FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2018/2019

Trabajo de Fin de Máster

TITULO: *El abandono en la Facultad de Estadística de la UCM. Prediciendo para mejorar*

***Alumno*:** Jorge Blanco Iglesias

***Tutor*:** Javier Portela García-Miguel

Septiembre de 2019

UNIVERSIDAD COMPLUTENSE
MADRID

A mi familia.

Siempre en

apoyo constante.

Resumen

El abandono de los programas de grado constituye uno de los problemas centrales de la gestión universitaria actual. Se puede abordar y tratar de explicar desde perspectivas descriptivas, pero sin lugar a duda constituye un enfoque diferente e interesante tratar de predecirlo antes de que se produzca. En este trabajo, tomando como punto de partida los datos académicos del Grado en Estadística Aplicada de la Universidad Complutense de Madrid, se llevarán a cabo labores de predicción basadas en algoritmos de Machine Learning que ayuden a definir si, efectivamente, es posible predecir el abandono universitario en base al rendimiento académico y qué asignaturas podrían ejercer de principales predictores del abandono tras el primer curso académico.

Palabras clave: Abandono universitario; Predicción del Abandono; Machine Learning; Grado en Estadística; Universidad Complutense de Madrid

Abstract

The dropout from Bachelor studies is one of the main problems the university management has to face nowadays. It can be dealt from several and different descriptive perspectives, even though it might be more interesting to approach it by forecasting it before it happens. In this paper, taking into account the academic records and data from the Bachelor's Degree in Applied Statistics, there will be developed forescast tasks based in Machine Learning algorithms that would help to define whether it is actually posible to forecast the university dropout base don the academic records and whic subjects or courses might work as the main predictors after the first academic year.

Keywords: University Dropout; Dropout Forecasting; Machine Learning; Degree in Statistics; Universidad Complutense de Madrid

ÍNDICE

1. INTRODUCCIÓN

La universidad, en su concepción más amplia y extensa, es actualmente considerada como un espacio de conocimiento, innovación e integración de diversos estamentos que trabajan con un objetivo común: colaborar en el desarrollo personal y profesional de los impulsores de la sociedad del futuro, a la par que se genera el conocimiento necesario para hacerla posible.

No obstante, esta visión resulta relativamente reciente, ya que la universidad española ha experimentado una profunda transformación en los últimos 40 años, donde ha transitado desde una lógica de espacio producido y orientado por y para las élites, a una lógica de espacio de masas (Elias Andreu y Daza Pérez, 2014), donde la inclusión y la provisión de oportunidades para todos los alumnos y alumnas, sin importar origen ni condición social, se ha articulado como el mantra fundamental del modelo actual.

Este cambio de paradigma introdujo toda una serie de nuevas problemáticas que, si bien ya gozaban de una cierta importancia e interés, no representaban un aspecto fundamental de la política universitaria. Entre estas cuestiones, se encuentra, como no podía ser de otra manera, el abandono universitario, dado el volumen nada despreciable que supone sobre el porcentaje de matriculados para las distintas cohortes de alumnos que han venido entrando en la universidad española en los últimos años.

El abandono universitario, tal y como lo define el Ministerio de Educación y Formación Profesional, se produce en el momento en que un alumno que se ha matriculado en un curso escolar, no lo vuelve a hacer en el periodo correspondiente a los dos años académicos posteriores a dicha última matriculación (SIIU, n.d.). Aunque este fenómeno no resulta exclusivo del primer año de estudio de una materia o grado, sí que es cierto que es habitualmente tras ese primer año cuando este se produce con mayor intensidad, produciéndose un goteo paulatino desde el segundo año de estudio hasta configurar el grueso de alumnos de una cohorte de entrada que nunca finaliza un grado.

Este abandono, una vez se produce, puede suponer o bien abandono completo del Sistema Universitario Español (SUE), o bien simplemente la continuación dentro del propio sistema, pero en otro programa, ya sea en la misma universidad o en cualquier otra dentro del SUE (SIIU, n.d.). No obstante, con datos aislados de una única facultad, a menudo resulta difícil discernir si el alumno o alumna continúa dentro del propio sistema, dada la inexistencia de datos armonizados por parte de las autoridades competentes.

Con todas estas cuestiones en mente, esta investigación, objeto del Trabajo de Fin de Máster (TFM) para el Máster de Minería de Datos de Inteligencia de Negocio de la Universidad Complutense de Madrid (UCM), persigue realizar una caracterización del abandono en el Grado en Estadística Aplicada de la Facultad de Estudios Estadísticos de la UCM, para la posterior elaboración de modelos de predicción que permitan identificar las potenciales situaciones de abandono en dicho grado, basadas en el rendimiento de los alumnos y alumnas del grado durante el año académico de ingreso.

Para ello, se dispone de un conjunto de datos de la matriculación y el rendimiento académico de todos los estudiantes del programa de grado cuya cohorte de entrada está situada entre los cursos académicos 2009/2010 y 2017/2018, recopilados por el Centro de Inteligencia Institucional (CII - SIDI) de la universidad y proporcionados por el Observatorio del Estudiante, unidad de reciente creación y cuyo objetivo consiste en llevar “a cabo estudios de interés sobre los estudiantes universitarios, obteniendo una imagen lo más fehaciente de la realidad universitaria” (“Observatorio del Estudiante - Universidad Complutense de Madrid,” n.d.).

Así, en primer lugar, se expondrá la relevancia del tema y la justificación del mismo, seguido de un breve marco teórico, para posteriormente abordar los objetivos del proyecto en relación a la cuestión a tratar. A continuación, se expondrá la metodología empleada para la consecución del proyecto, para posteriormente proceder a la elaboración y ajuste de modelos orientados a la predicción del abandono y finalizar con las conclusiones que se puedan desprender del desarrollo del trabajo.

El resultado último esperado de esta investigación, realizada en colaboración con el Observatorio del Estudiante de la UCM, consiste en la elaboración de un modelo que, en base a la información del rendimiento académico de cada alumno durante el primer curso, sea capaz de predecir si se va a producir abandono temprano o no. Se espera también que todo este desarrollo se constituya como el génesis de una herramienta que pueda resultar de utilidad a la universidad para alertar sobre potenciales situaciones de abandono y contribuir a desarrollar políticas internas específicas para dar respuesta a estas antes de que se produzcan.

La novedad de este proyecto reside en que, aunque se han realizado experiencias similares en otras universidades, como es el caso de la Universidad de Barcelona (Rovira et al., 2017), es la primera vez, que se tenga referencia, que se va a llevar a cabo algo parecido no solo en la facultad, sino a nivel de toda la UCM, por lo que se espera que después de la finalización de esta investigación, la herramienta tenga un recorrido mayor y cristalice en una propuesta concreta para la universidad.

2. RELEVANCIA DEL TEMA Y JUSTIFICACIÓN

En este apartado, se va a realizar un recorrido por el interés que se desprende del estudio del abandono en las universidades públicas de la Comunidad de Madrid, para posteriormente descender al nivel de la UCM y presentar la justificación de por qué centrarse específicamente en el Grado en Estadística Aplicada.

Así, ¿por qué resulta relevante realizar una aproximación al abandono en la Universidad Complutense de Madrid inicialmente? Tomando una perspectiva general, la realidad del abandono presenta implicaciones en, al menos, tres dimensiones de la esfera universitaria (Ruè, 2014):

- **Económica:** Cuando se menta el abandono, uno de los ejes principales que suele ordenar el debate se sitúa en torno a la eficiencia de la inversión pública que se realiza en los y las universitarias, y cómo esta se traduce en términos de retorno a la sociedad. Los recursos son limitados, y cada alumno que no acaba un grado, supone un porcentaje de financiación que, en el largo plazo, no se verá revertido de forma explícita en la generación de riqueza asociada a dicha formación.
- **Prestigio institucional:** En un contexto de competencia entre universidades, donde la entrada de actores privados ha vuelto mucho más exigente la diversidad de la oferta formativa disponible y ha dispersado la demanda, ningún ente contempla como positivo y en términos estratégicos presentar altas tasas de abandono, ya que puede generar incertidumbre y miedo a los potenciales alumnos que se encuentren contemplando la posibilidad de cursar sus estudios en alguno de estos centros.
- **Condiciones del alumnado:** Desde un punto de vista exclusivamente de un alumno o alumna que cursa un programa universitario y lo lleva a término, el hecho de que una parte importante de sus compañeros se vayan descolgando y no acaben puede representar una desventaja en términos de percepción de la utilidad del grado en cuestión, interrumpiendo el destino de recursos a los efectos de continuación de la enseñanza del mismo.

Cada una de estas tres dimensiones tiene una correspondencia con distintos niveles de las estructuras sociales, cuya lectura de la problemática difiere en la esencia, pero donde el rango de respuestas o soluciones canalizan hacia un orden de cosas común: dar otro paso más en pro de la calidad general universitaria.

Una de las maneras de conseguir obtener respuestas a las dificultades que plantean estos niveles consiste en caracterizar el problema, analizar tendencias y buscar áreas de oportunidad de mejora a través de la colaboración entre los estamentos de la comunidad universitaria.

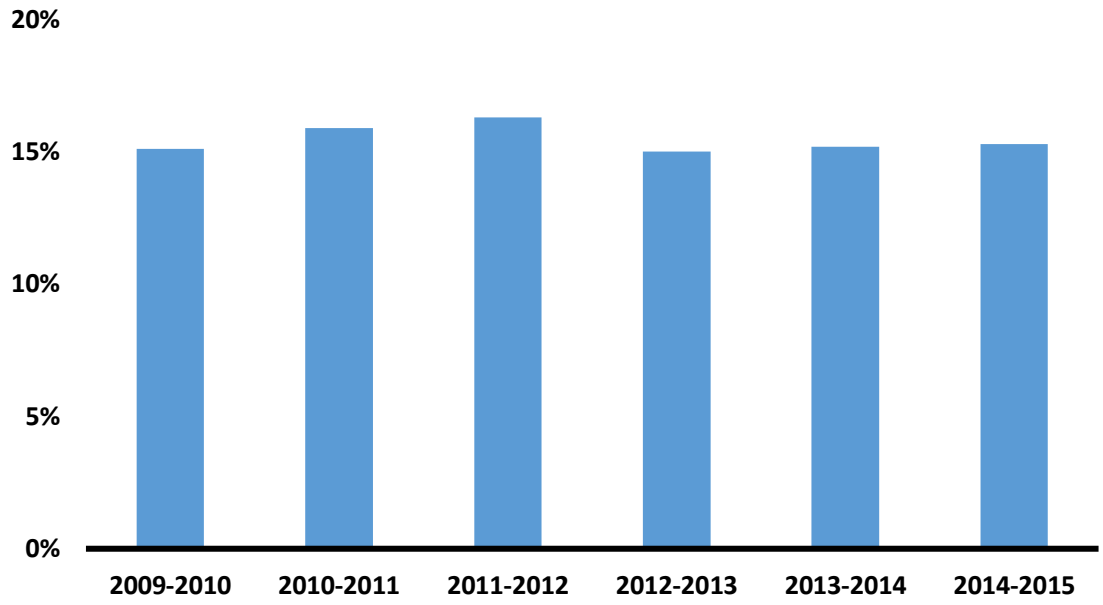
En la UCM, este último aspecto resulta crucial dadas las dimensiones de la institución en términos de trabajadores, alumnos, oferta de grados y volumen de actividad investigadora, donde la mejora en uno de los aspectos planteados puede tener una traducción directo en el resto de niveles a través de la generación de una inercia positiva.

2.1 ABANDONO EN LA COMUNIDAD DE MADRID: SITUACIÓN DE LAS UNIVERSIDADES PÚBLICAS

La aprobación del Plan Bolonia y su posterior implementación, de manera obligatoria a partir del año 2010, de grados, dobles grados y másteres como sustitutos de las antiguas diplomaturas, licenciaturas y planes de ingeniería técnica y superior, impuso un cambio en el panorama universitario sin precedentes. Algunas de sus principales virtudes, como la evaluación continua o la atención más personalizada a los alumnos, tenían la ocasión de probar su eficacia a pesar de las reticencias mostradas por los distintos estamentos de la comunidad universitaria.

Un primer acercamiento a la problemática se puede realizar para el conjunto de universidades públicas de la Comunidad de Madrid.

Figura 1. Evolución del porcentaje de abandono, universidades públicas CAM. Cohortes entrada entre cursos 2009-2010 y 2014-2015



Fuente: SIIU

La figura 1 provee de una panorámica sobre el estado del abandono universitario en el período seleccionado. Esencialmente, se puede observar cómo el abandono presenta unas cifras estables a lo largo del tiempo, oscilando entre el 15% y el 17%.

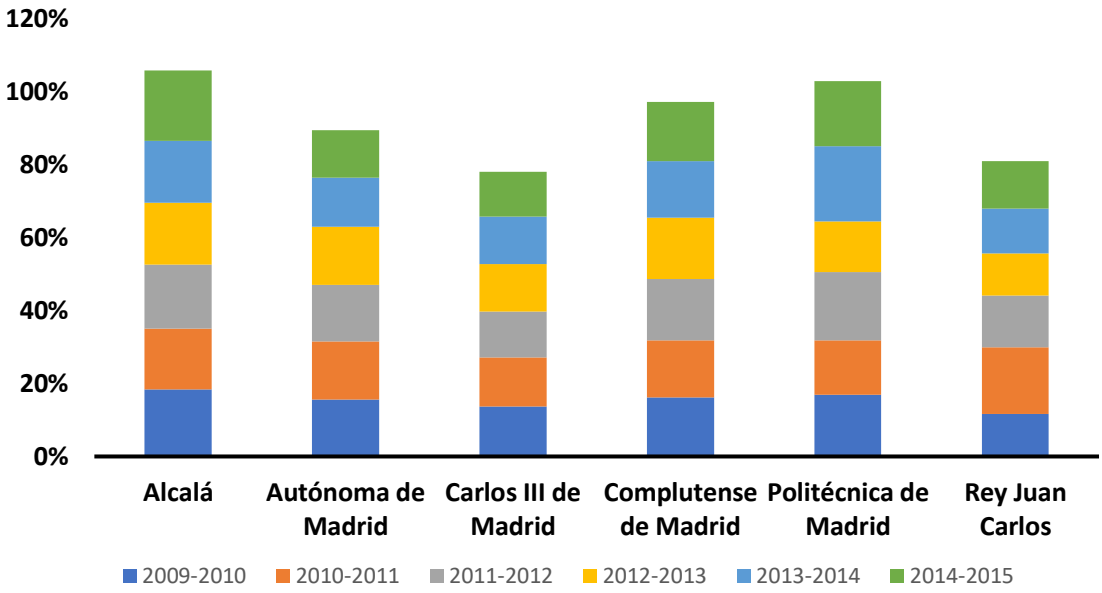
Estas cifras hacen pensar que la situación del abandono se encuentra ciertamente estabilizada en unas cifras que no varían demasiado entre años, aunque sí que parece que, para las dos últimas cohortes de entrada, 2012-2013 y 2013-2014, su intensidad podría resultar algo menor, del orden de 1,5 puntos menos que respecto a 2010-2011 y 2011-2012.

Aunque sin ser demasiado significativa esta diferencia, y a falta de tener una serie temporal con mayor alcance, se podría hablar de un ligero descenso en el abandono, posiblemente producido por un efecto de aprendizaje y rodaje tanto por parte de las

universidades, que entraron en dinámicas más estables y ya habían pasado un período de adaptación a los nuevos planes de estudios, como por parte de los y las estudiantes, donde existía un mayor grado de conocimiento y definición de la realidad de los planes de estudio.

Cuando se observa la cuestión del abandono para cada una de las universidades públicas – UCM, Autónoma, Carlos III, Rey Juan Carlos, Alcalá y Politécnica – se puede ver cómo este indicador no se distribuye homogéneamente entre las mismas.

Figura 2. Porcentaje acumulado de abandono, universidades públicas CAM. Cohortes entrada entre cursos 2009-2010 y 2014-2015



Fuente: SIIU

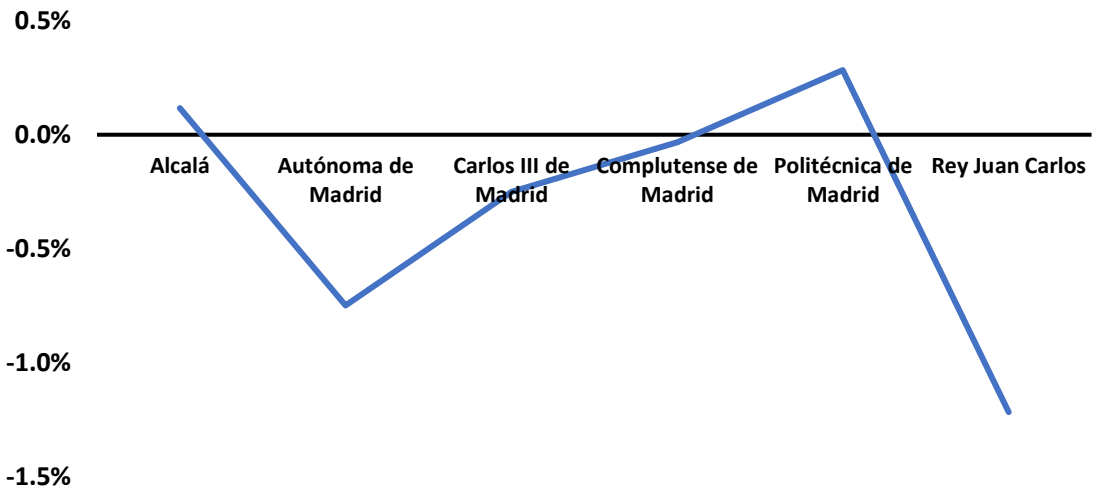
La figura 2, muestra el estado del abandono y su evolución en las citadas universidades públicas. Como era de esperar, entre la universidad con cifras más altas se encuentra la Universidad Politécnica de Madrid (UPM), cuyas enseñanzas mayoritarias son de Ingeniería y Arquitectura, las habitualmente percibidas como de mayor dificultad y donde existe un mayor porcentaje de alumnos y alumnas que abandonan.

Sin embargo, llama la atención que el primer lugar global lo copa la Universidad de Alcalá de Henares (UAH) y, en el tercer lugar, no muy lejos de la UPM, se puede encontrar a la UCM. Este hecho resulta llamativo en tanto que ambas universidades tienen un perfil de planes de estudios más generalista y heterogéneo, por lo que no tendría sentido achacar la problemática del abandono exclusivamente a la dificultad de las ingenierías.

Frente a estas tres universidades, se encuentran la Universidad Autónoma de Madrid (UAM), la Universidad Carlos III (UC3M) y la Universidad Rey Juan Carlos (URJC), cuyos niveles de abandono, estables en torno al 13%-14% de media cada año, resultan sensiblemente inferiores.

De la figura 1 se desprendía que en los últimos tres años se podía haber producido un ligero descenso del abandono en los últimos tres años de los que existen datos. La figura 3 provee de una panorámica sobre esta cuestión respecto a las universidades elegidas.

Figura 3. Comportamiento del abandono por universidad en los últimos tres años respecto a la media del periodo entre 2009-2010 y 2014-2015



Fuente: SIIU

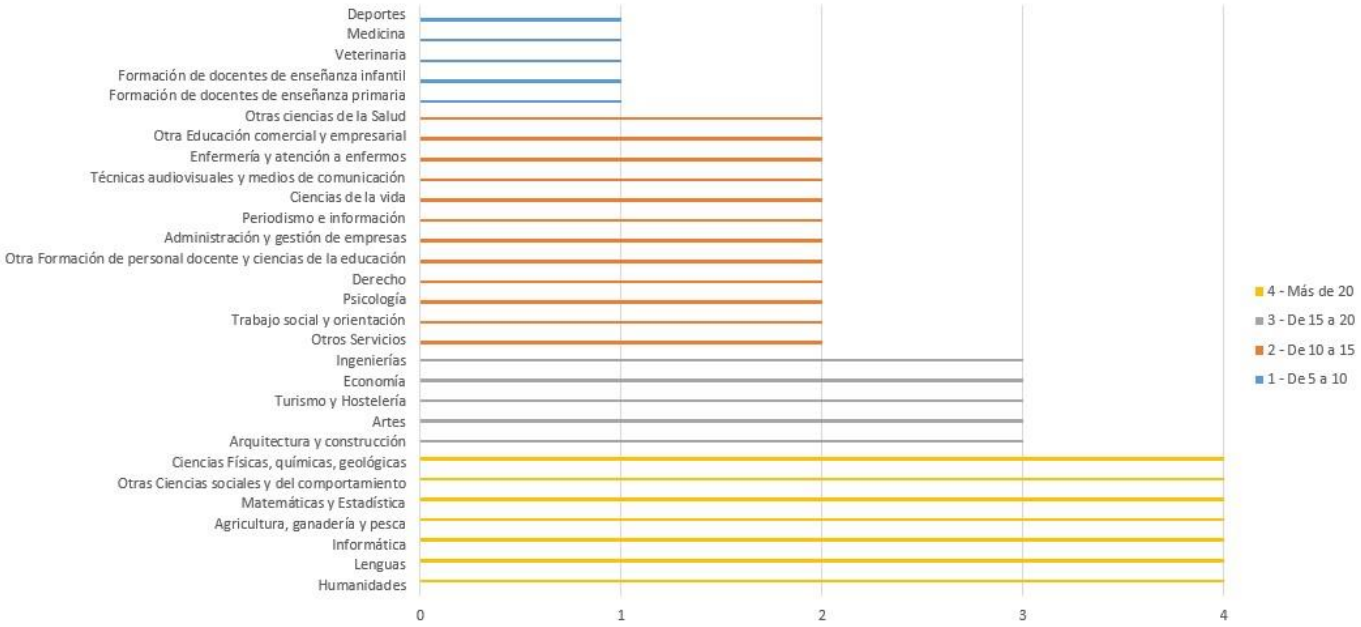
En los últimos tres años, tan solo la UAH y la UPM han aumentado sus tasas de abandono, pero en unas cifras tan cercanas a cero (0,12% y 0.28% respectivamente), que se podrían considerar prácticamente residuales. Por el lado de las universidades que sí que se han encontrado una reducción del abandono, destacan tanto la UAM, como especialmente la URJC, con un descenso respectivo del 0,75% y el 1,22%.

Tras un acercamiento global a la cuestión del abandono en las universidades públicas de la Comunidad de Madrid, el siguiente paso natural consiste en centrar el foco que se produce por áreas de conocimiento y, más específicamente, por materias, lo cual permitirá entender aquella donde el abandono se produce con mayor intensidad.

En la Comunidad de Madrid, dada la cantidad de oferta existente por parte de las seis universidades públicas, se pueden encontrar la mayoría, sino todas, de las materias que prevé el Sistema Universitario Español, que es lo que presenta la figura 4.

Para hacer más comprensible un gráfico que aborda tantas modalidades de estudios, se ha tomado el promedio de abandono en los años objeto de análisis, y se ha discretizado en cuatro categorías que permiten intuir la anteriormente comentada intensidad del abandono: de 5% a 10% de tasa de abandono; de 10% a 15%; de 15% a 20%; más de 20%.

Figura 4. Intensidad del abandono por materias para las universidades públicas de la CAM. Promedio del periodo entre 2009-2010 y 2014-2015



Fuente: SIIU

El resultado presentado en la figura 4 muestra resultados esperados e inesperados por igual o, al menos, que se salen de las concepciones más habituales y extendidas.

En primer lugar, en el grupo que presenta mayores tasas de abandono, por encima del 20%, existe una mezcla de carreras de Ciencias y Humanidades, a las que se suma Informática. Este hecho es el que posiblemente explique el por qué la UPM no se encuentra en el primer lugar en términos de abandono, y la UAH y la UCM se encuentren tan cercanas a estas.

Como titular en este apartado, se podría aseverar que las ingenierías no son las carreras universitarias con mayor abandono de la Comunidad de Madrid, a pesar de que estos planes de estudios sí que se localizan dentro del tercer grupo de abandono, el situado entre el 15% y el 20%.

En el lado opuesto, con tasas de abandono de tan solo entre el 5% y el 10%, se encuentran materias muy asociadas con la vocación y el cuidado, como son las Enseñanzas Docentes en infantil y primaria, Veterinaria y Medicina, además de los Deportes.

2.2. ABANDONO EN LA UNIVERSIDAD COMPLUTENSE. UN ZOOM SOBRE LA RAMA DE CIENCIAS E INGENIERÍA

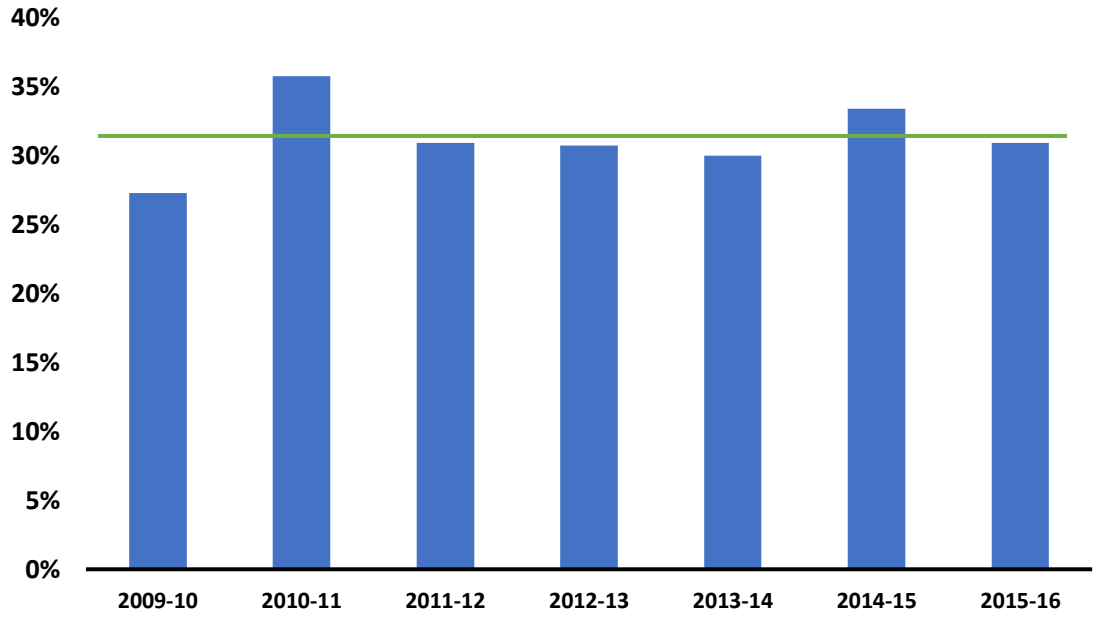
Como se ha podido observar en el apartado anterior, entre las carreras que mayor abandono presentan en la Comunidad de Madrid, están aquellas pertenecientes a las ramas tanto de Ciencias como de Ingeniería y Arquitectura. Es por ello que, en esta última parte del informe, relativa específicamente a la UCM, se va a realizar un enfoque

concreto sobre estos dos grupos de materias de conocimiento de la universidad, en aras de profundizar un poco más en el comportamiento que existe en términos de abandono para las mismas en la universidad.

Dado que la UCM tiene mayor recorrido y trayectoria con carreras de ciencias, y el número de ingenierías que oferta no es demasiado amplio, el hecho de analizar juntas las ramas de Ciencias e Ingeniería no resulta en un maremágnum de datos difícil de desentrañar.

En la figura 5, se realiza un acercamiento general al abandono en las citadas ramas de la UCM. En esta ocasión, los datos provenientes de SIDI proveen de datos sobre una cohorte de entrada más, la correspondiente a 2015-2016, por lo que los datos presentados a continuación tienen un alcance temporal ligeramente mayor.

Figura 5. Abandono medio agregado para los planes de estudio de Ciencias e Ingeniería de la UCM. Periodo entre 2009-2010 y 2015-2016

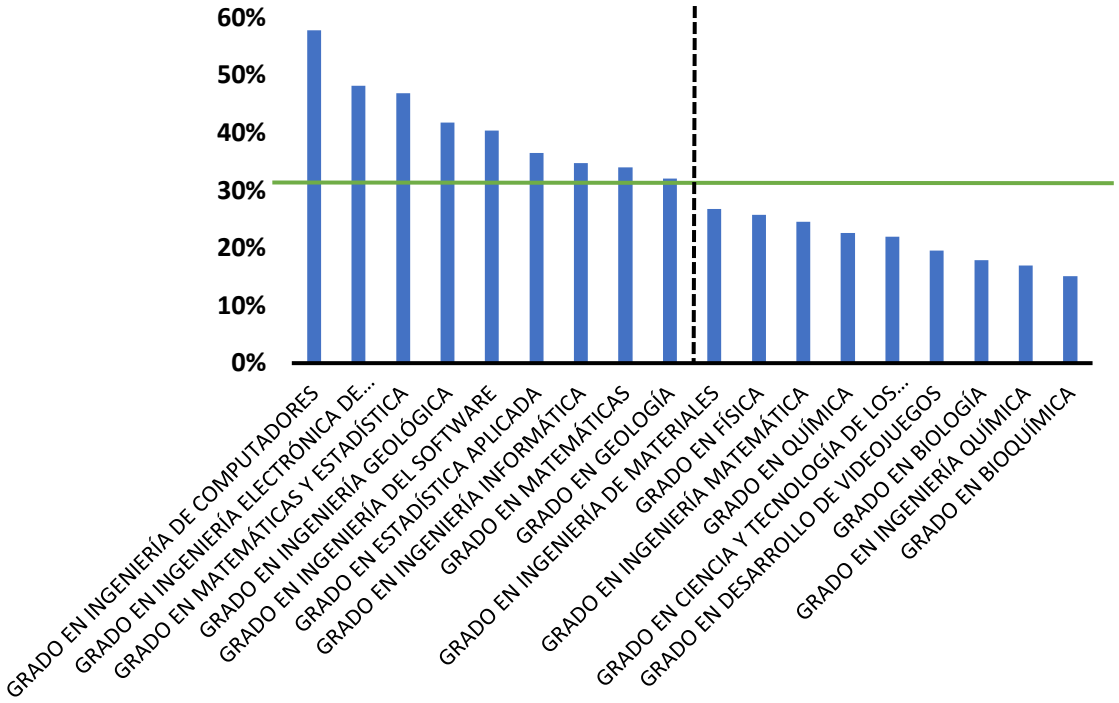


Fuente: SIDI

La figura 5 revela una realidad que ya se podía intuir en base a los resultados de la Comunidad de Madrid: la media de abandono anual se sitúa en unas cifras notablemente altas. La media del periodo, que viene marcada por la línea que atraviesa las barras, se sitúa en un imponente 31,31%, dejando algún año, como el 2010-2011, por encima del 35%.

Cuando se observa la problemática por planes de estudio, que es lo que se puede observar en la figura 6, se percibe que hay ocho planes de estudios que se encuentran por encima de la media del periodo, de nuevo marcada con una línea horizontal. Estas carreras son todas las ingenierías, salvo la Química, Matemática y de Materiales, además de Matemáticas, Estadística Aplicada, Geología y Matemáticas y Estadística.

Figura 6. Abandono medio desagregado por planes de estudio de Ciencias e Ingeniería de la UCM. Periodo entre 2009-2010 y 2015-2016



En esta ocasión, para la UCM, parece existir un sesgo en el abandono derivado de la dificultad propia asociada a las ingenierías, especialmente las impartidas en la Facultad de Informática, además de las disciplinas directamente relacionadas con la Estadística y las Matemáticas es su manera más pura o de enfoque de ciencia tradicional.

Por otro lado, ante este panorama, las tasas de abandono del resto de disciplinas, más relacionadas con las ciencias naturales y elementales (Físicas, Químicas), aunque en niveles todavía altos, no parecen representar una necesidad tan inmediata en su abordaje como las primeras.

La situación que se desprende de lo expuesto en ambos gráficos correspondientes a la Universidad Complutense plantea, en cierta medida, un dilema, pues en un contexto de cierto descenso del abandono, o al menos en una estabilización del mismo, como se podía observar en la figura 3, es posible que sea difícil seguir ahondando en un descenso general de las tasas de abandono universitario sin previamente intervenir en estas áreas, que se demuestran en clara contraposición a carreras que tradicionalmente se han considerado como de mayor vocación.

3. OBJETIVOS DE INVESTIGACIÓN

Una vez expuesta la pertinencia y necesidad de realización de este estudio, el siguiente paso consiste en plantear cuáles son los objetivos que van a guiar la realización del mismo, los cuales estarán eminentemente influenciados por el rendimiento académico de los alumnos y alumnas, tal y como se pondrá de relieve en el marco teórico.

Objetivo general

Elaborar modelos de predicción de perfiles de alumnos que tienen mayores probabilidades de abandonar el Grado en Estadística Aplicada en la Universidad Complutense de Madrid, en base a su rendimiento académico durante el primer año.

Objetivos específicos

-Encontrar si existen patrones o tipologías de alumnos en base a su trayectoria académica y su modalidad de abandono en la facultad.
-Identificar las asignaturas clave que suponen la diferencia entre abandonar o continuar el programa de estudios.
-Discernir si la nota de acceso a la universidad se puede emplear como predictor del abandono.
-Elaborar recomendaciones que, en base a los objetivos anteriores, pongan a las distintas instancias de la facultad sobre aviso por riesgo de abandono del estudiantado.

A continuación, se procede a exponerse un breve marco teórico sobre la cuestión del abandono universitario para, posteriormente, proceder a explicar la metodología y las técnicas de análisis que se emplearán a lo largo del trabajo.

4. MARCO TEÓRICO

La cuestión del abandono universitario no constituye un objeto de estudio especialmente novedoso desde el punto de vista de la descripción del evento, así como de la búsqueda de sus múltiples causas y razones por las cuales este se produce, aglutinando explicaciones de índole cognitiva, de origen social o estrictamente económicas (Benítez *et al.*, 2017).

La aplastante lógica de los problemas de eficiencia económica que este abandono genera para las entidades financiadoras del sistema universitario – ministerios e instituciones educativas varias – pues, en el largo plazo, la inversión realizada en el alumnado probablemente nunca se verá traducida en un aporte a la sociedad desde el campo financiado (Ruè, 2014). No solo eso, sino que también representa un problema para el estudiante que decide abandonar, pues en la medida en que decide invertir tiempo y dinero en su formación universitaria, asumiendo en ocasiones un alto coste de oportunidad en relación con la economía familiar, abandonar el plan de estudios que se encuentra cursando implica la no recuperación de esos recursos invertidos ni a corto ni medio plazo.

De esta manera, la reducción de dicho abandono comporta toda una serie de externalidades positivas que van en pro tanto del alumnado como de la propia universidad, facultad o plan de estudios al que este afecta con mayor o menor intensidad.

Por el lado del empleo de variables de rendimiento como elementos predictivos principales, esta decisión se sustenta en que “una variable recurrentemente estudiada ha sido el rendimiento académico temprano del estudiante. […] Varias investigaciones han constatado que el rendimiento ha probado su influencia en la toma de decisiones sobre permanencia en los estudios matriculados” Tuero *et al.* (2018: 136). Aunque no es la única de las posibilidades, el rendimiento académico constituye, por tanto, una perspectiva adecuada de análisis para un problema de abandono.

Sin embargo, podría parecer que un enfoque desde el rendimiento académico tampoco supone una gran novedad en el panorama del estudio del abandono universitario. ¿Qué resulta, por tanto, novedoso en este campo de estudio? Es precisamente ahí donde la predicción y las técnicas de Machine Learning que facilitan las labores de predicción toman un papel de absoluta relevancia para ampliar conocimiento y disponibilidad de herramientas en un campo incipiente: el empleo de algoritmos de predicción para localizar *a priori* a todos aquellos alumnos que estén en claro riesgo de retirarse del plan de estudios en el que se encuentran matriculados tras el primer año de carrera. A pesar de que ya existen trabajos que apuntan en esta dirección (Ortiz *et al.*, 2017; Rovira *et al.*, 2017), todavía no parecen existir esfuerzos institucionales serios para sistematizar la generación de perfiles de estudiantes en potencial riesgo de abandono y plantear programas de acción que puedan intervenir *a priori*.

Dado todo lo anterior, se entiende que un objeto de investigación como el que plantea este trabajo puede resultar de enorme utilidad no solo para la facultad de cuyos datos

se nutre, sino también la propia universidad, promoviendo la ampliación de proyectos de este calado.

5. FUENTE DE DATOS Y METODOLOGÍA

5.1 MATRIZ DE DATOS

Dado que el foco del trabajo se encuentra en perfilar y predecir la tipología de alumnos con mayor propensión al abandono en el Grado de Estadística Aplicada de la UCM, tal y como se ha comentado en la introducción, se cuenta con un conjunto de datos con la información académica de 1134 alumnos correspondiente a los matriculados en la facultad entre los cursos 2009/2010 y 2017/2018. Dicha matriz de datos, producto de la explotación bruta de las bases de la universidad, cuenta con las siguientes variables:

- Identificador anonimizado del alumno [Identificador]
- Género del alumno: Hombre o Mujer [Sexo]
- Código de la asignatura cursada [Asignatura]
- Curso de acceso a la universidad [Curso_acces]
- Curso en que se recibe docencia de la asignatura (una misma asignatura no superada se puede cursar en cursos distintos) [Curso_asig]
- Nota para la convocatoria de febrero (ordinaria) [Nota_feb]
- Nota para la convocatoria de junio (ordinaria) [Nota_jun]
- Nota para la convocatoria de septiembre (extraordinaria) [Nota_sept]
- Nota de acceso a la universidad [Nota_PAU]

El conjunto de datos facilitado se encuentra en formato “long”, es decir, para cada alumno localizado con su propio identificador, los datos de matriculación en las diferentes asignaturas están organizados por filas, de tal manera que la información de un único alumno o alumna puede ocupar más de una línea dentro de la matriz, tal y como se puede observar en la Tabla 1 para el caso cuyo ID es igual a 1. Hay que destacar que, en la matriz de datos original, tan solo se podían encontrar registros de asignaturas matriculadas. Si no había habido matriculación en una o varias asignaturas, estas simplemente no aparecían dentro del registro de cada alumno.

Tabla 1 – Extracto de la matriz de datos original proporcionada por SIDI (UCM)

ID	Asignatura	Sexo	Curso_ asig	Curso_ acces	Nota_ feb	Nota_ jun	Nota_ sept	Nota_ PAU
1	801584	2	201112	201112	7	0	0	7
1	801585	2	201112	201112	0	2.5	0	7
1	801586	2	201112	201112	5	0	0	7
1	801587	2	201112	201112	0	0	0	7
1	801589	2	201112	201112	7.6	0	0	7
1	801590	2	201112	201112	0	1	0	7
1	801598	2	201112	201112	0	2	5	7

Fuente: SIDI

Dado que la estructura de la matriz de datos disponible responde a las necesidades de la universidad y no a las del trabajo, que requiere de una estructura “wide”, es decir, una única línea por caso, para poder analizar y aplicar los modelos estadísticos adecuadamente, antes de comenzar a realizar ningún tipo de análisis, se ha llevado a cabo un laborioso y meticuloso trabajo de transformación de la misma.

En primer lugar, se han calculado, en el formato “long” del archivo, tres variables nuevas sobre cada asignatura: por un lado, la nota máxima obtenida por el alumno en el conjunto de todas las convocatorias de una asignatura durante su primer año (“notaMax”); por otro lado, se ha creado también la variable “cursada”, que indica el número de convocatorias a las que se ha presentado un alumno en una asignatura durante un año, las cuales pueden ser 0, 1 o 2 (como máximo, convocatoria ordinaria y extraordinaria); en último lugar, se ha creado lo que se ha venido a llamar el “Estado de la asignatura” (“estadoAsig”), cuyas categorías indican si el alumno ha aprobado una asignatura (1), ha suspendido (2), no se ha presentado (3) o no la ha cursado nunca (4).

Posteriormente, se han filtrado todos aquellos casos que no permitiesen calcular posteriormente si habían producido abandono o no tras el primer año de matriculación, esto es: aquellos alumnos que hubiesen comenzado la carrera antes del curso 2009-2010, dado que solo aparece la información de las asignaturas que estaban cursando del dicho curso en adelante y no existía información sobre su primer año; los alumnos y alumnas de las cohortes 2016-2017 y 2017-2018, ya que no resulta posible conocer si han causado abandono al no disponer de información sobre su tercer año de matriculación (en la práctica, este no se había producido todavía).

A continuación, se ha aplicado la transformación de la estructura de la base de datos del ya comentado formato “long” a “wide” con el paquete de R “reshape”, tomando como variable de unión “Identificador” y obteniendo como resultado una matriz donde, para cada asignatura, se encuentran presentes las tres variables nuevas anteriores bajo el formato “estadoAsig.801XXX”, “notaMax.801XXX” y “cursada.801XXX”.

Adicionalmente, dado que se corría el riesgo de repetir la cohorte de acceso, la nota obtenida en la PAU y el sexo de cada estudiante por cada asignatura en el formato “wide”, estos se han almacenado en matrices aparte para ser incorporados más tarde a la matriz definitiva.

Finalmente, se ha obtenido la matriz en formato “wide”, de tal manera que, aunque el alumnado no hubiese estado matriculado en alguna asignatura, a diferencia de en el formato “long”, sí que aparecerían las asignaturas nunca matriculadas (presentes con valores ausentes “NA”), a las que posteriormente se les ha aplicado el valor “4” en “estadoAsig”. Bajo este formato, ya se ha podido pasar a calcular si el alumno ha causado o no abandono tras el primer año de carrera, según la definición del Ministerio de Educación, y que ha quedado recogido en la variable dicotómica “Abandono”, con valor 1 si este se ha producido, y valor 0 en caso contrario. También se han calculado, como medidas auxiliares, las notas medias obtenidas por caso en cada semestre del primer año (“notaMedia1Q” y “notaMedia2Q”), y la nota media global del primer año de carrera (“NotaMedia1A”).

De esta manera, como presenta la Tabla 2, se obtiene una estructura de la base de datos con la siguiente estructura (solo se presenta 2 asignaturas, ya que la matriz de datos tiene del orden de 38 columnas), correspondiente a las asignaturas cursadas durante el primer año de carrera:

Tabla 2 – Extracto de la matriz de datos transformada para su uso en el estudio

Identificador	cursada.801584	notaMax.801584	estadoAsig.801584	cursada.801585	notaMax.801585	estadoAsig.801585	cursada.801586
1	1	7	1	1	2.5	2	1
2	2	5	1	2	1.5	2	0
3	1	6	1	1	0.5	2	0
4	1	7.5	1	0	0	3	1

almacenaCursoEntrada	almacenaSexo	almacenaPAU	Abandono	notaMedia1Q	notaMedia2Q	NotaMedia1A
201112	2	7	1	3.92	0.7	2.31
201314	2	5.74	0	2.36	1.5	1.93
201314	1	5.46	0	3.6	0.1	1.85
201516	1	5.16	1	2.84	0	1.42

Tras todas las transformaciones comentadas, llevadas a cabo con R a través de la IDE RStudio, se ha creado un archivo en formato .csv listo para ser exportado a otras plataformas, como SAS y SAS Miner.

A continuación, se detallan las asignaturas de primer año del Grado en Estadística Aplicada con sus códigos correspondientes:

Tabla 3 – Relación de asignaturas y códigos. Primer Curso Grado en Estadística Aplicada (UCM)

1º Semestre	2º Semestre
801580 - Descripción y Exploración de Datos	801581 - Azar y Probabilidad
801584 - Fuentes y Técnicas de Recogida de Información en Investigación Social y de Mercados	801583 - Software Estadístico I
801586 - Programación I	801585 - Estadística Económica
801588 - Métodos Matemáticos para Estadística I	801587 - Programación II
801589 - Métodos Matemáticos para Estadística II	801590 - Métodos Matemáticos para Estadística III

5.2 METODOLOGÍA DE INVESTIGACIÓN

Como metodología global de análisis, se ha empleado la propuesta por el Instituto SAS para la realización de proyectos con la lógica de la Minería de Datos (SAS, n.d.). Esta metodología, conocida como SEMMA, plantea cinco fases de tratamiento de los datos, desde la selección de los datos hasta la evaluación de las operaciones llevadas a cabo con ellos. SEMMA, por tanto, son las siglas de:

- Sampling (Muestreo): Selección de los datos a analizar y partición de los mismos en porciones de entrenamiento-validación-test, si procede, para garantizar la mayor aleatoriedad posible en el proceso de ejecución de modelos.
- Explore (Exploración): Búsqueda de relaciones iniciales para orientar la búsqueda y futura elaboración de modelos, a partir de herramientas tales como distribuciones, cruces de categorías o búsqueda de correlaciones.
- Modify (Modificación): Ejecución de las operaciones necesarias en los datos para cumplir con las condiciones *a priori* que requieren los diversos métodos, como la

búsqueda de valores extremos u *outliers*, el reemplazo de extremos o la imputación de valores ausentes.

- Model (Modelización): Creación y ejecución de modelos de entrenamiento necesarios para la predicción de la variable objetivo, variando todos aquellos parámetros necesarios para la realización de las técnicas de Machine Learning correspondientes.
- Assess (Evaluación): Prueba de los modelos para comprobar la calidad de sus predicciones y compararlos entre sí. La manera habitual de llevarla a cabo es a través de validación cruzada repetida para evaluar las condiciones de sesgo y varianza entre modelos y ejecutar comparaciones a través de *boxplot* o gráficos de caja.

Si bien la metodología SEMMA muestra una lógica de trabajo que permite pasar del dato bruto a toda una serie de predicciones con respecto a la variable objetivo o dimensión deseada, esta requiere de todo un conjunto de técnicas de Machine Learning para poder ejecutar las predicciones deseadas con respecto al abandono del alumnado del Grado en Estadística Aplicada, además de la ya tradicional Regresión Logística, que servirá como termómetro para evaluar los modelos y calcular las probabilidades de abandono o no en términos descriptivos.

De esta manera, de acuerdo a los objetivos planteados, se emplearán todo un conjunto de técnicas multivariantes y de Machine Learning, y que serán presentadas en los próximos subapartados.

5.2.1 Regresión Logística

La regresión logística permite "predecir la probabilidad de ocurrencia de un evento (Y = 1) […] a partir de los valores que presenten una serie de variables independientes categóricas y/o continuas analizadas" (Cea d'Ancona, 2002: 129). El rango de valores siempre es positivo, oscilando entre 0 y 1, donde la probabilidad de ocurrencia y no ocurrencia del evento se representa mediante las siguientes expresiones matemáticas, donde:

$$P(Y=0) = \frac{e^{-\sum_{K=1}^{K}\beta_K X_K}}{1+e^{-\sum_{K=1}^{K}\beta_K X_K}}$$

$$P(Y=1) = \frac{1}{1+e^{-\sum_{K=1}^{K}\beta_K X_K}}$$

De esta manera, se suele representar la probabilidad de ocurrencia del evento frente a la probabilidad de que dicho evento no ocurra, bajo la forma de cociente en lo que viene a llamarse *odds*:

$$Odds = \frac{P(Y=1)}{1-P(Y=1)} = \frac{P(Y=1)}{P(Y=0)}$$

Cuando los *odds* de un evento que se encuentra con una condición se ponen en relación con el mismo evento bajo otra condición, se tienen los *odds-ratio*, cuya finalidad consiste en evaluar cuál es el cambio que se produce por la variación de ambas condiciones.

Finalmente, para estimar los coeficientes de la regresión logística (β_K), se emplea el criterio de "máxima verosimilitud", lo que implica la máxima probabilidad de ocurrencia del evento que se pretende predecir.

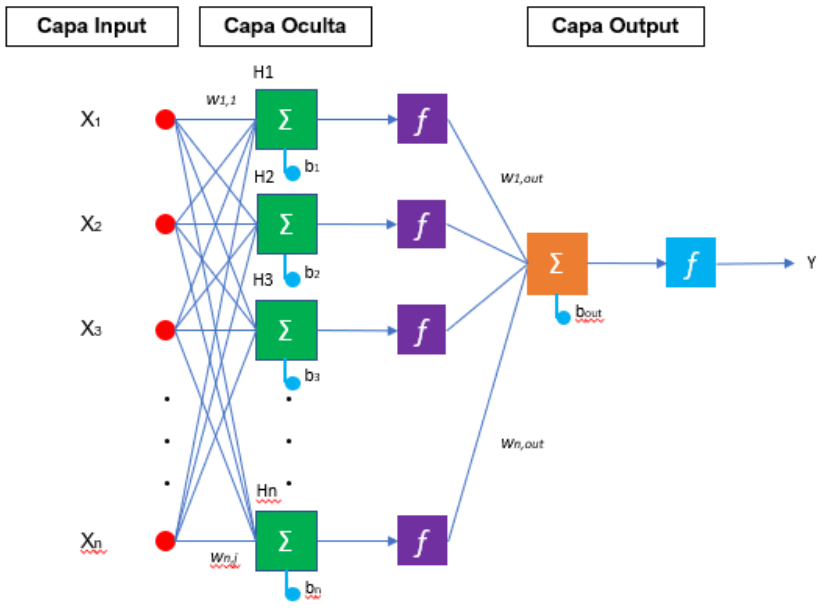
5.2.2 Redes Neuronales

Las redes neuronales son un conjunto de unidades de neuronas donde cada una de ellas "toma como entradas las salidas de las neuronas de las capas antecesoras, cada una de esas entradas se multiplica por un peso, se agregan los resultados parciales y mediante una función de activación se calcula la salida. Esta salida es a su vez es entrada de la neurona a la que precede" (Calvo, 2017).

La función de activación habitualmente es creciente, positiva y acotada entre [0,1], empleándose mayoritariamente la tangente hiperbólica (tanh).

Las redes neuronales utilizan procesos de aprendizaje no supervisado para "encontrar relaciones no lineales entre conjuntos de datos" (Pérez y Fernández, 2007). Funcionan como un modelo de caja negra y se pueden emplear bien para predicción de tendencias o también como clasificadoras (Caicedo y López, 2009).

Figura 7. Esquema red neuronal artificial



Fuente: Elaboración propia a partir de Portela (2019)

De esta manera, la elaboración de la ecuación resultado de una red neuronal se rige por la siguiente fórmula:

$$z_k = \sum_j (w_{jk} y_j) - \theta_k = \sum_j \left(w_{jk} f_j \left(\sum_i (w_{ij} x_i) - \theta_j \right) \right) - \theta_k$$

Los parámetros de aprendizaje que habrá que controlar en el seno de la red, de acuerdo con las características incluidas en el paquete "Caret" de R, serán:

- Número de nodos de la red (size)
- Tasa de aprendizaje (decay)
- Número de iteraciones de la red (itera)

5.2.3 Árboles de decisión

Los árboles de decisión constituyen una técnica de clasificación de individuos donde, a partir de un nodo original o raíz que contiene el conjunto global de datos, este se va dividiendo en subsiguientes nodos hijos donde van cayendo los individuos en base a características compartidas y a partir de reglas simples, buscando la mayor homogeneidad posible dentro de cada nodo (Medina y Ñique, 2017: 167).

En definitiva, el objetivo de un árbol de decisión consiste en "jerarquizar o repartir un grupo heterogéneo de características (atributos) de cosas o sujetos en grupos más pequeños y homogéneos" (Castillo-Rojas *et al.*, 2014: 354).

Si bien es cierto que los árboles de decisión se pueden emplear de manera aislada como técnica, en el desarrollo de este trabajo tomarán importancia en tanto que conformarán la base de otras técnicas de agregación de estos, como son Random Forest, Bagging y Gradient Boosting, expuestos en los próximos apartados.

5.2.4 Random Forest y Bagging

La técnica de Random Forest consiste en la modelización de múltiples árboles de decisión incorrelados para posteriormente promediarlos (Medina y Ñique, 2017: 170). Así, se obtienen árboles dependientes de un vector de la muestra, con la misma distribución en todos los árboles que conforman el bosque.

La manera en que se lleva a cabo consiste en tomar un porcentaje (n) de la muestra total de entrenamiento (N) con reemplazo (un mismo elemento de la muestra puede tomar parte en la misma más veces), de tal manera que en las sucesivas iteraciones realizadas para cada árbol, la predicción se ajuste incluyendo por sorteo un número (m) de variables menor que el total de las variables input (M), garantizando así el mayor grado de incorrelación posible entre los distintos árboles generados.

Bagging constituye un caso particular de Random Forest, donde el número de variables tomadas en cuenta es igual al total de variables existentes como input.

Al igual que con las redes neuronales, existen una serie de parámetros a tener en cuenta para el entrenamiento del algoritmo, bajo el paraguas de nuevo del paquete "Caret" de R:

- Número de variables a sortear (mtry)
- Número de árboles que se ejecutan en cada iteración (ntree)
- Tamaño de la muestra que se reemplaza (sampsize)
- Tamaño mínimo de las hojas finales de cada árbol (nodesize)

5.2.5 Gradient Boosting

El algoritmo Gradient Boosting, descrito por Friedman en 2001, trata de construir árboles de decisión de manera repetida, modificando en cada iteración las predicciones iniciales y buscando la reducción de los residuos en el sentido del vector de error (Friedman, 2001).

Así, el empleo del algoritmo Gradient Boosting "supone necesariamente la especificación de tres parámetros importantes: la ratio de aprendizaje o parámetro de contracción, la profundidad de los árboles de decisión (número de cortes o divisiones de los árboles desde un nodo terminal hasta el nodo raíz) y el número de árboles de decisión." (Pozuelo et al., 2018).

Los parámetros a controlar, nuevamente bajo el paquete "Caret" de R, son:

- Número de árboles a generar en cada iteración (n.trees)
- Profundidad máxima de los árboles (max.depth)
- Tasa de aprendizaje del algoritmo (shrinkage)
- Tamaño mínimo del nodo terminal de cada árbol (n.minobsinnode)

Adicionalmente, y dada la gran popularidad de la que ha gozado en los últimos años, se empleará una mejora del paquete estándar de R para ejecutar Gradient Boosting, el cual se llama "Xgboost". Este paquete introduce como novedad la regularización orientada a evitar el sobreajuste de los modelos de entrenamiento, y cuya aplicación se materializa a través de los parámetros *gamma* y *lambda*, que actúan como penalización en función del número de hojas y la predicción en cada una de ellas (Portela, 2019).

5.2.6 Ensamblado de modelos

El ensamblado de modelos consiste, básicamente, en realizar un promedio de los modelos obtenidos bajo las técnicas anteriores con la intención de reducir el error obtenido en todas las técnicas implicadas en el proceso de ensamblado, llevando a que se obtenga un menor error que con las otras técnicas por separado.

5.2.7 Proceso de remuestreo y validación cruzada. Uso de boxplot

Para la correcta ejecución de las técnicas, facilitar su comparabilidad y resultar los más exhaustivos y certeros posibles, dejando de lado la posibilidad de la influencia de efectos aleatorios, en la mayoría de los casos se llevará a cabo el proceso de la validación cruzada repetida, que permite dividir la muestra empleada en pequeñas submuestras que se van alternando de manera consecutiva como conjuntos de entrenamiento-test,

repitiéndose este proceso tantas veces como semillas se indiquen a los algoritmos. Todo ello producirá una variabilidad entre modelos mensurable en términos de sesgo y varianza, que serán mostrados y comparados para su evaluación en gráficos de caja o boxplot.

De esta manera, la revisión gráfica en estos boxplot facilitará la tarea de encontrar los modelos con una mayor capacidad predictiva o, lo que es lo mismo, los que en el momento de la predicción y clasificación hierren menos.

5.2.8 Matriz de confusión y medidas asociadas

Finalmente, cuando se establezca un modelo como ganador dentro de una técnica o un conjunto de datos, se presentará su matriz de confusión, que permitirá evaluar si las predicciones realizadas coinciden o no con las observaciones reales y calcular a partir de ella distintas medidas como la exactitud (accuracy), la sensibilidad o la especificidad.

5.3 DESCRIPCIÓN Y DEPURACIÓN DE LOS DATOS

Una vez introducidas las técnicas necesarias para llevar a cabo los modelos correspondientes para predecir el abandono en el grado objeto de estudio, en este apartado se procede a describir los datos y explicar el escaso tratamiento que se ha tenido que realizar de los mismos para cumplir con los objetivos de investigación.

5.3.1 Descripción de datos

En primer lugar, en las tablas 4 y 5 se encuentran los estadísticos descriptivos tanto para las variables de intervalo como para las variables continuas, obtenidos a través de SAS Enterprise Manager.

Tabla 4. Estadísticos descriptivos. Variables continuas

Variable	Ausente	N	Mínimo	Máximo	Media	D.E.	Asimetría	Curt.
NotaMedia1A	0	564	0	9.77	3.09	2.22	0.65	-0.35
almacenaPAU	114	450	5	9.42	5.94	0.73	1.70	3.67
notaMax.801580	0	564	0	10.00	4.72	2.72	-0.35	-0.60
notaMax.801581	0	564	0	10.00	2.38	2.83	0.89	-0.45
notaMax.801583	0	564	0	10.00	2.23	2.92	0.98	-0.39
notaMax.801584	0	564	0	10.00	5.35	2.45	-0.94	0.08
notaMax.801585	0	564	0	10.00	2.71	2.74	0.61	-0.82
notaMax.801586	0	564	0	10.00	3.14	3.38	0.49	-1.27
notaMax.801587	0	564	0	10.00	2.14	3.28	1.18	-0.15
notaMax.801588	0	564	0	10.00	2.78	2.85	0.65	-0.86
notaMax.801589	0	564	0	10.00	3.31	2.91	0.36	-1.07
notaMax.801590	0	564	0	10.00	2.13	2.81	1.06	-0.14
notaMedia1Q	1	563	0	9.68	3.86	2.26	0.18	-0.75
notaMedia2Q	0	564	0	9.86	2.32	2.42	0.93	-0.09

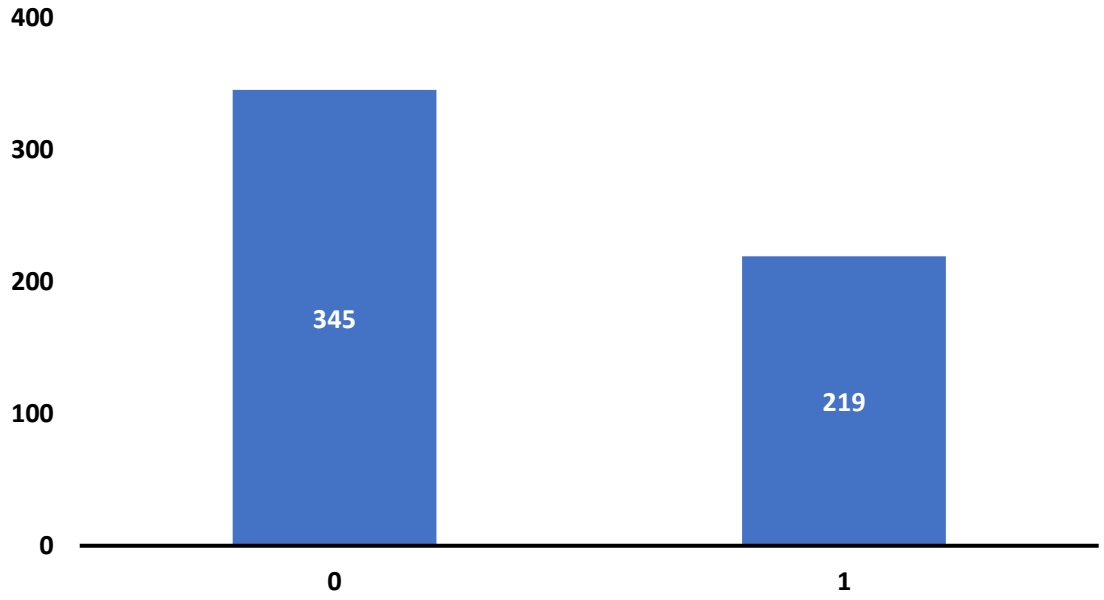
Tabla 5. Estadísticos descriptivos. Variables de clase

Variable	Niveles	Ausente
Abandono	2	0
almacenaCursoEntrada	7	0
almacenaSexo	2	0
cursada_801580	3	0
cursada_801581	3	0
cursada_801583	3	0
cursada_801584	3	0
cursada_801585	3	0
cursada_801586	3	0
cursada_801587	3	0
cursada_801588	3	0
cursada_801589	3	0
cursada_801590	3	0
estadoAsig_801580	4	0
estadoAsig_801581	4	0
estadoAsig_801583	4	0
estadoAsig_801584	3	0
estadoAsig_801585	4	0
estadoAsig_801586	4	0
estadoAsig_801587	4	0
estadoAsig_801588	4	0
estadoAsig_801589	4	0
estadoAsig_801590	4	0

Como se puede observar, salvo para la variable “almacenaPAU”, que contiene la nota de acceso a la universidad del alumnado, no existen valores ausentes en el resto de las variables. En el caso de la nota de la PAU, los casos ausentes no han sido proporcionados por la universidad, mientras que en el resto de variables, el propio proceso de creación de variables nuevas, que compone la casi totalidad del conjunto de datos de las asignaturas del primer año, propicia que, en efecto, no existan valores perdidos, lo cual facilita enormemente las labores de depuración de los datos.

En el ámbito de las variables categóricas, hay que destacar especialmente la variable objetivo, puesto que de su definición y correcta aplicación dependerá la capacidad predictiva de todos los modelos que se lleven a cabo. La manera en que se distribuye esta variable se puede observar en la Figura 8.

Figura 8. Frecuencias variable objetivo "Abandono"



Por tanto, del conjunto global de la muestra, habrá un total de 219 alumnos y alumnas que habrán causado abandono tras el primer año de carrera, mientras que 345 son los que al menos se matricularon un curso más después del primer año universitario lo que implica que, para la definición de abandono que se ha construido en esta muestra, se produce un 38.83% de abandono sobre el total de alumnos matriculados.

5.3.2 Depuración del conjunto de datos

En cuanto a la depuración del conjunto de datos, también realizada en el entorno SAS Enterprise Miner, siguiendo el proceso habitual, se ha procedido a evaluar, en primer lugar, la existencia o no de valores atípicos o extremos que pudiesen tener algún tipo de influencia sobre los datos. No obstante, tras realizar las pruebas correspondientes, tanto del rango intercuartílico como de la desviación estándar, no se ha encontrado la existencia de dichos valores.

Por ello, el siguiente paso que se ha llevado a cabo ha sido realizar una imputación de los valores ausentes de las notas PAU de los alumnos y alumnas con la media de todos los casos, dada la imposibilidad de conocer las puntuaciones si no han sido proporcionadas previamente. En este sentido, la media global no se verá alterada.

Finalmente, se ha aplicado el nodo de transformación de variables para evaluar si alguna de las variables continuas que se toman como input presenta algún tipo de relación oculta con la variable objetivo a través del procedimiento de correlación máxima. El programa no ha arrojado ninguna sugerencia de transformación, por lo que se ha procedido a continuar con las mismas variables iniciales.

Una vez depurados los datos, se puede proceder a realizar la descripción del comportamiento del alumnado en relación con el objeto de estudio.

6. EXPLORACIÓN DE LA MATRIZ DE DATOS. SECUENCIAS DE ESTADOS DE LAS ASIGNATURAS

Una vez depurados los datos, y continuando con la lógica de seguir indagando sobre los mismos antes de modelizarlos a través de las técnicas descritas, es momento de poner en práctica la razón de ser por la cual se ha modificado sustancialmente la base de datos original para dotarla de una estructura muy concreta.

Así, una de las ventajas de la transformación realizada sobre la base de datos bajo el formato de generación de "estados" de cada asignatura, es que se permite estudiar, de manera longitudinal, cuál ha sido el comportamiento de los alumnos en términos de éxito o fracaso a lo largo de las materias que componen el plan de estudios del Grado en Estadística Aplicada.

Una aproximación inicial de estas características permite, por un lado, trazar *a priori* tipologías de alumnos en base al mencionado fracaso y, por otro lado, aproximar aquellas asignaturas que representan una mayor dificultad o pueden tener mayor influencia en un posible abandono, de tal manera que todo ello pueda ser contrastado más adelante con los modelos que se desarrollen.

Para realizar esta aproximación, se ha empleado el paquete "TraMineR" de R, concebido precisamente para analizar secuencias de estado en sets de datos longitudinales, con especial énfasis en la visualización de dichas secuencias (Gabadinho *et al.*, 2011).

El objetivo de este apartado, por tanto, consiste en visualizar gráficamente el recorrido del alumnado a través de las asignaturas de toda la carrera, en primer lugar, para posteriormente centrar la atención en su comportamiento durante el primer curso y, en última instancia, hallar grupos que permitan agrupar pautas de abandono, siempre con la función "seqplot" del mencionado paquete de R.

En la Figura 9, se pueden observar las citadas trayectorias de la muestra completa final, estudiante a estudiante, que se han obtenido tras las transformaciones realizadas. Se recuerda que, a efectos de interpretación de los gráficos, las categorías de las variables "estadoAsig.801XXX" son:

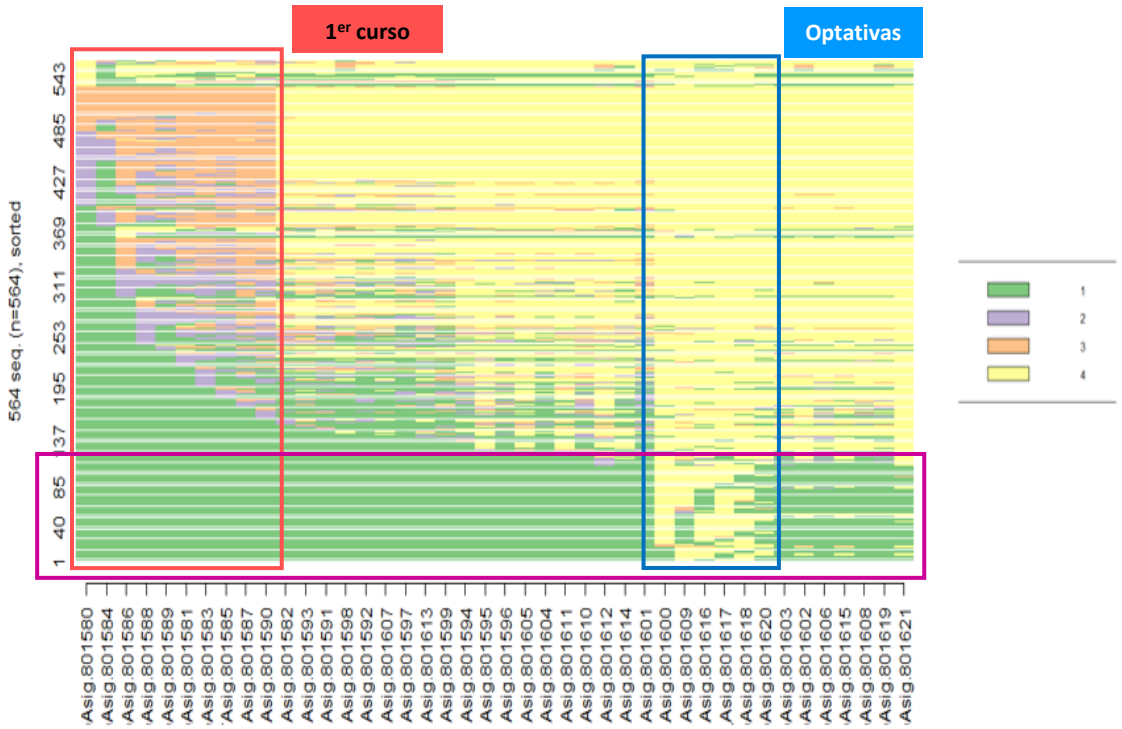
- 1: Aprobada
- 2: Suspensa
- 3: No presentado
- 4: Nunca matriculada

Dado que la inmensa mayoría de los y las estudiantes se matriculan el primer año de la casi totalidad de las asignaturas correspondientes al primer curso, resulta fácilmente perceptible el éxito o fracaso global obtenido por el alumnado en base a un criterio estricto de rendimiento académico en una lectura longitudinal.

Aproximadamente, en el tercio inferior, señalado en color violeta, se podrían encontrar aquellos alumnos y alumnas que habrían superado la carrera sin problemas en el tiempo

estipulado, con pocas o ninguna asignatura pendiente, dado que predomina el color verde (categoría “1”, asignatura aprobada).

Figura 9. Variación de los estados de las asignaturas. Plan de estudios completo. Grado en Estadística Aplicada



En el tercio intermedio, con algo más de amplitud que en el caso anterior, se encontrarían aquellas personas que, en el momento de la extracción de los datos, o bien no habían acabado la carrera, varios de ellos con asignaturas todavía pendientes o no presentadas, o bien personas susceptibles de haber producido abandono en un momento más tardío al primer año de carrera, los cuales quedarían fuera del objeto de este estudio.

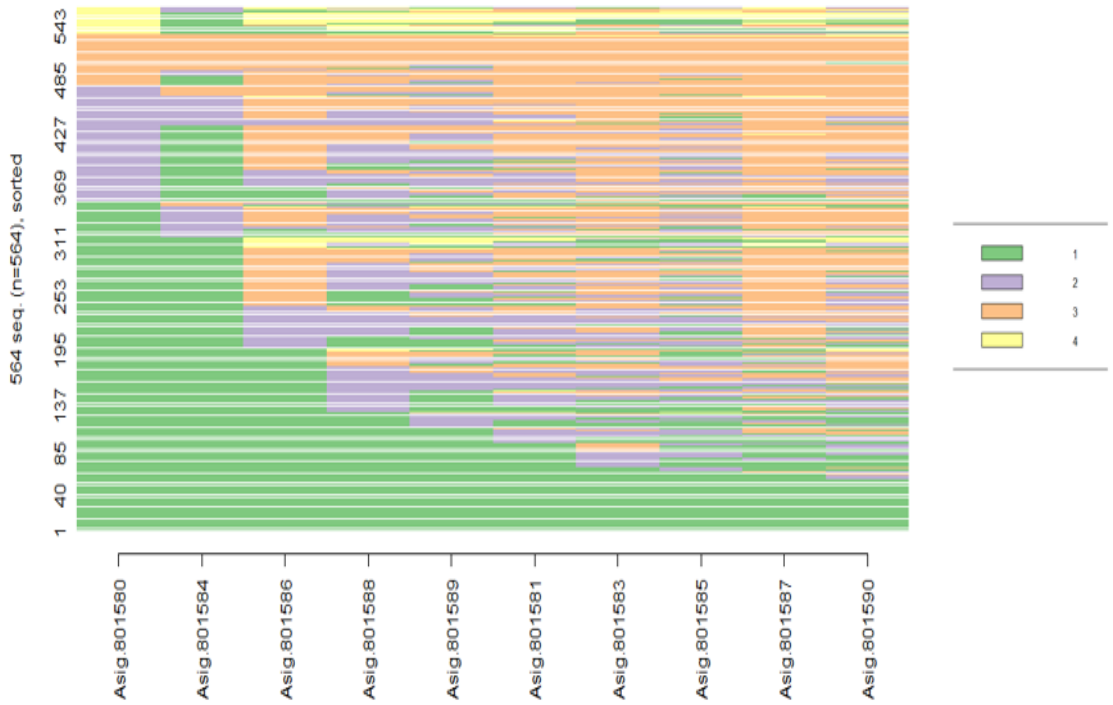
Finalmente, en el tercio superior de la Figura 9, se encontrarán con mayor probabilidad todas aquellas personas que, efectivamente, hayan causado abandono tras el primer año de matriculación, ya que predomina esencialmente el color amarillo, asociado a la categoría “4”, lo que indicaría asignaturas que nunca han sido matriculadas correspondientes tanto al segundo como al tercer curso, además de los colores morado y naranja en las asignaturas de primer año, colores asociados a las categorías “2” y “3”, lo que indicaría suspenso y no presentado respectivamente.

Cabe destacar el hueco amarillo situado a la derecha de la imagen, el cual realiza un corte transversal muy evidente. Este se corresponde con los códigos de las asignaturas optativas, cuya situación hace que no tengan que ser todas ellas matriculadas a lo largo del recorrido académico, sino tan solo tres como máximo.

Una lectura transversal de la Figura 9 permite observar el rendimiento del conjunto de estudiantes de la muestra curso a curso. Así, se pasará a centrar la atención en el curso

objeto de estudio, que se encuentra señalado con el rectángulo rojo y comprende las asignaturas cuyos códigos se encuentran entre “584” y “588”.

Figura 10. Variación de los estados de las asignaturas. Primer Curso. Grado en Estadística Aplicada



La Figura 10 representa precisamente un extracto sobre el primer curso de Estadística, que es lo mismo que se presenta en la Figura 9, pero aislado.

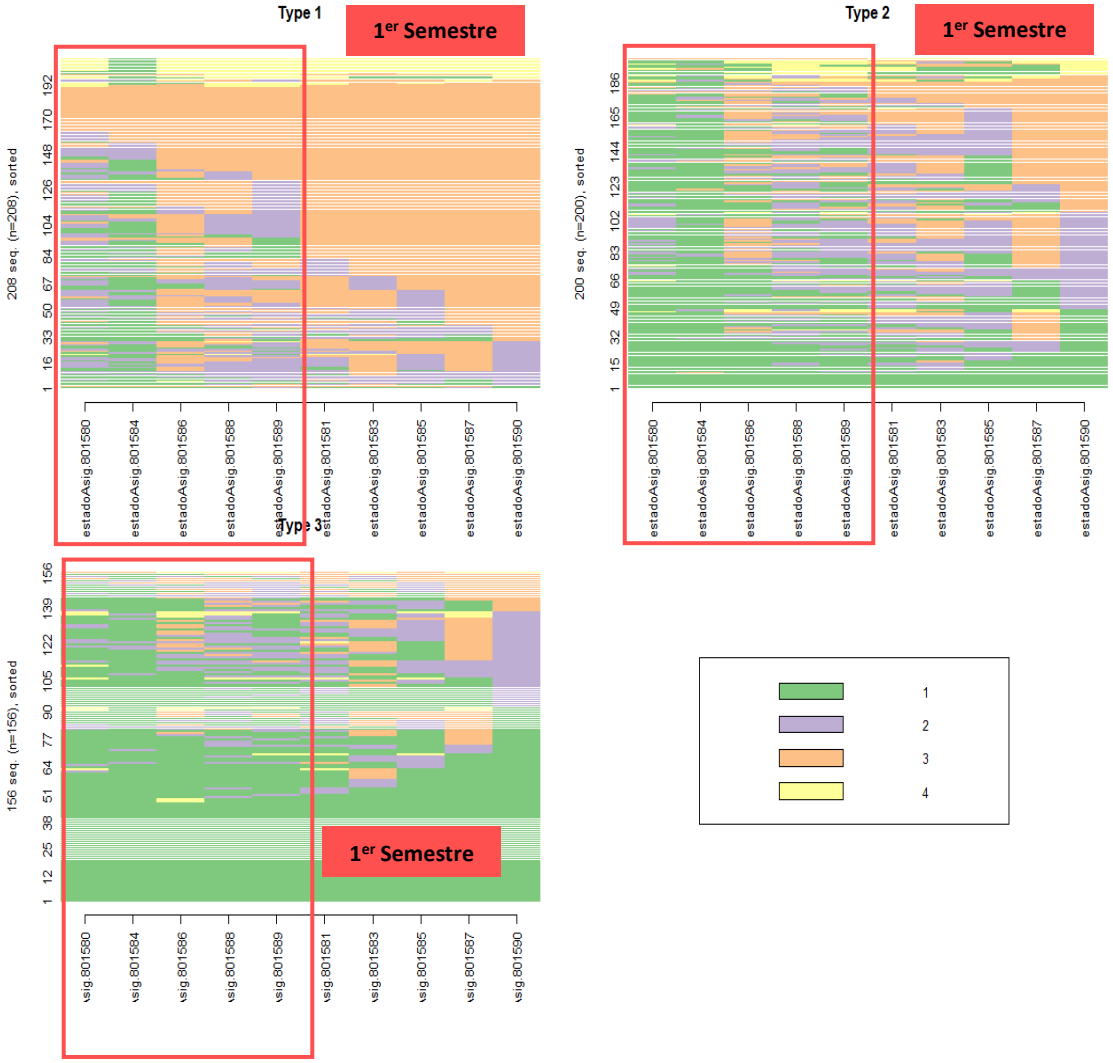
En una mirada rápida, resulta sencillo percibir que las asignaturas con un mayor número de aprobados, en color verde, son tanto las acabadas en “584” (*Fuentes y Técnicas de Recogida de Información en Investigación Social y de Mercados*) como “580” (*Descripción y Exploración de Datos*), mientras que no suponen pocas las que *a priori* parecen presentar una dificultad mayor dada la cantidad de suspensos y no presentados, pudiéndose destacar las asignaturas cuyos códigos finalizan en “590” (*Métodos Matemáticos para la Estadística III*) y “583” (*Software Estadístico I*).

Para tratar de hacer más comprehensiva la información presente en la Figura 10, y con la finalidad de hallar tipologías de alumnos con respecto a su rendimiento en el primer año de la carrera, tal y como se avanzaba en la introducción de este apartado, se han construido clústeres jerárquicos por el método de Ward mediante la función “agnes()” de R para, después de realizar una serie transformaciones internas sobre la matriz de entrada, habilitar la obtención de individuos agrupados en base a la variación de sus estados de asignatura.

Tras la construcción y representación de varias posibilidades en cuanto a número de clústeres, se ha optado por la realización del clustering con tres grupos, los cuales

permiten observar mejor las variaciones entre individuos y establecer diferencias de rendimiento en el seno del primer curso, tal y como se muestra en la Figura 11.

Figura 11. Variación de los estados de las asignaturas. Clústeres Primer Curso. Grado en Estadística Aplicada



Los clústeres generados contienen un total de 208, 200 y 156 individuos cada uno de ellos.

En el primero, se puede observar que es donde, *a priori*, parecen quedar agrupados aquellos que se podría entender que podrían causar baja transcurridos dos años desde la matriculación, ya que apenas se observan asignaturas aprobadas y predomina la presencia de asignaturas no presentadas, pareciendo incluso aumentar la no presentación a las evaluaciones a partir del segundo semestre, que comienza a partir de la asignatura finalizada con el código “581” (*Azar y Probabilidad*), y hacia la derecha. El hecho de que este primer grupo esté compuesto por 208 estudiantes, muy cercano al número de 219 que se ha considerado que causan abandono tras el primer año, parece indicar que, efectivamente, la mayoría de los que abandonen quedarán encuadrados en este primer grupo.

El segundo grupo parece estar conformado por un pequeño grupo de personas que no han tenido problema alguno para afrontar el curso, mientras que la mayoría de los 200 integrantes de este grupo tienen una o varias asignaturas suspensas o no presentadas, pero sin un patrón homogéneo que caracterice este no-aprobado. De este grupo deberían surgir aquellos que acaban abandonando y que no se encuentran incluidos dentro del clúster 1.

El tercer grupo muestra con cierta claridad a los estudiantes con menores dificultades para superar las asignaturas en el momento de extracción de los datos, pudiéndose observar varios de ellos sin ninguna asignatura suspensa y varios suspensos localizados en la asignatura *Métodos Matemáticos para la Estadística III*. Aquí, por tanto, podría parecer que esta porción del alumnado no debería causar baja a lo largo del desarrollo del plan de estudios o, al menos, no abandono tras el primer año.

Este análisis visual basado en la representación de las variaciones de los estados ha permitido realizar una panorámica sobre la trayectoria académica del alumnado del Grado en Estadística Aplicada. Se ha conseguido mostrar que parecen existir una serie de patrones de agrupamiento ciertamente relevantes que posiblemente clasifiquen a los y las alumnas en cuanto a un posible futuro abandono tras el primer año cursado. Sin embargo, aseverar que un estudiante queda encuadrado en un determinado clúster no constituye una condición *sine qua non* para afirmar que abandonará o no, ya que esa cuestión queda reservada para los modelos de predicción que se desarrollarán a partir del siguiente apartado.

7. MODELADO DE LA PREDICCIÓN DEL ABANDONO EN EL GRADO DE ESTADÍSTICA APLICADA EN LA UCM

Tras los preámbulos, la exposición de la metodología a seguir y la exploración inicial de los datos, el siguiente paso consiste en proceder a elaborar y evaluar los modelos de predicción correspondientes, empleando para ello las técnicas descritas en el apartado 5.2.

La elaboración de los modelos no se realizará en bruto con todas las variables disponibles en el conjunto de datos final, sino que, tal y como se suele indicar para el modelado con técnicas de Machine Learning, ha de realizarse una selección previa de variables mediante distintas técnicas con la finalidad de evitar sobrecargar los modelos con variables irrelevantes para la dimensión objetivo que puedan introducir ruido y alteren la capacidad predictiva de estos (Portela, 2019).

Así, el proceso para hallar los modelos que mejor permitan predecir el abandono de primer año en el Grado en Estadística Aplicada será el siguiente:

- En primer lugar, se realizará la mencionada selección de variables mediante distintas técnicas, para hallar las variables más relevantes con respecto al objeto de estudio.
- En segundo lugar, se hará un recorrido por todas las técnicas expuestas, modificando los parámetros descritos también en el apartado 5.2. para cada una de ellas, con la finalidad de hallar el modelo más competitivo en cuanto a sesgo y varianza, tras realizar validación cruzada repetida con diferentes semillas, en términos de la tasa de fallos (porcentaje de casos mal clasificados) y el área bajo la curva ROC (sensibilidad del modelo frente a la especificidad).
- Finalmente, se realizará una comparación de los mejores modelos por cada técnica para seleccionar el modelo global que permita hallar con un menor error el abandono del estudiantado en el mencionado Grado.

De esta manera, se procede realizar la selección inicial de variables para posteriormente elaborar y evaluar los mejores modelos.

7.1 SELECCIÓN INICIAL DE VARIABLES

Para la realización del filtrado de variables inicial, se ejecutarán, en primer lugar, algunas técnicas, concretamente cuatro, con menor complejidad que los modelos finales de predicción, cuya finalidad será establecer un ranking de variables seleccionadas que aparezcan una mayor cantidad de veces repartidas entre todas las técnicas.

Cabe destacar que, para esta selección de variables, y de ahora en adelante en el trabajo, no se emplearán las variables creadas “notaMedia1Q”, “notaMedia2Q” ni “notaMedia1A”, ya que algunas pruebas preliminares indican que, debido a la alta correlación entre el valor de estas variables y las de las notas máximas de las asignaturas en cada tramo del curso, se corre el riesgo de que los algoritmos opten por estas variables y dejen de lado las de las asignaturas individuales. Ello haría difícil, de cara a las recomendaciones finales, afinar asignatura a asignatura, resultando en una

perdida de la granularidad buscada para trazar las pautas de conducta en términos de rendimiento académico del alumnado.

Así, y en primer lugar, se llevan a cabo varias regresiones logísticas, con el procedimiento de selección y entrada de variables paso-a-paso o *stepwise*, repetida varias veces con semillas diferentes (un total de 235 veces, lo que implica 235 semillas consecutivas), a través de la macro de SAS *%randomselectlog*, facilitada por el profesor Portela (2019), con todas las variables continuas y de clase del conjunto de datos. La salida de esta macro es una tabla que indica la cantidad de veces que una combinación de variables ha resultado elegida a lo largo de todas las regresiones logísticas ejecutadas.

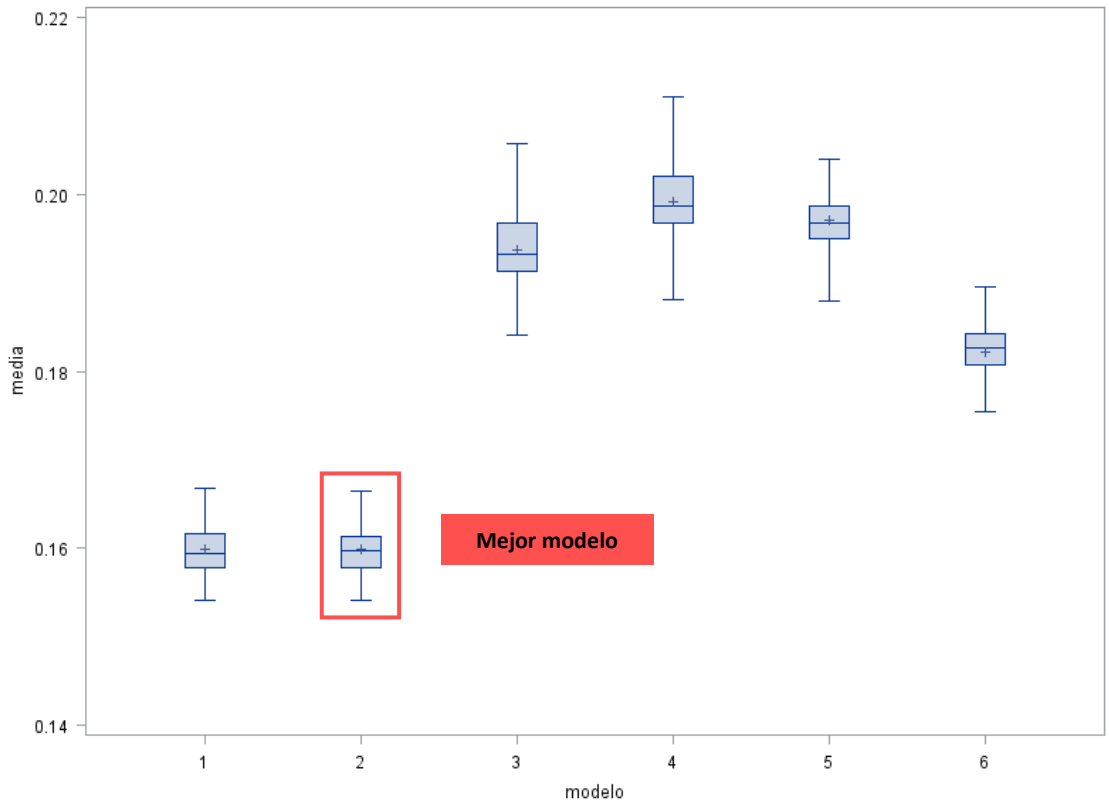
Tabla 6. Repetición de combinaciones por %randomselectlog. Mayor número de repeticiones

Efecto	Frecuencia Cuenta	Porcentaje de Frecuencia Total
cursada_801588 estadoAsig_801585 notaMax_801580 notaMax_801589 notaMax_801590	14	5.9322
cursada_801588 notaMax_801580 notaMax_801585 notaMax_801589 notaMax_801590	12	5.08475
cursada_801588 notaMax_801580 notaMax_801583 notaMax_801589 notaMax_801590	8	3.38983
cursada_801588 estadoAsig_801583 notaMax_801580 notaMax_801588	6	2.54237
cursada_801588 estadoAsig_801585 notaMax_801580 notaMax_801583 notaMax_801589	6	2.54237
cursada_801588 notaMax_801580 notaMax_801583 notaMax_801588	6	2.54237
cursada_801588 notaMax_801580 notaMax_801589 notaMax_801590	5	2.11864
cursada_801588 notaMax_801580 notaMax_801583 notaMax_801588 notaMax_801590	4	1.69492
cursada_801588 notaMax_801580 notaMax_801586 notaMax_801589 notaMax_801590	4	1.69492

La Tabla 6 muestra, por tanto, las combinaciones de variables que han aparecido un mayor número de veces tras ejecutar la macro y que servirán de referencia para la próxima operación, la cual consiste en probar las combinaciones de variables a través de validación cruzada repetida para poder evaluar cómo se comportan estas en términos de sesgo y varianza con respecto a la tasa de fallos. Para ello, se probarán todas las combinaciones que se han repetido al menos en seis ocasiones.

La realización de este proceso de elaboración de modelos logísticos por validación cruzada repetida implicará la partición de la muestra en 10 grupos, aplicándose 20 repeticiones con semillas consecutivas, gracias a la macro *%cruzadalogistica*, también ejecutada bajo el entorno SAS. La salida se puede comprobar en la Figura 12.

Figura 12. Tasa de fallos de modelos *%cruzadalogistica* con mayores frecuencias en la Tabla 6



De entre los modelos probados, tanto el 1 como el 2, que se corresponden con los que habían obtenido mayores frecuencias con *%randomselectlog*, son lo que presentan un menor sesgo y, prácticamente, también varianza con respecto al resto de modelos probados. Con este empate técnico, se ha optado por el modelo 2 ya que parece presentar un sesgo global ligeramente más reducido.

Una vez elegido el modelo que parece resultar más ventajoso en regresión logística por validación cruzada repetida, se procede a probar, todo en SAS Enterprise Miner, un árbol simple de decisión (profundidad = 6; tamaño de hoja = 25; sin poda), así como un único modelo de Gradient Boosting (iteraciones = 50; shrinkage = 0.1), sin validación cruzada y un nodo de selección de variables para comprobar, tentativamente, qué variables entienden estos modelos que son aquellas que tienen una mayor relación o presentan una mayor importancia con respecto a la variable objetivo.

Del proceso anterior, junto con la operación llevada a cabo con la regresión logística, se obtiene la Tabla 7, que indica todas aquellas variables que aparecen resaltadas en al menos dos técnicas diferentes.

Tabla 7. Relación de técnicas probadas, variables seleccionadas y número de apariciones

Variables Escogidas	Regresión Logística	Gradient Boosting	Árbol de Clasificación	Selección de variables	Cuenta
notaMax_801580	notaMax_801580	notaMax_801580	notaMax_801580	notaMax_801580	4
notaMax_801585	notaMax_801585	notaMax_801585	notaMax_801585	notaMax_801585	4
cursada_801585	-	cursada_801585	cursada_801585	cursada_801585	3
cursada_801588	cursada_801588	cursada_801588	-	cursada_801588	3
notaMax_801588	-	notaMax_801588	notaMax_801588	notaMax_801588	3
notaMax_801590	notaMax_801590	notaMax_801590	-	notaMax_801590	3
notaMax_801589	notaMax_801589	notaMax_801589	notaMax_801589	-	3
IMP_almacenaPAU	-	IMP_almacenaPAU	-	IMP_almacenaPAU	2
almacenaCursoEntrada	-	almacenaCursoEntrada	-	almacenaCursoEntrada	2
cursada_801589	-	cursada_801589	-	cursada_801589	2
estadoAsig_801581	-	estadoAsig_801581	-	estadoAsig_801581	2
estadoAsig_801586	-	estadoAsig_801586	-	estadoAsig_801586	2
notaMax_801586	-	notaMax_801586	notaMax_801586	-	2

De la tabla anterior se podrían deducir, por tanto, tres conjuntos reducidos de datos diferentes, en función de la cantidad de apariciones que realizan las variables en las diferentes técnicas. La decisión que se toma en primera instancia, de esta manera, es escoger el conjunto menos restrictivo de los tres posibles, es decir, el conformado por todas aquellas variables que aparecen en al menos dos técnicas distintas, un total de 13, con el fin de no restringir posibles efectos de variables menores pero que pudiesen tener algún tipo de influencia igualmente de cara al abandono.

Una vez tomada la decisión de la conformación de los conjuntos de variables que servirán como input para los modelos, se puede proceder a la ejecución y evaluación de estos mediante las técnicas de Machine Learning anunciadas.

El medio de ejecución de los algoritmos será a través de R, en el entorno RStudio, con parte de las funciones de validación cruzada repetida proporcionadas, una vez más, por Portela (2019). En la mayoría de las técnicas, se partirá de una serie de pruebas base con los parámetros que correspondan a cada técnica para, posteriormente, ir afinando los resultados iniciales obtenidos y tratar de conseguir el modelo más competitivo en términos de sesgo-varianza.

Con los modelos ganadores, se realizará una descripción de sus niveles de acierto a través de las métricas que se desprendan de la matriz de confusión de cada modelo, tales como la sensibilidad o la especificidad.

Finalmente, tras haber realizado un recorrido exhaustivo por las técnicas propuestas con el conjunto de 13 variables, a continuación se volverán a ejecutar los modelos con las mismas combinaciones de parámetros en cada técnica, pero en esta ocasión, con el conjunto más restringido de 7 variables, simplemente poniendo en perspectiva todos los

modelos y eligiendo aquel que posea un mejor comportamiento, para posteriormente pasar a evaluar las variables más relevantes en seno del mismo.

7.2 REGRESIÓN LOGÍSTICA

En primer lugar, se lleva a cabo la regresión logística, ya que esta constituye el modelo lineal básico a partir del cual resultaría óptimo comparar con el resto de modelos de Machine Learning, más capaces de captar relaciones no lineales, además de ofrecer un modelo con un mayor grado de interpretabilidad.

Así, como excepción frente al resto de métodos a aplicar, se llevará a cabo esta parte con el software SAS Base, a través de las macro *%cruzadalogistica*, facilitada por Portela (2019) para elegir el mejor modelo de entre todas las intersecciones de variables posibles presentes en la Tabla 7, mientras que se aplicará el procedimiento *proc logistic* para obtener los parámetros y medidas de ajuste del mejor modelo de regresión logística.

También, como excepción frente a la estructura de análisis planteada en el punto anterior, en esta técnica se probarán los tres posibles conjuntos de variables obtenidos en la Tabla 7.

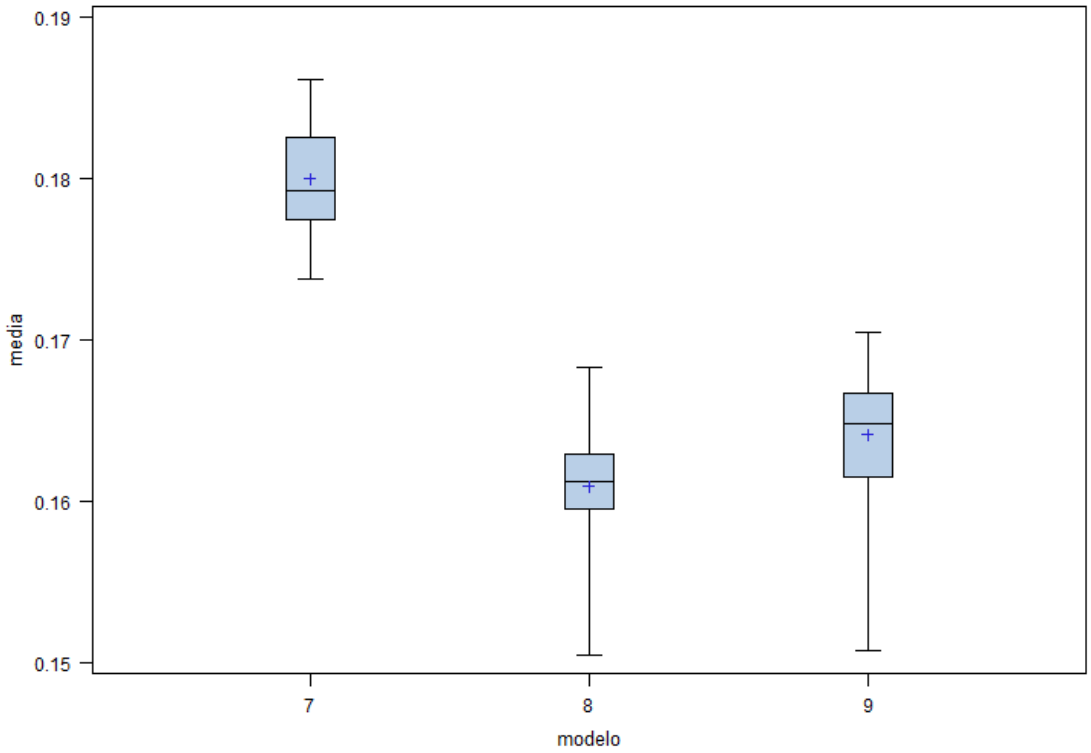
Por tanto, se ejecutan un total de 3 modelos por validación cruzada repetida, mediante la macro *%cruzadalogistica*, con 10 grupos de partición y 40 semillas consecutivas diferentes, variando las semillas de inicio, como se muestra en la Tabla 8.

Tabla 8. Relación de modelos de Regresión Logística probados

Logística1	Logística2	Logística3
notaMax_801580	notaMax_801580	notaMax_801580
notaMax_801585	notaMax_801585	notaMax_801585
	cursada_801585	cursada_801585
	cursada_801588	cursada_801588
	notaMax_801588	notaMax_801588
	notaMax_801590	notaMax_801590
	notaMax_801589	notaMax_801589
		IMP_almacenaPAU
		almacenaCursoEntrada
		cursada_801589
		estadoAsig_801581
		estadoAsig_801586
		notaMax_801586

A continuación, se muestra el boxplot generado para cada una de las combinaciones de variables con las que se ha realizado la regresión logística:

Figura 13. Tasa de fallos. Modelos de Regresión Logística según combinaciones de la Tabla 8



De todos los modelos probados, mostrados en la Figura 13, se puede observar que tanto la segunda como la tercera combinación son las más competitivas en términos de sesgo y varianza con respecto a la tasa de fallos. Se podría plantear que el modelo más competitivo es el modelo 8 o "Logística2", con el menor sesgo y la menor varianza, cuya mediana indica que el 50% de las repeticiones realizadas estarían por debajo de un 16.5% de casos mal clasificados, algo por debajo del modelo 9 o "Logística3".

Tabla 9. Resumen del proceso de Regresión Logística por selección Stepwise del modelo "Logística2"

Step	Effect		DF	Number	Score	Wald	Pr > ChiSq
	Entered	Removed		In	Chi-Square	Chi-Square	
1	**cursada_801585**		2	1	237.0309		<.0001
2	**notaMax_801588**		1	2	60.9828		<.0001
3	**cursada_801588**		2	3	21.117		<.0001
4	**notaMax_801580**		1	4	17.4426		<.0001
5	**notaMax_801585**		1	5	9.956		0.0016
6	**notaMax_801589**		1	6	6.6581		0.0099
7	**notaMax_801590**		1	7	5.9612		0.0146
8		**notaMax_801588**	1	6		2.2217	0.1361

De la Tabla 9 se deduce que, de las siete variables introducidas en el modelo, llevado a cabo por el procedimiento "stepwise", finalmente el método de SAS ha optado por tomar como relevantes o significativas seis de ellas (p<.05):

- **cursada_801585 y notaMax801585:** Cuántas veces se ha cursado y la nota máxima obtenida en la asignatura “Estadística Económica”.
- **cursada_801588:** Cuántas veces se ha cursado la asignatura “Métodos Matemáticos para Estadística I”.
- **notaMax801580:** Máxima nota obtenida en la asignatura “Descripción y Exploración de Datos”.
- **notaMax801589:** Máxima nota obtenida en la asignatura “Métodos Matemáticos para Estadística II”.
- **notaMax801590:** Máxima nota obtenida en la asignatura “Métodos Matemáticos para Estadística III”.

A continuación, en la Tabla 10, se muestra el análisis de máxima verosimilitud de los estimadores o parámetros asociados a la Regresión Logística.

Tabla 10. Análisis de máxima verosimilitud y estimadores del modelo “Logística2”

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Exp(B)
Intercept		1	1.3228	0.3578	13.667	0.0002	
cursada_801585	**0**	1	0.8989	0.2481	13.1221	0.0003	6.192
cursada_801585	**1**	1	0.0256	0.1988	0.0165	0.8977	2.586
cursada_801588	**0**	1	0.4964	0.2231	4.949	0.0261	5.502
cursada_801588	**1**	1	0.7123	0.2084	11.6806	0.0006	6.827
notaMax_801580		1	-0.2121	0.0596	12.6561	0.0004	0.809
notaMax_801585		1	-0.2017	0.078	6.692	0.0097	0.817
notaMax_801590		1	-0.2605	0.0816	10.1882	0.0014	0.771
notaMax_801589		1	-0.1894	0.0576	10.8259	0.001	0.827

Como se puede observar, hay un único estimador que no es estadísticamente significativo (categoría 1 de “cursada_801585”), mientras que en el conjunto de las asignaturas cuya importancia viene mediada por la nota máxima, tomando como referencia la asignatura “Descripción y Exploración de Datos” (801580), por cada punto que se incrementa dicha nota, se reducen un 23.61% las probabilidades de causar abandono tras el primer año cursado del grado. Sin embargo, para el caso de no asistir a ninguna convocatoria en la asignatura “Estadística Económica” (801585), las probabilidades de abandonar se disparan en un 83.85% frente a acudir a dos convocatorias en un mismo año.

De esta manera, en Regresión Logística, y una vez elaborados y analizados los modelos con todas las combinaciones posibles de variables, se puede observar que el conjunto más competitivo es aquel conformado por tan solo siete variables, con una tasa de fallos de 0.1609 o, lo que es lo mismo, un 16.1% de casos mal clasificados en la predicción.

7.3 REDES NEURONALES

Las redes neuronales, como se ha comentado en el apartado de metodología, pueden ser francamente útiles para hallar relaciones intrínsecas no lineales en conjuntos de complejidad diversa.

No obstante, requieren de un ajuste cuidadoso para evitar la sobreparametrización excesiva con el conjunto de datos de entrenamiento y que luego puedan fallar estrepitosamente con el conjunto de datos de test, dando lugar a una capacidad predictiva mermada con respecto a un buen ajuste.

Para ello, dentro de los parámetros a ajustar en la red neuronal dentro del paquete "Caret" de R, con el procedimiento por "avNNet", toma especial importancia el número de nodos ocultos que puede soportar la red en la única capa oculta entre la capa de entrada y de salida de la misma.

La determinación del valor del número de nodos ocultos en la red dependerá, por tanto, del número global de casos, ya que el número de nodos y, en consecuencia, el número de parámetros derivados de la construcción de la red ha de guardar una relación de casos por parámetro no menor de 5, y hasta un supuesto ideal de 25 casos por parámetro.

¿Cómo calcular el número ideal de nodos? Existe una fórmula que permite calcular esta relación aproximada, de la manera que sigue:

$$N\ parametros\ =\ h(k+1)+h+1 \mid h{:}\ nodos\ ocultos, k{:}\ variables\ input$$

Como resultado inicial, y asumiendo un número mínimo de 9 casos por parámetro, lo que resulta en un máximo de 62.7 parámetros, se encuentra que la red neuronal que se quiere construir podría soportar hasta un total de cuatro nodos ocultos. Por tanto, lo ideal sería probar distintas configuraciones de la red neuronal desde dos hasta cuatro nodos ocultos para el caso concreto de este conjunto de datos. No obstante, Portela (2019) indica que, para posibles relaciones no lineales, el mínimo número de nodos a probar debería ser tres, lo que acota sustancialmente el número de pruebas que han de realizarse.

Tabla 11. Relación de variables input, nodos ocultos y máximo N de parámetros en la red neuronal

Input (k)	Nodos ocultos (h)	Parámetros (N)
13	3	46
13	4	61
13	5	76

En la tabla 11 se puede observar que, con el conjunto de datos planteado, tan solo se puede plantear redes neuronales de 3 y 4 nodos ocultos ya que, con la casuística de 5 nodos, se excedería el umbral máximo de parámetros para la relación de casos por variable anteriormente mencionada.

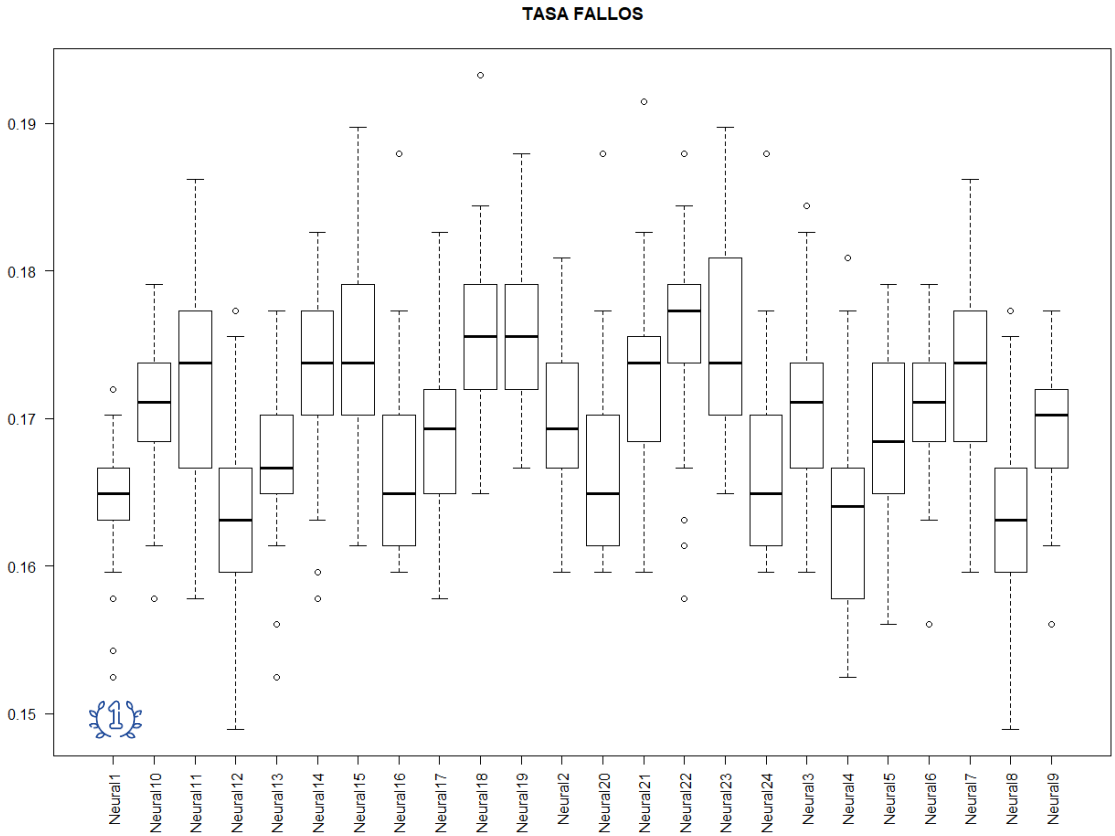
A continuación, en la Tabla 12 se muestran los modelos más relevantes llevados a cabo con las redes neuronales, siempre por validación cruzada repetida, con 10 grupos y, en este caso, 30 semillas, empleando la función “cruzadaavnnetbin”.

Tabla 12. Relación de modelos probados. Redes Neuronales mediante validación cruzada repetida

Modelo	Nodos ocultos	Decay	Iteraciones	Modelo	Nodos ocultos	Decay	Iteraciones
Neural1	3	0.001	100	**Neural13**	4	0.001	100
Neural2	3	0.005	100	**Neural14**	4	0.005	100
Neural3	3	0.01	100	**Neural15**	4	0.01	100
Neural4	3	0.1	100	**Neural16**	4	0.1	100
Neural5	3	0.001	200	**Neural17**	4	0.001	200
Neural6	3	0.005	200	**Neural18**	4	0.005	200
Neural7	3	0.01	200	**Neural19**	4	0.01	200
Neural8	3	0.1	200	**Neural20**	4	0.1	200
Neural9	3	0.001	500	**Neural21**	4	0.001	500
Neural10	3	0.005	500	**Neural22**	4	0.005	500
Neural11	3	0.01	500	**Neural23**	4	0.01	500
Neural12	3	0.1	500	**Neural24**	4	0.1	500

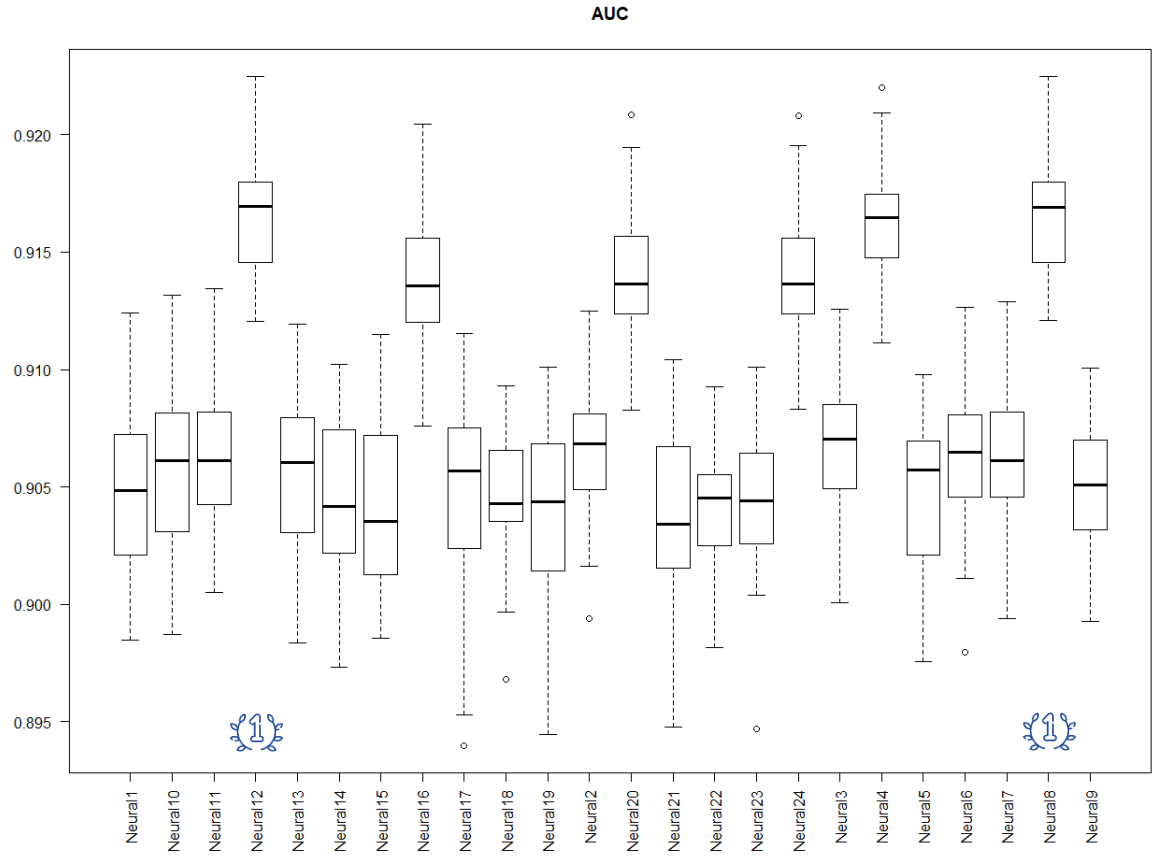
Así, derivado de estas combinaciones principales con 3 y 4 nodos, se obtienen los siguientes gráficos de cajas que expresan, de nuevo, las relaciones sesgo-varianza tanto para la tasa de fallos como para el área bajo la curva ROC.

Figura 14. Tasa de fallos. Modelos de Redes Neuronales



Observando la tasa de fallos, se encuentra que el modelo que parece ser más competitivo es “Neural1” ya que, si bien es cierto que no presenta el menor sesgo de todos, con alrededor del 16,5% de casos mal clasificados, la varianza de este modelo es sin duda la menor de todos los modelos probados. Esto le convertiría, *a priori*, en el mejor modelo de Redes Neuronales para el conjunto de datos probados.

Figura 15. Área bajo la curva ROC. Modelos de Redes Neuronales



Cuando se comprueba el comportamiento de los modelos en cuanto al área bajo la curva ROC, la realidad cambia la fotografía antes descrita, puesto que el modelo “Neural1” no se muestra tan competitivo en la relación entre la sensibilidad y la especificidad, donde parecen imponerse los modelos “Neural12” y “Neural8”, ya que no existe diferencia entre ellos. No en vano, son los modelos que habían obtenido un menor sesgo en mediana en cuanto a la tasa de fallos.

De esta manera, se considera que, al mostrarse claramente ventajosos en AUC, con una de las menores varianzas, y acercarse tanto al mejor en cuanto a la tasa de fallos, se tomarán los modelos “Neural12” y “Neural 8” como ganadores en el apartado de Redes Neuronales. Por simplificar, se tomará solamente para las futuras operaciones el modelo “Neural8”.

Una vez elegido el mejor modelo de regresión logística, pasa a exponerse la matriz de confusión asociada al mismo, a través del método de la librería “Caret”, “confusionMatrix”.

A continuación, se muestran tanto la matriz de confusión del mejor modelo, como las medidas asociada a la misma:

Tabla 13. Matriz y medidas de confusión. Mejor modelo Redes Neuronales (Neural8)

Matriz de confusión				
			Exactitud (accuracy)	0.8304
	Referencia		**Sensibilidad**	0.788
Predicción	**No**	**Yes**	**Especificidad**	0.8574
No	8874	1393	**VPP**	0.7781
Yes	1476	5176	**VPN**	0.8646

Las medidas obtenidas en la Red Neuronal son muy parecidas a las obtenidas por la Regresión Logística donde, en términos de exactitud, hay un ligero empeoramiento a partir del tercer decimal. De esta manera, se obtiene que se clasificarán correctamente como alumnos o alumnas en situación de abandono en un 78.8% de los casos, mientras que se predecirá correctamente como estudiantes que continúan tras el primer año en un 85.74% de las ocasiones.

7.4 RANDOM FOREST Y BAGGING

En este apartado se van a ejecutar, de nuevo, diferentes combinaciones de los parámetros expuestos en el apartado 5.2.4. tanto para Random Forest como para Bagging, de nuevo por el procedimiento de validación cruzada repetida, también con 30 repeticiones y submuestreo en 10 grupos, empleando la función “cruzadarfbin”.

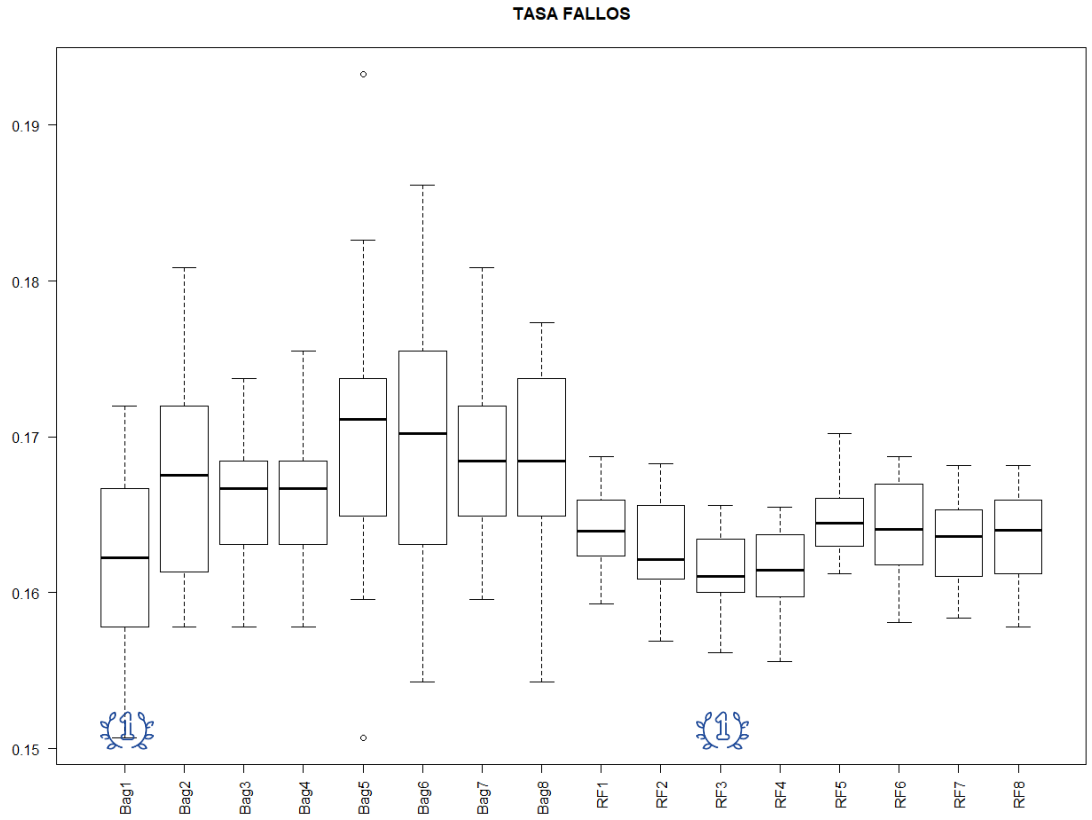
Tabla 14. Relación de modelos probados. Random Forest y Bagging mediante validación cruzada repetida

Modelo	mtry	nodesize	ntree	sampsize
RF1	4	10	50	450 (0.8)
RF2	2	10	100	450 (0.8)
RF3	2	10	500	450 (0.8)
RF4	2	10	1000	450 (0.8)
RF5	2	20	50	507 (0.9)
RF6	2	20	100	507 (0.9)
RF7	2	20	500	507 (0.9)
RF8	2	20	1000	507 (0.9)
Modelo	**mtry**	**nodesize**	**ntree**	**sampsize**
Bag1	13	10	50	450 (0.8)
Bag2	13	10	100	450 (0.8)
Bag3	13	10	500	450 (0.8)
Bag4	13	10	1000	450 (0.8)
Bag5	13	20	50	507 (0.9)
Bag6	13	20	100	507 (0.9)
Bag7	13	20	500	507 (0.9)
Bag8	13	20	1000	507 (0.9)

En Random Forest, uno de los parámetros más importantes que hay que configurar y afinar es el número de variables aleatorias (mtry) que intervienen en la ejecución del algoritmo. Para no complejizar la presentación de los modelos probados, presentes en la Tabla 14, inicialmente se han probado todas las combinaciones de número de variables (de 2 hasta 12) con el número de árboles, el tamaño de la hoja final de los árboles y el porcentaje de reemplazo, eligiendo para ello las combinaciones con mayor exactitud (accuracy), para finalmente ajustar y registrar una única combinación de número de variables y poder comparar los mejores modelos entre sí.

Como Bagging constituye un caso particular de Random Forest, en tanto que el parámetro “mtry” es igual al número de variables que se introducen inicialmente, este no ha exigido ninguna prueba previa.

Figura 16. Tasa de fallos. Modelos de Random Forest y Bagging

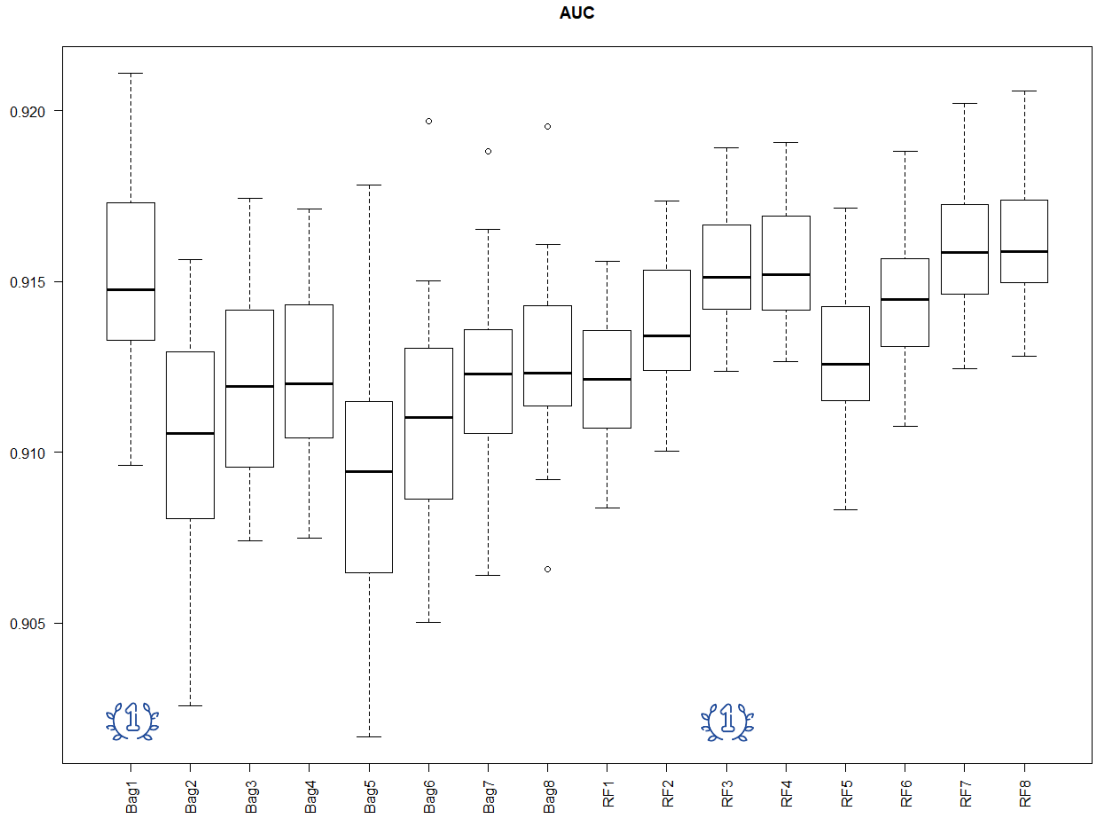


En la figura 16 se pueden observar los modelos ejecutados con ambas técnicas. En Bagging, el modelo que se muestra más competitivo en términos de sesgo en cuanto a la tasa de fallos es “Bag1”, que finalmente se toma como el mejor realizado con esta técnica. Esto se debe a que, aunque presenta una varianza superior frente a otros modelos, aunque no mucho mayor, presenta un sesgo menor en probablemente más de medio punto.

En Random Forest, la elección del mejor modelo por la tasa de fallos resulta más evidente, en tanto que es “RF3” el que, de entre los dos con menor sesgo, plantea una ligera menor varianza.

Por el lado del área bajo la curva ROC, cuyos modelos se encuentran en la Figura 17, en Bagging, el modelo que se destaca claramente es “Bag1”, destacándose en cuanto a un área global mayor bajo dicha curva, con un valor de 0.915 en su mediana. Sin embargo, en Random Forest, la cuestión se encuentra ligeramente más disputada, ya que se podría hablar de empate técnico entre “RF7” y “RF8”, con una ligera ventaja para este último. Esto hace que no coincidan los modelos entre la tasa de fallos y el área bajo la curva ROC. Sin embargo, dada la escasa diferencia en esta cifra del modelo “RF3” con los anteriores y, al presentar este una menor varianza, además del mejor rendimiento por el lado de la tasa de fallos o clasificación errónea, se toma como mejor modelo global de Random Forest a “RF3”.

Figura 17. Área bajo la curva ROC. Modelos de Random Forest y Bagging



Una vez diferenciados los mejores modelos de Bagging y Random Forest, de cara a plasmar la matriz de confusión y sus medidas asociadas del mejor de este apartado, se puede tomar como modelo ganador “RF3” ya que, en términos globales, presenta un mejor rendimiento tanto en la tasa de fallos como en el área bajo la curva ROC.

Tabla 15. Matriz y medidas de confusión. Mejor modelo Random Forest y Bagging (RF3)

Matriz de confusión				
			Exactitud (accuracy)	0.8436
	Referencia		**Sensibilidad**	0.7679
Predicción	**No**	**Yes**	**Especificidad**	0.8917
No	9229	1525	**VPP**	0.8182
Yes	1121	5045	**VPN**	0.8582

Por el lado de las medidas asociadas a la matriz de confusión, se puede observar que, en este modelo, la exactitud (accuracy) es algo más alta que en el caso de las Redes Neuronales, motivado por el significativo aumento de la especificidad, donde un 89.17% de los casos que se indica que permanecerán en el grado, efectivamente no abandonarán. En el caso de la sensibilidad, en un 76.79% de las ocasiones en que se prediga que un alumno abandonará, esto efectivamente así sucederá.

7.4 GRADIENT BOOSTING Y XGBOOST

De entre todos los algoritmos probados, tanto Gradient Boosting como Xgboost se les presupone una mayor potencia y capacidad predictiva, puesto que presentan una gran adaptabilidad a los datos y van corrigiendo las predicciones en el sentido de la reducción del error.

Bajo este paraguas, se procede a presentar los modelos principales y más relevantes de ambos algoritmos que se han empleado para elaborar los modelos de predicción subsiguientes, en formato de validación cruzada repetida, con 30 repeticiones (semillas consecutivas) y partición de la muestra en 10 grupos de validación cruzada, gracias a las funciones “gradientboostingbin” y “xgboostbin”, desarrolladas por Portela (2019).

Al igual que se ha hecho en la sección de Random Forest y Bagging, antes de elegir los modelos finales que compiten entre sí, se ha realizado una parrilla de modelos con shrinkage entre 0.001, 0.01, 0.05 y 0.1 para el total de combinaciones mostradas, eligiéndose para los ajustes finales aquellos cuya exactitud (accuracy) fuese mayor.

Tabla 16. Relación de modelos probados. Gradient Boosting mediante validación cruzada repetida

Modelo	n.trees	max.depth	shrinkage	n.minobsinnode
GBM1	100	4	0.05	5
GBM2	100	4	0.05	10
GBM3	100	4	0.05	20
GBM4	1000	4	0.01	5
GBM5	1000	4	0.01	10
GBM6	1000	4	0.01	20
GBM7	5000	4	0.001	5
GBM8	5000	4	0.001	10
GBM9	5000	4	0.001	20
GBM10	100	5	0.05	5
GBM11	100	5	0.05	10
GBM12	100	5	0.05	20
GBM13	1000	5	0.01	5
GBM14	1000	5	0.01	10
GBM15	1000	5	0.01	20
GBM16	5000	5	0.001	5
GBM17	5000	5	0.001	10
GBM18	5000	5	0.001	20

La diferencia esencial entre Gradient Boosting normal y Xgboost, como se adelantaba en el apartado 5.2.5., es la posibilidad de llevar a cabo la regularización, hecho que, en esencia, diferencia las combinaciones presentes en las Tablas 16 y 17 con la introducción de los parámetros Gamma y Lambda.

Para realizar la regularización, se ha tomado el modelo más competitivo de Xgboost y se han probado varias combinaciones de Gamma y Lambda hasta obtener la combinación más ventajosa con mayor *accuracy*.

Tabla 17. Relación de modelos probados. Xgboost mediante validación cruzada repetida

Modelo	n.trees	max.depth	shrinkage	n.minobsinnode	Gamma	Lambda
XGB1	100	6	0.05	5	0	0
XGB2	100	5	0.05	5	0	0
XGB3	100	4	0.05	5	0	0
XGB4	1000	6	0.05	5	0	0
XGB5	1000	5	0.05	10	0	0
XGB6	1000	4	0.05	20	0	0
XGB7	5000	6	0.1	5	0	0
XGB8	5000	5	0.1	10	0	0
XGB9	5000	4	0.1	20	0	0
XGB10	100	6	0.03	5	0	0
XGB11	100	5	0.03	10	0	0
XGB12	100	4	0.01	20	0	0
XGB13	1000	6	0.05	5	0	0
XGB14	1000	5	0.05	10	0	0
XGB15	1000	4	0.01	20	0	0
XGB16	5000	6	0.03	5	0	0
XGB17	5000	5	0.1	10	0	0
XGB18	5000	4	0.03	20	0	0
XGB19	100	5	0.05	10	1	1
XGB20	100	5	0.05	10	0	1

Una vez obtenidos los modelos, se muestran los resultados de los diferentes ajustes en términos de tasa de fallos para pasar a analizar su relación sesgo-varianza y seleccionar el más competitivo de todos ellos.

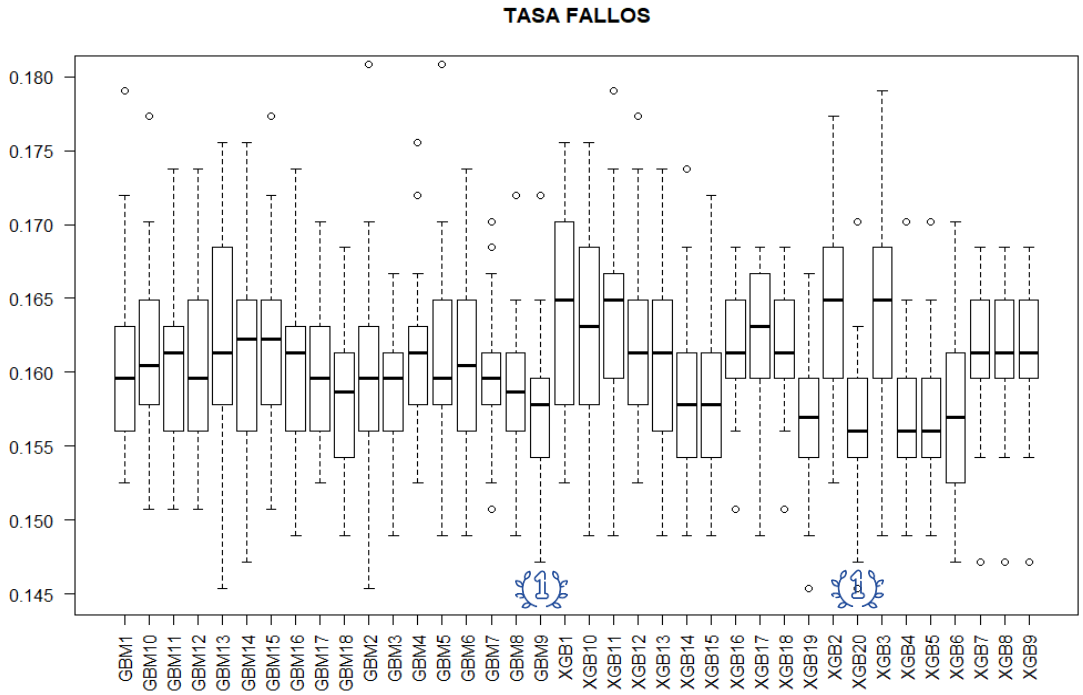
Cuando de la tasa de fallos se trata, los modelos ejecutados tanto de Gradient Boosting como de Xgboost parecen comportarse ligeramente mejor que en el resto de las técnicas anteriormente empleadas, especialmente este último, encontrándose varias de sus combinaciones por debajo del umbral del 16% de la tasa de casos mal clasificados.

Así, en la Figura 18 se puede comprobar el rendimiento de cada combinación de la pareja de algoritmos. En Gradient Boosting normal, el mejor modelo, aquel que presenta menor sesgo (que no varianza), se corresponde con “GBM9”.

En el caso de Xgboost, el modelo que aparenta ser más competitivo es “XGB20”, dado que su iteración con menor sesgo es el menor de todos, además de tener también la

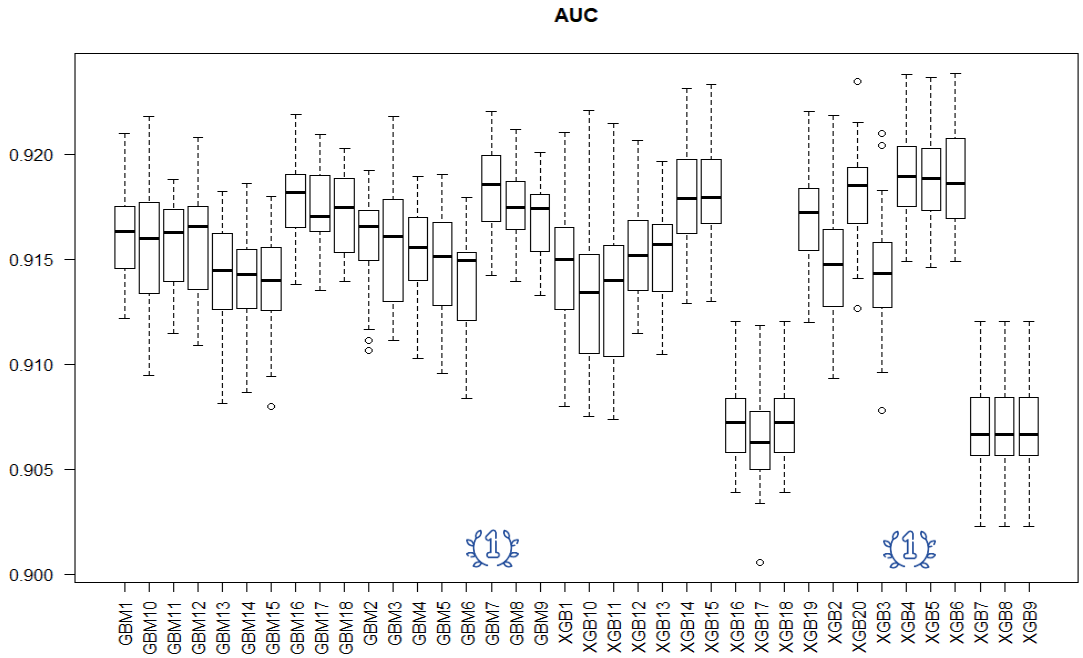
menor mediana en sesgo compartida con otros modelos, añadido a una de las menores varianzas. No obstante, la diferencia con otros modelos es tan sutil y reducida que no se podría afirmar que estos son necesariamente peores que el propuesto.

Figura 18. Tasa de fallos. Modelos de Gradient Boosting y Xgboost



Tomando ahora en consideración el área bajo la curva ROC, de los resultados presentes en la Figura 20, se aprecia que en, Gradient Boosting, la mayor relación sensibilidad/especificidad la presenta el modelo “GBM7”, mientras que en Xgboost, el mejor con una ligerísima ventaja es “XGB4”.

Figura 19. Área bajo la curva ROC. Modelos de Gradient Boosting y Xgboost



Como en ambos casos, para cada una de las métricas se consideran modelos ganadores cuatro de ellos diferentes entre sí, se opta por elegir uno que gana a los demás en términos globales, y no es nada más ni nada menos que el modelo “XGB20”, pues el hecho de que sea el modelo ganador por la tasa de fallos y su varianza en área bajo la curva ROC sea menor que en “XGB4”, decanta la balanza por este modelo.

Tabla 18. Matriz y medidas de confusión. Mejor modelo Gradient Boosting y Xgboost (XGB20)

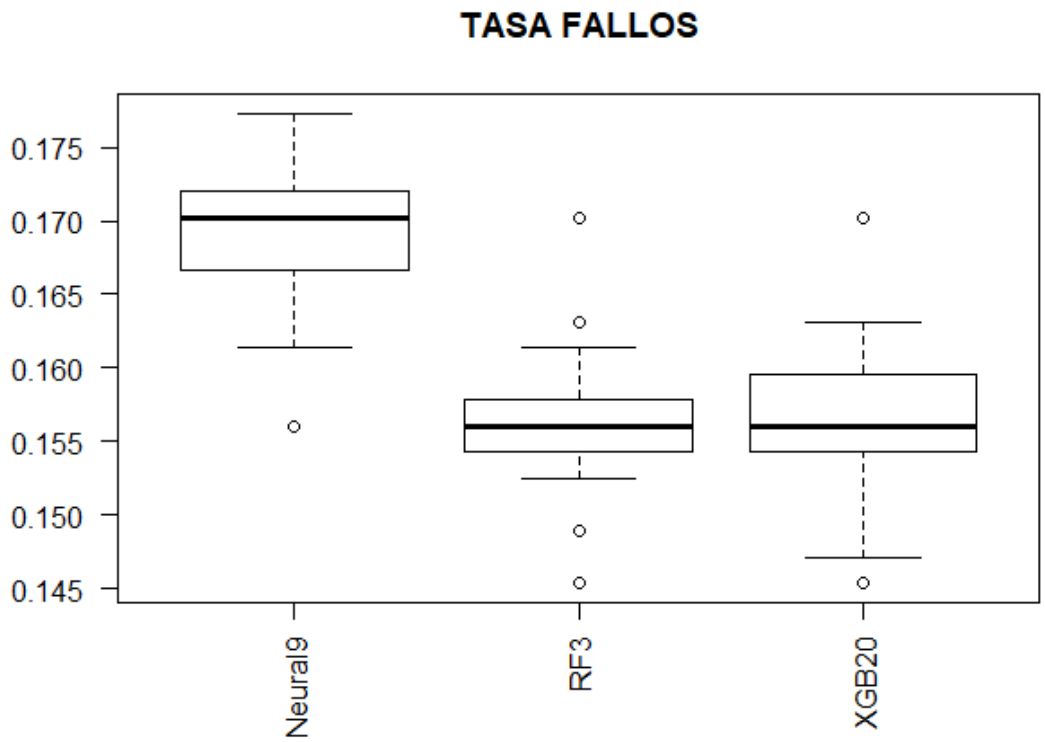
Matriz de confusión				
			Exactitud (accuracy)	0.8433
	Referencia		**Sensibilidad**	0.7732
Predicción	**No**	**Yes**	**Especificidad**	0.8877
No	9188	1490	**VPP**	0.8138
Yes	1162	5080	**VPN**	0.8605

Las medidas asociadas a la matriz de confusión revelan, una vez más, una relación de fallos y aciertos en varias combinaciones muy pareja con el resto de técnicas expuestas. Así, de los casos predichos que causarán abandono después del primer año, con este modelo, el abandono se produciría un 77.32% de las ocasiones, mientras que de los que se predigan como no abandono, se acertará en un 88.77% de los casos, con una *accuracy* global del modelo de 0.8433.

8. DISCUSIÓN DE RESULTADOS

Una vez ejecutados todos los algoritmos y modelos propuestos a lo largo del apartado 7 para el conjunto de variables con mayor tamaño, es momento de poner en perspectiva los modelos ganadores de cada técnica con la finalidad de hallar el modelo con la mayor capacidad predictiva para anticipar el abandono tras el primer año en el Grado en Estadística Aplicada de la UCM, en base al rendimiento académico de ese primer año.

Figura 20. Tasa de fallos. Mejores modelos

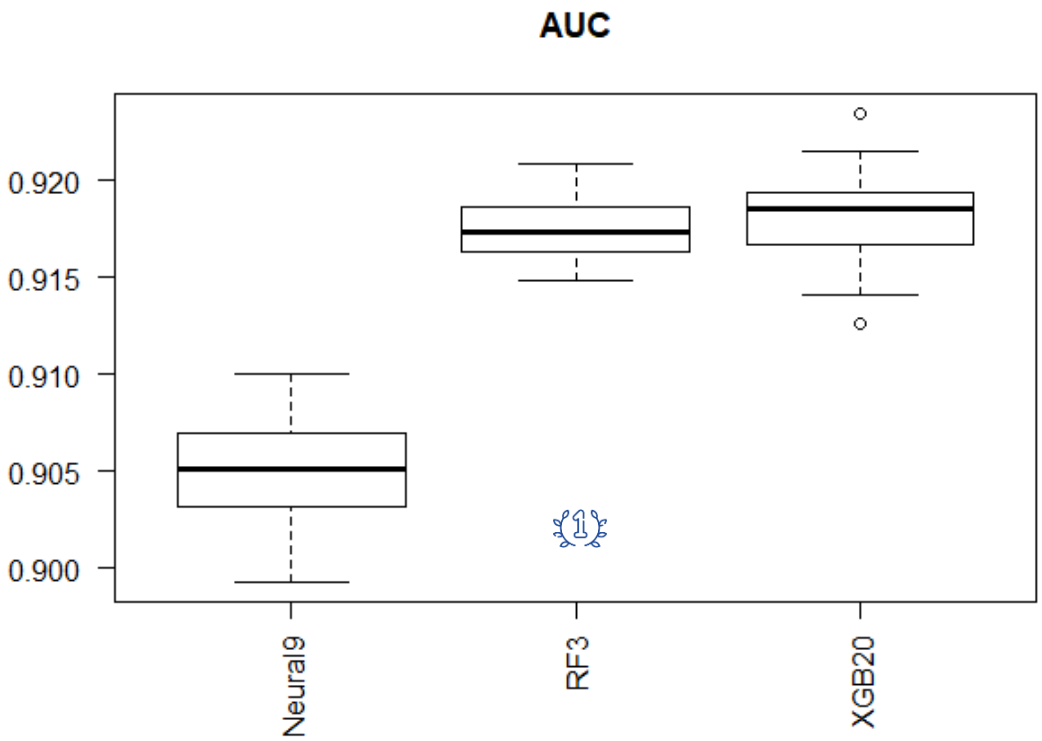


Puestos uno al lado del otro, de entre los mejores modelos de todas las técnicas contrastadas, presentes en la Figura 20, se encuentra que tanto el modelo por Random Forest como por Xgboost presentan una tasa de fallos mediana muy similar, pero la balanza podría decantarse por el ajuste “RF3”, dado que, en términos de varianza, este modelo es sensiblemente más compacto que “XGB20”.

De esta manera, tanto la Regresión Logística como la Red Neuronal se muestran claramente menos competitivas frente a Random Forest y Xgboost, comportamiento que también se repite en el área bajo la curva ROC.

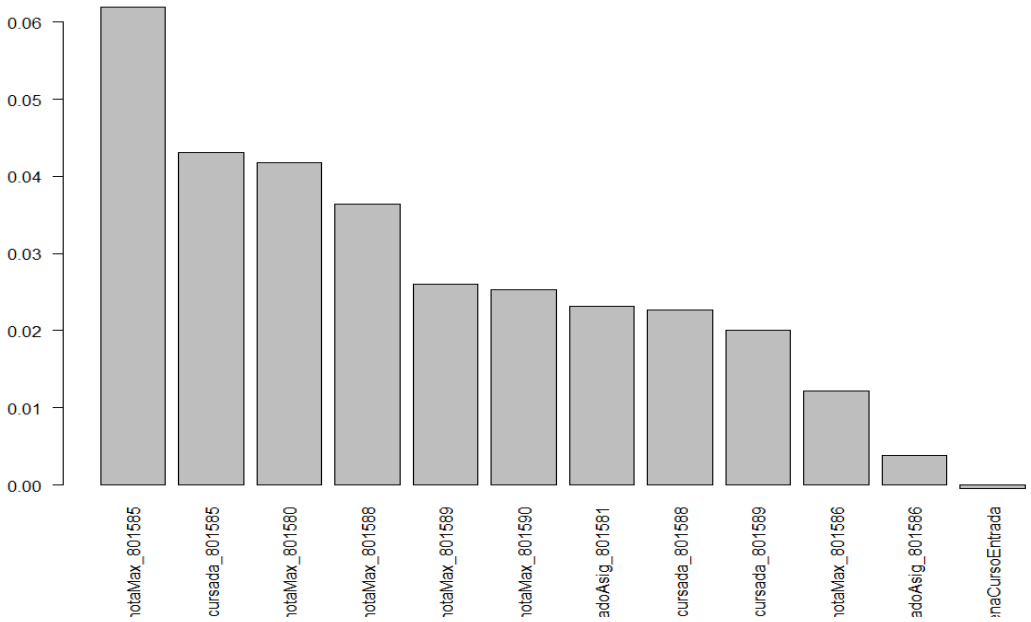
Para dirimir entonces cuál es el mejor modelo global, dada la igualdad aparentemente mostrada en la tasa de fallos, se acude nuevamente al área bajo la curva ROC, observable en la Figura 21. De nuevo, se percibe una relativa igualdad entre ambos modelos anteriormente mencionados. Sin embargo, aquí también la competición se resuelve finalmente resolver en favor de “RF3”, dada su menor varianza en relación con “XGB20”, por lo que se declara este modelo como el mejor de entre todos los probados.

Figura 21. Área bajo la curva ROC. Mejores modelos



Al tratarse el modelo ganador de un algoritmo basado en árboles de decisión, se puede obtener un gráfico de importancia de las variables, esto es, cuáles son aquellas variables que realizan un mayor aporte al modelo, de entre todas las posibles, para predecir la posibilidad de abandono, presente en la Figura 22.

Figura 22. Importancia variables del mejor modelo compuesto por 13 variables (RF3)

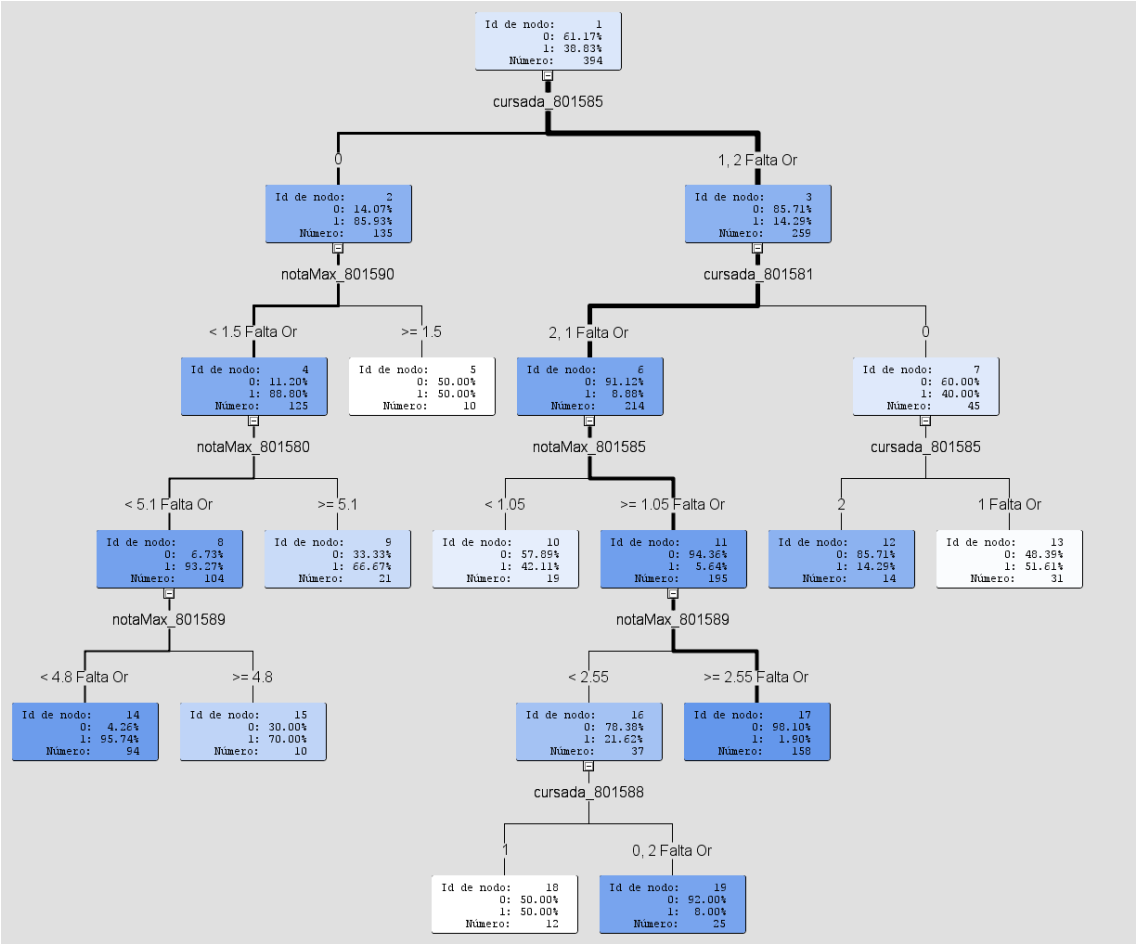


Del gráfico de barras anterior, se puede entender que, por orden descendente, las variables que más aportan al mejor modelo desarrollado (RF3) son:

1º - notaMax_801585: Nota máxima en todas las convocatorias de la asignatura "Estadística Económica".
2 º- cursada_801585: Cantidad de convocatorias cursadas de la asignatura "Estadística Económica".
3 º - notaMax_801580: Nota máxima de la asignatura "Descripción y Exploración de Datos".
4º - notaMax_801588: Nota máxima de la asignatura "Métodos Matemáticos para Estadística I".

Este ranking de variables más importantes proporciona una idea aproximada de qué tres asignaturas del primer curso resultan más cruciales para predecir el abandono. Sin embargo, resulta complicado inferir los niveles o valores de las anteriores variables a partir de los cuales una o un estudiante se orienta hacia el abandono o la permanencia. Para ello, se elabora un árbol de decisión que permite observar, a través de las reglas de división que plantea en cada nodo, el porcentaje de probabilidad de abandono del alumnado.

Figura 23. Árbol de decisión simple



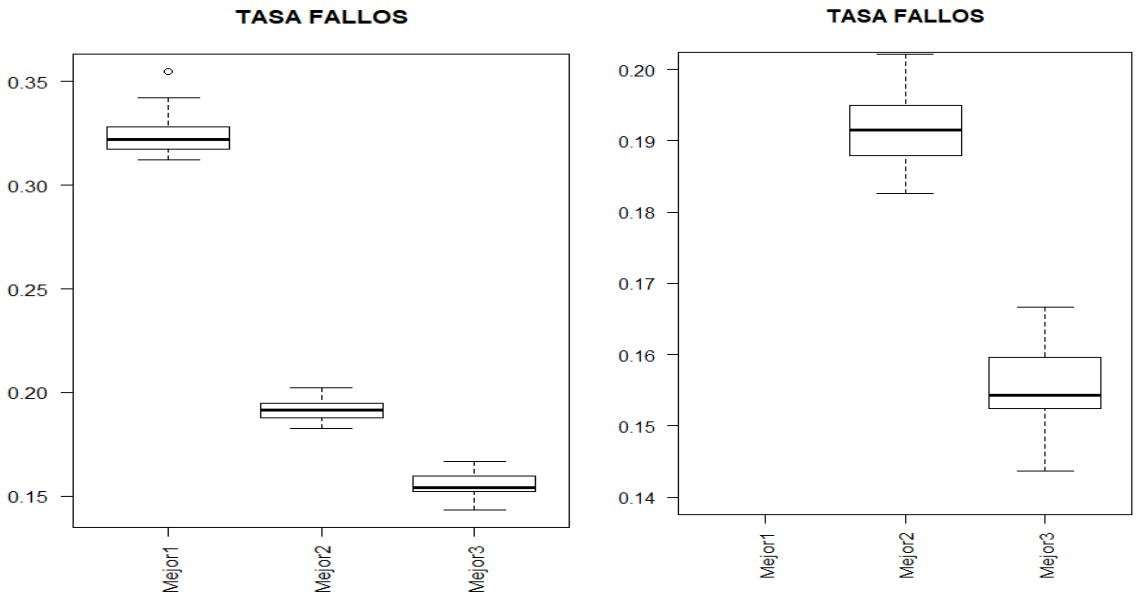
Así, en la Figura 24, se puede observar la influencia que tiene, una vez más, la variable cursada_801585, referente a la asignatura “Estadística Económica”. Aquellos alumnos que se presentan a alguna convocatoria iniciarán el camino hacia una permanencia casi segura, dependiente del curso de alguna otra asignatura como “Azar y Probabilidad”, además de su rendimiento en otras asignaturas.

Por último en este apartado, se ha decidido repetir el modelo ganador con un escalonamiento de variables en orden cronológico de existencia de las mismas: primero, solo un modelo con la nota de la PAU; el segundo modelo agrega todas las variables del primer semestre al anterior; finalmente, el tercer modelo agrega todas las variables relacionadas con el segundo semestre.

Esta secuencia de introducción de variables, que aparece representada en la Figura 24, permite vislumbrar algunas cuestiones:

- Es muy difícil ser preciso en la predicción del abandono tan solo con la nota de la prueba de acceso a la universidad.
- Con la información del primer semestre del curso, aunque no se obtiene el mejor modelo en términos de sesgo y varianza para el conjunto de datos test, se observar un acercamiento muy razonable a la tasa de fallos mínima posible.
- Con todas las variables posibles introducidas, conteniendo la información de ambos semestres, se obtiene un modelo muy similar, si no igual al ganador con la selección de las variables realizada, lo que indica que se ha hecho una aproximación más que correcta con el número preciso de variables.

Figura 24. Modelo ganador con introducción secuencial de variables en orden cronológico de aparición durante el primer año



8.1 MODELADO DEL SEGUNDO CONJUNTO CON MÁS VARIABLES

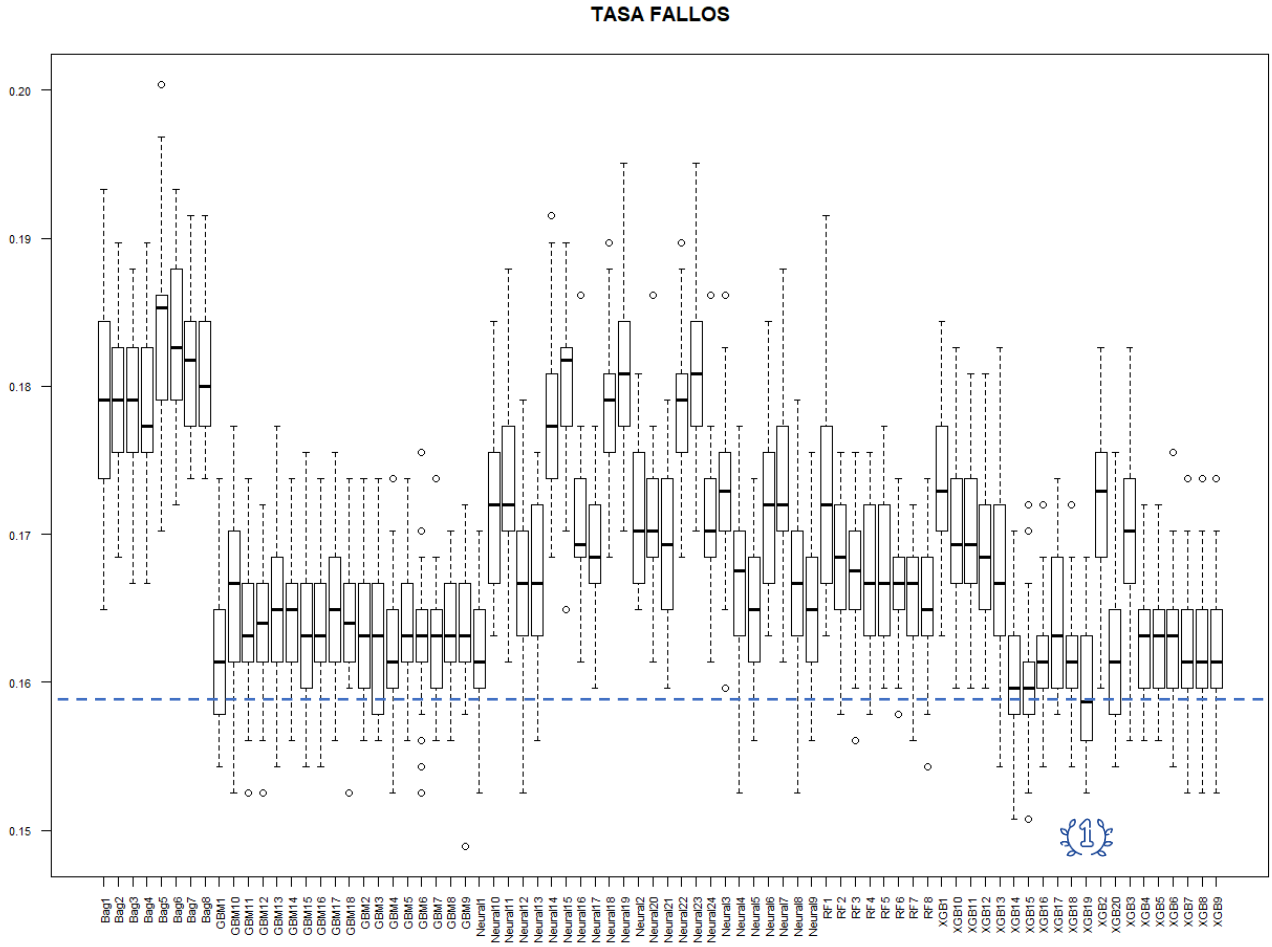
Como se había anunciado en el apartado 7.1, en primer lugar, se han ejecutado todos los modelos correspondientes al conjunto de variables menos restrictivo, aquel cuya propuesta incluía 13 variables.

Una vez hecho esto, se han comparado los mejores modelos y, del ganador de ellos, se han obtenido medidas como la importancia de sus variables para tratar de discernir cuáles son, en efecto, las asignaturas y sus variantes con mayor incidencia en el abandono.

El siguiente paso, por tanto, consiste en llevar a cabo el mismo proceso pero, en esta ocasión, con tan solo 7 variables, que constituye la siguiente combinación con más variables de las tablas 7 y 8. Esto se lleva a cabo con la finalidad de comprobar si, efectivamente, el modelo menos restrictivo genera sobreparametrización en los modelos ganadores debido al mayor número de variables que tomaba en cuenta.

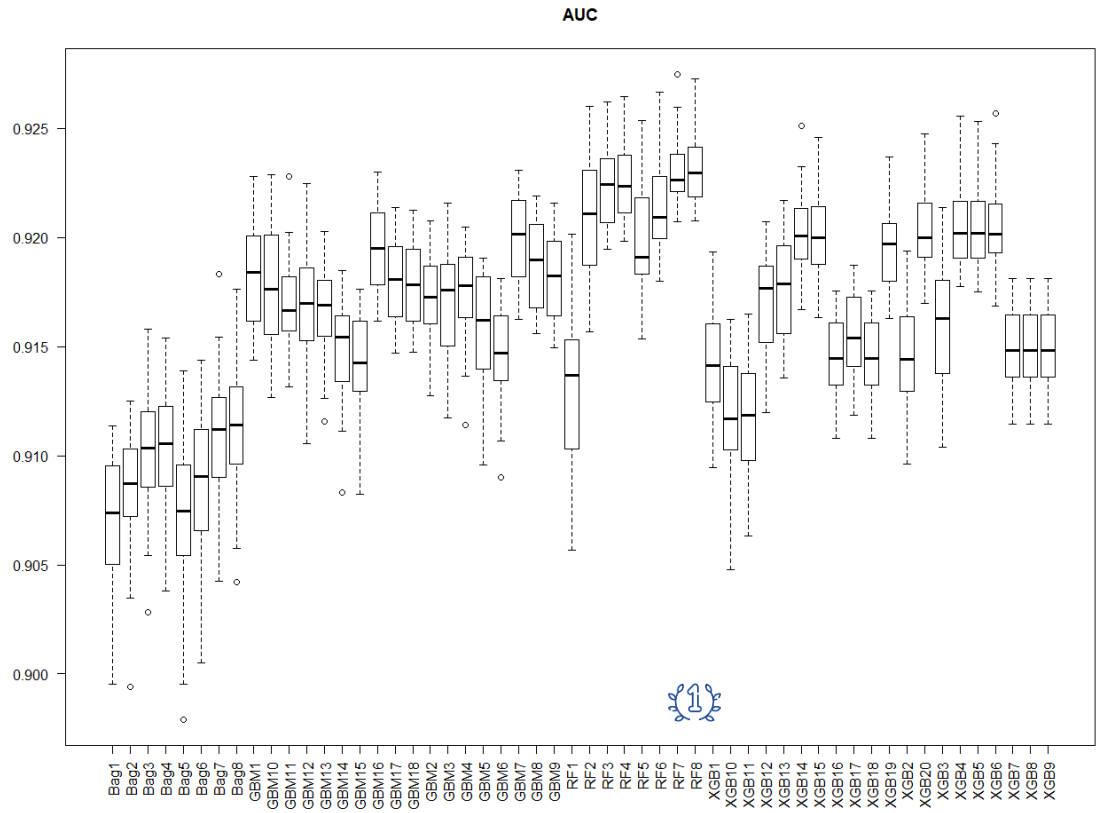
Así, la manera de proceder en este apartado consistirá en repetir el mismo proceso llevado a cabo en los anteriores, para elegir directamente un ganador global, del cual se obtendrán tanto la importancia de las variables que intervienen mayoritariamente en la predicción del abandono, así como su matriz de confusión.

Figura 25. Tasa de fallos. Todos los modelos para el conjunto de datos de 7 variables



Los primeros resultados se pueden observar en la Figura 25, donde se muestran, efectivamente, todos los modelos configurados en los puntos anteriores, pero con una combinación diferente de variables. En esta ocasión, por el lado de la tasa de fallos, se observa que destaca claramente el modelo “XGB19”, situándose por debajo de un 16% de tasa de fallos.

Figura 26. Área bajo la curva ROC. Todos los modelos para el conjunto de datos de 7 variables



En el área bajo la curva ROC, observable en la Figura 26, ya sin los modelos de Redes Neuronales, ya que presentan algunos de los peores rendimientos posibles, se encuentra que el modelo con mejor relación sensibilidad/especificidad es “RF8”, un modelo de Random Forest.

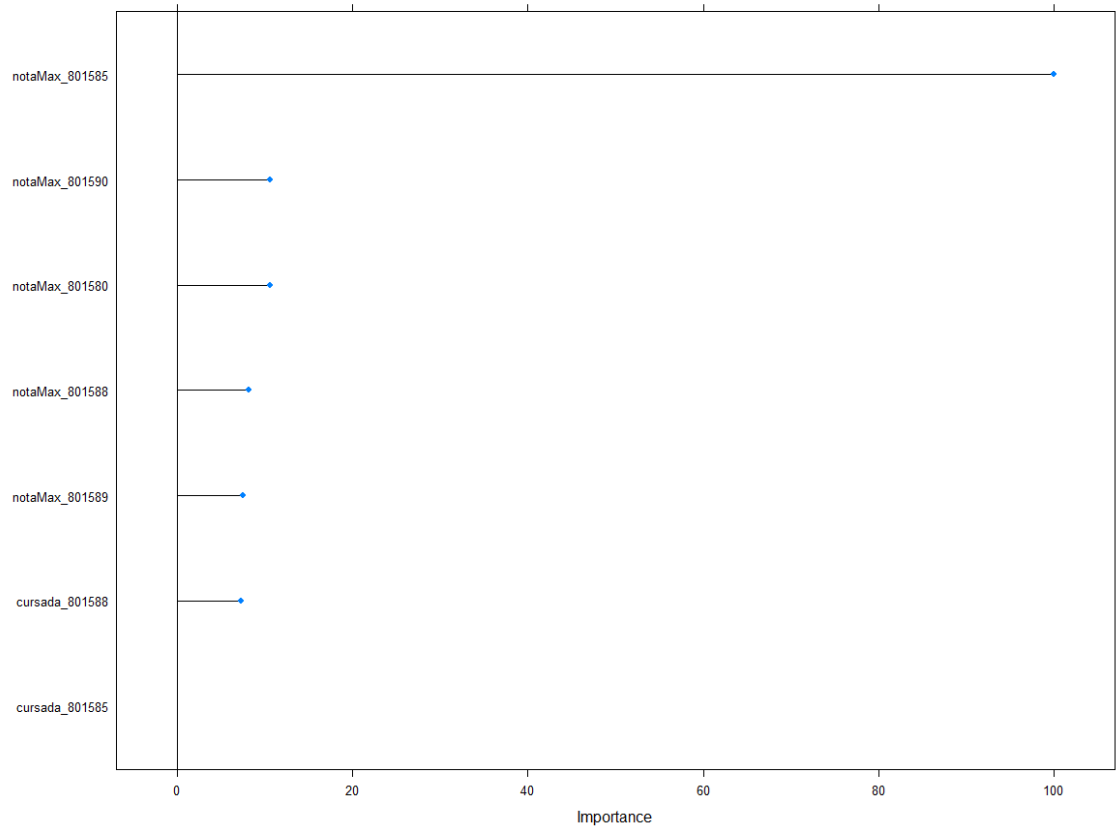
Sin embargo, una vez más, al encontrarse “XGB19” muy cerca en ROC y tener la menor tasa de fallos, se toma este modelo como el ganador de todos cuantos se han probado en esta segunda combinación de variables.

En la Figura 27 se puede observar cuáles son las variables que más han influido en la construcción del modelo, donde se repite un patrón parecido al que se podía encontrar en el conjunto de 13 variables:

1º - **notaMax_801585:** Nota máxima en todas las convocatorias de la asignatura “Estadística Económica”.
2º - notaMax_801580: Nota máxima de la asignatura “Descripción y Exploración de Datos”.

3º - notaMax_801590: Nota máxima de la asignatura “Métodos Matemáticos para Estadística III”.
4º - notaMax_801588: Nota máxima de la asignatura “Métodos Matemáticos para Estadística II”.

Figura 27. Importancia de las variables del mejor modelo del conjunto compuesto por 7 variables (XGB19)



De la importancia de las variables en los modelos de 13 y de 7 variables, se deduce que existe un consenso absoluto en lo determinante de la nota de la asignatura “Estadística Económica” para, a partir de ella, y en orden variable, gozar las demás de una importancia variable, pero siempre presentes. La única salvedad la constituye la cantidad de veces que se cursa “Estadística Económica” en el modelo de 7 variables, puesto que pasa a resultar absolutamente irrelevante bajo el modelo de Xgboost (XGB19).

En cualquier caso, se puede observar que, al menos por tasa de fallos, el mejor modelo de 7 variables no es capaz de superar al mejor de 13 variables, salvo lo ya visto en Regresión Logística, donde esto sí que ha sido así, aunque solo en una comparación directa a través de ese propio método.

8.2 ENSAMBLADO DE MODELOS

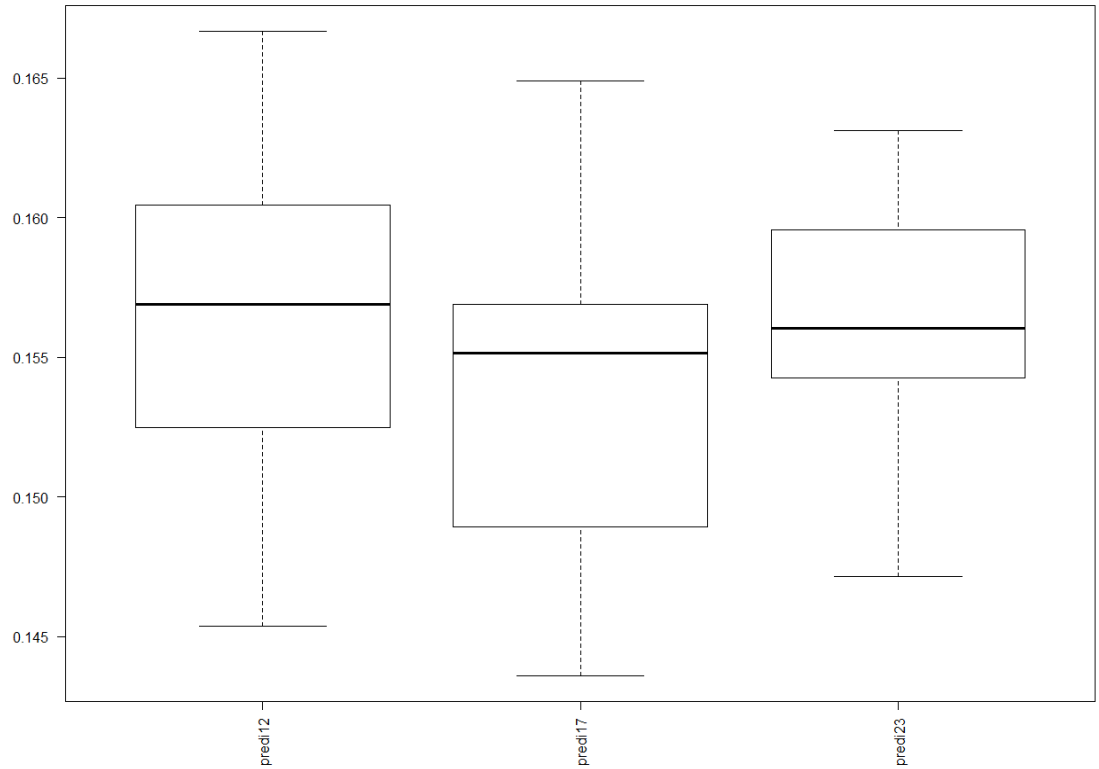
Una vez examinados los modelos más relevantes para los conjuntos de variables mayoritarios, sería conveniente realizar una última prueba con el conjunto que ha salido mejor parado (el de 13 variables) para examinar si sería posible reducir aún más el sesgo de los modelos obtenidos hasta ahora.

Para ello, se acude al ensamblado de modelos, los cuales deberían permitir reducir el sesgo un poco más. Así, se realizan pruebas con todas las combinaciones posibles de las técnicas empleadas hasta el momento, y se obtiene que los tres modelos más competitivos son los que se ilustran en la Tabla 19:

Tabla 19. Selección de mejores combinaciones por el método de Ensamblado

Modelo	Composición
Predi12	Logística + Xgboost
Predi17	Random Forest + Gradient Boosting
Predi23	Logística + Random Forest + Xgboost

Figura 28. Representación de los mejores modelos de ensamblado para el conjunto de 13 variables



El ensamblado de modelos muestra que la combinación más competitiva sitúa el modelo "predi17" como el mejor obtenido hasta ahora en el conjunto de las pruebas globales efectuadas con el conjunto de 13 variables, con tan solo un 15.5% de casos mal clasificados expresados a través de la tasa de fallos.

9. CONCLUSIONES Y RECOMENDACIONES FINALES

A lo largo de este trabajo, que constituye la prueba final para la consecución del Máster de Minería de Datos e Inteligencia de Negocio de la Universidad Complutense de Madrid, en primer lugar, se ha realizado un recorrido sobre la situación del abandono universitario en global para las universidades públicas de la Comunidad de Madrid, para posteriormente centrar el foco en la UCM y posteriormente desvelar los altos niveles de abandono que se producen en el Grado en Estadística Aplicada de la propia universidad.

Se ha establecido la necesidad de abordar este problema no solo exclusivamente desde las perspectivas descriptivas tradicionales, sino desde la elaboración de predicciones en base al rendimiento académico que traten de preverlo en vez de conocer exclusivamente las razones subjetivas que han llevado al alumnado a abandonar el programa de grado.

De esta manera, se ha llevado a cabo un completo estudio que, a través de la ejecución de las principales técnicas de Machine Learning cursadas durante el máster, y gracias a la base de datos del rendimiento académico de los y las estudiantes entre los años 2009 y 2018, proporcionada por la universidad a través del Observatorio del Estudiante, ha permitido obtener resultados relevantes en cuanto a la predicción del abandono tras el primer año cursado con una tasa de error reducida (menos del 16% de casos mal clasificados).

Gracias a ello, tanto a través de la regresión logística como de los mejores modelos obtenidos con 7 y 13 variables, producto de una selección exhaustiva previa de variables, se obtiene que las asignaturas que más influyen, en orden descendente, de cara a un potencial abandono de un estudiante tras el primer año son:

Tabla 20. Asignaturas más influyentes en el abandono del Grado en Estadística Aplicada

Asignatura	Período
Estadística Económica	2º Semestre
Descripción y Exploración de Datos	1er Semestre
Métodos Matemáticos para Estadística I	1er Semestre
Métodos Matemáticos para Estadística III	2º Semestre
Métodos Matemáticos para Estadística II	1er Semestre

Con todo ello, y una vez localizadas las asignaturas con mayor capacidad predictiva, sabedores además de que tan solo con la información del primer semestre del alumno, mostrada en la Figura 24, ya se podría obtener con un reducido nivel de error si el alumno o alumna se encuentra en riesgo de abandono se considera que habría que llevar a cabo las siguientes recomendaciones en el seno de la facultad en caso de ponerse en práctica un sistema de alerta temprana de abandono:

- Establecer un sistema de banderas en las bases de datos institucionales o en el Campus Virtual, que beban directamente de los resultados de las notas que obtienen los y las estudiantes en GEA, inmediatamente después del primer semestre del curso, de tal manera que se lance una alerta al coordinador/a del

grado si ha existido un suspenso en alguna de las asignaturas que figuran en la Tabla 20.

- Compartir con el mentor/a del alumno de nuevo ingreso dicha información, si este se encuentra en dicho programa, y realizar encuentros de seguimiento personal por parte del coordinador/a y mentor/a con el alumno en riesgo.
- Oferta de experiencias de refuerzo similares al “Taller Matemático” que ya se viene realizando, en grupos reducidos de 3 o 4 personas como máximo, para individualizar y realizar el mayor seguimiento posible.

Aunque si bien es cierto que el abandono en esta facultad no depende exclusivamente del factor del rendimiento académico, como se demuestra en el estudio por encuesta que se realizó en el año 2017 a alumnos que habían producido abandono temprano en la facultad (Espínola, 2017), sí que es cierto que todas estas medidas podrían servir de coadyuvante para que aquella parte del estudiantado que sí que se sienta con la intención de abandonar por sus resultados académicos pueda tener una segunda oportunidad y continuar en el programa, lo que significaría la justificación de la inversión realizada ya que no habría una pérdida de recursos y esfuerzos empleados por ambas partes, institución y estudiante.

No obstante, algunas de las razones reflejadas en dicho estudio, tales como el haber llegado de rebote o el desinterés en la materia quizá constituyan una de las principales limitaciones del trabajo, pues con los datos disponibles resulta imposible realizar aproximaciones basadas en aspectos subjetivos de la persona. Sin embargo, los resultados basados en las predicciones realizadas pueden constituir un indicador de tendencia sobre lo atractivo del programa para estudiantes cuya vocación no se encuentra alineada con el mismo.

Por tanto, se considera que se han cumplido los objetivos del trabajo, puesto que se han encontrado tipologías de alumnos por su rendimiento académico, se ha visto que la nota de PAU por sí sola no aporta la información necesaria para predecir el abandono, se han determinado las asignaturas más influyentes para poner en preaviso sobre un posible riesgo de abandono y, por último, se han elaborado algunas recomendaciones sobre cómo poner en práctica el modelo desarrollado.

Finalmente, como trabajo futuro, no cabe duda que la continuación de la recogida de datos es un elemento intrínseco y necesario para afinar las predicciones y cuantificar de manera más certera las probabilidades de abandono que tengan los y las matriculadas en el Grado, además de la posibilidad de sistematizar y extender el estudio realizado al conjunto de las carreras de la Universidad, pues las estructuras de datos a las que se pueden acceder son las mismas dado que se gestionan de manera centralizada.

10. BIBLIOGRAFÍA

Benítez, J. T. B., Cabrera Pérez, L., Hernández Cabrera, J. A., Álvarez Pérez, P., y González Afonso, M. (2017). Variables psicológicas y educativas en el abandono universitario. *Electronic Journal of Research in Education Psychology*, *6*(16).

Caicedo Bravo, E. F., y López Sotelo, J. A. (2009). *Una aproximación práctica a las redes neuronales artificiales*. Programa Editorial Universidad del Valle.

Calvo, D. (2017). Definición de red neuronal artificial. Acceso: 8 septiembre de 2019, desde http://www.diegocalvo.es/definicion-de-red-neuronal/

Castillo-Rojas, W., Medina-Quispe, F., & Meneses-Villegas, C. (2014). Modelo aumentado de árbol de decisión utilizando mapas autoorganizados. *Ingeniare. Revista Chilena de Ingeniería*, *22*(3), 351–362.

Cea d'Ancona, M. A. (2002). *Análisis multivariable. Teoría y práctica en la investigación social*. Síntesis.

Elias Andreu, M., & Daza Pérez, L. (2014). Sistema de Becas y Equidad Participativa en la Universidad. *Revista de La Asociación de Sociología de La Educación*, *7*(1), 233–251.

Espínola, M. R. (2017). *Análisis de la tasa de abandono de primer año del Grado en Estadística Aplicada. Evolución y proyección*.

Friedman, J. H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine*. *Source: The Annals of Statistics* (Vol. 29).

Gabadinho, A., Ritschard, G., Müller, N. S., & Studer, M. (2011). Analyzing and Visualizing State Sequences in R with TraMineR. *Journal of Statistical Software*, *40*(4), 1–37.

Medina Merino, R. F., & Ñique Chacón, C. I. (2017). Bosques Aleatorios como Extensión de los Árboles de Clasificación con los Programas R y Python. *Interfases*, (10), 165–189.

Observatorio del Estudiante - Universidad Complutense de Madrid. (n.d.). Acceso 6 septiembre de 2019, desde https://www.ucm.es/observatorio-del-estudiante

Ocaris Pérez Ramírez, F., & Fernández Castaño, H. (2007). Las Redes Neuronales y la Evaluación del Riesgo de Crédito. *Revista Ingenierías - Universidad de Medellín*, *6*(10), 77–91.

Ortiz, J. M., Rua, A., & Bilbao-Calabuig, P. (2017). APLICACIÓN DE ÁRBOLES DE CLASIFICACIÓN A LA DETECCIÓN PRECOZ DE ABANDONO EN LOS ESTUDIOS UNIVERSITARIOS DE ADMINISTRACIÓN Y DIRECCIÓN DE EMPRESAS. *Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA*, *18*, 177–201.

Portela, J. (2019). Materiales Asignatura "Técnicas de Machine Learning" - Máster Minería de Datos e Inteligencia Negocio (UCM).

Pozuelo Campillo, J., Martínez Vargas, J., & Carmona Ibáñez, P. (2018). Analysis of the algorithm Gradient Boosting Machine (GBM) in business failure prediction. *Revista Espanola de Financiacion y Contabilidad*, *47*(4), 507–532.

Rovira, S., Puertas, E., & Igual, L. (2017). Data-driven system to predict academic grades and dropout. *PLOS ONE*, *12*(2), e0171207.

Ruè, J. (2014). El abandono universitario: variables, marcos de referencia y políticas de calidad. *REDU. Revista de Docencia Universitaria*, *12*(2), 281.

SAS. (n.d.). Data Mining and SEMMA - Data Mining Using SAS(R) Enterprise Miner(TM): A Case Study Approach, Third Edition. Acceso: 12 enero de 2019, desde http://support.sas.com/documentation/cdl/en/emcs/66392/HTML/default/viewer.htm#n0pejm83csbja4n1xueveo2uoujy.htm

SIIU. (n.d.). Sistema Integrado de Información Universitaria (SIIU) - Ministerio de Educación, Cultura y Deporte. Acceso: 11 enero 2019, desde http://www.educacionyfp.gob.es/servicios-al-ciudadano-mecd/estadisticas/educacion/universitaria/siiu.html

Tuero Herrero, E., Cervero, A., Esteban, M., & Bernardo, A. (2018). ¿POR QUÉ

ABANDONAN LOS ALUMNOS UNIVERSITARIOS? VARIABLES DE INFLUENCIA EN EL PLANTEAMIENTO Y CONSOLIDACIÓN DEL ABANDONO. *Educación XX1*, *21*(2).

ANEXO I – Relación de asignaturas del Grado en Estadística Aplicada (UCM)

1º Cuatrimestre	2º Cuatrimestre
Primer Curso	
801580 - Descripción y Exploración de Datos	801581 - Azar y Probabilidad
801584 - Fuentes y Técnicas de Recogida de Información en Investigación Social y de Mercados	801583 - Software Estadístico I
801586 - Programación I	801585 - Estadística Económica
801588 - Métodos Matemáticos para Estadística I	801587 - Programación II
801589 - Métodos Matemáticos para Estadística II	801590 - Métodos Matemáticos para Estadística III
Segundo Curso	
801582 - Estimación I	801593 - Estimación II
801591 - Matemáticas con Ordenador	801598 - Estudio y Depuración de Datos
801592 - Probabilidad y Procesos Dinámicos	801607 - Técnicas de Optimización
801597 - Bases de Datos: Diseño y Gestión	801613 - Sistema Estadístico e Indicadores Económicos
801599 - Software Estadístico II	
Tercer Curso	
801594 - Diseños Muestrales	801595 - Diseños Muestrales Avanzados y Estadísticas Oficiales
801596 - Diseño de Experimentos	801605 - Técnicas Estadísticas Multidimensionales II
801604 - Técnicas Estadísticas Multidimensionales I	801611 - Aplicaciones Estadísticas a la Industria
801610 - Simulación y Líneas de Espera	801612 - Métodos Avanzados de Diseño de Experimentos
801614 - Investigación Comercial y Análisis de Mercados: Procedimientos y Aplicaciones	801601 - Métodos de Predicción Lineal
	Optativa I
Cuarto Curso	
801603 - Series Temporales	801602 Técnicas Avanzadas de Predicción
801606 - Técnicas de Segmentación y Tratamiento de Encuestas	801615 - Métodos Econométricos en Economía y Finanzas
801608 - Metodología 6 para la Mejora de la Calidad	Optativa III
801619 - Aplicaciones Estadísticas en Ciencias de la Salud	801621 - Trabajo Fin de Grado (TFG)
Optativas / Optional Courses	
801600 - Inglés para Fines Específicos	
801609 - Taller de Algoritmos	
801616 - Introducción a la Economía Aplicada	
801617 - Fundamentos de Empresa y Marketing	
801618 - Entorno Económico en la Empresa	
801620 - Demografía	

ANEXO II – CÓDIGO SAS BASE EMPLEADO

1. MACROS

```
/* VALIDACIÓN CRUZADA LOGÍSTICA PARA VARIABLES DEPENDIENTES BINARIAS

*****************************************************************
**********
                                                    PARÁMETROS
*****************************************************************
**********

BÁSICOS

archivo=          archivo de datos
vardepen=         variable dependiente binaria
categor=          lista de variables independientes categóricas
conti=                  lista de variables independientes continuas Y
TODAS LAS INTERACCIONES
ngrupos=          número de grupos validación cruzada
sinicio=          semilla inicial para repetición
sfinal=                 semilla final para repetición
objetivo=
      tasafallos,sensi,especif,porcenVN,porcenFN,porcenVP,porcenFP,pre
cision,tasaciertos


El archivo final se llama final. La variable media es la media del
obejtivo en todas las pruebas de validación cruzada
(habitualmente tasa de fallos).

*/

%macro
cruzadalogistica(archivo=,vardepen=,conti=,categor=,ngrupos=,sinicio=,
sfinal=,objetivo=tasafallos);
title ' ';
data final;run;
/* Bucle semillas */
%do semilla=&sinicio %to &sfinal;
      data dos;set &archivo;u=ranuni(&semilla);
      proc sort data=dos;by u;run;
      data dos (drop=nume);
      retain grupo 1;
      set dos nobs=nume;
      if _n_>grupo*nume/&ngrupos then grupo=grupo+1;
      run;
      data fantasma;run;
      %do exclu=1 %to &ngrupos;
            data tres;set dos;if grupo ne &exclu then vardep=&vardepen;
            proc logistic data=tres noprint;/*<<<<<******SE PUEDE
QUITAR EL NOPRINT */
            %if (&categor ne) %then %do;class &categor;model
vardep=&conti &categor ;%end;
            %else %do;model vardep=&conti;%end;
            output out=sal p=predi;run;
            data sal2;set sal;pro=1-predi;if pro>0.5 then pre11=1; else
pre11=0;
            if grupo=&exclu then output;run;
            proc freq data=sal2;tables pre11*&vardepen/out=sal3;run;
```

```
            data estadisticos (drop=count percent pre11 &vardepen);
            retain vp vn fp fn suma 0;
            set sal3 nobs=nume;
            suma=suma+count;
            if pre11=0 and &vardepen=0 then vn=count;
            if pre11=0 and &vardepen=1 then fn=count;
            if pre11=1 and &vardepen=0 then fp=count;
            if pre11=1 and &vardepen=1 then vp=count;
            if _n_=nume then do;
            porcenVN=vn/suma;
            porcenFN=FN/suma;
            porcenVP=VP/suma;
            porcenFP=FP/suma;
            sensi=vp/(vp+fn);
            especif=vn/(vn+fp);
            tasafallos=1-(vp+vn)/suma;
            tasaciertos=1-tasafallos;
            precision=vp/(vp+fp);
            F_M=2*Sensi*Precision/(Sensi+Precision);
            output;
            end;
            run;

            data fantasma;set fantasma estadisticos;run;
      %end;
      proc means data=fantasma sum noprint;var &objetivo;
      output out=sumaresi sum=suma mean=media;
      run;
      data sumaresi;set sumaresi;semilla=&semilla;
      data final (keep=suma media semilla);set final sumaresi;if
suma=. then delete;run;
%end;
proc print data=final;run;
%mend;

/* LA MACRO RANDOMSELECTlog REALIZA UN MÉTODO STEPWISE
REPETIDAS VECES CON DIFERENTES ARCHIVOS TRAIN.

LA SALIDA INCLUYE UNA TABLA DE FRECUENCIAS
DE LOS MODELOS QUE APARECEN SELECCIONADOS EN LOS DIFERENTES
ARCHIVOS TRAIN

LOS MODELOS QUE SALEN MÁS VECES SON POSIBLES CANDIDATOS A PROBAR
CON VALIDACIÓN CRUZADA

listclass=lista de variables de clase ATENCIÓN, EN ESTA LISTA SOLO
PONER VARIABLES
                  QUE SE VAYAN A USAR (BIEN COMO EFECTOS PRINCIPALES O
INTERACCIONES)
vardepen=variable dependiente
modelo=modelo
sinicio=semilla inicio
sfinal=semilla final
fracciontrain=fracción de datos train
directorio=directorio para archivos basura

EL ARCHIVO QUE CONTIENE LOS EFECTOS SE LLAMA SALEFEC.
SE INCLUYE EN EL LOG EL LISTADO PARA PODER COPIAR Y PEGAR
(A VECES LA VENTANA OUTPUT ESTÁ LIMITADA)

*/
```

```
%macro
randomselectlog(data=,listclass=,vardepen=,modelo=,sinicio=,sfinal=,fr
acciontrain=,directorio=);
options nocenter linesize=256;
proc printto print="&directorio\kk.txt";run;
data;file "&directorio\cosa2.txt" ;run;
%do semilla=&sinicio %to &sfinal;
proc surveyselect data=&data rate=&fracciontrain out=sal1234
seed=&semilla;run;

%if &listclass ne %then %do;
ods output type3=parametros;
proc logistic data=sal1234;
      class &listclass;
    model &vardepen= &modelo/ selection=stepwise;
run;
data parametros;length effect $20. modelo $ 20000;retain modelo "
";set parametros end=fin;effect=cat(' ',effect);
if _n_ ne 1 then modelo=catt(modelo,' ',effect);if fin then
do;variable=modelo;output;end;
run;
%end;
%else %do;
ods output  Logistic.ParameterEstimates=parametros;
proc logistic data=sal1234;
        model &vardepen= &modelo/ selection=stepwise;
run;
%end;
ods graphics off;
ods html close;
data;file "&directorio\cosa2.txt" mod;set parametros;
%if &listclass ne %then %do; put variable @@;%end;
%else %do; if _n_ ne 1 then put variable @@;%end;
run;
%end;
proc printto ;run;
data todos;
infile "&directorio\cosa2.txt";
length efecto $ 400;
input efecto @@;
if efecto ne 'Intercept' then output;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;

data todos;
infile "&directorio\cosa2.txt";
length efecto $ 200;
input efecto $ &&;
run;
proc freq data=todos;tables efecto /out=sal;run;
proc sort data=sal;by descending count;
proc print data=sal;run;
data sal;set sal;put efecto;run;
%mend;
```

2.DESARROLLO

```
proc freq data=tfm.Primeranodep;run;
proc contents data=tfm.Primeranodep out=tfm.sal;run;quit;
data;set tfm.sal;put name @@;run;
options mprint=0;

/* MACRO RANDOMSELECTLOG */

options nonotes;
%randomselectlog(data=tfm.Primeranodep,
listclass=almacenaCursoEntrada almacenaSexo cursada_801580
cursada_801581 cursada_801583 cursada_801584 cursada_801585
cursada_801586 cursada_801587 cursada_801588 cursada_801589
cursada_801590 estadoAsig_801580 estadoAsig_801581 estadoAsig_801583
estadoAsig_801584 estadoAsig_801585 estadoAsig_801586
estadoAsig_801587 estadoAsig_801588 estadoAsig_801589
estadoAsig_801590,
vardepen=Abandono,
modelo=almacenaCursoEntrada almacenaSexo cursada_801580 cursada_801581
cursada_801583 cursada_801584 cursada_801585 cursada_801586
cursada_801587 cursada_801588 cursada_801589 cursada_801590
estadoAsig_801580 estadoAsig_801581 estadoAsig_801583
estadoAsig_801584 estadoAsig_801585 estadoAsig_801586
estadoAsig_801587 estadoAsig_801588 estadoAsig_801589
estadoAsig_801590 IMP_almacenaPAU notaMax_801580 notaMax_801581
notaMax_801583 notaMax_801584 notaMax_801585 notaMax_801586
notaMax_801587 notaMax_801588 notaMax_801589 notaMax_801590,
sinicio=12345,sfinal=12580,fracciontrain=0.8);

options notes;
/* PROBAMOS LOS mejores MODELOS CON LOGÍSTICA */

%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti= notaMax_801580 notaMax_801585 notaMax_801589 notaMax_801590,
categor=cursada_801588 ,
ngrupos=10,sinicio=12345,sfinal=12365);
data final1;set final;modelo=1;
%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti=notaMax_801580 notaMax_801589 notaMax_801590,
categor=estadoAsig_801585 cursada_801588,
ngrupos=10,sinicio=12345,sfinal=12365);
data final2;set final;modelo=2;
%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti=notaMax_801580 notaMax_801588,
categor=cursada_801588 estadoAsig_801583,
ngrupos=10,sinicio=12345,sfinal=12365);
data final3;set final;modelo=3;
%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti= notaMax_801580 notaMax_801588 notaMax_801589,
categor=cursada_801588 estadoAsig_801583,
ngrupos=10,sinicio=12345,sfinal=12365);
data final4;set final;modelo=4;
%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti= notaMax_801580 notaMax_801583 notaMax_801588 notaMax_801589,
categor=cursada_801588,
```

```
ngrupos=10,sinicio=12345,sfinal=12365);
data final5;set final;modelo=5;
%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti= notaMax_801580 notaMax_801583 notaMax_801589 notaMax_801590,
categor=cursada_801588,
ngrupos=10,sinicio=12345,sfinal=12365);
data final6;set final;modelo=6;

data union;set final1 final2 final3 final4 final5 final6;
proc boxplot data=union;plot media*modelo;run;

/*PRUEBA DE LOS CRUCES DE VARIABLES CON LOGÍSTICA*/
options nonotes;

%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti= notaMax_801580 notaMax_801585,
categor=,
ngrupos=10,sinicio=12345,sfinal=12385);
data final7;set final;modelo=Set1;

%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti= notaMax_801580 notaMax_801585 notaMax_801588 notaMax_801590
notaMax_801589,
categor=cursada_801585 cursada_801588,
ngrupos=10,sinicio=12345,sfinal=12385);
data final8;set final;modelo=Set2;

%cruzadalogistica
(archivo=tfm.Primeranodep,vardepen=Abandono,
conti= notaMax_801580 notaMax_801585 notaMax_801588 notaMax_801590
notaMax_801589 notaMax_801586,
categor=almacenaCursoEntrada cursada_801589 estadoAsig_801581
estadoAsig_801586 cursada_801585 cursada_801588,
ngrupos=10,sinicio=12345,sfinal=12385);
data final9;set final;modelo=Set3;

data union;set final7 final8 final9;
proc boxplot data=union;plot media*modelo;run;

options notes;
/****REGRESIÓN LOGÍSTICA SIMPLE****/

ods output SelectedEffects=efectos;run;
proc logistic data=tfm.Primeranodep descending;
class cursada_801585 cursada_801588;
    model Abandono= cursada_801585 cursada_801588 notaMax_801580
notaMax_801585 notaMax_801588 notaMax_801590 notaMax_801589
/ selection=stepwise
;
proc print data=efectos;run;
data;set efectos;put effects ;run;

proc means data = final8;
var media;
run;
```

ANEXO III – CÓDIGO R EMPLEADO

1. TRANSFORMACIÓN DE LA BASE DE DATOS Y TRAMINER

```
library(psych)
library(foreign)
library(TraMineR)
library(cluster)

primerAño <- matriz_estadistica[, -c(12,13)]

primerAño <- rbind(subset(primerAño, Asignatura == 801580), subset(primerAño,
Asignatura == 801581), subset(primerAño, Asignatura == 801583)
                , subset(primerAño, Asignatura == 801584), subset(primerAño, Asignatura
== 801585), subset(primerAño, Asignatura == 801586)
                , subset(primerAño, Asignatura == 801587), subset(primerAño, Asignatura
== 801588), subset(primerAño, Asignatura == 801589)
                , subset(primerAño, Asignatura == 801590))

primerAño <- dfOrder(primerAño, 1)

#Nota máxima asignatura del año en curso

contador <- 1

for (i in primerAño$Identificador){

  primerAño$notaMax[contador] = max(primerAño$Nota_feb[contador],
primerAño$Nota_jun[contador], primerAño$Nota_sept[contador])

  contador = contador + 1

}

#Estado Asignatura

primerAño$estadoAsig <- primerAño$Sexo

contador <- 1

for (k in primerAño$estadoAsig){


  if(primerAño$notaMax[contador] >= 5){

    primerAño$estadoAsig[contador] = 1

  }

  if(primerAño$notaMax[contador] > 0 & primerAño$notaMax[contador] < 5){
```

```
    primerAño$estadoAsig[contador] = 2
  }
  if(primerAño$notaMax[contador] == 0){
    primerAño$estadoAsig[contador] = 3
    #print(k)
  }
  contador = contador + 1
}
#Número convocatorias
contador<-1
contador_aux <- 0
for (i in primerAño$Asignatura){
  contador_aux = 0
  if(primerAño$estadoAsig[contador] == 1 || primerAño$estadoAsig[contador] == 2){
    if(primerAño$Nota_feb[contador] != 0){
      contador_aux = contador_aux + 1
    }
    if(primerAño$Nota_jun[contador] != 0){
      contador_aux = contador_aux + 1
    }
    if(primerAño$Nota_sept[contador] != 0){
      contador_aux = contador_aux + 1
    }
    primerAño$cursada[contador] = contador_aux
  }
  if(primerAño$estadoAsig[contador] == 3){
```

```
    primerAño$cursada[contador] = 0

  }

  contador = contador + 1

}

#Almacenamiento de variables para su posterior incorporación en la matriz en formato
"wide"

contador <- 2
id <- 1
almacenaCursoEntrada = c()

for (i in primerAño$Identificador){

  if (primerAño$Identificador[contador] != primerAño$Identificador[contador-1]){

    id = id + 1
    almacenaCursoEntrada[id] = primerAño$Curso_acces[contador]

  }

  contador = contador + 1

}

almacenaCursoEntrada1 <- data.frame(almacenaCursoEntrada)


contador <- 2
id <- 1
almacenaSexo = c()

for (i in primerAño$Identificador){

  if (primerAño$Identificador[contador] != primerAño$Identificador[contador-1]){

    id = id + 1
    almacenaSexo[id] = primerAño$Sexo[contador]

  }

  contador = contador + 1

}

almacenaSexo1 <- data.frame(almacenaSexo)

contador <- 2
```

```
id <- 1
almacenaPAU = c()

for (i in primerAño$Identificador){

  if (primerAño$Identificador[contador] != primerAño$Identificador[contador-1]){

    id = id + 1
    almacenaPAU[id] = primerAño$Nota_PAU[contador]

  }

  contador = contador + 1

}

almacenaPAU1 <- data.frame(almacenaPAU)

#Limpieza de la matriz para dejar las variables que se quieren transformar a formato
"wide"

primerAño <- primerAño[, -c(3, 5, 6, 7, 8, 9)]

#Aplicación de la función reshape() y unión de las variables almacenadas

primerAno1 <- reshape(primerAño, idvar = "Identificador", timevar = "Asignatura",
direction = "wide")

primerAno1 <- cbind(primerAno1, almacenaCursoEntrada, almacenaSexo,
almacenaPAU)


#####Asignatura801580#####

contador <- 1

for (k in primerAno1$estadoAsig.801580){

  if(is.na(primerAno1$notaMax.801580[contador])){

    primerAno1$estadoAsig.801580[contador] = 4

  }

  contador = contador + 1

}

#####Asignatura801581#####

contador <- 1
```

```
for (k in primerAno1$estadoAsig.801581){
  if(is.na(primerAno1$notaMax.801581[contador])){
    primerAno1$estadoAsig.801581[contador] = 4
  }
  contador = contador + 1
}
#####Asignatura801583#####
contador <- 1
for (k in primerAno1$estadoAsig.801583){
  if(is.na(primerAno1$notaMax.801583[contador])){
    primerAno1$estadoAsig.801583[contador] = 4
  }
  contador = contador + 1
}
#####Asignatura801584#####
contador <- 1
for (k in primerAno1$estadoAsig.801584){
  if(is.na(primerAno1$notaMax.801584[contador])){
    primerAno1$estadoAsig.801584[contador] = 4
  }
  contador = contador + 1
}
#####Asignatura801585#####
contador <- 1
for (k in primerAno1$estadoAsig.801585){
```

```
  if(is.na(primerAno1$notaMax.801585[contador])){
    primerAno1$estadoAsig.801585[contador] = 4
  }
  contador = contador + 1
}
#####Asignatura801586#####
contador <- 1
for (k in primerAno1$estadoAsig.801586){
  if(is.na(primerAno1$notaMax.801586[contador])){
    primerAno1$estadoAsig.801586[contador] = 4
  }
  contador = contador + 1
}

#####Asignatura801587#####
contador <- 1
for (k in primerAno1$estadoAsig.801587){
  if(is.na(primerAno1$notaMax.801587[contador])){
    primerAno1$estadoAsig.801587[contador] = 4
  }
  contador = contador + 1
}
#####Asignatura801588#####
contador <- 1
for (k in primerAno1$estadoAsig.801588){
  if(is.na(primerAno1$notaMax.801588[contador])){
```

```
    primerAno1$estadoAsig.801588[contador] = 4
  }
  contador = contador + 1
}
#####Asignatura801589#####
contador <- 1
for (k in primerAno1$estadoAsig.801589){
  if(is.na(primerAno1$notaMax.801589[contador])){
    primerAno1$estadoAsig.801589[contador] = 4
  }
  contador = contador + 1
}
#####Asignatura801590#####
contador <- 1
for (k in primerAno1$estadoAsig.801590){
  if(is.na(primerAno1$notaMax.801590[contador])){
    primerAno1$estadoAsig.801590[contador] = 4
  }
  contador = contador + 1
}
#En primer lugar, se rellenan todos los vacíos del curso de entrada con el curso de la
asignatura 801584
contador <- 1
for (i in primerAno1$almacenaCursoEntrada){
  if (is.na(i)){
    primerAno1$almacenaCursoEntrada[contador] =
primerAno1$Curso_asig.801584[contador]
```

```
  }

  contador = contador + 1

}

##A continuación, se eliminan todos aquellos casos de los cuales no tenemos
información sobre su año de entrada, ya que no se podrá calcular su abandono

primerAno2 <- primerAno1
primerAno2 <- subset(primerAno2, !is.na(Curso_asig.801584))

#Ahora, se eliminan los casos que no sirven para medir el abandono, que son aquellos
que accedieron en el curso 1617 y 1718

primerAno2 <- subset(primerAno2, almacenaCursoEntrada != "201617" &
almacenaCursoEntrada != "201718")

#Finalmente, se calcula aquellos alumnos que causaron abandono en este período.
Para ello, se toma el curso de entrada
#y se observa si existe matriculación en alguna asignatura dos años después
(básicamente, se observa si ha existido
#matriculación en alguna asignatura de segundo o tercero)

primerAno2$Abandono <- primerAno2$Identificador
matrizsinNA <- matriz_final2

write.csv(matrizsinNA, "C:/Users/NitroPC/Documents/Máster Minería de
Datos/TFM/TFM Project Agosto1/matrizsinNA.csv")

matrizsinNA <- csv.get("C:/Users/NitroPC/Documents/Máster Minería de
Datos/TFM/TFM Project Agosto1/matrizsinNA.csv", sep = ";", labels = 0)

contador <- 1

for (i in primerAno2$Identificador){

  if (0 == matrizsinNA$Curso.asig.801582[contador] & 0 ==
matrizsinNA$Curso.asig.801591[contador] & 0 ==
matrizsinNA$Curso.asig.801592[contador] & 0 ==
matrizsinNA$Curso.asig.801597[contador] & 0 ==
matrizsinNA$Curso.asig.801599[contador] & 0 ==
matrizsinNA$Curso.asig.801593[contador] & 0 ==
matrizsinNA$Curso.asig.801598[contador] & 0 ==
matrizsinNA$Curso.asig.801607[contador] & 0 ==
matrizsinNA$Curso.asig.801613[contador] & 0 ==
matrizsinNA$Curso.asig.801594[contador] & 0 ==
matrizsinNA$Curso.asig.801596[contador] & 0 ==
matrizsinNA$Curso.asig.801604[contador] & 0 ==
matrizsinNA$Curso.asig.801610[contador] & 0 ==
```

```
matrizsinNA$Curso.asig.801614[contador] & 0 ==
matrizsinNA$Curso.asig.801595[contador] & 0 ==
matrizsinNA$Curso.asig.801605[contador] & 0 ==
matrizsinNA$Curso.asig.801611[contador] & 0 ==
matrizsinNA$Curso.asig.801612[contador] & 0 ==
matrizsinNA$Curso.asig.801601[contador]){

   primerAno2$Abandono[contador] = 1

 }

 else{

   primerAno2$Abandono[contador] = 0

 }

 # if(matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801582[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801591[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801592[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801597[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801599[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801593[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801598[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801607[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801613[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801594[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801596[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801604[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801610[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801614[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801595[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801605[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801611[contador] &
```

```
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801612[contador] &
matriz1AsinNA$almacenaCursoEntrada[contador] !=
matriz1AsinNA$Curso_asig.801601[contador]){
  #
  #   primerAno2$Abandono[contador] = 1
  #
  # }

  contador = contador + 1

}

contador <- 1

for (i in primerAno2$Identificador){

  if(primerAno2$Abandono[contador] == 0 &
(matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801582[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801591[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801592[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801597[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801599[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801593[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801598[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801607[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801613[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801594[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801596[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801604[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801610[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801614[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801595[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801605[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
```

```
matrizsinNA$Curso.asig.801611[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801612[contador] |
matrizsinNA$almace0CursoEntrada[contador] ==
matrizsinNA$Curso.asig.801601[contador])){

    primerAno2$Abandono[contador] = 1

  }

  contador = contador + 1

}

#En primerAno2 - Notas 1Q y 2Q

notaMedia1Q <- 0
contador <- 1

for (i in primerAno2$Identificador){

  if(is.na(primerAno2$notaMax.801580[contador])){

    primerAno2$notaMax.801580[contador] = 0

  }

  if(is.na(primerAno2$notaMax.801584[contador])){

    primerAno2$notaMax.801584[contador] = 0

  }

  if(is.na(primerAno2$notaMax.801586[contador])){

    primerAno2$notaMax.801586[contador] = 0

  }

  if(is.na(primerAno2$notaMax.801588[contador])){

    primerAno2$notaMax.801588[contador] = 0

  }


  if(is.na(primerAno2$notaMax.801589[contador])){

    primerAno2$notaMax.801589[contador] = 0

  }
```

```
  notaMedia1Q = sum(primerAno2$notaMax.801580[contador],
primerAno2$notaMax.801584[contador], primerAno2$notaMax.801586[contador],
primerAno2$notaMax.801588[contador], primerAno2$notaMax.801589[contador])/5
  primerAno2$notaMedia1Q[contador] <- notaMedia1Q

  contador = contador + 1

}

notaMedia2Q <- 0
contador <- 1

for (i in primerAno2$Identificador){

  if(is.na(primerAno2$notaMax.801581[contador])){

    primerAno2$notaMax.801581[contador] = 0

  }

  if(is.na(primerAno2$notaMax.801583[contador])){

    primerAno2$notaMax.801583[contador] = 0

  }

  if(is.na(primerAno2$notaMax.801585[contador])){

    primerAno2$notaMax.801585[contador] = 0

  }

  if(is.na(primerAno2$notaMax.801587[contador])){

    primerAno2$notaMax.801587[contador] = 0

  }


  if(is.na(primerAno2$notaMax.801590[contador])){

    primerAno2$notaMax.801590[contador] = 0

  }

  notaMedia2Q = sum(primerAno2$notaMax.801581[contador],
primerAno2$notaMax.801583[contador], primerAno2$notaMax.801585[contador],
primerAno2$notaMax.801587[contador], primerAno2$notaMax.801590[contador])/5
  primerAno2$notaMedia2Q[contador] <- notaMedia2Q
```

```
  contador = contador + 1

}

#Nota Media primer curso

contador <- 1

for (i in primerAno2$Identificador){

  primerAno2$NotaMedia1A[contador] = (primerAno2$notaMedia1Q[contador] +
primerAno2$notaMedia2Q[contador])/2

  contador = contador + 1

}

#Eliminación curso asignaturas de "primerAno2"

primerAno2 <- primerAno2[, -c(2,6,10,14,18,22,26,30,34,38)]
table(primerAno2$Abandono)

# Output table
# 0   1
# 345 219

#Exportación del archivo en .csv

write.csv(primerAno2, "C:/Users/NitroPC/Documents/Máster Minería de
Datos/TFM/TFM Project Agosto1/primerAno2.csv")

###COMIENZA TRAMINER###

TRAMobject <- seqdef(matriz_final2, var = c("estadoAsig.801580",
"estadoAsig.801584", "estadoAsig.801586", "estadoAsig.801588",
                                    "estadoAsig.801589", "estadoAsig.801581",
"estadoAsig.801583", "estadoAsig.801585",
                                    "estadoAsig.801587", "estadoAsig.801590",
"estadoAsig.801582","estadoAsig.801593", "estadoAsig.801591",
                                    "estadoAsig.801598", "estadoAsig.801592",
"estadoAsig.801607", "estadoAsig.801597", "estadoAsig.801613",
"estadoAsig.801599",
                                    "estadoAsig.801594", "estadoAsig.801595",
"estadoAsig.801596", "estadoAsig.801605", "estadoAsig.801604",
                                    "estadoAsig.801611", "estadoAsig.801610",
"estadoAsig.801612", "estadoAsig.801614", "estadoAsig.801601",
                                    "estadoAsig.801600", "estadoAsig.801609",
"estadoAsig.801616", "estadoAsig.801617", "estadoAsig.801618",
                                    "estadoAsig.801620", "estadoAsig.801603",
"estadoAsig.801602", "estadoAsig.801606", "estadoAsig.801615",
```

```
                                "estadoAsig.801608", "estadoAsig.801619",
"estadoAsig.801621"))
names(matriz_finalTRAM)

seqiplot(TRAMobject, idxs = 1:564, border = NA, sortv = "from.start", with.legend =
"right", cex.legend = c(0.5))
seqiplot(TRAMobject, idxs = 1:564, border = NA, sortv = "from.start")

#Clusters

ccost <- seqsubm(TRAMobject, method = "CONSTANT", cval = 2)
cluster.OM <- seqdist(TRAMobject, method = "OM", sm = ccost)

clusterTRAM <- agnes(cluster.OM, diss = TRUE, method = "ward")
cluster4 <- cutree(clusterTRAM, k = 4)
cluster4 <- factor(cluster4, labels = c("Type 1", "Type 2", "Type 3", "Type 4"))
table(cluster4)

seqiplot(TRAMobject, group = cluster4, pbarw = T, border = NA, idxs = 1:564, sortv =
"from.end")

cluster3 <- cutree(clusterTRAM, k = 3)
cluster3 <- factor(cluster3, labels = c("Type 1", "Type 2", "Type 3"))
table(cluster3)

seqiplot(TRAMobject, group = cluster3, pbarw = T, border = NA, idxs = 1:564, sortv =
"from.end", with.legend = FALSE, par(las=2))

##Solo asignaturas primer año

TRAMobject1A <- seqdef(primerAno2, var = c("estadoAsig.801580",
"estadoAsig.801584", "estadoAsig.801586", "estadoAsig.801588",
                                "estadoAsig.801589", "estadoAsig.801581",
"estadoAsig.801583", "estadoAsig.801585",
                                "estadoAsig.801587", "estadoAsig.801590"))

seqiplot(TRAMobject1A, idxs = 1:564, border = NA, sortv = "from.start")

ccost1A <- seqsubm(TRAMobject1A, method = "CONSTANT", cval = 2)
cluster.OM1A <- seqdist(TRAMobject1A, method = "OM", sm = ccost1A)

clusterTRAM1A <- agnes(cluster.OM, diss = TRUE, method = "ward")

cluster4_1A <- cutree(clusterTRAM1A, k = 4)
cluster4_1A <- factor(cluster4_1A, labels = c("Type 1", "Type 2", "Type 3", "Type 4"))

seqiplot(TRAMobject1A, group = cluster4_1A, pbarw = T, border = NA, idxs = 1:564,
sortv = "from.end")

cluster3_1A <- cutree(clusterTRAM1A, k = 3)
cluster3_1A <- factor(cluster3_1A, labels = c("Type 1", "Type 2", "Type 3"))
```

```
seqiplot(TRAMobject1A, group = cluster3_1A, pbarw = T, border = NA, idxs = 1:564,
sortv = "from.end")

cluster5_1A <- cutree(clusterTRAM1A, k = 5)
cluster5_1A <- factor(cluster5_1A, labels = c("Type 1", "Type 2", "Type 3", "Type 4",
"Type 5"))

seqiplot(TRAMobject1A, group = cluster5_1A, pbarw = T, border = NA, idxs = 1:564,
sortv = "from.end")

cluster2_1A <- cutree(clusterTRAM1A, k = 2)
cluster2_1A <- factor(cluster2_1A, labels = c("Type 1", "Type 2"))

seqiplot(TRAMobject1A, group = cluster2_1A, pbarw = T, border = NA, idxs = 1:564,
sortv = "from.end")
```

2. CREACIÓN DE LA VARIABLE “ABANDONO”

```
#En primer lugar, se rellenan todos los vacíos del curso de entrada con el curso de la
asignatura 801584

contador <- 1

for (i in matriz_final1$almacenaCursoEntrada){

  if (is.na(i)){

    matriz_final1$almacenaCursoEntrada[contador] =
matriz_final1$Curso_asig.801584[contador]

  }

  contador = contador + 1

}

##A continuación, se eliminan todos aquellos casos de los cuales no tenemos
información sobre su año de entrada, ya que no se podrá calcular su abandono
##Para ello, se genera una matriz que contiene los casos de los casos faltantes, para
posteriormente restarlos en una matriz copiada de "matriz_final1" ("matriz_final2") [[No
era necesario]]
##Finalmente, tan solo era necesario aplicar la función "subset" y quedarse con todos
aquellos casos cuya matriculación sí apareciese en la asignatura 801484

matriz_final2 <- matriz_final1

matriz_final2 <- subset(matriz_final2, !is.na(Curso_asig.801584))

#Ahora, se eliminan los casos que no sirven para medir el abandono, que son aquellos
que accedieron en el curso 1617 y 1718
```

```
matriz_final2 <- subset(matriz_final2, almacenaCursoEntrada != "201617" &
almacenaCursoEntrada != "201718")

#Finalmente, se calcula aquellos alumnos que causaron abandono en este período.
Para ello, se toma el curso de entrada
#y se observa si existe matriculación en alguna asignatura dos años después
(básicamente, se observa si ha existido
#matriculación en alguna asignatura de segundo o tercero)

write.csv(matriz_final2, '/Users/jorgeblancoiglesias/Documents/Máster en Minería de
Datos/TFM/matriz_final2.csv')

matriz_final2$Abandono <- matriz_final2$Identificador


#Asignaturas de Tercero
contador <- 1

for (i in matriz_final2$Identificador){

  if (is.na(matriz_final2$CursoAsig.801594[contador]) &
is.na(matriz_final2$CursoAsig.801596[contador]) &
is.na(matriz_final2$CursoAsig.801604[contador]) &
is.na(matriz_final2$CursoAsig.801610[contador]) &
is.na(matriz_final2$CursoAsig.801614[contador]) &
is.na(matriz_final2$CursoAsig.801595[contador]) &
is.na(matriz_final2$CursoAsig.801605[contador]) &
is.na(matriz_final2$CursoAsig.801611[contador]) &
is.na(matriz_final2$CursoAsig.801612[contador]) &
is.na(matriz_final2$CursoAsig.801601[contador])){

    matriz_final2$Abandono[contador] = 1

  }

  else{

    matriz_final2$Abandono[contador] = 0

  }

  contador = contador + 1

}

#Asignaturas de Segundo y Tercero
contador <- 1

for (i in matriz_final2$Identificador){
```

```
  if (is.na(matriz_final2$Curso_asig.801582[contador]) &
is.na(matriz_final2$Curso_asig.801591[contador]) &
is.na(matriz_final2$Curso_asig.801592[contador]) &
is.na(matriz_final2$Curso_asig.801597[contador]) &
is.na(matriz_final2$Curso_asig.801599[contador]) &
is.na(matriz_final2$Curso_asig.801593[contador]) &
is.na(matriz_final2$Curso_asig.801598[contador]) &
is.na(matriz_final2$Curso_asig.801607[contador]) &
is.na(matriz_final2$Curso_asig.801613[contador]) &
is.na(matriz_final2$Curso_asig.801594[contador]) &
is.na(matriz_final2$Curso_asig.801596[contador]) &
is.na(matriz_final2$Curso_asig.801604[contador]) &
is.na(matriz_final2$Curso_asig.801610[contador]) &
is.na(matriz_final2$Curso_asig.801614[contador]) &
is.na(matriz_final2$Curso_asig.801595[contador]) &
is.na(matriz_final2$Curso_asig.801605[contador]) &
is.na(matriz_final2$Curso_asig.801611[contador]) &
is.na(matriz_final2$Curso_asig.801612[contador]) &
is.na(matriz_final2$Curso_asig.801601[contador])){

    matriz_final2$Abandono[contador] = 1

  }

  else{

    matriz_final2$Abandono[contador] = 0

  }

  contador = contador + 1

}

#Prueba solo con asignaturas de segundo curso

contador <- 1

for (i in matriz_final2$Identificador){

  if (is.na(matriz_final2$Curso_asig.801582[contador]) &
is.na(matriz_final2$Curso_asig.801591[contador]) &
is.na(matriz_final2$Curso_asig.801592[contador]) &
is.na(matriz_final2$Curso_asig.801597[contador]) &
is.na(matriz_final2$Curso_asig.801599[contador]) &
is.na(matriz_final2$Curso_asig.801593[contador]) &
is.na(matriz_final2$Curso_asig.801598[contador]) &
is.na(matriz_final2$Curso_asig.801607[contador]) &
is.na(matriz_final2$Curso_asig.801613[contador])){

    matriz_final2$Abandono1[contador] = 1
```

```
  }

  else{

    matriz_final2$Abandono1[contador] = 0

  }

  contador = contador + 1

}

table(matriz_final2$Abandono1)



################################################################################
################################################
######################### PREPARACIÓN DE LA MATRIZ PARA ELABORAR LOS
MODELOS DE MACHINE LEARNING ####################
############################## SE ELIMINA TODA LA INFORMACIÓN QUE NO
SEA DEL PRIMER AÑO ################################
################################################################################
################################################

matriz_final2$test1 <- matriz_final2$Abandono1

#which(colnames(matriz_final2) == "Identificador") ---> Para identificar el número de
columna de una variable

matriz_final3 <- matriz_final2[,-(50:173)]
matriz_final3 <- matriz_final3[,-(38:41)]
matriz_final3 <- matriz_final3[,-(26:29)]
matriz1A <- matriz_final3

names(matriz1A)

matriz1Ab <- matriz1A[,-c(2, 6, 10, 14, 18, 22, 26, 30, 34, 38)] #La matriz 1Ab no
contiene la información de los cursos de las asignaturas
```

3. FUNCIONES Y ALGORITMOS POR VALIDACIÓN CRUZADA REPETIDA

```
library(sas7bdat)

# **********************************
# CRUZADAS PARA ENSAMBLADO DEPENDIENTE BINARIA
# **********************************
# VALIDACIÓN CRUZADA REPETIDA Y BOXPLOT para
#
```

```
# LOGISTICA
# AVNNET
# RF
# GBM
# XGBM
# SVM

# ES NECESARIO QUE LA VARIABLE DEPENDIENTE ESTÉ CATEGORIZADA
# COMO Yes,No

definitivo <- read.sas7bdat('C:/Users/NitroPC/Documents/Máster Minería de
Datos/TFM/TFM Project Agosto1/Librería SAS Base/primerAnoDep.sas7bdat')

contador <- 1
definitivo$AbandonoStr <- definitivo$Abandono

for (i in definitivo$Abandono){

  if (i == 0){

    definitivo$AbandonoStr[contador] = "No"

  }

  else{

    definitivo$AbandonoStr[contador] = "Yes"

  }

  contador = contador + 1

}

# *********************************

library(dummies)
library(MASS)
library(reshape)
library(caret)
library(dplyr)
library(pROC)

# *********************************
# CRUZADA LOGISTICA
# *********************************

cruzadalogistica <- function(data=data,vardep=NULL,
 listconti=NULL,listclass=NULL,grupos=10,sinicio=12345,repe=10)
{
  library(dummies)
```

```
library(MASS)
library(reshape)
library(caret)
library(dplyr)
library(pROC)

if  (listclass !=c(""))
{
 for (i in 1:dim(array(listclass))) {
  numindi<-which(names(data)==listclass[[i]])
  data[,numindi]<-as.character(data[,numindi])
  data[,numindi]<-as.factor(data[,numindi])
 }
}

data[,vardep]<-as.factor(data[,vardep])

# Creo la formula para la logistica

if  (listclass!=c(""))
{
 koko<-c(listconti,listclass)
}  else   {
 koko<-c(listconti)
}

modelo<-paste(koko,sep="",collapse="+")
formu<-formula(paste(vardep,"~",modelo,sep=""))

formu
# Preparo caret

set.seed(sinicio)
control<-trainControl(method = "repeatedcv",number=grupos,repeats=repe,
 savePredictions = "all",classProbs=TRUE)

# Aplico caret y construyo modelo

regresion <- train(formu,data=data,
 trControl=control,method="glm",family = binomial(link="logit"))
preditest<-regresion$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

tasafallos<-function(x,y) {
 confu<-confusionMatrix(x,y)
 tasa<-confu[[3]][1]
 return(tasa)
```

```
  }

  # Aplicamos función sobre cada Repetición

  medias<-preditest %>%
   group_by(Rep) %>%
   summarize(tasa=1-tasafallos(pred,obs))

  # CalculamoS AUC  por cada Repetición de cv
  # Definimnos función

  auc<-function(x,y) {
   curvaroc<-roc(response=x,predictor=y)
   auc<-curvaroc$auc
   return(auc)
  }

  # Aplicamos función sobre cada Repetición

  mediasbis<-preditest %>%
   group_by(Rep) %>%
   summarize(auc=auc(obs,Yes))

  # Unimos la info de auc y de tasafallos

  medias$auc<-mediasbis$auc

 return(list(medias,preditest))

}

modelo1 <- cruzadalogistica(definitivo, "AbandonoStr", c("notaMax_801580",
"notaMax_801585", "cursada_801585", "cursada_801588", "notaMax_801588",
"notaMax_801590", "notaMax_801589", "notaMax_801586"),
                listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                grupos=10,sinicio=12345,repe=30)

modelo2 <- cruzadalogistica(definitivo, "AbandonoStr", c("notaMax_801580",
"notaMax_801585", "cursada_801585", "cursada_801588", "notaMax_801588",
"notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=67895,repe=30)

modelo3 <- cruzadalogistica(definitivo, "AbandonoStr", c("notaMax_801580",
"notaMax_801585", "cursada_801585", "cursada_801588", "notaMax_801588",
"notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=20147,repe=30)
```

```
modelo4 <- cruzadalogistica(definitivo, "AbandonoStr", c("notaMax_801580",
"notaMax_801585", "cursada_801585", "cursada_801588", "notaMax_801588",
"notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=93248,repe=30)

modelo5 <- cruzadalogistica(definitivo, "AbandonoStr", c("notaMax_801580",
"notaMax_801585", "cursada_801585", "cursada_801588", "notaMax_801588",
"notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=13485,repe=30)

modelo1_1 <- modelo1[[1]]
modelo2_1 <- modelo2[[1]]
modelo3_1 <- modelo3[[1]]
modelo4_1 <- modelo4[[1]]
modelo5_1 <- modelo5[[1]]

modelo1_1$modelo ="Logística1"
modelo2_1$modelo ="Logística2"
modelo3_1$modelo ="Logística3"
modelo4_1$modelo ="Logística4"
modelo5_1$modelo ="Logística5"

union1 <- rbind(modelo1_1, modelo2_1, modelo3_1, modelo4_1, modelo5_1)

boxplot(data=union1,tasa~modelo,main="TASA FALLOS")
boxplot(data=union1,auc~modelo,main="AUC")

# **********************************
# CRUZADA avNNet
# **************

cruzadaavnnetbin<-
 function(data=data,vardep="vardep",
  listconti="listconti",listclass="listclass",grupos=4,sinicio=1234,repe=5,
  size=c(5),decay=c(0.01),repeticiones=5,itera=100,trace=FALSE)
 {

  # Preparación del archivo

  # b)pasar las categóricas a dummies

  if  (listclass!=c(""))
  {
   databis<-data[,c(vardep,listconti,listclass)]
   databis<- dummy.data.frame(databis, listclass, sep = ".")
  }  else   {
```

```
 databis<-data[,c(vardep,listconti)]
}

# c)estandarizar las variables continuas

# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

means <-apply(databis[,listconti],2,mean)
sds<-sapply(databis[,listconti],sd)

# Estandarizo solo las continuas y uno con las categoricas

datacon<-scale(databis[,listconti], center = means, scale = sds)
numerocont<-which(colnames(databis)%in%listconti)
databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

databis[,vardep]<-as.factor(databis[,vardep])

formu<-formula(paste(vardep,"~.",sep=""))

# Preparo caret

set.seed(sinicio)
control<-trainControl(method = "repeatedcv",number=grupos,repeats=repe,
 savePredictions = "all",classProbs=TRUE)

# Aplico caret y construyo modelo

avnnetgrid <-  expand.grid(size=size,decay=decay,bag=FALSE)

avnnet<- train(formu,data=databis,
 method="avNNet",linout = FALSE,maxit=itera,repeats=repeticiones,
 trControl=control,tuneGrid=avnnetgrid,trace=trace)

print(avnnet$results)

preditest<-avnnet$pred

preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

tasafallos<-function(x,y) {
 confu<-confusionMatrix(x,y)
 tasa<-confu[[3]][1]
 return(tasa)
}

# Aplicamos función sobre cada Repetición
```

```
  medias<-preditest %>%
   group_by(Rep) %>%
   summarize(tasa=1-tasafallos(pred,obs))

  # CalculamoS AUC  por cada Repetición de cv
  # Definimnos función

  auc<-function(x,y) {
   curvaroc<-roc(response=x,predictor=y)
   auc<-curvaroc$auc
   return(auc)
  }

  # Aplicamos función sobre cada Repetición

  mediasbis<-preditest %>%
   group_by(Rep) %>%
   summarize(auc=auc(obs,Yes))

  # Unimos la info de auc y de tasafallos

  medias$auc<-mediasbis$auc

   return(list(medias,preditest))

 }

modelo6 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.001),
repeticiones = 30, itera = 100)

modelo7 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.005),
repeticiones = 30, itera = 100)

modelo8 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.01),
repeticiones = 30, itera = 100)
```

```
modelo9 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.1),
repeticiones = 30, itera = 100)

modelo10 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.001),
repeticiones = 30, itera = 200)

modelo11 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.005),
repeticiones = 30, itera = 200)

modelo12 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.01),
repeticiones = 30, itera = 200)

modelo13 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.1),
repeticiones = 30, itera = 200)

modelo14 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.001),
repeticiones = 30, itera = 500)

modelo15 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
```

```
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.005),
repeticiones = 30, itera = 500)

modelo16 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.01),
repeticiones = 30, itera = 500)

modelo17 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(3), decay = c(0.1),
repeticiones = 30, itera = 500)

modelo6_1 <- modelo6[[1]]
modelo6_1$modelo ="Neural1"

modelo7_1 <- modelo7[[1]]
modelo7_1$modelo ="Neural2"

modelo8_1 <- modelo8[[1]]
modelo8_1$modelo ="Neural3"

modelo9_1 <- modelo9[[1]]
modelo9_1$modelo ="Neural4"

modelo10_1 <- modelo10[[1]]
modelo10_1$modelo ="Neural5"

modelo11_1 <- modelo11[[1]]
modelo11_1$modelo ="Neural6"

modelo12_1 <- modelo12[[1]]
modelo12_1$modelo ="Neural7"

modelo13_1 <- modelo13[[1]]
modelo13_1$modelo ="Neural8"

modelo14_1 <- modelo14[[1]]
modelo14_1$modelo ="Neural9"

modelo15_1 <- modelo15[[1]]
modelo15_1$modelo ="Neural10"
```

```
modelo16_1 <- modelo16[[1]]
modelo16_1$modelo ="Neural11"

modelo17_1 <- modelo17[[1]]
modelo17_1$modelo ="Neural12"

union2 <- rbind(modelo6_1, modelo7_1, modelo8_1, modelo9_1, modelo10_1,
              modelo11_1, modelo12_1, modelo13_1, modelo14_1, modelo15_1,
modelo16_1, modelo17_1)

boxplot(data=union2,tasa~modelo,main="TASA FALLOS")
boxplot(data=union2,auc~modelo,main="AUC")

modelo18 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.001),
repeticiones = 30, itera = 100)

modelo19 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.005),
repeticiones = 30, itera = 100)

modelo20 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.01),
repeticiones = 30, itera = 100)

modelo21 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.1),
repeticiones = 30, itera = 100)

modelo22 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                         listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
```

```
                    grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.001),
repeticiones = 30, itera = 200)

modelo23 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.005),
repeticiones = 30, itera = 200)

modelo24 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.01),
repeticiones = 30, itera = 200)

modelo25 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.1),
repeticiones = 30, itera = 200)

modelo26 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.001),
repeticiones = 30, itera = 500)

modelo27 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.005),
repeticiones = 30, itera = 500)

modelo28 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.01),
repeticiones = 30, itera = 500)
```

```
modelo29 <- cruzadaavnnetbin(definitivo, "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                        listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                        grupos=10,sinicio=12345,repe=30, size = c(4), decay = c(0.1),
repeticiones = 30, itera = 500)

modelo18_1 <- modelo18[[1]]
modelo18_1$modelo ="Neural13"

modelo19_1 <- modelo19[[1]]
modelo19_1$modelo ="Neural14"

modelo20_1 <- modelo20[[1]]
modelo20_1$modelo ="Neural15"

modelo21_1 <- modelo21[[1]]
modelo21_1$modelo ="Neural16"

modelo22_1 <- modelo22[[1]]
modelo22_1$modelo ="Neural17"

modelo23_1 <- modelo23[[1]]
modelo23_1$modelo ="Neural18"

modelo24_1 <- modelo24[[1]]
modelo24_1$modelo ="Neural19"

modelo25_1 <- modelo25[[1]]
modelo25_1$modelo ="Neural20"

modelo26_1 <- modelo26[[1]]
modelo26_1$modelo ="Neural21"

modelo27_1 <- modelo27[[1]]
modelo27_1$modelo ="Neural22"

modelo28_1 <- modelo28[[1]]
modelo28_1$modelo ="Neural23"

modelo29_1 <- modelo29[[1]]
modelo29_1$modelo ="Neural24"

union2 <- rbind(modelo6_1, modelo7_1, modelo8_1, modelo9_1, modelo10_1,
          modelo11_1, modelo12_1, modelo13_1, modelo14_1, modelo15_1,
modelo16_1, modelo17_1,

modelo18_1,modelo19_1,modelo20_1,modelo21_1,modelo22_1,modelo23_1,modelo2
4_1,modelo25_1,
          modelo26_1,modelo27_1,modelo28_1,modelo29_1)
```

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


# *********************************
# CRUZADA Random Forest
# ******************************


cruzadarfbin<-
 function(data=data,vardep="vardep",
  listconti="listconti",listclass="listclass",
  grupos=4,sinicio=1234,repe=5,nodesize=20,
  mtry=2,ntree=50,replace=TRUE,sampsize=1)
 {


  # Preparación del archivo

  # b)pasar las categóricas a dummies

  if  (listclass!=c(""))
  {
   databis<-data[,c(vardep,listconti,listclass)]
   databis<- dummy.data.frame(databis, listclass, sep = ".")
  }  else   {
   databis<-data[,c(vardep,listconti)]
  }

  # c)estandarizar las variables continuas

  # Calculo medias y dtipica de datos y estandarizo (solo las continuas)

  means <-apply(databis[,listconti],2,mean)
  sds<-sapply(databis[,listconti],sd)

  # Estandarizo solo las continuas y uno con las categoricas

  datacon<-scale(databis[,listconti], center = means, scale = sds)
  numerocont<-which(colnames(databis)%in%listconti)
  databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

  databis[,vardep]<-as.factor(databis[,vardep])

  formu<-formula(paste("factor(",vardep,")~.",sep=""))

  # Preparo caret

  set.seed(sinicio)
  control<-trainControl(method = "repeatedcv",number=grupos,repeats=repe,
```

```
  savePredictions = "all",classProbs=TRUE)

 # Aplico caret y construyo modelo

 rfgrid <-expand.grid(mtry=mtry)

     if  (sampsize==1)
 {
   rf<- train(formu,data=databis,
  method="rf",trControl=control,
  tuneGrid=rfgrid,nodesize=nodesize,replace=replace,ntree=ntree)
 }

else  if  (sampsize!=1)
 {
   rf<- train(formu,data=databis,
  method="rf",trControl=control,
  tuneGrid=rfgrid,nodesize=nodesize,replace=replace,sampsize=sampsize,
  ntree=ntree)
 }

 print(rf$results)

 preditest<-rf$pred

 preditest$prueba<-strsplit(preditest$Resample,"[.]")
 preditest$Fold <- sapply(preditest$prueba, "[", 1)
 preditest$Rep <- sapply(preditest$prueba, "[", 2)
 preditest$prueba<-NULL

 tasafallos<-function(x,y) {
  confu<-confusionMatrix(x,y)
  tasa<-confu[[3]][1]
  return(tasa)
 }

 # Aplicamos función sobre cada Repetición

 medias<-preditest %>%
  group_by(Rep) %>%
  summarize(tasa=1-tasafallos(pred,obs))

 # CalculamoS AUC  por cada Repetición de cv
 # Definimnos función

 auc<-function(x,y) {
  curvaroc<-roc(response=x,predictor=y)
  auc<-curvaroc$auc
  return(auc)
 }
```

```
  # Aplicamos función sobre cada Repetición

  mediasbis<-preditest %>%
   group_by(Rep) %>%
   summarize(auc=auc(obs,Yes))

  # Unimos la info de auc y de tasafallos

  medias$auc<-mediasbis$auc

   return(list(medias,preditest))

 }

modelo30 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                         listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                         grupos=10,sinicio=12345,repe=30,nodesize=10,
                         mtry=4,ntree=50,replace=TRUE,sampsize=450) #Mejor con mtry = 4

modelo30.1 <- modelo30[[1]]

modelo31 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                         listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                         grupos=10,sinicio=12345,repe=30,nodesize=10,
                         mtry=2,ntree=100,replace=TRUE,sampsize=450) #Mejor con mtry = 2

modelo31.1 <- modelo31[[1]]


modelo32 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                         listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                         grupos=10,sinicio=12345,repe=30,nodesize=10,
                         mtry=2,ntree=500,replace=TRUE,sampsize=450) #Mejor con mtry = 2

modelo32.1 <- modelo32[[1]]

modelo33 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                         listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                         grupos=10,sinicio=12345,repe=30,nodesize=10,
```

```
                mtry=2,ntree=1000,replace=TRUE,sampsize=450) #Mejor con mtry = 2

modelo33.1 <- modelo33[[1]]

modelo34 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                grupos=10,sinicio=12345,repe=30,nodesize=20,
                mtry=2,ntree=50,replace=TRUE,sampsize=507) #Mejor con mtry = 2

modelo34.1 <- modelo34[[1]]

modelo35 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                grupos=10,sinicio=12345,repe=30,nodesize=20,
                mtry=2,ntree=100,replace=TRUE,sampsize=507) #Mejor con mtry = 2

modelo35.1 <- modelo35[[1]]

modelo36 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                grupos=10,sinicio=12345,repe=30,nodesize=20,
                mtry=2,ntree=500,replace=TRUE,sampsize=507) #Mejor con mtry = 2

modelo36.1 <- modelo36[[1]]

modelo37 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                grupos=10,sinicio=12345,repe=30,nodesize=20,
                mtry=2,ntree=1000,replace=TRUE,sampsize=507) #Mejor con mtry = 2

modelo37.1 <- modelo37[[1]]

modelo30.1$modelo ="RF1"
modelo31.1$modelo ="RF2"
modelo32.1$modelo ="RF3"
modelo33.1$modelo ="RF4"
modelo34.1$modelo ="RF5"
modelo35.1$modelo ="RF6"
modelo36.1$modelo ="RF7"
```

```
modelo37.1$modelo ="RF8"


union3 <- rbind(modelo30.1, modelo31.1, modelo32.1, modelo33.1, modelo34.1,
modelo35.1, modelo36.1
              , modelo37.1)

boxplot(data=union3,tasa~modelo,main="TASA FALLOS")
boxplot(data=union3,auc~modelo,main="AUC")

union.prov <- rbind(union1, union2, union3)

boxplot(data=union.prov,tasa~modelo,main="TASA FALLOS", par(las=2))
boxplot(data=union.prov,auc~modelo,main="AUC", par(las=2))

##########BAGGING############

modelo38 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=10,
                    mtry=13,ntree=50,replace=TRUE,sampsize=450)

modelo38.1 <- modelo38[[1]]

modelo39 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=10,
                    mtry=13,ntree=100,replace=TRUE,sampsize=450) #Mejor con mtry = 2

modelo39.1 <- modelo39[[1]]


modelo40 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=10,
                    mtry=13,ntree=500,replace=TRUE,sampsize=450) #Mejor con mtry = 2

modelo40.1 <- modelo40[[1]]

modelo41 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
```

```
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=10,
                    mtry=13,ntree=1000,replace=TRUE,sampsize=450) #Mejor con mtry =
2

modelo41.1 <- modelo41[[1]]

modelo42 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=20,
                    mtry=13,ntree=50,replace=TRUE,sampsize=507) #Mejor con mtry = 2

modelo42.1 <- modelo42[[1]]

modelo43 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=20,
                    mtry=13,ntree=100,replace=TRUE,sampsize=507) #Mejor con mtry = 2

modelo43.1 <- modelo43[[1]]

modelo44 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=20,
                    mtry=13,ntree=500,replace=TRUE,sampsize=507) #Mejor con mtry = 2

modelo44.1 <- modelo44[[1]]

modelo45 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801580", "notaMax_801585", "cursada_801585", "cursada_801588",
"notaMax_801588", "notaMax_801590", "notaMax_801589", "notaMax_801586"),
                    listclass = c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,nodesize=20,
                    mtry=13,ntree=1000,replace=TRUE,sampsize=507) #Mejor con mtry =
2

modelo45.1 <- modelo45[[1]]

modelo38.1$modelo ="Bag1"
modelo39.1$modelo ="Bag2"
```

```
modelo40.1$modelo ="Bag3"
modelo41.1$modelo ="Bag4"
modelo42.1$modelo ="Bag5"
modelo43.1$modelo ="Bag6"
modelo44.1$modelo ="Bag7"
modelo45.1$modelo ="Bag8"

union4 <- rbind(modelo38.1, modelo39.1,modelo40.1, modelo41.1, modelo42.1,
modelo43.1, modelo44.1,
            modelo45.1)

boxplot(data=union4,tasa~modelo,main="TASA FALLOS")
boxplot(data=union4,auc~modelo,main="AUC")

union.prov <- rbind(union3,union4)

boxplot(data=union.prov,tasa~modelo,main="TASA FALLOS", par(las=2))
boxplot(data=union.prov,auc~modelo,main="AUC", par(las=2))

# ****************************************************************
# gbm : parámetros

#
#    Number of Boosting Iterations (n.trees, numeric)
#    Max Tree Depth (max.depth, numeric)
#    Shrinkage (shrinkage, numeric)
#    Min. Terminal Node Size (n.minobsinnode, numeric)
#
# ****************************************************************



cruzadagbmbin<-
 function(data=data,vardep="vardep",
  listconti="listconti",listclass="listclass",
  grupos=4,sinicio=1234,repe=5,
  n.minobsinnode=20,shrinkage=0.1,n.trees=100,interaction.depth=2)
 {

  # Preparación del archivo

  # b)pasar las categóricas a dummies

  if  (listclass!=c(""))
  {
   databis<-data[,c(vardep,listconti,listclass)]
   databis<- dummy.data.frame(databis, listclass, sep = ".")
  }  else   {
   databis<-data[,c(vardep,listconti)]
  }
```

```
# c)estandarizar las variables continuas

# Calculo medias y dtipica de datos y estandarizo (solo las continuas)

means <-apply(databis[,listconti],2,mean)
sds<-sapply(databis[,listconti],sd)

# Estandarizo solo las continuas y uno con las categoricas

datacon<-scale(databis[,listconti], center = means, scale = sds)
numerocont<-which(colnames(databis)%in%listconti)
databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

databis[,vardep]<-as.factor(databis[,vardep])

formu<-formula(paste("factor(",vardep,")~.",sep=""))

# Preparo caret

set.seed(sinicio)
control<-trainControl(method = "repeatedcv",number=grupos,repeats=repe,
 savePredictions = "all",classProbs=TRUE)

# Aplico caret y construyo modelo



 gbmgrid <-expand.grid(n.minobsinnode=n.minobsinnode,
  shrinkage=shrinkage,n.trees=n.trees,
  interaction.depth=interaction.depth)

gbm<- train(formu,data=databis,
 method="gbm",trControl=control,
 tuneGrid=gbmgrid,distribution="bernoulli",verbose=FALSE)

print(gbm$results)

preditest<-gbm$pred


preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

tasafallos<-function(x,y) {
 confu<-confusionMatrix(x,y)
 tasa<-confu[[3]][1]
 return(tasa)
}
```

```
  # Aplicamos función sobre cada Repetición

  medias<-preditest %>%
   group_by(Rep) %>%
   summarize(tasa=1-tasafallos(pred,obs))

  # CalculamoS AUC  por cada Repetición de cv
  # Definimnos función

  auc<-function(x,y) {
   curvaroc<-roc(response=x,predictor=y)
   auc<-curvaroc$auc
   return(auc)
  }

  # Aplicamos función sobre cada Repetición

  mediasbis<-preditest %>%
   group_by(Rep) %>%
   summarize(auc=auc(obs,Yes))

  # Unimos la info de auc y de tasafallos

  medias$auc<-mediasbis$auc

   return(list(medias,preditest))

 }

modelo46 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                          listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                          listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                          grupos=10,sinicio=12345,repe=30,

n.minobsinnode=5,shrinkage=0.05,n.trees=100,interaction.depth=4)
modelo46.1 <- modelo46[[1]]



modelo47 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=10,shrinkage=0.05,n.trees=100,interaction.depth=4)
```

```
modelo47.1 <- modelo47[[1]]


modelo48 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=20,shrinkage=0.05,n.trees=100,interaction.depth=4)

modelo48.1 <- modelo48[[1]]

modelo49 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=5,shrinkage=0.01,n.trees=1000,interaction.depth=4)

modelo49.1 <- modelo49[[1]]

modelo50 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=10,shrinkage=0.01,n.trees=1000,interaction.depth=4)

modelo50.1 <- modelo50[[1]]

modelo51 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=20,shrinkage=0.01,n.trees=1000,interaction.depth=4)

modelo51.1 <- modelo51[[1]]

modelo52 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
```

```
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=5,shrinkage=0.001,n.trees=5000,interaction.depth=4)

modelo52.1 <- modelo52[[1]]

modelo53 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,

n.minobsinnode=10,shrinkage=0.001,n.trees=5000,interaction.depth=4)

modelo53.1 <- modelo53[[1]]

modelo54 <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,

n.minobsinnode=20,shrinkage=0.001,n.trees=5000,interaction.depth=4)

modelo54.1 <- modelo54[[1]]

modelo46bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=5,shrinkage=0.05,n.trees=100,interaction.depth=5)


modelo47bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=10,shrinkage=0.05,n.trees=100,interaction.depth=5)

modelo48bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
```

```
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=20,shrinkage=0.05,n.trees=100,interaction.depth=5)

modelo49bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=5,shrinkage=0.01,n.trees=1000,interaction.depth=5)

modelo50bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=10,shrinkage=0.01,n.trees=1000,interaction.depth=5)

modelo51bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=20,shrinkage=0.01,n.trees=1000,interaction.depth=5)

modelo52bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    n.minobsinnode=5,shrinkage=0.001,n.trees=5000,interaction.depth=5)

modelo53bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
```

```
n.minobsinnode=10,shrinkage=0.001,n.trees=5000,interaction.depth=5)

modelo54bis <- cruzadagbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,

n.minobsinnode=20,shrinkage=0.001,n.trees=5000,interaction.depth=5)

modelo46.1$modelo <- "GBM1"
modelo47.1$modelo <- "GBM2"
modelo48.1$modelo <- "GBM3"
modelo49.1$modelo <- "GBM4"
modelo50.1$modelo <- "GBM5"
modelo51.1$modelo <- "GBM6"
modelo52.1$modelo <- "GBM7"
modelo53.1$modelo <- "GBM8"
modelo54.1$modelo <- "GBM9"
modelo46bis[[1]]$modelo <- "GBM10"
modelo47bis[[1]]$modelo <- "GBM11"
modelo48bis[[1]]$modelo <- "GBM12"
modelo49bis[[1]]$modelo <- "GBM13"
modelo50bis[[1]]$modelo <- "GBM14"
modelo51bis[[1]]$modelo <- "GBM15"
modelo52bis[[1]]$modelo <- "GBM16"
modelo53bis[[1]]$modelo <- "GBM17"
modelo54bis[[1]]$modelo <- "GBM18"

union5 <- rbind(modelo46.1, modelo47.1, modelo48.1, modelo49.1, modelo50.1,
modelo51.1, modelo52.1,
          modelo53.1, modelo54.1, modelo46bis[[1]], modelo47bis[[1]],
modelo48bis[[1]], modelo49bis[[1]]
          , modelo50bis[[1]], modelo51bis[[1]], modelo52bis[[1]], modelo53bis[[1]],
modelo54bis[[1]])

boxplot(data=union5,tasa~modelo,main="TASA FALLOS")
boxplot(data=union5,auc~modelo,main="AUC")

union.prov <- rbind(union1, union2, union3,union4, union5)

boxplot(data=union.prov,tasa~modelo,main="TASA FALLOS", par(las=2))
boxplot(data=union.prov,auc~modelo,main="AUC", par(las=2))


# ************************************************************
# xgboost: parámetros
```

```
   # nrounds (# Boosting Iterations)
    # max_depth (Max Tree Depth)
    # eta (Shrinkage)
    # gamma (Minimum Loss Reduction)
    # colsample_bytree (Subsample Ratio of Columns)
    # min_child_weight (Minimum Sum of Instance Weight)
    # subsample (Subsample Percentage)
#
# PONER linout = FALSE
# ****************************************************************

cruzadaxgbmbin<- function(data=data,vardep="vardep",
  listconti="listconti",listclass="listclass",
  grupos=4,sinicio=1234,repe=5,
  min_child_weight=20,eta=0.1,nrounds=100,max_depth=2,
  gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)
 {

  # Preparación del archivo

  # b)pasar las categóricas a dummies

  if  (listclass!=c(""))
  {
   databis<-data[,c(vardep,listconti,listclass)]
   databis<- dummy.data.frame(databis, listclass, sep = ".")
  }  else   {
   databis<-data[,c(vardep,listconti)]
  }

  # c)estandarizar las variables continuas

  # Calculo medias y dtipica de datos y estandarizo (solo las continuas)

  means <-apply(databis[,listconti],2,mean)
  sds<-sapply(databis[,listconti],sd)

  # Estandarizo solo las continuas y uno con las categoricas

  datacon<-scale(databis[,listconti], center = means, scale = sds)
  numerocont<-which(colnames(databis)%in%listconti)
  databis<-cbind(datacon,databis[,-numerocont,drop=FALSE ])

  databis[,vardep]<-as.factor(databis[,vardep])

  formu<-formula(paste("factor(",vardep,")~.",sep=""))

  # Preparo caret

  set.seed(sinicio)
  control<-trainControl(method = "repeatedcv",number=grupos,repeats=repe,
```

```
 savePredictions = "all",classProbs=TRUE)

# Aplico caret y construyo modelo

xgbmgrid <-expand.grid( min_child_weight=min_child_weight,
eta=eta,nrounds=nrounds,max_depth=max_depth,
gamma=gamma,colsample_bytree=colsample_bytree,subsample=subsample)

xgbm<- train(formu,data=databis,
 method="xgbTree",trControl=control,
 tuneGrid=xgbmgrid,objective = "binary:logistic",verbose=FALSE,
 alpha=alpha,lambda=lambda,lambda_bias=lambda_bias)

print(xgbm$results)

preditest<-xgbm$pred


preditest$prueba<-strsplit(preditest$Resample,"[.]")
preditest$Fold <- sapply(preditest$prueba, "[", 1)
preditest$Rep <- sapply(preditest$prueba, "[", 2)
preditest$prueba<-NULL

tasafallos<-function(x,y) {
 confu<-confusionMatrix(x,y)
 tasa<-confu[[3]][1]
 return(tasa)
}

# Aplicamos función sobre cada Repetición

medias<-preditest %>%
 group_by(Rep) %>%
 summarize(tasa=1-tasafallos(pred,obs))

# CalculamoS AUC  por cada Repetición de cv
# Definimnos función

auc<-function(x,y) {
 curvaroc<-roc(response=x,predictor=y)
 auc<-curvaroc$auc
 return(auc)
}

# Aplicamos función sobre cada Repetición

mediasbis<-preditest %>%
 group_by(Rep) %>%
 summarize(auc=auc(obs,Yes))

# Unimos la info de auc y de tasafallos
```

```
  medias$auc<-mediasbis$auc

   return(list(medias,preditest))

 }

modelo55 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                      listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                      listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                     grupos=10,sinicio=12345,repe=30,
                     min_child_weight=5,eta=0.05,nrounds=100,max_depth=6,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo56 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                      listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                      listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                      grupos=10,sinicio=12345,repe=30,
                      min_child_weight=5,eta=0.05,nrounds=100,max_depth=5,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo57 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                      listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                      listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                      grupos=10,sinicio=12345,repe=30,
                      min_child_weight=5,eta=0.05,nrounds=100,max_depth=4,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo58 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                      listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                      listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                      grupos=10,sinicio=12345,repe=30,
                      min_child_weight=10,eta=0.05,nrounds=100,max_depth=6,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)
```

```
modelo59 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=10,eta=0.05,nrounds=100,max_depth=5,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo60 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=10,eta=0.05,nrounds=100,max_depth=4,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo61 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=20,eta=0.1,nrounds=100,max_depth=6,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo62 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=20,eta=0.1,nrounds=100,max_depth=5,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo63 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
```

```
                    min_child_weight=20,eta=0.1,nrounds=100,max_depth=4,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo64 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=5,eta=0.03,nrounds=300,max_depth=6,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo65 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=5,eta=0.03,nrounds=300,max_depth=5,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo66 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=5,eta=0.01,nrounds=300,max_depth=4,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo67 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=10,eta=0.05,nrounds=300,max_depth=6,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo68 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
```

```
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=10,eta=0.01,nrounds=300,max_depth=5,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo69 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=10,eta=0.01,nrounds=300,max_depth=4,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo70 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=20,eta=0.03,nrounds=300,max_depth=6,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo71 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=20,eta=0.1,nrounds=300,max_depth=5,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo72 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=20,eta=0.03,nrounds=300,max_depth=4,
```

```
gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=0,lambda_bias=0)

modelo55[[1]]$modelo = "XGB1"
modelo56[[1]]$modelo = "XGB2"
modelo57[[1]]$modelo = "XGB3"
modelo58[[1]]$modelo = "XGB4"
modelo59[[1]]$modelo = "XGB5"
modelo60[[1]]$modelo = "XGB6"
modelo61[[1]]$modelo = "XGB7"
modelo62[[1]]$modelo = "XGB8"
modelo63[[1]]$modelo = "XGB9"
modelo64[[1]]$modelo = "XGB10"
modelo65[[1]]$modelo = "XGB11"
modelo66[[1]]$modelo = "XGB12"
modelo67[[1]]$modelo = "XGB13"
modelo68[[1]]$modelo = "XGB14"
modelo69[[1]]$modelo = "XGB15"
modelo70[[1]]$modelo = "XGB16"
modelo71[[1]]$modelo = "XGB17"
modelo72[[1]]$modelo = "XGB18"
modelo73[[1]]$modelo = "XGB19"
modelo74[[1]]$modelo = "XGB20"

#Con regularización gamma = 1, lambda = 1

modelo73 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=10,eta=0.05,nrounds=100,max_depth=5,

gamma=1,colsample_bytree=1,subsample=1,alpha=0,lambda=1,lambda_bias=0)

#Con regularización gamma = 5, lambda = 10

modelo74 <- cruzadaxgbmbin(data=definitivo,vardep="AbandonoStr",
                    listconti=c("notaMax_801580", "notaMax_801585",
"cursada_801585", "cursada_801588", "notaMax_801588", "notaMax_801590",
"notaMax_801589", "notaMax_801586"),
                    listclass=c("almacenaCursoEntrada", "cursada_801589",
"estadoAsig_801581", "estadoAsig_801586"),
                    grupos=10,sinicio=12345,repe=30,
                    min_child_weight=10,eta=0.05,nrounds=100,max_depth=5,

gamma=0,colsample_bytree=1,subsample=1,alpha=0,lambda=1,lambda_bias=0)
```

```
union6 <-
rbind(modelo55[[1]],modelo56[[1]],modelo57[[1]],modelo58[[1]],modelo59[[1]],modelo60
[[1]]

,modelo61[[1]],modelo62[[1]],modelo63[[1]],modelo64[[1]],modelo65[[1]],modelo66[[1]]

,modelo67[[1]],modelo68[[1]],modelo69[[1]],modelo70[[1]],modelo71[[1]],modelo72[[1]]
            ,modelo73[[1]],modelo74[[1]])

union.prov <- rbind(union5, union6)

boxplot(data=union6,tasa~modelo,main="TASA FALLOS", par(las=2))
boxplot(data=union6,auc~modelo,main="AUC", par(las=2))

boxplot(data=union.prov,tasa~modelo,main="TASA FALLOS", par(las=2), ylim =
c(0.12, 0.2))
boxplot(data=union.prov,auc~modelo,main="AUC", par(las=2))



##Tablas de confusión##

confusionMatrix(modelo90[[2]][["pred"]],modelo90[[2]][["obs"]],"Yes")

confusionMatrix(modelo5[[2]][["pred"]],modelo5[[2]][["obs"]],"Yes")

confusionMatrix(modelo14[[2]][["pred"]],modelo14[[2]][["obs"]],"Yes")

confusionMatrix(modelo32[[2]][["pred"]],modelo32[[2]][["obs"]],"Yes")

confusionMatrix(modelo73[[2]][["pred"]],modelo73[[2]][["obs"]],"Yes")

mejoresModelos <- rbind(modelo14_1, modelo32.1, modelo74[[1]])

boxplot(data=mejoresModelos,tasa~modelo,main="TASA FALLOS", par(las=2))
boxplot(data=mejoresModelos,auc~modelo,main="AUC", par(las=2))

# IMPORTANCIA DE VARIABLES RF3

listconti = c("notaMax_801580", "notaMax_801585", "cursada_801585",
"cursada_801588", "notaMax_801588", "notaMax_801590", "notaMax_801589",
"notaMax_801586")
listclass= c("almacenaCursoEntrada", "cursada_801589", "estadoAsig_801581",
"estadoAsig_801586")
vardep="AbandonoStr"

set.seed(12345)
rfgrid<-expand.grid(mtry=c(2))

control<-trainControl(method = "cv",number=10,savePredictions = "all",
                      classProbs=TRUE)
```

```
rf<-
train(factor(AbandonoStr)~notaMax_801580+notaMax_801585+cursada_801585+curs
ada_801588+notaMax_801588+notaMax_801590+notaMax_801589+notaMax_80158
6+almacenaCursoEntrada+cursada_801589+estadoAsig_801581+estadoAsig_801586
,definitivo,
        method="rf",trControl=control,tuneGrid=rfgrid,
        linout = FALSE,ntree=500,nodesize=10,replace=TRUE,
        importance=TRUE)

rf


final<-rf$finalModel

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

barplot(tabla$MeanDecreaseAccuracy,names.arg=rownames(tabla), las=2)

arbol <-
train(factor(AbandonoStr)~notaMax_801580+notaMax_801585+cursada_801585+curs
ada_801588+notaMax_801588+notaMax_801590+notaMax_801589+notaMax_80158
6+almacenaCursoEntrada+cursada_801589+estadoAsig_801581+estadoAsig_801586
,definitivo,
            method="rpart",trControl=control,parms = list(split = "gini"),
            tuneLength = 2)

rpart.plot::prp(arbol$finalModel, box.palette = "Reds", tweak = 1.2)

#Modelo ganador bruto

ganador1 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("IMP_almacenaPAU", "cursada_801583"),
                    listclass = c(""),
                    grupos=10,sinicio=12345,repe=30,nodesize=10,
                    mtry=2,ntree=500,replace=TRUE,sampsize=450)

ganador2 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801584",
                                                    "notaMax_801586",
                                                    "notaMax_801589",
                                                    "notaMax_801590",
                                                    "notaMax_801588",
                                                    "IMP_almacenaPAU"),
                    listclass = c("cursada_801584",
                                "cursada_801586" ,
                                "cursada_801589" ,
                                "cursada_801580" ,
                                "cursada_801588" ,
```

```
                    "estadoAsig_801584" ,
                    "estadoAsig_801586" ,
                    "estadoAsig_801589" ,
                    "estadoAsig_801580" ,
                    "estadoAsig_801588"),
                grupos=10,sinicio=12345,repe=30,nodesize=10,
                mtry=2,ntree=500,replace=TRUE,sampsize=450)

ganador3 <- cruzadarfbin(data = definitivo, vardep = "AbandonoStr", listconti =
c("notaMax_801584",
                                                    "notaMax_801585",
                                                    "notaMax_801586",
                                                    "notaMax_801587",
                                                    "notaMax_801589",
                                                    "notaMax_801590",
                                                    "notaMax_801580",
                                                    "notaMax_801581",
                                                    "notaMax_801583",
                                                    "notaMax_801588",
                                                    "IMP_almacenaPAU"),
                listclass = c("cursada_801584",
                    "cursada_801585",
                    "cursada_801586" ,
                    "cursada_801587" ,
                    "cursada_801589" ,
                    "cursada_801590" ,
                    "cursada_801580" ,
                    "cursada_801581" ,
                    "cursada_801583" ,
                    "cursada_801588" ,
                    "estadoAsig_801584" ,
                    "estadoAsig_801585" ,
                    "estadoAsig_801586" ,
                    "estadoAsig_801587" ,
                    "estadoAsig_801589" ,
                    "estadoAsig_801590" ,
                    "estadoAsig_801580" ,
                    "estadoAsig_801581" ,
                    "estadoAsig_801583" ,
                    "estadoAsig_801588" ,
                    "almacenaCursoEntrada",
                    "almacenaSexo"),
                grupos=10,sinicio=12345,repe=30,nodesize=10,
                mtry=2,ntree=500,replace=TRUE,sampsize=450)

ganador1[[1]]$modelo <- "Mejor1"
ganador2[[1]]$modelo <- "Mejor2"
ganador3[[1]]$modelo <- "Mejor3"

unionMejores <- rbind(ganador1[[1]],ganador2[[1]],ganador3[[1]])
```

```
boxplot(data=unionMejores,tasa~modelo,main="TASA FALLOS", par(las=2), ylim =
c(0.14,0.2))

boxplot(data=modelo73[[1]],tasa~modelo,main="TASA FALLOS", par(las=2), cex.axis
= 1)
```