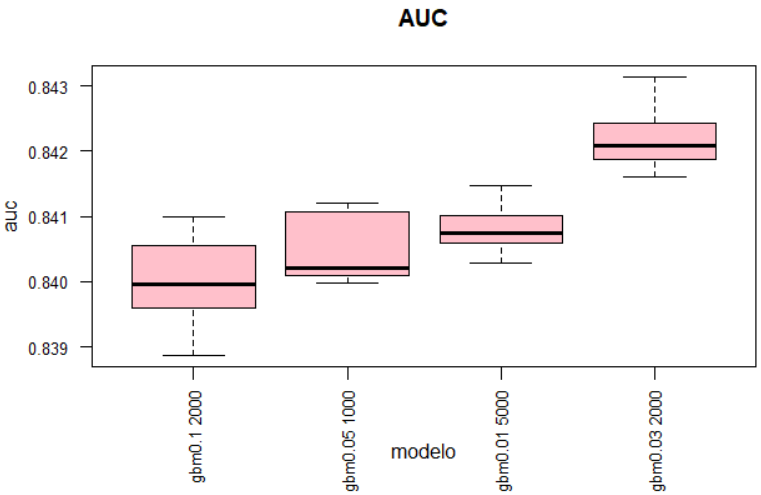
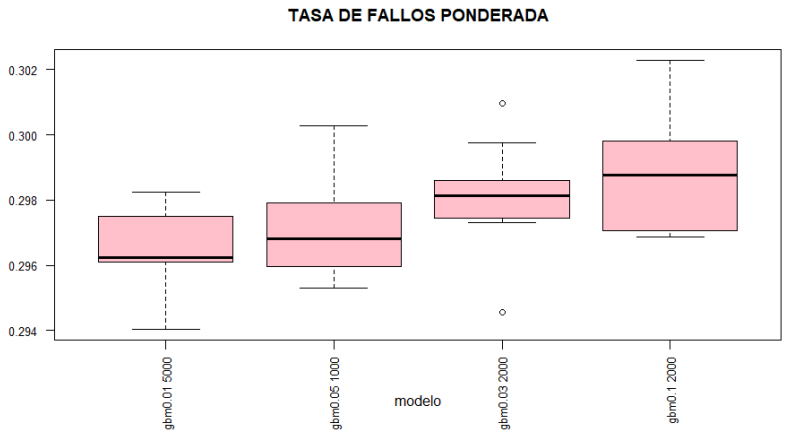


Ilustración 21: Tasa de fallos ponderada. Gradient Boosting.



En este caso, se tienen dos modelos claramente superiores al resto que son gbm0.01 5000 y gbm0.1 2000. El gbm0.01 5000 es el mejor en cuanto a tasa de fallos y el segundo mejor en cuanto a AUC, mientras que el modelo gbm0.03 2000 es el mejor en AUC y el tercer mejor en cuanto a tasa de fallos ponderada. Por tanto, se elige el modelo gbm0.01 5000 dado que 1 y 2 en ranking es mejor que 1 y 3 en ranking.
Este modelo se caracteriza por los valores siguientes de los hiperparámetros: N.trees = 5000, Nodesize = 400, shrinkage = 0.01.

4.7. XGBOOST

Los parámetros que se van a tunear en Xgboost son los siguientes:

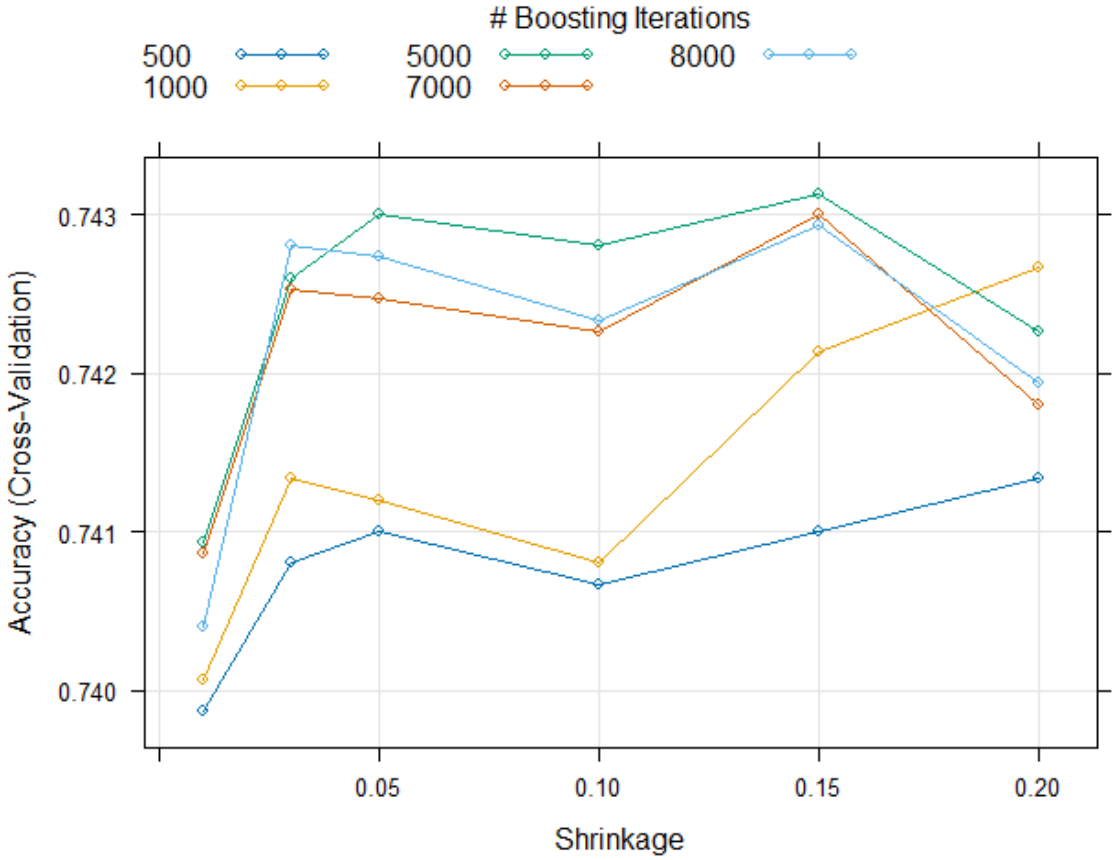
- **Nrounds**: Número de árboles de decisión a construir.
- **Eta**: Parámetro de aprendizaje que reduce la influencia de cada árbol en el modelo final. Al multiplicar las predicciones de cada árbol por un valor pequeño, mejora la estabilidad del modelo y disminuye el riesgo de sobreajuste, similar al shrinkage de Gradient Boosting.

Hay otros parámetros como **Min_child_weight** (similar al N.minobsinnode de Gradient Boosting) que se fija desde el principio en 400 de acuerdo con el valor óptimo encontrado en el tuneo del minibucket en el árbol de decisión simple, y **Colsample_bytree** (similar al Mtry en Bagging y Random Forest) y **Subsample** (similar a Sampsize en Bagging y Random Forest) que se dejan fijados en 1 en primera instancia.

Una vez encontrados algunos modelos susceptibles de ser elegidos como óptimos, se harán pruebas modificando los hiperparámetros que inicialmente se han fijado: colsample_bytree y Subsample. De esta forma se extraen conclusiones más solventes y precisas.

Por tanto, en primer lugar, se tunean los parámetros mencionados con anterioridad a partir del gráfico siguiente:

Ilustración 22: Tuneo hiperparámetros. Xgboost.



Antes de continuar es importante recalcar que para llegar a este gráfico de tuneo se han ido moldeando los resultados. En un principio se probó con valores de Nrounds de entre 100 y 5000, pero se observó que conforme se aumentaba el número de iteraciones, aumentaba la tasa de aciertos, y valores pequeños no eran buenos, por lo que se eliminó el valor de 100 y se añadieron 7000 y 8000. Asimismo, con el shrinkage se probó con valores pequeños hasta 0.15 y se observó una tendencia creciente por lo que se añadió el valor de 0.2.

De esta forma, observando los cambios, se ve que unas iteraciones superiores a 7000 y 8000 no parecen dar mejores resultados debido a que la línea correspondiente a Nrounds = 5000 da una tasa de acierto superior, y un valor del shrinkage de 0.2 da una tasa de acierto inferior.

Por tanto, de ese gráfico se extraen varios posibles modelos que se comparan bajo validación cruzada repetida a partir del AUC y de la tasa de fallos ponderada.

Tabla 33: Modelos de Xgboost (1ªParte)

MODELO	NROUNDS	MIN_CHILD_WEIGHT	SHRINKAGE
Xgbm_1	5000	400	0.05
Xgbm_2	5000	400	0.15
Xgbm_3	7000	400	0.15
Xgbm_4	8000	400	0.15

Ilustración 24: AUC Xgboost. (1ªParte)

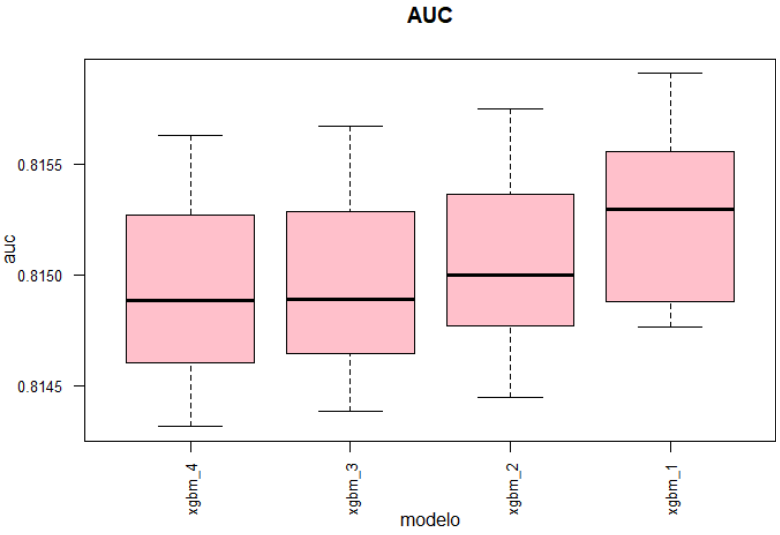
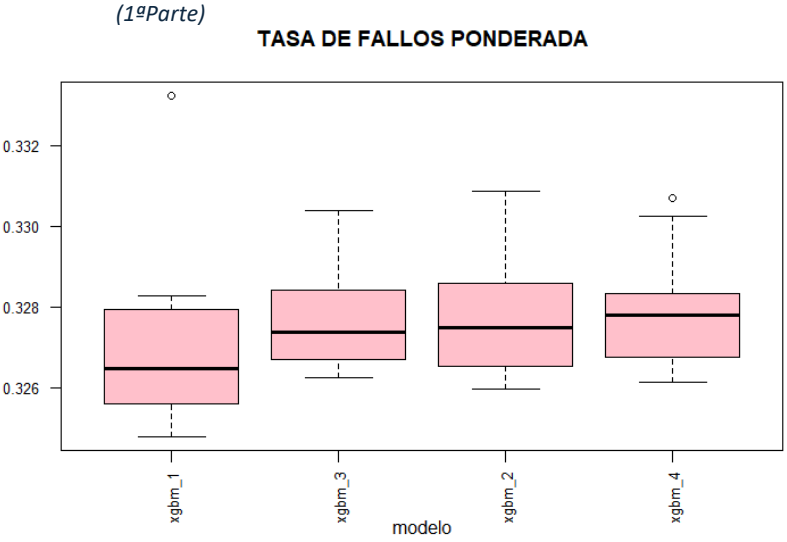


Ilustración 23: Tasa de fallos ponderada. Xgboost (1ªParte)



Atendiendo a ambos gráficos, se observa que el modelo ganador en este caso sería el Xgbm_1 ya que tanto para AUC como para la tasa de fallos ponderada es el mejor modelo, al presentar el AUC más alto y la tasa de fallos más baja. Este modelo se caracteriza por:

- Nrounds = 5000
- Min_child_weight = 400
- Shrinkage = 0.05
- Colsample_bytree = 1
- Subsample = 1

Para este modelo ganador se generan 3 modelos más:

► Sorteo de variables y no observaciones → Tuneo del hiperparámetro Colsample_bytree.

► Sorteo de observaciones y no de variables → Tuneo del hiperparámetro Subsample

► Sorteo de observaciones y variables → Tuneo de los hiperparámetros Subsample y Colsample_bytree

Esto se hace con el fin de saber si, metiendo las partes de Bagging (sorteo de observaciones) y de Random Forest (sorteo de variables), cambian o no los resultados y se obtiene un modelo mejor.

Para determinar el porcentaje de observaciones (Subsample) que se van a sortear se tienen en cuenta el valor de Sampsize que se obtuvo óptimo en Bagging. Se tenía un valor de 7000, por lo que se sortean un 50% de las observaciones. En cuanto al porcentaje de variables (Colsample_bytree) se sortea el 25% ya que, teniendo inicialmente 16 variables, se elegirían de esta forma 4 coincidiendo con las que se elegían en el modelo de Random Forest.

Tabla 34: Modelos Xgboost (2ªParte)

MODELO	NROUNDS	MIN_CHILD_WEIGHT	SHRINKAGE	COLSAMPLE_BYTREE	SUBSAMPLE
Xgbm_1	5000	400	0.05	1	1
Xgbm_5	5000	400	0.05	0.25	1
Xgbm_6	5000	400	0.05	1	0.5
Xgbm_7	5000	400	0.05	0.25	0.5

A continuación, se muestran los gráficos de cajas y bigotes de la tasa de fallos ponderada y el AUC tras la aplicación de validación cruzada repetida. Se muestran los modelos de la *Tabla 34* excepto el modelo Xgbm_6 que no se mostrará en los gráficos siguientes ya que tras los resultados era un modelo mucho peor con una tasa de fallos ponderada de 0.40 y un AUC de 0.76 y los gráficos no se veían de forma correcta.

Ilustración 25: AUC. Xgboost (2ªParte)

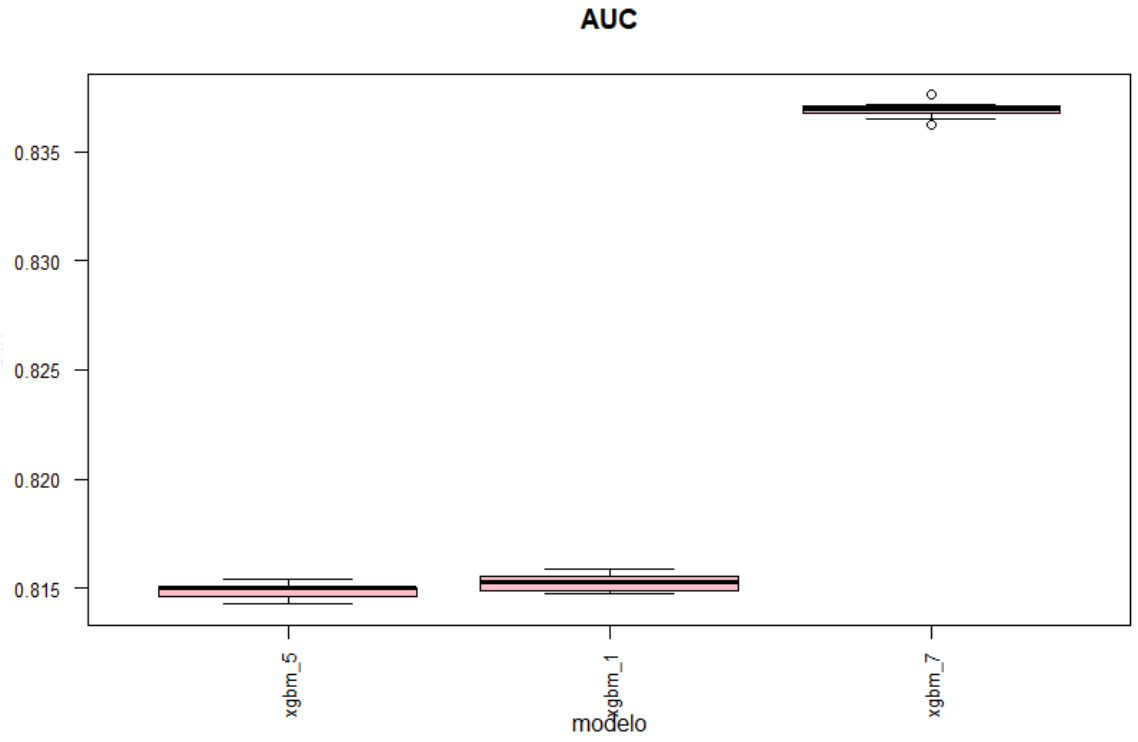
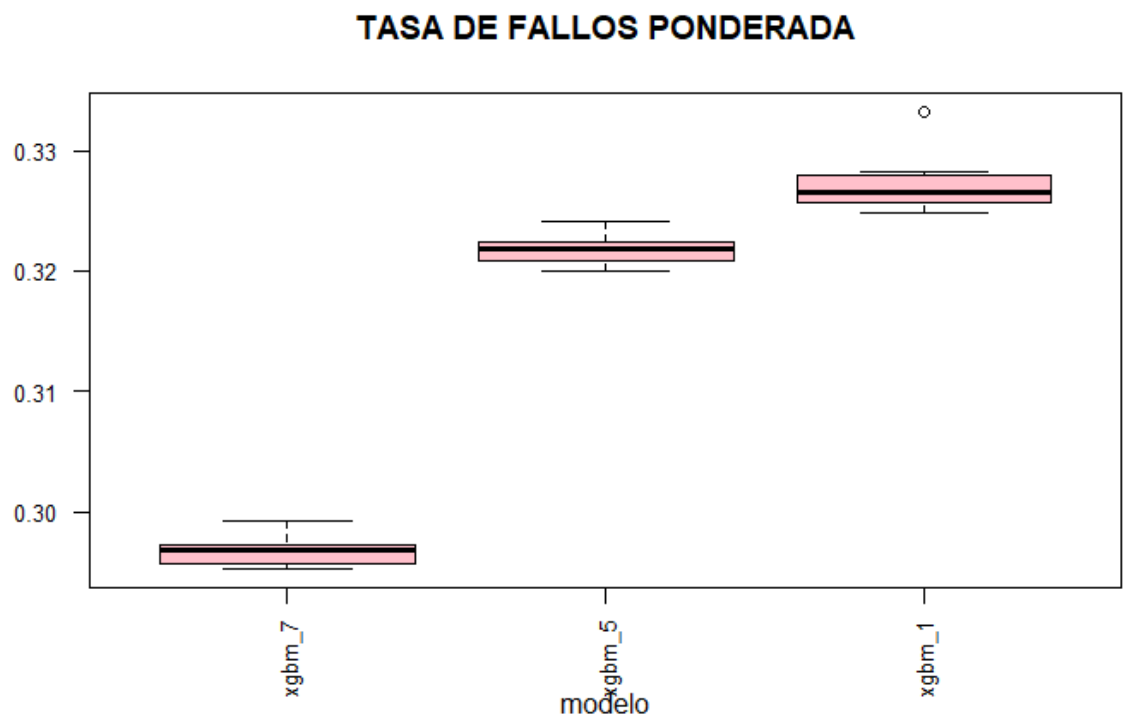


Ilustración 26:Tasa de fallos ponderada. Xgboost (2ªParte)



Se observa que el sorteo simultáneo de observaciones y de variables que se hace con el modelo Xgbm_7 resulta que es el mejor en cuanto a AUC y el mejor en cuanto a tasa de fallos ponderada con bastante diferencia con respecto a los demás, por lo que será elegido como óptimo. Se caracteriza por los siguientes valores de los hiperparámetros: Nrounds = 5000, Min_child_weight = 400, shrinkage = 0.05, colsample_bytree = 0.25, Subsample = 0.5.

4.8. SUPPORT VECTOR MACHINE (SVM)

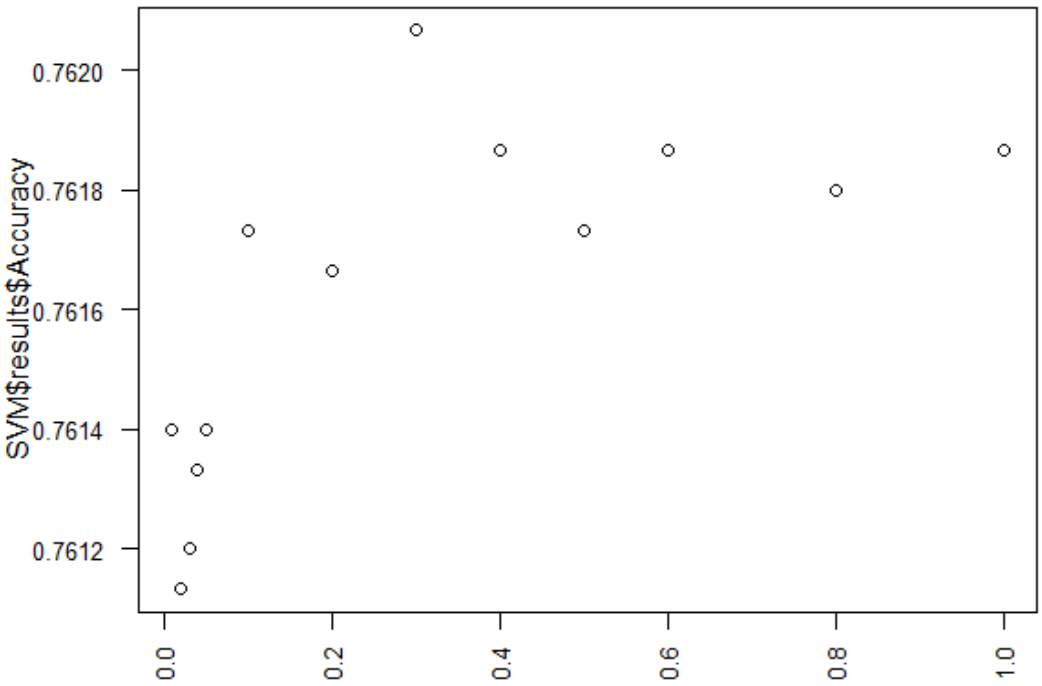
Para todos los modelos de Support Vector Machine se va a elegir un único modelo a partir de los gráficos de tuneo de hiperparámetros en lugar de probar con varias combinaciones de hiperparámetros y luego comparar cada modelo mediante AUC y tasa de fallos tras aplicarles validación cruzada repetida y posteriormente quedarse con uno. Se toma esta decisión debido a que por motivos computacionales es inviable probar con tantos modelos, por lo que se mostrará en cada uno de los tipos de SVM el gráfico de tuneo de hiperparámetros.

4.8.1. SVM LINEAL

En SVM Lineal el único hiperparámetros que ha de ser tuneado es **C** que controla el nivel de regularización del modelo para lograr un buen equilibrio entre la minimización del error y la maximización del margen.

En un primer análisis se probó con valores de C de entre 0 y 10 y se observó que los valores máximos se encontraban entre 0 y 1, por lo que se redujo la parrilla para verlo correctamente.

Ilustración 27: Tuneo hiperparámetros SVM Lineal



En este caso, atendiendo a la *Figura 27*, la máxima tasa de aciertos se alcanza con un valor de C de 0.3, por lo que será el valor elegido para el modelo de SVM lineal.

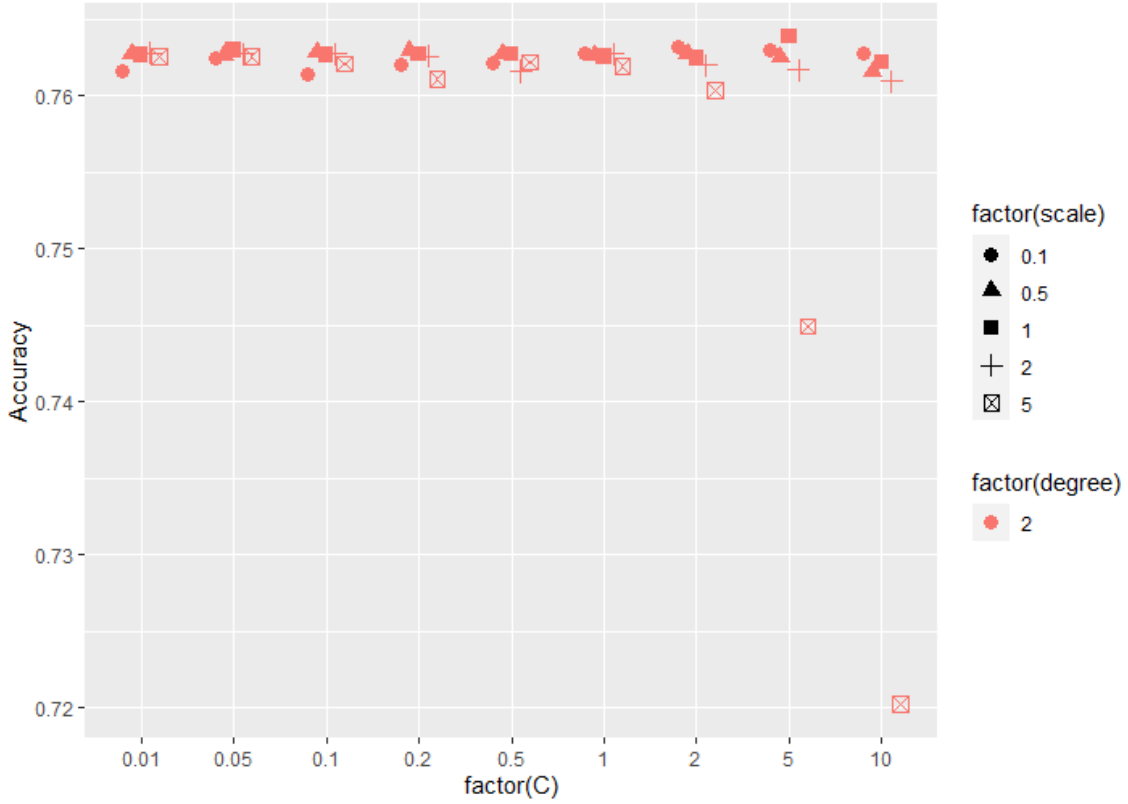
4.8.2. SVM POLINOMIAL

En el SVM Polinomial se tienen tres hiperparámetros: degree, scale y C. El hiperparámetro degree se va a fijar de primeras en dos debido a que el proceso es muy lento computacionalmente hablando. Se toma esta decisión sabiendo que si la separación de clases es lineal el SVM Lineal es mejor, y si la separación no es lineal siempre va a funcionar mejor el SVM Radial.

Por tanto, los hiperparámetros a tunear son:

- **Scale**: Controla la influencia de cada punto de entrenamiento. Un valor pequeño de este hiperparámetro implica una influencia mayor.
- **C**: Es el hiperparámetro de regularización. Controla la compensación entre el margen máximo de separación y el error de clasificación en el conjunto de entrenamiento. Un valor alto de C intenta clasificar todas las muestras de entrenamiento lo que puede llevar a un menor margen y la posibilidad de sobreajustar.

Ilustración 28: Tuneo hiperparámetros SVM Polinomial



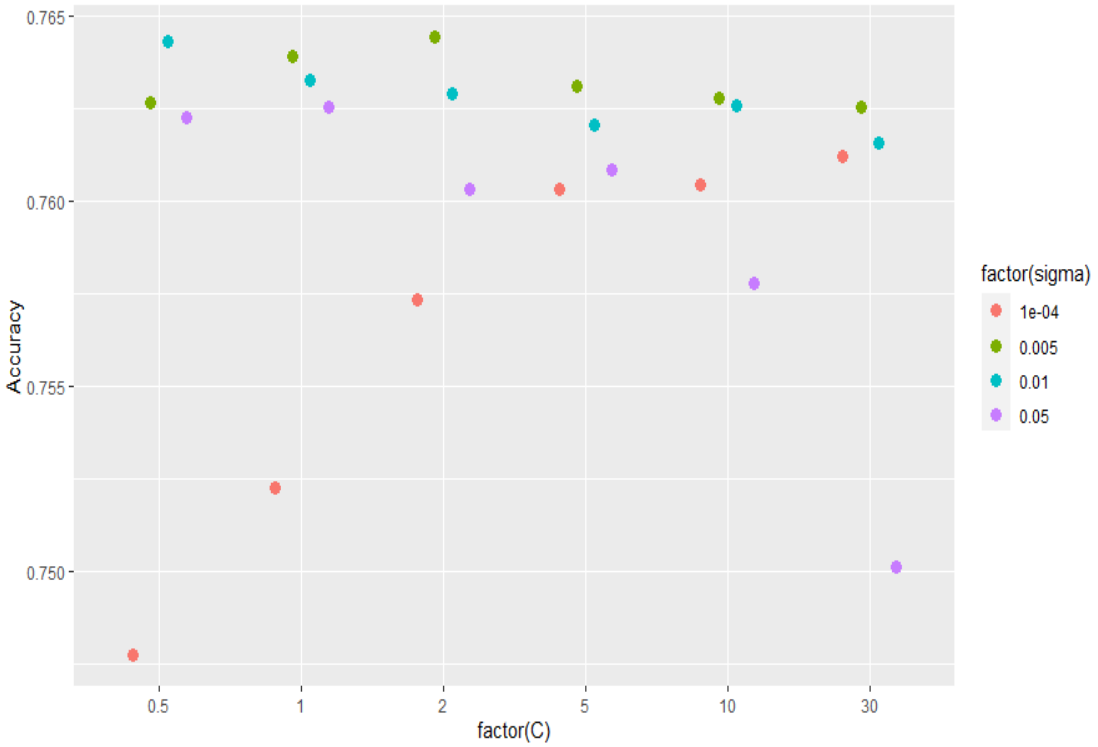
Aquellos hiperparámetros con los que se alcanza una tasa de aciertos máxima, de acuerdo con la *Figura 28*, son los siguientes: C = 5 y scale = 1.

4.8.3. SVM RADIAL (SBF)

En este algoritmo se van a tunear dos hiperparámetros:

- **C**: Controla la regularización del modelo. Un valor alto puede llevar a un menor margen y posible sobreajuste, y un valor bajo permite más errores de clasificación, pero maximiza el margen y puede mejorar la generalización del modelo.
- **Sigma**: Controla la forma del Kernel RBF y la influencia de cada punto de datos. Valores pequeños tienen una influencia más localizada y valores grandes tienen una influencia más grande.

Ilustración 29: Tuneo hiperparámetros SVM SBF



A la vista del gráfico anterior, se obtienen los hiperparámetros que han de ser fijados para este algoritmo. Se eligen C = 2 y sigma = 0.005, ya que con estos valores se alcanza una mayor tasa de aciertos.

4.9. MÉTODOS DE ENSAMBLADO

Una vez identificados los mejores modelos para cada uno de los algoritmos, se hará una comparación de todos ellos, y se ensamblarán los mejores en cuanto a AUC y tasa de fallos ponderada.

En primera lugar, se muestran los gráficos de cajas y bigotes en cuanto a AUC y tasa de fallos ponderada para seleccionar los mejores modelos que van a ser ensamblados.

Ilustración 30: AUC. Comparación Modelos.

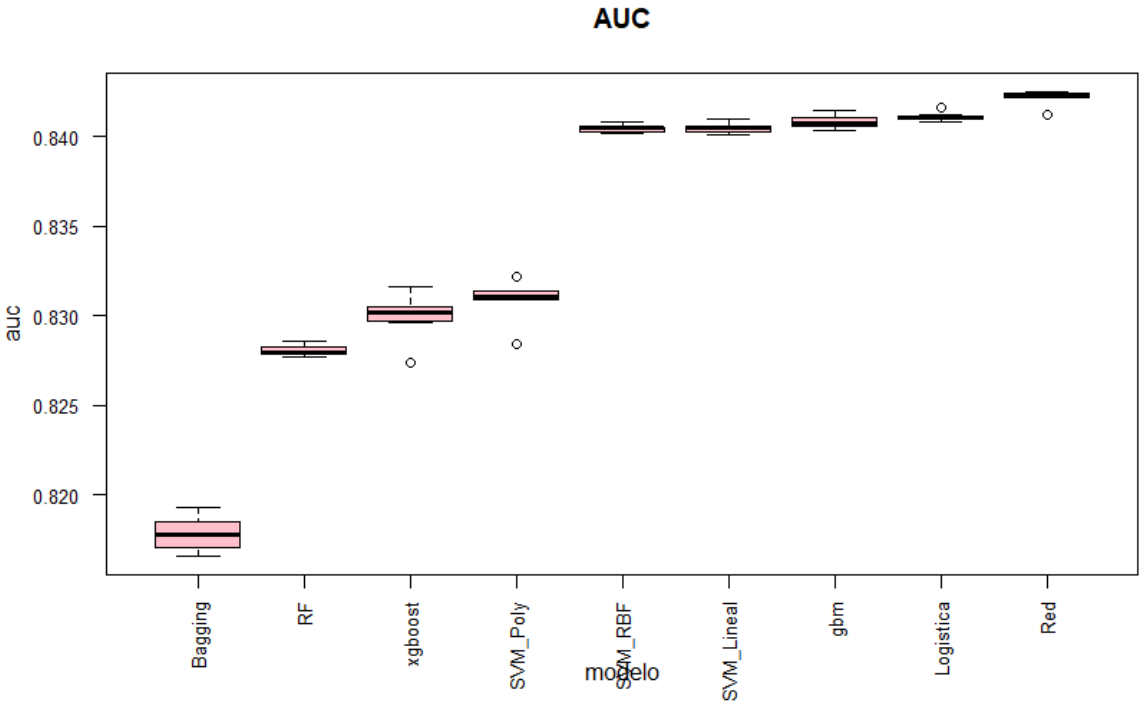
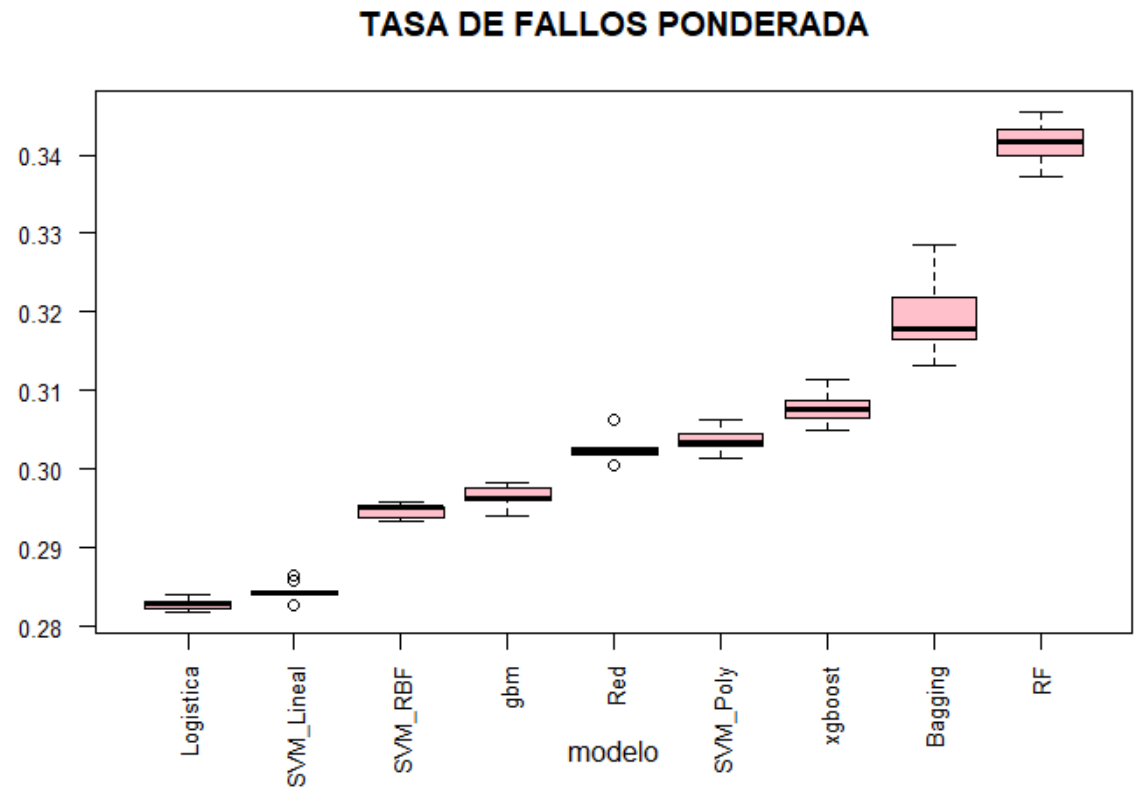


Ilustración 31:Tasa de fallos ponderada. Comparación Modelos.



Observando los gráficos se podría decir que la regresión logística es el que mejor rendimiento ofrece si se tiene en cuenta a la vez el AUC y la tasa de fallos ponderada. Si,

por el contrario, solo se tuviera en cuenta el AUC, se diría que el mejor modelo es el de red neuronal.
Para llevar a cabo el ensamblado se van a elegir los cinco mejores modelos que son los siguientes atendiendo a los resultados anteriores:

- Red neuronal.
- Regresión logística.
- SVM_Lineal.
- SVM_RBF.
- Gradient Boosting.

En este caso se construirán modelos de ensamblados mediante medias. Cabe destacar que además de con la media se pueden hacer con otras funciones de agregación como el máximo o el mínimo.

Se construyen 26 modelos diferentes: diez modelos compuestos por dos, diez modelos compuestos por tres, cinco modelos compuestos por cuatro y, por último, el ensamblado de los cinco a la vez.

A continuación, se muestran los gráficos de cajas y bigotes en cuanto a AUC y tasa de fallos ponderada tras la aplicación de la validación cruzada repetida:

Ilustración 32: AUC. Comparación Métodos Ensamblado.

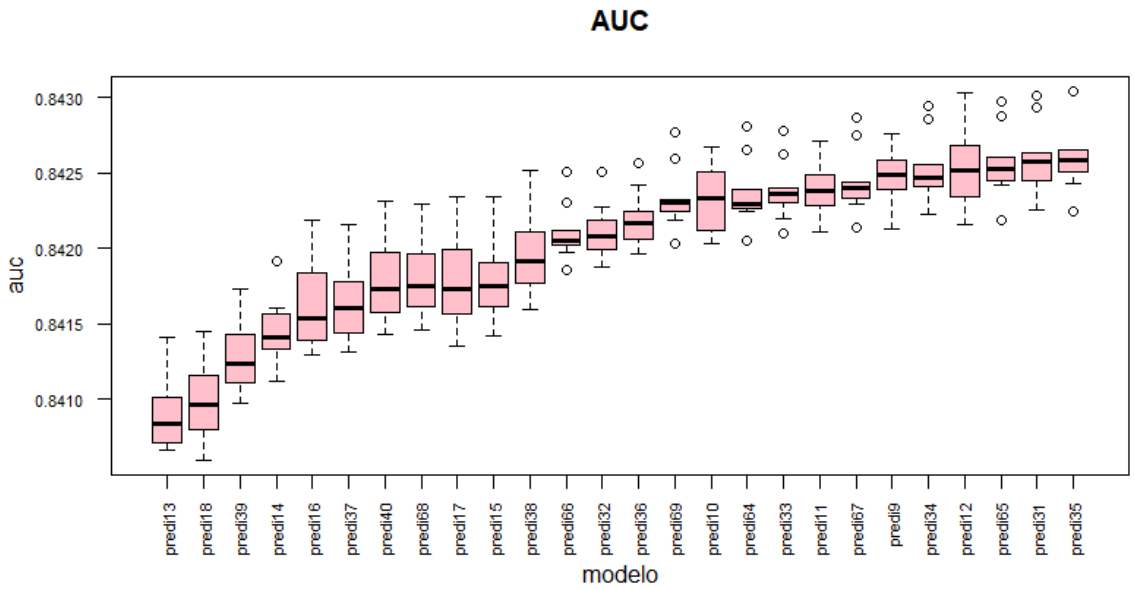


Ilustración 33: Tasa de fallos ponderada. Comparación Métodos Ensamblado.

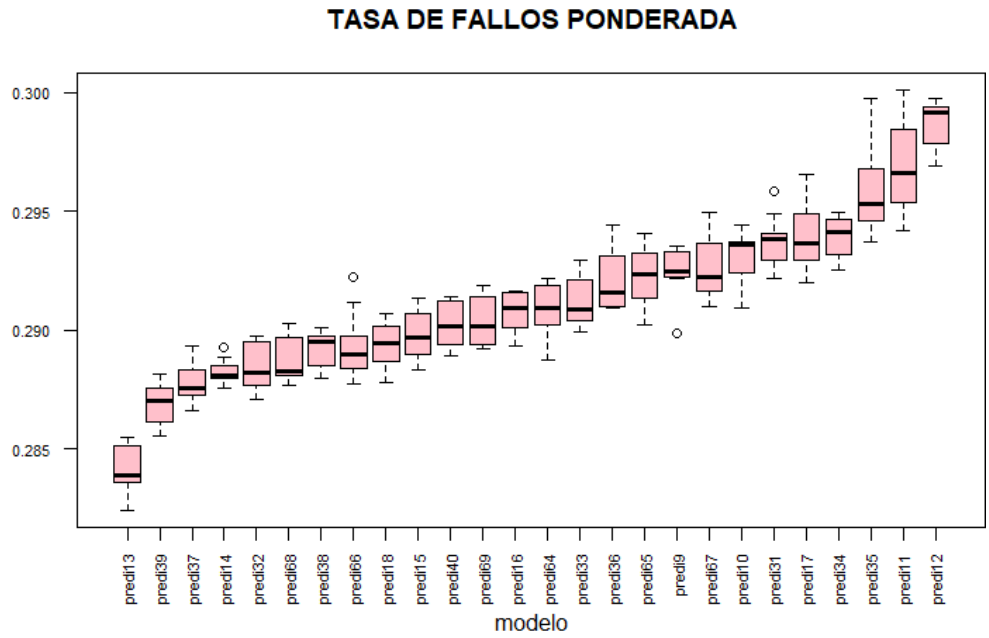


Tabla 35: Definición Modelos de Ensamblado

MODELO	DEFINICIÓN
Predi9	(Logística + Red) / 2
Predi10	(Red + SVM_Lineal) / 2
Predi11	(Red + SVM_Radial) / 2
Predi12	(Red + Gradient Boosting) / 2
Predi13	(Logística + SVM_Lineal) / 2
Predi14	(Logística + SVM_Radial) / 2
Predi15	(Logística + Gradient Boosting) / 2
Predi16	(Gradient Boosting + SVM_Lineal) / 2
Predi17	(Gradient Boosting + SVM_Radial) / 2
Predi18	(SVM_Lineal + SVM_Radial) / 2
Predi31	(Red + Logística + Gradient Boosting) / 3
Predi32	(Red + Logística + SVM_Lineal) / 3
Predi33	(Red + Logística + SVM_Radial) / 3
Predi34	(Red + Gradient Boosting + SVM_Lineal) / 3
Predi35	(Red + Gradient Boosting + SVM_Radial) / 3
Predi36	(Red + SVM_Lineal + SVM_Radial) / 3
Predi37	(Logística + Gradient Boosting + SVM_Lineal) / 3
Predi38	(Logística + Gradient Boosting + SVM_Radial) / 3
Predi39	(Logística + SVM_Lineal + SVM_Radial) / 3
Predi40	(Gradient Boosting + SVM_Lineal + SVM_Radial) / 3
Predi64	(Red + Logística + Gradient Boosting + SVM_Lineal) / 4
Predi65	(Red + Logística + Gradient Boosting + SVM_Radial) / 4
Predi66	(Red + Logística + SVM_Radial + SVM_Lineal) / 4
Predi67	(Red + SVM_Radial + Gradient Boosting + SVM_Lineal) / 4
Predi68	(SVM_Radial + Logística + Gradient Boosting + SVM_Lineal) / 4
Predi69	(Red + Logística + Gradient Boosting + SVM_Lineal + SVM_Radial) / 5

Haciendo una evaluación de los visto en los gráficos, el Predi13 es el modelo que menor AUC y el que menor tasa de fallos ponderada presenta, por lo que dependiendo de a qué criterio le diésemos más importancia, sería un buen modelo o no.

Por otra parte, el rango de valores en los que se mueve tanto el AUC como la tasa de fallos ponderada no es muy grande, por lo que se podría decir que el Predi9 formado por la combinación de red y regresión logística, funciona bien en AUC y con una tasa de fallos ponderada en torno al 0.29, no parece ser una mala opción.

Para obtener una decisión final acerca de qué modelo da mejores resultados, se añaden a los gráficos anteriores los modelos elegidos para ensamblar: red, regresión, gradient boosting, SVM_Lineal, SVM_RBF.

Ilustración 34: AUC. Comparación Modelos y Ensamblados.

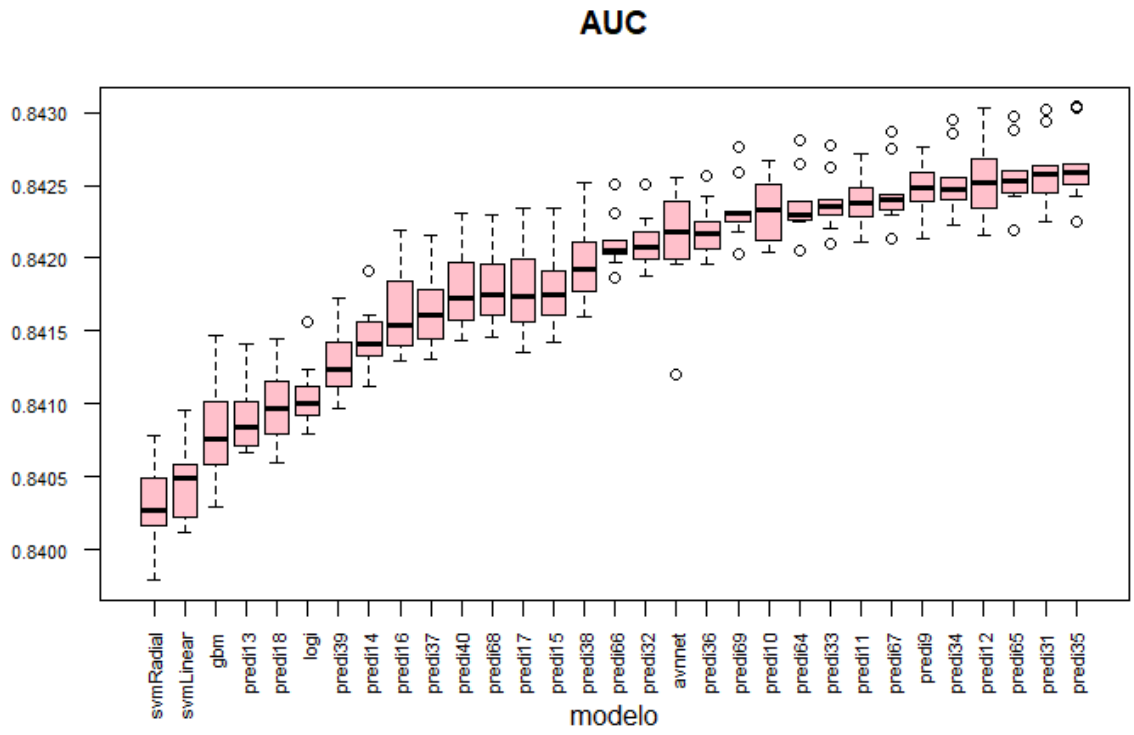
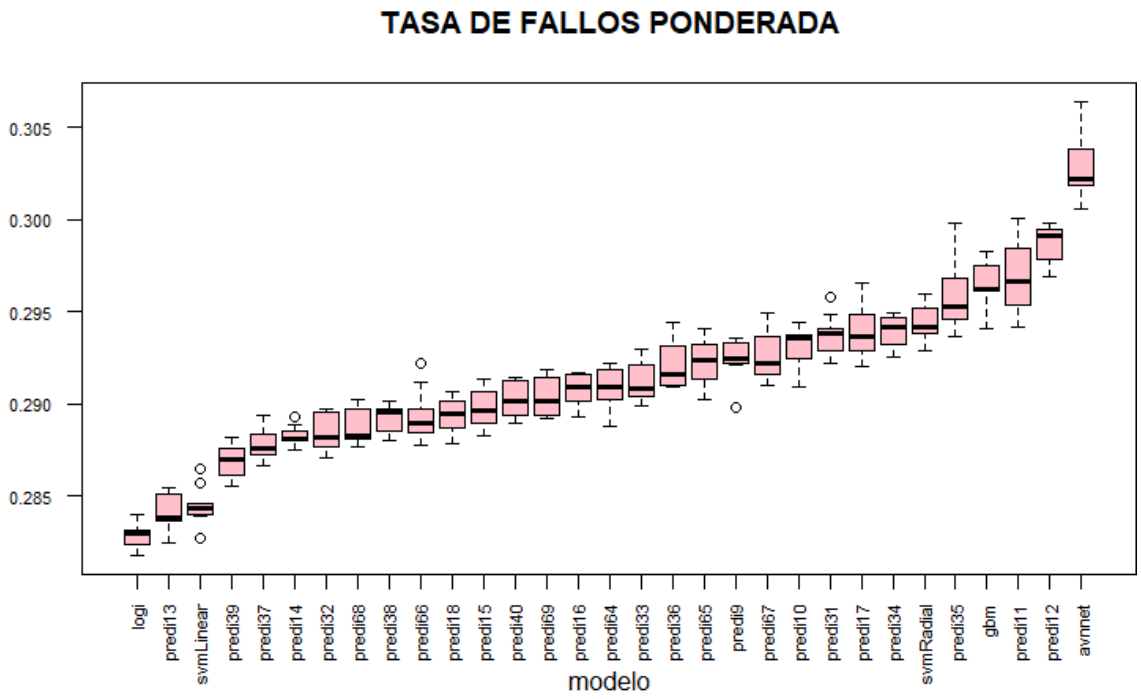


Ilustración 35: Tasa de fallos ponderada. Comparación Modelos y Ensamblados.



Nótese que los modelos de ensamblado no han mejorado al mejor modelo en cuanto a tasa de fallos ponderada, pero si consiguen mejorar el AUC con respecto a los mejores modelos, siendo el Predi35 el que consigue una mayor efectividad, seguido de Predi31 y Predi65.

4.10. MODELO FINAL

Vistos los gráficos del apartado anterior, aquel modelo con el que se alcanza una mayor efectividad es con la regresión logística. Con los hiperparámetros que se han probado en los diferentes algoritmos, que no han sido muy amplios debido a los problemas computacionales, no se han logrado mejores resultados. Quizás probando con más combinaciones de hiperparámetros se podrían conseguir mejores resultados, cosa que con la logística ya no se puede más ya que está en su óptimo y con los demás no se tiene garantizado que estén en su óptimo.

Por consiguiente, se opta por elegir la logística debido a las ventajas de interpretación que tiene y al ser un modelo más sencillo.

Por otra parte, observando la *Figura 34* y la *Figura 35* en las que se muestran los gráficos de cajas y bigotes de todos los algoritmos y ensamblados en cuanto a AUC y tasa de fallos ponderada y se le da la misma importancia de decisión a ambas medidas, en cómputo general se elige modelo óptimo la regresión logística.

Una vez elegido el modelo de regresión logística, se pasa a una evaluación de este mediante diferentes métricas.

Inicialmente se contaba con una base de datos de 319795 observaciones, de las cuales fueron muestreadas 15000, obteniendo una muestra balanceada de los datos, es decir, 7500 observaciones “Yes” y 7500 observaciones “No”.

Esta muestra de 15000 observaciones se divide a continuación en train (70%) y validación (30%). El resto de observaciones que no han sido utilizadas anteriormente, se utilizarán como datos test para evaluar el modelo entrenado con el conjunto de datos de entrenamiento, es decir, el conjunto de datos test se compone de 304795 observaciones.

Por tanto, se sacarán tres matrices de confusión, una para cada uno de los conjuntos de datos. Las dos primeras (train y validación) necesitan reconversión mediante la aplicación de unos pesos para tener los mismos porcentajes en la variable objetivo de la población original. Los pesos son los mismos que se explicaron en el apartado 3.1:

- P0 = (7500/292422)
- P1 = (7500/27373)

La tercera matriz de confusión también necesita una reconversión, pero los pesos son diferentes a las dos anteriores, ya que como se cogieron 7500 observaciones de la categoría “No” y 7500 observaciones de la categoría “Yes”, los números resultantes no son las proporciones de la población. En la población originariamente había un 9% de unos y en este conjunto de datos hay menos de ese 9% de unos, por lo que hay que recalcular los pesos para este conjunto de datos:

- P0 = (284922/292422)
- P1 = (19873/27373)

La matriz de confusión representa a los individuos clasificados por categorías de la variable objetivo, de forma que indicará cuántos han sido predichos correctamente y cuántos erróneamente de cada una de las categorías.

A partir de la matriz de confusión se pueden sacar diferentes medidas de evaluación como la sensibilidad, la especificidad, la tasa de fallos ponderada o la tasa de aciertos.

Conjunto de datos de entrenamiento:

Se tiene el 70% de 15000, es decir, 10500 observaciones:

Tabla 36: Matriz de confusión. Train (1)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	4016	1214
	YES	1280	3990

Se aplican los pesos para recuperar la población original:

Tabla 37: Matriz de confusión. Train (2)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	4016/(P0)	1214/(P1)
	YES	1280/(P0)	3990/(P1)

P1 = (7500/27373)

P0 = (7500/292422)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	156582	4431
	YES	49907	14562

$$Sensibilidad = \frac{VP}{VP + FN} = \frac{14562}{14562 + 4431} = 0.7667$$

$$Especificidad = \frac{VN}{VN + FP} = \frac{156582}{156582 + 49907} = 0.7583$$

$$TasaFallosPonderada = \frac{(3 * 4431) + (1 * 49907)}{225482} = 0.2803$$

$$TasaAciertos = \frac{156582 + 14562}{225482} = 0.7590$$

Conjunto de datos de validación

Se tiene el 30% de 15000, es decir, 4500 observaciones:

Tabla 38: Matriz de confusión. Validación (1)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	1670	540
	YES	534	1756

Se aplican los pesos para recuperar la población original:

Tabla 39: Matriz de confusión. Validación (2)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	1670/(P0)	540/(P1)
	YES	534/(P0)	1756/(P1)

P1 = (7500/27373)

P0 = (7500/292422)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	65113	1971
	YES	20820	6409

$$Sensibilidad = \frac{VP}{VP + FN} = \frac{6409}{6409 + 1971} = 0.7648$$

$$Especificidad = \frac{VN}{VN + FP} = \frac{65113}{65113 + 20820} = 0.7577$$

$$TasaFallosPonderada = \frac{(3 * 1971) + (1 * 20820)}{94313} = 0.2834$$

$$TasaAciertos = \frac{65113 + 6409}{94313} = 0.7583$$

Conjunto de datos test

Se tienen 319975 – 15000 = 304795 observaciones:

Tabla 40: Matriz de confusión. Test (1)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	196807	3265
	YES	88115	16608

Tabla 41: Matriz de confusión. Test (2)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	213692/P0	4571/P1
	YES	71230/P0	15302/P1

P1 = (19873/27373)

P0 = (284922/ 292422)

MATRIZ DE CONFUSIÓN		REALIDAD	
		NO	YES
PREDICCION	NO	219317	6296
	YES	73105	21077

$$Sensibilidad = \frac{VP}{VP + FN} = \frac{21077}{21077 + 6296} = 0.7699$$

$$Especificidad = \frac{VN}{VN + FP} = \frac{219317}{219317 + 73105} = 0.75$$

$$TasaFallosPonderada = \frac{(3 * 6296) + (1 * 73105)}{319795} = 0.2877$$

$$TasaAciertos = \frac{219317 + 21077}{319795} = 0.7517$$

A continuación, se presenta una tabla resumen de todas las medidas de evaluación obtenidas en los tres conjuntos de datos:

Tabla 42: Métricas de evaluación

	SENSIBILIDAD	ESPECIFICIDAD	TASA DE FALLOS PONDERADA	TASA DE ACIERTOS
TRAIN	0.7667	0.7583	0.2803	0.7590
VALIDACIÓN	0.7648	0.7577	0.2834	0.7583
TEST	0.7699	0.75	0.2877	0.7517

La sensibilidad mide la proporción de verdaderos positivos que son correctamente identificados por el modelo, mientras que la especificidad mide la proporción de verdaderos negativos que son correctamente identificados. Tanto la sensibilidad para los tres conjuntos de datos como la especificidad para los tres conjuntos de datos se aprecian valores similares, lo que es un indicativo de que el modelo de regresión logística es robusto y con unos valores de entorno al 75% le hace ser un modelo con buena capacidad predictiva.

Una cosa similar ocurre con la tasa de fallos ponderada y la tasa de aciertos. Por lo tanto, se entiende que el modelo es bueno y estable.

Por último, se muestra la tabla de estimación de parámetros del modelo de regresión logística:

Tabla 43: Modelo de regresión logística. Estimación de parámetros

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.92706	0.12097	7.663	1.81e-14
Age	0.91873	0.03080	29.946	< 2e-16
Diabetic.Yes	0.64728	0.06216	10.123	< 2e-16
Stroke.Yes	135.318	0.11011	13.121	< 2e-16
Sex.Female	-0.73015	0.05051	-14.301	< 2e-16
GenHealth.Excellent	-215.661	0.13651	-16.152	< 2e-16

GenHealth.VeryGood	-162.883	0.12007	-13.931	< 2e-16
GenHealth.Good	-105.871	0.11556	-9.025	< 2e-16
Smoking.Yes	0.39812	0.04941	8.157	3.43e-16
KidneyDisease.Yes	0.66040	0.10535	6.633	3.29e-11
Asthma.Yes	0.26161	0.06991	3.606	0.000311
GenHealth.Fair	-0.52701	0.11872	-4.319	1.57e-05
SkinCancer.Yes	0.27579	0.07487	3.533	0.000411
MentalHealth	0.07861	0.02707	2.000	0.045485
AlcoholDrinking.Yes	0.36062	0.11079	-4.745	2.08e-06
DiffWalking.Yes	0.22466	0.06659	2.385	0.017075
Race.Black	-0.27120	0.10032	-2.870	0.004100

Atendiendo al p – valor se confirma que todas las variables son significativas en el modelo y, por tanto, no es necesario eliminar ninguna de ellas ni probar con otro set de variables más reducido.

Con respecto al signo de los coeficientes estimados se puede observar que la mitad tienen signo negativo y la otra mitad signo positivo, se interpretan:

- **Age** → Signo positivo, indica que, a mayor edad, mayor probabilidad de que una persona padezca una enfermedad cardíaca.
- **Diabetic.Yes** → Signo positivo. Ser diabético aumenta la probabilidad de sufrir enfermedad cardíaca comparado con no ser diabético.
- **Stroke.Yes** → Signo positivo. Haber tenido un derrame cerebral incrementa significativamente las probabilidades de sufrir enfermedad cardíaca comparado con no haber tenido un derrame cerebral.
- **Sex.Female** → Signo negativo. Ser mujer reduce las probabilidades de sufrir enfermedad cardíaca en comparación con ser hombre.
- **GenHealth.Excellent** → Signo negativo. Tener una salud general excelente reduce las probabilidades de sufrir una enfermedad cardíaca comparado con tener una salud general pobre.
- **GenHealth.VeryGood** → Signo negativo. Tener una salud general muy buena reduce las probabilidades de sufrir enfermedad cardíaca con respecto a tener una salud general pobre.
- **GenHealth.Good** → Signo negativo. Tener una salud general buena disminuye la probabilidad de tener una enfermedad cardíaca comparado con tener una salud general pobre.
- **Smoking.Yes** → Signo positivo. Ser fumador aumenta la probabilidad de sufrir una enfermedad cardíaca con respecto a no ser fumador.
- **KidneyDisease.Yes** → Signo positivo. Tener una enfermedad renal aumenta la probabilidad de tener una enfermedad cardíaca con respecto a no tener enfermedad renal.
- **Asthma.Yes** → Signo positivo. Tener asma incrementa la probabilidad de sufrir enfermedad cardíaca comparado con no tener asma.

- **GenHealth.Fair** → Signo negativo. Tener una salud general justa reduce la probabilidad de sufrir enfermedad cardíaca comparado con tener una salud general pobre.
- **SkinCancer.Yes** → Signo positivo. Tener cáncer de piel aumenta la probabilidad de sufrir una enfermedad cardíaca en comparación con no tener cáncer de piel.
- **MentalHealth** → Signo positivo. Por cada día más que una persona no se encuentra mentalmente bien, aumentan las probabilidades de sufrir una enfermedad cardíaca. Esto quiere decir que peores condiciones de salud mental están asociadas a una mayor probabilidad de tener enfermedad cardíaca.
- **AlcoholDrinking.Yes** → Signo positivo. Consumir alcohol aumenta la probabilidad de sufrir una enfermedad cardíaca comparado con no consumir alcohol.
- **DiffWalking.Yes** → Signo positivo. Tener dificultades para caminar aumenta la probabilidad de sufrir una enfermedad cardíaca con respecto a no tener problemas para caminar.
- **Race.Black** → Signo negativo. Ser de raza negra reduce la probabilidad de tener enfermedad cardíaca comparado con respecto a no ser de raza negra.

En cuanto a las cuatro variables de GenHealth, todas ellas son negativas unas más que otras como manda el sentido común, ya que cuanto mejor sea la salud general, más positivo es el número que sale y, por consiguiente, menor probabilidad de sufrir una enfermedad cardíaca.

Por último, se muestran cuáles de las variables anteriores tienen más importancia en el modelo mediante el análisis tipo II. Se utiliza para contrastar si las variables del modelo son significativas o no, es decir, evalúa cuánto empeora el modelo al quitar una variable. De esta forma se obtiene un ranking de la utilidad de las variables.

Tabla 44: Importancia de variables

	LRChisq	**P - valor**
Age	1070.66	<2.2e-16
Diabetic.Yes	104.71	<2.2e-16
Stroke.Yes	212.47	<2.2e-16
Sex.Female	209.76	<2.2e-16
GenHealth.Excellent	293.78	<2.2e-16
GenHealth.VeryGood	219.00	<2.2e-16
GenHealth.Good	89.35	<2.2e-16
Smoking.Yes	66.51	3.49E-13
KidneyDisease.Yes	47.17	6.50E-09
Asthma.Yes	13.04	0.0003045
GenHealth.Fair	19.39	1.06E-02

SkinCancer.Yes	12.62	0.0003816
MentalHealth	4.00	0.0453978
AlcoholDrinking.Yes	23.17	1.48E-03
DiffWalking.Yes	5.69	0.0170171
Race.Black	8.29	0.0039949

A la vista de la *Tabla 44*, se puede afirmar que las variables más importantes y, por tanto, más influyentes a la hora de sufrir o no una enfermedad cardíaca son: Age, Stroke.Yes (haber sufrido un derrame cerebral), GenHealth.Excellent (tener una salud excelente), GenHealth.VeryGood (tener una salud muy buena), Sex.Female (ser mujer) y Diabetic.Yes (ser diabético). El resto de variables son menos importantes pero significativas en el modelo de regresión logística.

5. CONCLUSIONES

El propósito fundamental de este trabajo es establecer la asociación entre la presencia de enfermedades cardiovasculares, como el infarto de miocardio o la enfermedad coronaria, y diversos factores de riesgo.

En primer lugar, se analizó la relación entre la variable HeartDisease y todas las demás variables cualitativas mediante pruebas de independencia. Los resultados indicaron que todas las variables cualitativas estaban asociadas con el hecho de haber sufrido un infarto de miocardio o enfermedad coronaria. Asimismo, se investigó la relación entre HeartDisease y las cinco variables cuantitativas, revelando que todas ellas estaban asociadas a la variable respuesta.

Posteriormente, tras tomar una muestra balanceada de los datos, se procedió a crear variables dummies, aumentando el número de variables de 18 a 27. De estas 27 variables, se seleccionaron 16 como variables independientes mediante un proceso de selección de variables basado en regresión logística.

Con este set de variables, se evaluaron diversos modelos de Machine Learning, ajustando sus hiperparámetros para optimizar su rendimiento. Cabe recordar que, debido a limitaciones computacionales, no fue posible probar todas las combinaciones posibles, lo que sugiere que los resultados podrían mejorarse con recursos adicionales.

Después del ajuste de todos los modelos y la construcción de modelos ensamblados, se concluyó que el modelo de regresión logística era el más adecuado debido a su buen desempeño en términos de tasa de fallos ponderada y AUC, así como sus ventajas en cuanto a interpretación y simplicidad en comparación con otros modelos.

Con este modelo se alcanzaron buenos valores de sensibilidad y especificidad, en torno al 75% en cualquier conjunto de datos, lo cual indica su capacidad para detectar correctamente tanto a los individuos verdaderamente enfermos como a los verdaderamente no enfermos. Lo mismo se observó con la tasa de fallos ponderada y la tasa de aciertos.

Por otra parte, entre el resto de modelos probados, el modelo de red neuronal muestra un buen rendimiento, por lo que, de haber probado con muchas más combinaciones de los hiperparámetros, cosa que no se ha podido hacer debido a los problemas computacionales mencionados en su momento, se habría conseguido mejorar tanto en AUC como en tasa de fallos ponderada.

Así como para la red neuronal, para el resto de modelos de Machine Learning tuneados, se podrían mejorar haciendo un grid de todos los hiperparámetros con todos y probando con el mayor número posible de combinaciones. El único contratiempo es que supone un coste computacional demasiado elevado.

En el apartado del modelo final se eligió mejor modelo la regresión logística y se sacaron las variables más importantes y el signo que tenían cada una en el modelo. La variable más importante es la edad y con signo positivo por lo que a más edad, mayores son las probabilidades de sufrir una enfermedad cardíaca. Le sigue en importancia dos variables de GenHealth (GenHealth.Excellent y GenHealth.VeryGood) con signo negativo, lo que quiere decir que cuanta mejor salud se tenga menores son las probabilidades de sufrir una enfermedad cardíaca.

En conclusión, la regresión logística se presenta como una herramienta eficaz para la clasificación y predicción en escenarios clínicos y de diagnóstico. Su capacidad para proporcionar un equilibrio adecuado entre sensibilidad y especificidad garantiza decisiones diagnósticas precisas, reduciendo tanto los falsos positivos como los falsos negativos.

Desde el punto de vista clínico, es vital que los pacientes se sometan a chequeos médicos regulares para monitorear los factores de riesgo identificados. Adoptar un estilo de vida saludable, que incluya una dieta equilibrada y ejercicio físico regular, es fundamental para reducir el riesgo de enfermedades cardíacas.

Además, dejar de fumar y moderar el consumo de alcohol son recomendaciones cruciales, ya que estos hábitos tienen un impacto significativo en la salud cardiovascular. Gestionar adecuadamente otras condiciones de salud crónicas, como la enfermedad renal y el asma, también es importante para disminuir el riesgo general de enfermedades cardíacas.

Es importante promover la educación sobre los factores de riesgo y cómo manejarlos para la toma óptima de decisiones para la salud.

Por último, personalizar el plan de cuidado según las características del paciente, utilizando modelos predictivos, puede aumentar la efectividad a la hora de tratar con una enfermedad de este tipo. Por tanto, con un enfoque basado en datos, es posible mejorar significativamente la prevención y el manejo de enfermedades cardíacas, ofreciendo una atención más efectiva y centrada en el paciente.

BIBLIOGRAFÍA

• BERTOMEU, V. and CASTILLO-CASTILLO, J., 2008. Situación de la enfermedad cardiovascular en España. Del riesgo a la enfermedad. *Revista Española de Cardiología Suplementos,* 8(5), pp.2E-9E.

• BISHOP, C.M., 2006. Pattern recognition and machine learning. *Springer google schola,* 2, pp.645-678.

• BLOG DE CARDIOLOGÍA DE LOS HOSPITALES QUIRÓNSALUD ALICANTE, MURCIA, TORREVIEJA Y VALENCIA, , Enfermedades cardíacas frecuentes. https://www.quironsalud.com/blogs/es/corazon-salud/enfermedades-cardiacas-frecuentes#:~:text=Son%20aquellos%20problemas%20afectan%20al,dolor%20y%20sin%20s%C3%ADntomas%20evidentes.

• BREIMAN, L., 1996. Bagging predictors. *Machine Learning,* 24**,** pp.123-140.
CDC, Cómo evaluar el índice de masa corporal. https://www.cdc.gov/healthyweight/spanish/assessing/index.html#:~:text=Si%20su%20IMC%20es%20menos,dentro%20del%20rango%20de%20obesidad.

• CDC, b. Know Your Risk for Heart Disease.

• DOMINGO F. RASILLA, , Relación de variables. https://personales.unican.es/rasillad/docencia/G14/TEMA_3/relacion_entre_una_variable_cualitativa_otra_cuantitativa.html#suposici%C3%B3n-1-son-las-dos-muestras-independientes.

• FRIEDMAN, J.H., 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics,* , pp. 1189–1232.

• R. MOHAMMED, J. RAWASHDEH and M. ABDULLAH, 2020. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results, *2020 11th International Conference on Information and Communication Systems (ICICS)* 2020, pp. 243–248.

• RODRIGO, C.P., 2013. Current mapping of obesity. *Nutricion hospitalaria,* 28(5), pp. 21–31.

• Apuntes de la asignatura de Técnicas de Segmentación y Tratamiento de Encuestas de la profesora Aida Calviño Martínez. Curso 2022-2023. Facultad de Estudios Estadísticos. Grado en Estadística Aplicada. UCM.

• Apuntes de la asignatura Técnicas de Machine Learning del profesor Javier Portela García-Miguel. Curso 2023-2024. Facultad de Estudios Estadísticos. Máster en Minería de Datos e Inteligencia de Negocios. UCM.

ANEXO

Comprobación missing, estandarización y creación de variables dummy:

```
> listconti<- c("BMI", "PhysicalHealth", "MentalHealth","Age",
"SleepTime")
> listclass<-c("Smoking", "AlcoholDrinking", "Stroke",
"DiffWalking", "Sex", "Race", "Diabetic", "PhysicalActivity",
"GenHealth", "Asthma", "KidneyDisease", "SkinCancer")
> vardep<-"HeartDisease"
> base<-base[,c(listconti,listclass,vardep)]
> library(naniar)
> gg_miss_var(base)
> # Estandarizamos variables cuantitativas
> means <-apply(base[,listconti],2,mean,na.rm=TRUE)
> sds<-sapply(base[,listconti],sd,na.rm=TRUE)
> base2<-scale(base[,listconti], center = means, scale = sds)
> base<-data.frame(cbind(base2,base[,c(listclass,vardep)]))
> # Creamos las dummy de las variables categóricas
> basebis<-dummy.data.frame(base, listclass, sep = ".")
> variables = c("BMI", "PhysicalHealth", "MentalHealth", "Age", "SleepTime",
"Smoking.Yes", "AlcoholDrinking.Yes","Stroke.Yes", "DiffWalking.Yes", "Sex.Female",
"Race.Indian","Race.Asian", "Race.Black", "Race.Hispanic", "Race.Other", "Race.White",
"Diabetic.Yes", "PhysicalActivity.Yes", "GenHealth.Excellent", "GenHealth.Fair",
"GenHealth.Good", "GenHealth.Poor", "GenHealth.VeryGood", "Asthma.Yes",
"KidneyDisease.Yes", "SkinCancer.Yes","HeartDisease")
> basebis <- basebis[variables] # pasamos de 37 variables a 27
```

Selección de variables bajo regresión logística:

```
> ## SBF
> filtro<-sbf(x,y,sbfControl = sbfControl(functions = rfSBF,
method = "cv", verbose = FALSE))
> a<-dput(filtro$optVariables)
> length(a) ## 23 de 27
> ## RFE
> set.seed(12345)
> control <- rfeControl(functions=rfFuncs, method="cv", number=5)
> results <- rfe(x, y, sizes=c(1:15), rfeControl=control)
> cosa<-as.data.frame(results$results)
> ggplot(cosa,aes(y=Accuracy, x=Variables))+geom_point()+geom_line()+
+   scale_y_continuous(breaks = cosa$Accuracy) +
+   scale_x_continuous(breaks = cosa$Variables)+labs(title="RFE")
> cosa2<-cosa[c(5:16),]
> ggplot(cosa2,aes(y=Accuracy, x=Variables))+geom_point()+geom_line()+
+   scale_y_continuous(breaks = cosa$Accuracy) +
+   scale_x_continuous(breaks = cosa$Variables)+labs(title="RFE")
> selecrfe<-results$optVariables[1:9]
> dput(selecrfe) # visualizamos las variables
> ## STEPWISE AIC
> library(MASS)
> full<-glm(HeartDisease~.,data=archivo1,family = binomial(link="logit"))
> null<-glm(HeartDisease~1,data=archivo1,family = binomial(link="logit"))
> selec1<-stepAIC(null,scope=list(upper=full),
direction="both",family = binomial(link="logit"),trace=FALSE)
> vec<-(names(selec1[[1]]))
> length(vec) # 21 - 1 variables
> dput(vec)
> ## STEPWISE BIC
> full<-glm(HeartDisease~.,data=archivo1,family = binomial(link="logit"))
> null<-glm(HeartDisease~1,data=archivo1,family = binomial(link="logit"))
> selec1<-stepAIC(null,scope=list(upper=full),
direction="both",family = binomial(link="logit"),trace=FALSE,k=9.6)
> vec<-(names(selec1[[1]]))
> length(vec) # 18 variables-1
> dput(vec)
> ## BORUTA
> out.boruta <- Boruta(HeartDisease~., data = archivo1)
> print(out.boruta)
> summary(out.boruta)
> sal<-data.frame(out.boruta$finalDecision)
> sal2<-sal[which(sal$out.boruta.finalDecision=="Confirmed"),
drop=FALSE]
> dput(row.names(sal2))
> length(dput(row.names(sal2))) # 23 variables
> ## MMPC
> mmpc1 <- MMPC(vardep, archivo1, max_k = 3, hash = TRUE,
                test = "testIndLogistic")
> a<-dput(names(archivo1[,c(mmpc1@selectedVars)]))
> length(a)
> ## SES
```

```
> SES1 <- SES(vardep, archivo1, max_k = 2, hash = TRUE,
            test = "testIndLogistic")
> SES1@selectedVars
> dput(names(archivo1[,c(SES1@selectedVars)]))
> a<-dput(names(archivo1[,c(SES1@selectedVars)]))
> length(a)
> ## STEPWISE AIC REPETIDO
> source("funcion steprepetido binaria.R")
> lista<-steprepetidobinaria(data=archivo1,vardep=c("HeartDisease"),
listconti=nombres1,sinicio=12345,sfinal=12385,porcen=0.8,
criterio="AIC")
> tabla<-lista[[1]]
> dput(lista[[2]][[1]])
> ## STEPWISE BIC REPETIDO
> lista<-steprepetido(data=archivo1,vardep=vardep,
listconti= nombres1,sinicio=12345,sfinal=12385,porcen=0.8,criterio="BIC")
> tabla<-lista[[1]]
> dput(lista[[2]][[1]])
```

Regresión logística. Comparación a través de validación cruzada repetida. Boxplots tasa de fallos ponderada y AUC (ILUSTRACIÓN 8, ILUSTRACIÓN 9):

```
> source("cruzadas avNNet y log binaria 2.R")
> data <- basebis
> medias1<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias1$modelo="STEPAIC"
> medias2<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias2$modelo="STEPBIC"
> medias3<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias3$modelo="Boruta"
> medias4<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias4$modelo="RFE"
> medias5<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias5$modelo="MXM
> medias6<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias6$modelo="STEPREP-AIC"
> medias7<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias7$modelo="STEPREP-BIC"
> medias8<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias8$modelo="SES"
> medias9<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
```

```
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias9$modelo="SBF"
> union1<-rbind(medias1,medias2,medias3,medias4,medias5,medias6,
medias7,medias8,medias9)
> par(cex.axis=1.5,las=2)
> boxplot(data=union1,col="pink", auc~modelo,main="AUC")
> boxplot(data=union1, col="pink", tasa~modelo, main= "TASA DE FALLOS PONDERADA"
```

Tuneo red neuronal

```
> set.seed(12345)
> nnetgrid <-  expand.grid(size=c(1,3,4,6,8),decay=c(0.01,0.1,0.001),
bag=F)
> completo<-data.frame()
> listaiter<-c(10,20,50,100,200,300,500,1000,2000,3000,4000,5000,6000)
> for (iter in listaiter)
 {
   rednnet<- train(HeartDisease~Age+Diabetic.Yes+Stroke.Yes+Sex.Female+
              GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+
              Smoking.Yes+KidneyDisease.Yes+Asthma.Yes+GenHealth.Fair+
             SkinCancer.Yes+MentalHealth+AlcoholDrinking.Yes+
              DiffWalking.Yes+Race.Black, data=data,method="avNNet",
             linout = FALSE,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$results)
 }
> completo<-completo[order(completo$Accuracy),]
> ggplot(completo, aes(x=factor(itera), y=Accuracy,
color=factor(decay),pch=factor(size))) +
  geom_point(position=position_dodge(width=0.5),size=3)
```

Red neuronal. Comparación a través de validación cruzada repetida. Boxplots tasa de fallos ponderada y AUC (ILUSTRACIÓN 11, ILUSTRACIÓN 12):

```
> medias10<-cruzadaavnnetbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes","Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=5,size=c(6),decay=c(0.01),repeticiones=5,itera=20)
> medias10$modelo="Red6 0.01 20"
> medias11<-cruzadaavnnetbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=5,size=c(3),decay=c(0.001),repeticiones=5,itera=90)
> medias11$modelo="Red3 0.001 90"
> medias12<-cruzadaavnnetbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=5, size=c(4),decay=c(0.01),repeticiones=5,itera=300)
> medias12$modelo="Red4 0.01 300"
> medias13<-cruzadaavnnetbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent",  "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"), listclass=c(""),grupos=4,
sinicio=1234,repe=5,size=c(3),decay=c(0.01),repeticiones=5,itera=5000)
> medias13$modelo="Red3 0.01 5000"
> union1<-rbind(medias10,medias11,medias12,medias13)
> par(cex.axis=1.5,las=2)
> uni7 <- union1
> uni7$modelo <- with(uni7, reorder(modelo, auc, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni7,auc~modelo,main="AUC",col="pink")
> uni8 <- union1
> uni8$modelo <- with(uni8,reorder(modelo,tasa,mean))
> uni8$modelo <- with(uni8, reorder(modelo, tasa, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni8,auctasa~modelo,main="TASA DE FALLOS PONDERADA",col="pink")
```

Árbol de clasificación. Tuneo Minibucket (TABLA 28)

```
> for (minbu in seq(from=6000, to=11000, by=100))
 {
   print(minbu)
   cat("/n")
   control<-trainControl(method = "cv",number=4,
   classProbs=TRUE,savePredictions = "all")
   arbolgrid <-  expand.grid(cp=c(0))

   arbolcaret<-
train(HeartDisease~Age+Diabetic.Yes+Stroke.Yes+Sex.Female+GenHealth.Excellent
                    +GenHealth.VeryGood+GenHealth.Good+Smoking.Yes+KidneyDisease.Yes
                    +Asthma.Yes+GenHealth.Fair+SkinCancer.Yes+MentalHealth
                    +AlcoholDrinking.Yes+DiffWalking.Yes+Race.Black,data=basebis,
                    method="rpart",minbucket=minbu,trControl=control,tuneGrid=arbolgrid)

   sal<-arbolcaret$pred

   salconfu<-confusionMatrix(sal$pred,sal$obs)
   print(salconfu)

   curvaroc<-roc(response=sal$obs,predictor=sal$Yes)
   auc<-curvaroc$auc
   print(auc)

 }
```

Tuneo Bagging (ILUSTRACIÓN 13)

➔ Tuneo Ntree:

```
> set.seed(123)
> rfbis<-randomForest(HeartDisease~Age+Diabetic.Yes+Stroke.Yes+
Sex.Female+GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+
Smoking.Yes+KidneyDisease.Yes+Asthma.Yes+GenHealth.Fair+
SkinCancer.Yes+MentalHealth+AlcoholDrinking.Yes+DiffWalking.Yes+
Race.Black,data=data,mtry=16,ntree=2000,nodesize=400,
replace=TRUE)
> plot(rfbis$err.rate[,1])
```

➔ **Tuneo Sampsize. Comparación a través de validación cruzada repetida. Boxplots tasa de fallos ponderada y AUC (ILUSTRACIÓN 14, ILUSTRACIÓN 15):**

```
 > medias1<-cruzadarfbin(data=data,vardep="HeartDisease",listconti=c("Age", "Diabetic.Ye
 s", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes","Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes",
"MentalHealth","AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),
listclass=c(""),grupos=5,sinicio=1234,repe=20,nodesize=400,
mtry=16,ntree=300,replace=TRUE,sampsize=1000)
> medias1$modelo="bagging1000"
> medias2<-cruzadarfbin(data=data,vardep="HeartDisease",listconti=c("age", "Diabetic.Yes
", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes","Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes",
"MentalHealth", AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),
listclass=c(""), grupos=5,sinicio=1234,repe=20,nodesize=400,
mtry=16,ntree=300,replace=TRUE,sampsize=11250)
> medias2$modelo="bagging11250"
> medias3<-cruzadarfbin(data=data,vardep="HeartDisease",listconti=c("Age", "Diabetic.Yes
", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes",
"MentalHealth","AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),
listclass=c(""), grupos=5,sinicio=1234,repe=20,nodesize=400,
mtry=16,ntree=300,replace=TRUE,sampsize=10000)
> medias3$modelo="bagging10000"
> medias4<-cruzadarfbin(data=data,vardep="HeartDisease",listconti=c("Age", "Diabetic.Yes
", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes",
"MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),
listclass=c(""), grupos=5,sinicio=1234,repe=20,nodesize=400,
mtry=16,ntree=300,replace=TRUE,sampsize=8000)
> medias4$modelo="bagging8000"
> medias5<-cruzadarfbin(data=data,vardep="HeartDisease",listconti=c("Age", "Diabetic.Yes
", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes",
"MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),
listclass=c(""), grupos=5,sinicio=1234,repe=20,nodesize=400,
mtry=16,ntree=300,replace=TRUE,sampsize=7000)
> medias5$modelo="bagging7000"
```

```
> medias6<-cruzadarfbin(data=data,vardep="HeartDisease",listconti=c("Age", "Diabetic.Yes
", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes",
"MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),
listclass=c(""), grupos=5,sinicio=1234,repe=20,nodesize=400,
mtry=16,ntree=300,replace=TRUE,sampsize=5000)
> medias6$modelo="bagging5000"
> medias7<-cruzadarfbin(data=data,vardep="HeartDisease",listconti=c("Age", "Diabetic.Yes
", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes",
"MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),
listclass=c(""), grupos=5,sinicio=1234,repe=20,nodesize=400,
mtry=16,ntree=300,replace=TRUE,Sampsize=3000)
> medias7$modelo="bagging3000"
> union1<-rbind(medias1,medias2,medias3,medias4,medias5,medias6,medias7)
> par(cex.axis=1.5,las=2)
> uni7 <- union1
> uni7$modelo <- with(uni7, reorder(modelo, auc, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni7,auc~modelo,main="AUC",col="pink")
> uni8 <- union1
> uni8$modelo <- with(uni8,reorder(modelo,tasa,mean))
> uni8$modelo <- with(uni8, reorder(modelo, tasa, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni8,auctasa~modelo,main="TASA DE FALLOS PONDERADA",col="pink")
```

Tuneo Random Forest

➔ **Tuneo Mtry (TABLA 30)**

```
> set.seed(12345)
> rfgrid<-expand.grid(mtry=c(3,4,5,6,7,8,9,10,11,12,13,14,15,16))
> control<-trainControl(method = "cv",number=4,savePredictions = "all",classProbs=TRUE)
> rf<- train(HeartDisease~Age+Diabetic.Yes+Stroke.Yes+Sex.Female+
GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+Smoking.Yes+KidneyDisease.Yes+Asth
ma.Yes+GenHealth.Fair+SkinCancer.Yes+
MentalHealth+AlcoholDrinking.Yes+DiffWalking.Yes+Race.Black,
data=data, method="rf",trControl=control,tuneGrid=rfgrid,
linout = FALSE,nodesize=400,replace=TRUE,
importance=TRUE)
> rf
```

➔ **Tuneo Ntree (ILUSTRACIÓN 16)**

```
> rfbis<-randomForest(HeartDisease~Age+Diabetic.Yes+Stroke.Yes+
Sex.Female+GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+
Smoking.Yes+KidneyDisease.Yes+Asthma.Yes+GenHealth.Fair+
SkinCancer.Yes+MentalHealth+AlcoholDrinking.Yes+DiffWalking.Yes+
Race.Black,data=data, mtry=4,ntree=2000,nodesize=400,replace=TRUE)
> plot(rfbis$err.rate[,1])
```

➔ **Tuneo Sampsize. Comparación a través de validación cruzada repetida. Boxplots tasa de fallos ponderada y AUC (ILUSTRACIÓN 17, ILUSTRACIÓN 18):**

```
> medias1<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10,
nodesize=400,mtry=4,ntree=500,replace=TRUE,sampsize=1000)
> medias1$modelo="rf4 1000"
> medias2<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=10,nodesize=400,mtry=4,ntree=500,replace=TRUE,
sampsize=3000)
> medias2$modelo="rf4 3000"
> medias3<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),
grupos=4,sinicio=1234,repe=10,nodesize=400, mtry=4,ntree=500,
replace=TRUE,sampsize=5000)
> medias3$modelo="rf4 5000"
> medias4<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
```

```
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=10,nodesize=400,mtry=4,ntree=500,replace=TRUE,
sampsize=7000)
> medias4$modelo="rf4 7000"
> medias5<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes",
"GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes", "DiffWalking.
Yes",
"Race.Black"),listclass=c(""),grupos=4,sinicio=1234,repe=10,
nodesize=400,mtry=4,ntree=500,replace=TRUE,sampsize=8000)
> medias5$modelo="rf4 8000"
> medias6<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=10,nodesize=400,mtry=4,ntree=500,replace=TRUE,
sampsize=10000)
> medias6$modelo="rf4 10000"
> medias7<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=10,nodesize=400,mtry=4,ntree=500,replace=TRUE,
sampsize=11250)
> medias7$modelo="rf4 11250"
> union1<-rbind(medias1,medias2,medias3,medias4,medias5,medias6,medias7)
> par(cex.axis=1.5,las=2)
> uni7 <- union1
> uni7$modelo <- with(uni7, reorder(modelo, auc, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni7,auc~modelo,main="AUC",col="pink")
> uni8 <- union1
> uni8$modelo <- with(uni8,reorder(modelo,tasa,mean))
> uni8$modelo <- with(uni8, reorder(modelo, tasa, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni8,auctasa~modelo,main="TASA DE FALLOS PONDERADA",col="pink")
```

Tuneo Gradient Boosting

➔ **Tuneo Ntree y shrinkage (ILUSTRACIÓN 19):**

```
> set.seed(12345)
> gbmgrid<-expand.grid(shrinkage=c(0.001, 0.01,0.03,0.05,0.1),
           n.minobsinnode=c(400),n.trees=c(100,1000,2000,5000,7000),
           interaction.depth=c(2))
> control<-trainControl(method = "cv",number=4,savePredictions = "all",
classProbs=TRUE)
> gbm<- train(HeartDisease~Age+Diabetic.Yes+Stroke.Yes+Sex.Female+
GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+Smoking.Yes+
KidneyDisease.Yes+Asthma.Yes+GenHealth.Fair+SkinCancer.Yes+MentalHealth+
AlcoholDrinking.Yes+DiffWalking.Yes+Race.Black,data=data,
method="gbm",trControl=control,tuneGrid=gbmgrid,distribution="bernoulli", bag.fraction=1
,verbose=FALSE)
> gbm
> plot(gbm)
```

➔ **Comparación a través de validación cruzada repetida. Boxplots tasa de fallos ponderada y AUC (ILUSTRACIÓN 20, ILUSTRACIÓN 21):**

```
> medias18<-cruzadagbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,repe=10,n.minobsi
nnode=400,shrinkage=0.1,n.trees=2000,
interaction.depth=2)
> medias18$modelo="gbm0.1 2000"
> medias19<-cruzadagbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),
grupos=4,sinicio=1234,repe=10,n.minobsinnode=400,shrinkage=0.05,
n.trees=1000,interaction.depth=2)
> medias19$modelo="gbm0.05 1000"
> medias20<-cruzadagbmbin(data=data, vardep="HeartDisease",
```

```
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent", "G
enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes",
"KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "
AlcoholDrinking.Yes", "DiffWalking.Yes","Race.Black"),listclass=c(""),grupos=4,sinicio=1
234,repe=10, n.minobsinnode=400,
shrinkage=0.01,n.trees=5000,interaction.depth=2)
> medias20$modelo="gbm0.01 5000"
> medias21<-cruzadagbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,repe=10,n.minobsi
nnode=400,shrinkage=0.03,n.trees=2000,
interaction.depth=2)
> medias21$modelo="gbm0.03 2000"
> union1<-rbind(medias18,medias19,medias20,medias21)
> par(cex.axis=1.5,las=2)
> uni7 <- union1
> uni7$modelo <- with(uni7, reorder(modelo, auc, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni7,auc~modelo,main="AUC",col="pink")
> uni8 <- union1
> uni8$modelo <- with(uni8,reorder(modelo,tasa,mean))
> uni8$modelo <- with(uni8, reorder(modelo, tasa, mean))
> par(cex.axis=0.8, las=2,mar=c(8,4,4,2)+0.1)
> boxplot(data=uni8,auctasa~modelo,main="TASA DE FALLOS PONDERADA",col="pink")
```

Tuneo Xgboost

➔ **Tuneo Nrounds y eta (ILUSTRACIÓN 22)**

```
> set.seed(12345)
> xgbmgrid<-expand.grid(min_child_weight=c(400),
eta=c(0.15,0.2,0.1,0.05,0.03,0.01), # learning rate
nrounds=c(500,1000,5000,7000,8000), # número de árboles
max_depth=6,gamma=0,colsample_bytree=1,subsample=1)
> control<-trainControl(method = "cv",number=4,
savePredictions = "all",classProbs=TRUE)
> xgbm<- train(HeartDisease~Age+Diabetic.Yes+Stroke.Yes+
Sex.Female+GenHealth.Excellent+GenHealth.VeryGood+
GenHealth.Good+Smoking.Yes+KidneyDisease.Yes+Asthma.Yes+
GenHealth.Fair+SkinCancer.Yes+MentalHealth+AlcoholDrinking.Yes+
DiffWalking.Yes+Race.Black,data=data,method="xgbTree",
trControl=control,tuneGrid=xgbmgrid,verbose=FALSE)
> xgbm
> plot(xgbm)
```

➔ **Comparación a través de validación cruzada repetida, Boxplots tasa de fallos ponderada y AUC (sin tunear Subsample y Colsample) (ILUSTRACIÓN 23, ILUSTRACIÓN 24):**

```
> medias30<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.05,nrounds=5000,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0)
> medias30$modelo="xgbm_1"
> medias31<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.15,nrounds=5000,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0)
> medias31$modelo="xgbm_2"
> medias32<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.15,nrounds=7000,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0)
> medias32$modelo="xgbm_3"
> medias33<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
```

```
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.15,nrounds=8000,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0)
> medias33$modelo="xgbm_4"
> union1<-rbind(medias30,medias31,medias32,medias33)
> ## Sacamos AUC
> uni7<-union1
> uni7$modelo <- with(uni7,reorder(modelo,auc, mean))
> par(cex.axis=0.8,las=2)
> boxplot(data=uni7, auc~modelo,main="AUC",col="pink")
> ## Sacamos tasa de fallos ponderada
> uni8 <- union1
> uni8$modelo <- with(uni8,reorder(modelo,tasa,mean))
> par(cex.axis=0.8,las=2)
> boxplot(data=uni8, tasa~modelo,main="TASA DE FALLOS PONDERADA",col="pink")
```

➔ **Comparación a través de validación cruzada repetida, Boxplots tasa de fallos ponderada y AUC . Modelos xgbm_1 y los tres en los que se tunean Colsample y Subsample (ILUSTRACIÓN 25, ILUSTRACIÓN 26):**

```
> medias30<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.05,nrounds=5000,max_depth=6,
gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0)
> medias30$modelo="xgbm_1"
> medias34<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.05,nrounds=5000,max_depth=6,
gamma=0,colsample_bytree=0.25,subsample=1,alpha=0,lambda=0)
> medias34$modelo="xgbm_5"
> medias35<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.05,nrounds=5000,max_depth=6,
gamma=0,colsample_bytree=1,subsample=0.5,alpha=0,lambda=0)
> medias35$modelo="xgbm_6"
> medias36<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female",
"GenHealth.Excellent", "GenHealth.VeryGood", "GenHealth.Good",
"Smoking.Yes", "KidneyDisease.Yes", "Asthma.Yes", "GenHealth.Fair",
"SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,sinicio=1234,
repe=10,min_child_weight=400,eta=0.05,nrounds=5000,max_depth=6,
gamma=0,colsample_bytree=0.25,subsample=0.5,alpha=0,lambda=0)
> medias36$modelo="xgbm_7"
> union1<-rbind(medias30,medias34,medias36)
> ## Sacamos AUC
> uni7<-union1
> uni7$modelo <- with(uni7,reorder(modelo,auc, mean))
> par(cex.axis=0.8,las=2)
> boxplot(data=uni7, auc~modelo,main="AUC",col="pink")
> ## Sacamos tasa de fallos ponderada
> uni8 <- union1
> uni8$modelo <- with(uni8,reorder(modelo,tasa,mean))
> par(cex.axis=0.8,las=2)
> boxplot(data=uni8, tasa~modelo,main="TASA DE FALLOS PONDERADA",col="pink")
```

Tuneo SVM_Lineal (ILUSTRACIÓN 27)

```
> set.seed(12345)
> SVMgrid<-expand.grid(C=c(0.01,0.02,0.03,0.04,0.05,0.8,0.1,0.2,0.3,
0.4,0.5,0.6,0.8,1))
> control<-trainControl(method = "cv",number=4,savePredictions = "all")
> SVM<- train(data=data,HeartDisease~Age+Diabetic.Yes+Stroke.Yes+
Sex.Female+GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+
Smoking.Yes+KidneyDisease.Yes+Asthma.Yes+GenHealth.Fair+SkinCancer.Yes+MentalHealth+Alco
holDrinking.Yes+DiffWalking.Yes+Race.Black,
method="svmLinear",trControl=control,tuneGrid=SVMgrid,verbose=FALSE)
```

Tuneo SVM_Polinomial (ILUSTRACIÓN 28)

```
> set.seed(12345)
> SVMgrid<-expand.grid(C=c(0.01,0.05,0.1,0.2,0.5,1,2,5,10),
```

```
degree=c(2),scale=c(0.1,0.5,1,2,5))
> control<-trainControl(method = "cv", number=4,
savePredictions = "all")
> SVM<- train(data=data,HeartDisease~Age+Diabetic.Yes+Stroke.Yes+
Sex.Female+GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+
Smoking.Yes+KidneyDisease.Yes+Asthma.Yes+GenHealth.Fair+
SkinCancer.Yes+MentalHealth+AlcoholDrinking.Yes+DiffWalking.Yes+
Race.Black,method="svmPoly",trControl=control,tuneGrid=SVMgrid,
verbose=FALSE)
> dat<-as.data.frame(SVM$results)
> library(ggplot2)
> # PLOT DE DOS VARIABLES CATEGÓRICAS, UNA CONTINUA
> ggplot(dat, aes(x=factor(C), y=Accuracy,
color=factor(degree),pch=factor(scale))) +
geom_point(position=position_dodge(width=0.5),size=3)
```

Tuneo SVM_Radial (ILUSTRACIÓN 29)

```
> set.seed(12345)
> SVMgrid<-expand.grid(C=c(0.5,1,2,5,10,30),
sigma=c(0.0001,0.005,0.01,0.05))
> control<-trainControl(method = "cv", number=4,
savePredictions = "all")
> SVM<- train(data=data,HeartDisease~Age+Diabetic.Yes+Stroke.Yes+
Sex.Female+GenHealth.Excellent+GenHealth.VeryGood+GenHealth.Good+
Smoking.Yes+KidneyDisease.Yes+Asthma.Yes+GenHealth.Fair+
SkinCancer.Yes+MentalHealth+AlcoholDrinking.Yes+DiffWalking.Yes+
Race.Black,method="svmRadial",trControl=control,tuneGrid=SVMgrid,
verbose=FALSE)
> dat<-as.data.frame(SVM$results)
> ggplot(dat, aes(x=factor(C), y=Accuracy,
color=factor(sigma)))+ geom_point(position=position_dodge(width=0.5),
size=3)
```

Comparación de modelos. AUC y tasa de fallos ponderada (ILUSTRACIÓN 30, ILUSTRACIÓN 31)

```
> medias12<-cruzadaavnnetbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth",
"AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234,repe=10, size=c(4),decay=c(0.01),repeticiones=5,itera=300)
> medias12$modelo="Red"
> medias17<-cruzadalogistica(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"G enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking. Yes", "Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10)
> medias17$modelo="Logistica"
> medias1<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes ", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent"
, "GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"), listclass=c(""), grupos=4,sinicio=1234,repe=10,
nodesize=400, mtry=16,ntree=300,replace=TRUE,sampsize=7000)
> medias1$modelo="Bagging"
> medias3<-cruzadarfbin(data=data,vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth", "AlcoholDrinking.Yes",
"DiffWalking.Yes", "Race.Black"),listclass=c(""), grupos=4,sinicio=1234,repe=10,
nodesize=400, mtry=4,ntree=500, replace=TRUE,sampsize=5000)
> medias3$modelo="RF"
> medias16<-cruzadagbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"G enHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth",
" AlcoholDrinking.Yes", "DiffWalking.Yes","Race.Black"),listclass=c(""),grupos=4,
sinicio=1 234,repe=10, n.minobsinnode=400, shrinkage=0.01,n.trees=5000,
interaction.depth=2)
> medias16$modelo="gbm"
> medias36<-cruzadaxgbmbin(data=data, vardep="HeartDisease",
listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth",
"AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234, repe=10,min_child_weight=400,eta=0.05,nrounds=5000,max_depth=6, gamma=0,
colsample_bytree=0.25,subsample=0.5,alpha=0,lambda=0)
> medias36$modelo="xgbm"
> medias50 <- cruzadasSVMbin(data=data, vardep="HeartDisease",
Listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
```

```
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth",
"AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234, repe=10, C=0.3)
> medias50$modelo="SVM_Lineal"
> medias60 <- cruzadasSVMbinPoly(data=data, vardep="HeartDisease",
Listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth",
"AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234, repe=10, C=5,degree=2,scale=1)
> medias60$modelo="SVM_Poly"
> medias70 <- cruzadasSVMbinRBF(data=data, vardep="HeartDisease",
Listconti=c("Age", "Diabetic.Yes", "Stroke.Yes", "Sex.Female", "GenHealth.Excellent",
"GenHealth.VeryGood", "GenHealth.Good", "Smoking.Yes", "KidneyDisease.Yes",
"Asthma.Yes", "GenHealth.Fair", "SkinCancer.Yes", "MentalHealth",
"AlcoholDrinking.Yes", "DiffWalking.Yes", "Race.Black"),listclass=c(""),grupos=4,
sinicio=1234, repe=10, C=2,sigma=0.005)
> medias70$modelo="SVM_RBF"
> union1<-rbind(medias12,medias17,medias1,medias3,medias16,medias36,medias50,medias60,
medias70)
> ## Sacamos AUC
> uni7<-union1
> uni7$modelo <- with(uni7,reorder(modelo,auc, mean))
> par(cex.axis=0.8,las=2)
> boxplot(data=uni7, auc~modelo,main="AUC",col="pink")
> ## Sacamos tasa de fallos ponderada
> uni8 <- union1
> uni8$modelo <- with(uni8,reorder(modelo,tasa,mean))
> par(cex.axis=0.8,las=2)
> boxplot(data=uni8, tasa~modelo,main="TASA DE FALLOS PONDERADA",col="pink")
```

Construcción Ensamblados.

```
> grupos<-4
> sinicio<-1234
> repe<-10
# REGRESIÓN LOGÍSTICA
> medias1<-cruzadalogistica(data=archivo,vardep=vardep,
listconti=listconti,listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe)
> medias1bis<-as.data.frame(medias1[1])
> medias1bis$modelo<-"Logistica"
> predi1<-as.data.frame(medias1[2])
> predi1$logi<-predi1$Yes
# RED NEURONAL
> medias2<-cruzadaavnnetbin(data=archivo,vardep=vardep,
listconti=listconti,listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe,
size=c(4),decay=c(0.01),repeticiones=5,itera=300)
> medias2bis<-as.data.frame(medias2[1])
> medias2bis$modelo<-"avnnet"
> predi2<-as.data.frame(medias2[2])
> predi2$avnnet<-predi2$Yes
# GRADIENT BOOSTING
> medias3<-cruzadagbmbin(data=archivo,vardep=vardep,
listconti=listconti,listclass=listclass,grupos=grupos,sinicio=sinicio,repe=repe,
n.minobsinnode=400,shrinkage=0.01,n.trees=5000,
interaction.depth=2)
> medias3bis<-as.data.frame(medias3[1])
> medias3bis$modelo<-"gbm"
> predi3<-as.data.frame(medias3[2])
> predi3$gbm<-predi3$Yes
# SVM LINEAL
> medias4<-cruzadaSVMbin(data=archivo,vardep=vardep,
listconti=listconti,listclass=listclass,grupos=grupos,
sinicio=sinicio,repe=repe,C=0.3)
> medias4bis<-as.data.frame(medias4[1])
> medias4bis$modelo<-"svmLinear"
> predi4<-as.data.frame(medias4[2])
> predi4$svmLinear<-predi4$Yes
# SVM RBF
> medias5<-cruzadaSVMbinRBF(data=archivo,vardep=vardep,
listconti=listconti,listclass=listclass,grupos=grupos,
sinicio=sinicio,repe=repe,C=2,sigma=0.005)
> medias5bis<-as.data.frame(medias5[1])
> medias5bis$modelo<-"svmRadial"
> predi5<-as.data.frame(medias5[2])
> predi5$svmRadial<-predi5$Yes
> union1<-rbind(medias1bis,medias2bis,medias3bis,medias4bis,medias5bis)
> unipredi<-cbind(predi1,predi2,predi3,predi4,predi5)
> unipredi<- unipredi[, !duplicated(colnames(unipredi))]

## SE CONSTRUYEN TODOS LOS ENSAMBLADOS
> unipredi$predi9<-(unipredi$logi+unipredi$avnnet)/2
> unipredi$predi10<-(unipredi$avnnet+unipredi$svmLinear)/2
> unipredi$predi11<-(unipredi$avnnet+unipredi$svmRadial)/2
```

```
> unipredi$predi12<-(unipredi$avnnet+unipredi$gbm)/2
> unipredi$predi13<-(unipredi$logi+unipredi$svmLinear)/2
> unipredi$predi14<-(unipredi$logi+unipredi$svmRadial)/2
> unipredi$predi15<-(unipredi$logi+unipredi$gbm)/2
> unipredi$predi16<-(unipredi$gbm+unipredi$svmLinear)/2
> unipredi$predi17<-(unipredi$gbm+unipredi$svmRadial)/2
> unipredi$predi18<-(unipredi$svmLinear+unipredi$svmRadial)/2
> unipredi$predi31<-(unipredi$avnnet+unipredi$logi+unipredi$gbm)/3
> unipredi$predi32<-(unipredi$avnnet+unipredi$logi+unipredi$svmLinear)/3
> unipredi$predi33<-(unipredi$avnnet+unipredi$logi+unipredi$svmRadial)/3
> unipredi$predi34<-(unipredi$avnnet+unipredi$gbm+unipredi$svmLinear)/3
> unipredi$predi35<-(unipredi$avnnet+unipredi$gbm+unipredi$svmRadial)/3
> unipredi$predi36<-(unipredi$avnnet+unipredi$svmLinear+unipredi$svmRadial)/3
> unipredi$predi37<-(unipredi$logi+unipredi$gbm+unipredi$svmLinear)/3
> unipredi$predi38<-(unipredi$logi+unipredi$gbm+unipredi$svmRadial)/3
> unipredi$predi39<-(unipredi$logi+unipredi$svmLinear+unipredi$svmRadial)/3
> unipredi$predi40<-(unipredi$gbm+unipredi$svmLinear+unipredi$svmRadial)/3
> unipredi$predi64<-(unipredi$avnnet+unipredi$logi+unipredi$gbm+unipredi$svmLinear)/4
> unipredi$predi65<-(unipredi$avnnet+unipredi$logi+unipredi$gbm+unipredi$svmRadial)/4
> unipredi$predi66<-(unipredi$avnnet+unipredi$logi+unipredi$svmLinear+unipredi$svmRadial
)/4
> unipredi$predi67<-(unipredi$avnnet+unipredi$gbm+unipredi$svmLinear+unipredi$svmRadial)
/4
> unipredi$predi68<-(unipredi$logi+unipredi$gbm+unipredi$svmLinear+unipredi$svmRadial)/4
> unipredi$predi69<-(unipredi$logi+unipredi$avnnet+unipredi$gbm+unipredi$svmRadial+unipr
edi$svmLinear)/5
```

Comparación Métodos de Ensamblado. AUC y tasa de fallos ponderada (ILUSTRACIÓN 32, ILUSTRACIÓN 33)

```
> listado<-c("predi9", "predi10", "predi11", "predi12", "predi13", "predi14", "predi15",
"predi16", "predi17", "predi18", "predi31", "predi32", "predi33", "predi34", "predi35",
"predi36", "predi37", "predi38", "predi39", "predi40", "predi64", "predi65", "predi66",
"predi67", "predi68", "predi69")
> N <- 319795
> p1 <- (7500/27373)
> p0 <- (1250/48737)
> tasafallos<-function(x,y) {
   confu<-confusionMatrix(x,y)
   tasa<-3*((confu[[2]][3])/(p1*N))+((confu[[2]][2])/(p0*N))
   return(tasa)
 }
> auc<-function(x,y) {
   curvaroc<-roc(response=x,predictor=y)
   auc<-curvaroc$auc
   return(auc)
 }
> repeticiones<-nlevels(factor(unipredi$Rep))
> unipredi$Rep<-as.factor(unipredi$Rep)
> unipredi$Rep<-as.numeric(unipredi$Rep)
> medias0<-data.frame(c())
> for (prediccion in listado)
 {
   unipredi$proba<-unipredi[,prediccion]
   unipredi[,prediccion]<-ifelse(unipredi[,prediccion]>0.5,"Yes","No")
   for (repe in 1:repeticiones)
   {
     paso <- unipredi[(unipredi$Rep==repe),]
     pre<-factor(paso[,prediccion])
     archi<-paso[,c("proba","obs")]
     archi<-archi[order(archi$proba),]
     obs<-paso[,c("obs")]
     tasa=tasafallos(pre,obs)
     t<-as.data.frame(tasa)
     t$modelo<-prediccion
     auc<-suppressMessages(auc(archi$obs,archi$proba))
     t$auc<-auc
     medias0<-rbind(medias0,t)
   }
 }

## TASA DE FALLOS PONDERADA
> medias0$modelo <- with(medias0,reorder(modelo,tasa, mean))
> par(cex.axis=0.7,las=2)
> boxplot(data=medias0,tasa~modelo,col="pink", main='TASA DE FALLOS PONDERADA')

## AUC
> medias0$modelo <- with(medias0,reorder(modelo,auc, mean))
> par(cex.axis=0.7,las=2)
> boxplot(data=medias0,auc~modelo,col="pink", main='AUC')
```

Comparación Modelos y Ensamblados. AUC y tasa de fallos ponderada (ILUSTRACIÓN 34, ILUSTRACIÓN 35)

```
> listado<- c("logi", "avnnet", "gbm","svmLinear",  "svmRadial", "predi9", "predi10",
"predi11", "predi12", "predi13", "predi14", "predi15", "predi16", "predi17", "predi18",
"predi31", "predi32", "predi33", "predi34", "predi35", "predi36", "predi37", "predi38",
"predi39", "predi40", "predi64", "predi65", "predi66", "predi67", "predi68", "predi69")
> N <- 319795
> p1 <- (7500/27373)
> p0 <- (1250/48737)
> tasafallos<-function(x,y) {
   confu<-confusionMatrix(x,y)
   tasa<-3*((confu[[2]][3])/(p1*N))+((confu[[2]][2])/(p0*N))
   return(tasa)
 }
> auc<-function(x,y) {
   curvaroc<-roc(response=x,predictor=y)
   auc<-curvaroc$auc
   return(auc)
 }
> repeticiones<-nlevels(factor(unipredi$Rep))
> unipredi$Rep<-as.factor(unipredi$Rep)
> unipredi$Rep<-as.numeric(unipredi$Rep)
> medias0<-data.frame(c())
> for (prediccion in listado)
 {
   unipredi$proba<-unipredi[,prediccion]
   unipredi[,prediccion]<-ifelse(unipredi[,prediccion]>0.5,"Yes","No")
   for (repe in 1:repeticiones)
   {
     paso <- unipredi[(unipredi$Rep==repe),]
     pre<-factor(paso[,prediccion])
     archi<-paso[,c("proba","obs")]
     archi<-archi[order(archi$proba),]
     obs<-paso[,c("obs")]
     tasa=tasafallos(pre,obs)
     t<-as.data.frame(tasa)
     t$modelo<-prediccion
     auc<-suppressMessages(auc(archi$obs,archi$proba))
     t$auc<-auc
     medias0<-rbind(medias0,t)
   }
 }

## TASA DE FALLOS PONDERADA
> medias0$modelo <- with(medias0,reorder(modelo,tasa, mean))
> par(cex.axis=0.7,las=2)
> boxplot(data=medias0,tasa~modelo,col="pink", main='TASA DE FALLOS PONDERADA')

## AUC
> medias0$modelo <- with(medias0,reorder(modelo,auc, mean))
> par(cex.axis=0.7,las=2)
> boxplot(data=medias0,auc~modelo,col="pink", main='AUC')
```